



Guidance, Navigation and Control for Autonomous Quadrotor Flight in an Agricultural Field: The Case of Vineyards

Adel Mokrane, Abdelaziz Benallegue, Amal Choukchou-Braham, A. El Hadri, Brahim Cherki

► To cite this version:

Adel Mokrane, Abdelaziz Benallegue, Amal Choukchou-Braham, A. El Hadri, Brahim Cherki. Guidance, Navigation and Control for Autonomous Quadrotor Flight in an Agricultural Field: The Case of Vineyards. *Sensors*, 2022, 22 (22), 10.3390/s22228865 . hal-03888937

HAL Id: hal-03888937

<https://hal.science/hal-03888937>

Submitted on 20 Dec 2022

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

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



Distributed under a Creative Commons Attribution 4.0 International License

Article

Guidance, Navigation and Control for Autonomous Quadrotor Flight in an Agricultural Field: The Case of Vineyards

Adel Mokrane ^{1,2,*} , Abdelaziz Benallegue ¹, Amal Choukchou-Braham ², Abdelhafid El Hadri ¹  and Brahim Cherki ²

¹ Laboratoire d'Ingénierie des Systèmes de Versailles, Université Paris-Saclay, UVSQ, LISV, 78124 Vélizy-Villacoublay, France

² Laboratoire d'Automatique de Tlemcen, University of Tlemcen, Tlemcen 13000, Algeria

* Correspondence: adel.mokrane@uvsq.fr; Tel.: +33-748-662-948

Abstract: In this paper, we present a complete and efficient solution of guidance, navigation and control for a quadrotor platform to accomplish 3D coverage flight missions in mapped vineyard terrains. Firstly, an occupancy grid map of the terrain is used to generate a safe guiding coverage path using an Iterative Structured Orientation planning algorithm. Secondly, way-points are extracted from the generated path and added to them trajectory's velocities and accelerations constraints. The constrained way-points are fed into a Linear Quadratic Regulator algorithm so as to generate global minimum snap optimal trajectory while satisfying both the pointing and the corridor constraints. Then, when facing unexpected obstacles, the quadrotor tends to re-plan its path in real-time locally using an Improved Artificial Potential Field algorithm. Finally, a geometric trajectory tracking controller is developed on the Special Euclidean group $SE(3)$. The aim of this controller is to track the generated trajectory while pointing towards predetermined direction using the vector measurements provided by the inertial unit. The performance of the proposed method is demonstrated through several simulation results. In particular, safe guiding paths are achieved. Obstacle-free optimal trajectories that satisfy the way-point position, the pointing direction, and the corridor constraints, are successfully generated with optimized platform snap. Besides, the implemented geometric controller can achieve higher trajectory tracking accuracy with an absolute value of the maximum error in the order of 10^{-3} m.

Keywords: guidance; navigation; trajectory generation; quadrotor; re-planning; trajectory tracking controller



Citation: Mokrane, A.; Benallegue, A.; Choukchou-Braham, A.; El Hadri, A.; Cherki, B. Guidance, Navigation and Control for Autonomous Quadrotor Flight in an Agricultural Field: The Case of Vineyards. *Sensors* **2022**, *22*, 8865. <https://doi.org/10.3390/s22228865>

Academic Editor: Gregor Klancar

Received: 3 October 2022

Accepted: 11 November 2022

Published: 16 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

A century ago, technology was introduced to agriculture; the first tractor was launched in 1913 [1]. Nowadays, mechanical automated machines have been extensively used in many agricultural tasks. This has led to increased productivity, and has reduced the amount of human labor. This may not be sufficient to satisfy the world's need for food, yet. Hence, in the beginning of the 1990s, the concept of Precision Agriculture (PA) was created. PA consists of specific management practices based on observation, measurement and response to inter and intra-field crop variability [2]. In PA, farmers and growers are able to increase productivity with quality by the means of agricultural systems in which the site-specific management practice, that divides the agricultural fields into zones or blocks, is implemented at the right place, with the right intensity, and at the right time [3]. This is particularly true in viticulture where it is possible to differentiate between zones of different quality of grapes and perform proper management. Precision Viticulture (PV) is a cyclic management approach that aims to use the available developed technologies for maximizing the oenological potential and optimizing agronomic inputs in the vineyard to minimize the costs and protect environmental sustainability.

Many regional strategic research agendas have been developed in the recent years. Most of them state that the use of automated robotic platforms will result in food security and improve agriculture efficiency. Although many research approaches have been conducted, few commercial solutions have come into existence recently [4]. Pruning, spraying, planting, harvesting, monitoring, and remote sensing are known as the field operations of such approaches where autonomous navigation is critical. Autonomous navigation, in general, comprises localization, mapping, motion control and path and trajectory planning. This latter is the crucial part to insure safe and efficient robot navigation in any environment.

Path and trajectory planning is an essential tool for automation and optimization of the aforementioned field operations. It is used to achieve a complete guidance, navigation and control (GNC) of automated machines during the execution of the field operation [5]. The field of path planning can be categorized mainly in two major aspects: the point-to-point optimization path planning and the coverage optimization path planning. In recent years, researchers and experts have concentrated their research on the first aspect; however, the study of the second one has relatively reduced.

Most of coverage optimization path planning algorithms deal with 2D field terrains. The aim of these algorithms is to optimally generate driving angles and sequence of tracks of 2D paths so that the field operations can be executed while reducing maneuvering over the field terrain and total time of the operation. Consequently, this may result in reducing soil compaction and fuel consumption. 2D coverage path planning approaches take into consideration that most of the agricultural fields are flat. However, ignoring the elevation changes affects the optimization process, which leads to an inadequate coverage path design [6]. The agricultural field characteristics have a substantial effect on the design and optimization of the coverage path approaches.

Vineyards terrain is an example of an intricate, unstructured, and fickle agricultural fields which can be challenging for robotic machines. Due to their steep slopes and limited room for maneuvering, the need to develop specific path and trajectory planning systems arises. Robotics for vineyards have been well investigated in the literature. Several applications, such as trunk recognition and localization [7], pruning [8], yield estimation [9] and irrigation [10], have been developed. Different algorithm approaches were used to perform the applications tasks. Graph-based approaches using A* and Dijkstra to solve navigation within the vineyard rows have been presented in [11–13]. Most of the algorithms were designed for Unmanned Ground Vehicles (UGVs). These latter have a higher autonomy for covering long distances and offer the benefit of carrying heavier sensor payloads for taking a variety of measurements simultaneously. However, due to their size, weight and intense usage, UGVs have drawbacks of soil compaction causing fertility problems. Besides, vineyards, which are planted in line rows with steep slope hills and/or plowed soil, represent higher complex conditions for UGVs. This has led researchers to develop efficient navigation solutions within vineyard lanes using UAV systems. To the best of our knowledge, few research studies have been proposed in this regard. The researchers in [14] reported a feasible approach to survey vineyards using UAVs. The approach described how navigation missions were achieved in three steps: offline definition of the workspace, computation of coverage paths, and mission execution. The experimental implementation of the approach was conducted and a brief analysis of the results was highlighted in [15]. In [16,17], a UAV was exploited perform in-field operations. An inter-row vine navigation strategy was built by combining a Rapidly-exploring Random Tree (RRT*) global path planner, a Dynamic-Window Approach (DWA) local path planner and a Model Predictive Control (MPC) trajectory tracker. Table 1, which summarizes the proposed approaches, reports that none of these approaches offer complete solutions to solve the autonomous navigation problem for UAVs in a vineyard.

Table 1. The GNC problems solved by the mentioned references.

Reference No.	UGV	UAV	2D	3D	Coverage Path Planning	Trajectory Optimization	Trajectory Tracking	Trajectory Re-Planning
[9]	X		X				X	
[11]	X		X		X			
[12]	X		X		X			
[13]	X		X		X	X		
[14]		X	X		X			
[15]		X	X		X	X		
[17]		X	X		X			
[16]		X		X		X	X	

The problem of 3D autonomous GNC for aerial vehicles in the agricultural field is very challenging and complex. In general, the problem cannot be solved directly. It can be decomposed into a multi-phase sub-problems: path planning, trajectory optimization, trajectory re-planning and trajectory tracking. Owing to the numerous challenges of each sub-problem, a feasibility screening among the possible already known technologies is conducted. In this paper, we attempt to design a complete and efficient solution, which will be tested in a robotic simulator, based on a combination of algorithms used for the first time in a PA scenario. These algorithms are accurately selected among those currently available in the literature aiming to identify the best combination of GNC algorithms. The solution is developed for 3D autonomous flying in vineyards. The proposed method generates optimal trajectories that allow the quadrotor to navigate within the vines rows while pointing toward them, simultaneously, for sake of performing remote sensing. An offline definition of the optimal trajectory is used. This task is typically performed in two consecutive steps. The first makes use of artificial Occupancy Grid Maps (OGMs) of the vineyards for generating safe coverage paths between the vines rows. The purpose of this step is to locate reference way-points with the aid of Digital Elevation Models (DEMs) of the vineyards terrain. The second step uses the extracted way-points to generate global paths that satisfy position, velocity and acceleration constraints at these way-points. Taking inspirations from [18], a Linear Quadratic Regulator (LQR) approach that minimizes the snap and treats the constraints at the way-points as soft, is adopted. In this way, the approach guarantees both a relaxation on satisfying the way-points constraints and a generation of stable position trajectories. When unknown obstacles are detected, an Artificial Potential Field (APF) re-planning strategy commands the UAV to move away from the obstacles. The UAV takes the current position (when obstacle detection occurs) as the start configuration and the next turning point in the global path as the target configuration. Afterwards, a geometric tracking controller, which is inspired from [19], is constructed to follow the predefined trajectories while simultaneously pointing towards prescribed direction (the vines in our case) along the whole flight. The control scheme is developed such that the vehicle orientation is reconstructed from the body measurement vectors provided by the Inertial Measurement Units (IMUs). Although the algorithms used were collected from the literature, minimum contributions were provided in our manuscript. These contributions were reported mainly in the trajectory optimization and the trajectory tracking phases:

1. An offline LQR position trajectory generator that satisfies not only the position and the acceleration constraints but even the velocity ones is designed. In this way, the corridor constraint, which keeps the quadrotor flying within the vine rows, is taken into consideration.
2. A new attitude control law that helps the quadrotor generate its attitude trajectory online using raw vector measurements is implemented. These vectors enable us to avoid the use of the desired attitude directly in the control law, as presented in most existing methods. The control law is used in a geometric trajectory controller that tracks the generated position and attitude trajectories.

The remainder of this paper is organized as follows. In Section 2, we present in detail the proposed GNC method used for the quadrotor. In Section 3, results are emphatically

provided. Section 4 discusses the findings of Section 3. Finally, the conclusion and further work are drawn in Section 5.

2. Materials and Methods

Broadly speaking, the UAV flight trajectory generation is divided into two modules: the initial path finding and the trajectory generation and optimization. The goal of the initial path finding is to plan passable collision-free paths. Due to the system dynamic feasibility and safety constraints, it is undesirable to provide the generated paths directly to the UAV in order to be executed. The trajectory generation and optimization aims to optimize the path from the former module while satisfying the vehicle dynamics feasibility. In addition, this module helps to achieve flight trajectories that guarantee higher degree of safety. In order to achieve safe and efficient offline flights, OGM and DEM representations of the vineyard environments are used. Figure 1 depicts an overview of the workflow used in this paper.

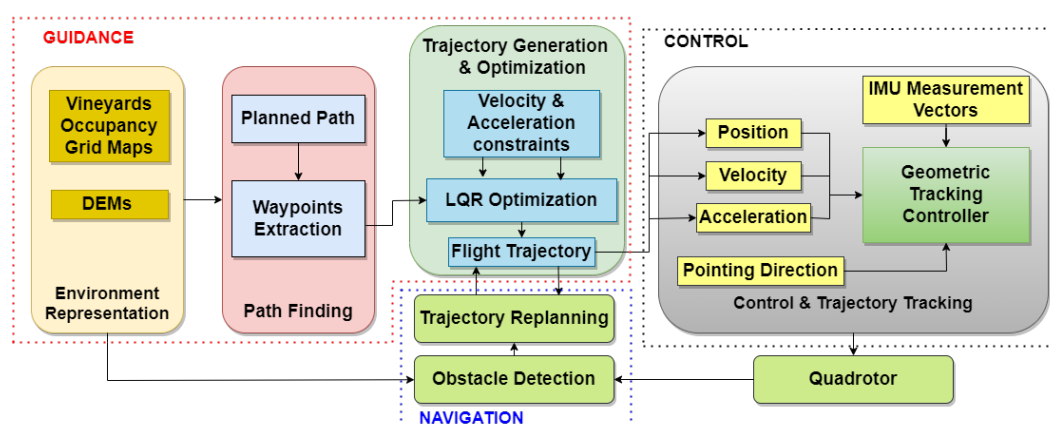


Figure 1. An overview of the GNC global structure.

The workflow above is mainly divided into five parts: UAV terrain mapping, path finding, trajectory generation and optimization, trajectory re-planning, and control and trajectory tracking.

2.1. UAV Terrain Mapping

Mapping is the process of modelling the robot's environment. It uses the observations from the robot's onboard sensors for constructing a consistent environment model called maps [9]. Maps are used by robotics vehicles for locating themselves and making motion plans. Aerial terrain mapping, which can be achieved using an UAV, is an essential feature for autonomous vehicles to perform navigation in unknown environments. The UAV acquires information about the terrain geometry and the existence of static and dynamic obstacles. In the literature, there are mainly three types of maps: grid maps [20], feature maps [21], and topological maps [22].

Occupancy Grid Maps (OGMs) are the mostly utilized type for robot's environment representation. They are simple and easy to maintain. They are used to store the obstacle information for sake of planning safe and stable geometric paths. OGMs decompose the environment into a grid of cells, each one either occupied, so that the agent cannot pass through it, or free, so that the agent can traverse it. OGM cannot be accurate; nevertheless, by selecting a small enough cell size they can offer all the necessary information. An example of OGM is the binary OGM that is adopted in this work where each cell is a binary random variable; 1 for free space and 0 for obstacle space (the vines in our case).

Another type of grid maps are Digital Elevation Maps (DEMs). They exhibit the relief of the terrain in digital format at regularly spaced horizontal intervals. DEMs are used in several applications like geospatial applications, 3D graphics displaying, terrain aspect, slope, and terrain profiles between selected points. They are sometimes referred to as

Digital Height Model (DHM) where the square grids are organized in columns and rows where each grid point refers to the height at that location.

2.2. Coverage Path Finding

Coverage path finding is the first step of trajectory generation. Coverage path finding consists of finding the path that covers any environment, while taking the motion restrictions and avoiding the collision with the obstacles present in that environment [23]. This step is computationally complex; hence, approximate or even heuristic solutions are used for solving it. The environment is divided into sub-regions, selecting a sequence of those sub-regions and then generating a path that traverses each sub-region in turn. Such methods take for granted the convex polygonal environment and perform exact cell decomposition, which is less time consuming, and the robot dimensions have exactly the same dimensions of one cell within the grid map. Several conditions are considered when performing coverage path finding [24]:

1. The robot must cover completely all points in its environment.
2. The robot must fill the environment without overlapping routes.
3. Continuous and sequential coverage process without repetition of the routes is entailed.
4. The robot must avoid obstacles (if present).
5. Motion routes, which are simple (straight lines or circles), must be used (for sake of simplicity).
6. Optimal path is obtained under predefined conditions.

However, taking into consideration all the above conditions in a complex environment is unfeasible in most situations.

This paper adopts the Iterative Structured Orientation Algorithm (ISOA) proposed in [25]. The ISOA is an optimal approach where its optimality is delineated in terms of the distance of the generated path. The reason behind choosing such an algorithm is that it makes use of OGMs and plans a complete coverage by using the approach of main lines. These are a beam of parallel lines with an orientation in the grid map. Straight lines with maximum length, which are bordered by the map and interrupted by the obstacles, are guaranteed by that orientation. The optimal continuous coverage path is achieved by connecting the beam of lines.

Given a modeled map $M = [M_{ij}]_{i=1..n, j=1..m} \in \mathbb{R}^{n \times m}$ as shown in Figure 2, where n is number of horizontal pixels and m is the number of vertical pixels with

$$M_{ij} = \begin{cases} 1, & \text{if pixel}(i, j) \text{ white} \\ 0, & \text{if pixel}(i, j) \text{ black} \end{cases} \quad (1)$$

The problem of finding an optimal path can be summarized as follows

1. Find the appropriate beam of parallel lines and their orientation;
2. Get the main segments;
3. Connect these segments with the auxiliary segments to form a continuous optimal coverage path.

The optimal path is used then for extracting the 2D local goals (way-points). Finally, by the use of the DEMs, the position along the z-axis is added to the way-points. The result of this step is a set of 3D local goals W required for trajectory generation. The flowchart of the approach is shown in Figure 3.

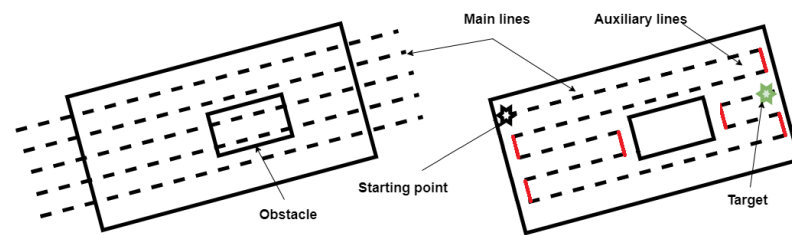


Figure 2. The map with both main lines (black) and auxiliary lines (red).

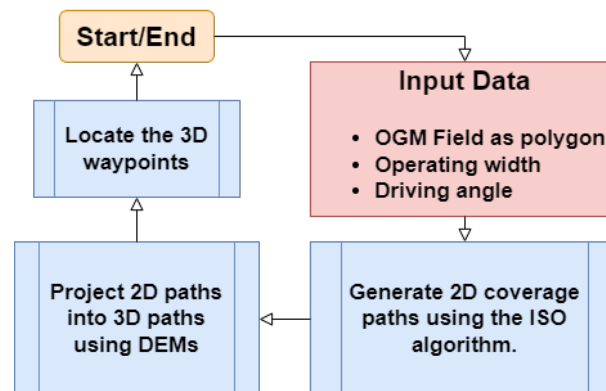


Figure 3. The 3D local goals extraction flowchart.

2.3. Trajectory Generation and Optimization

Trajectory generation algorithms are core problems in UAV control which allow them to attain their full autonomy. They can be categorized mainly into three classes. The algorithms in the first class can be regarded as primarily geometric. The process of planning a trajectory comprises a geometric path generation, first. Thereafter, the path is time parameterized while satisfying the UAV dynamic constraints. In this case, path primitives like lines [26] polynomials [27] and splines [28] are considered. In the second class, the algorithms are used for planning trajectories while minimizing a certain derivative of the UAV position (or combinations thereof). The differential flatness property of the UAV is used in such a way the trajectory feasibility depends on the derivatives. Here, minimum snap trajectory generation [29], weighted sum of derivative minimization [30], maneuver duration minimization [31] can be listed. Model predictive algorithms fall under this class. For instance in [32], a Learning Based Model Predictive Control (LBMPC) is implemented with a priori unknown trajectory. The algorithm showed its robustness, safety and convergence while used for catching a thrown ball. In the third class, the algorithms are used to solve an optimal control problem to generate numerical trajectories based on the full vehicle system dynamics. In [33], the Minimum Pontryagin's Principle is used to compute the UAV maneuvers. The minimum principle showed that the time-optimal trajectories are bang-bang in the thrust command. However, in the rotational rate control, they are bang-singular. In [34], numerical optimal control is used in order to solve a wide range of problems related to UAV systems.

2.3.1. Velocity and Acceleration Constraints

Trajectory generation must not only describe the desired trajectory accurately, but must also have smooth kinematics profiles for increasing the precision and the durability of the system, maintaining higher tracking accuracy while avoiding exciting natural modes of the mechanical structure or servo control system [35]. To do so, the trajectory must satisfy certain constraints such as the vehicle's physical limits, safety regulations and sensor specifications. In this work, we adopt a method of decoupling these limits into constraints on the trajectory's velocity and acceleration. For the way-points at the corners, the velocities should be slowed down to zero, but the vehicle should pass through the

way-points between the corners at a constant velocity. However, the acceleration should always be brought down to zero at every way-point.

2.3.2. LQR Optimization Trajectory Generation

The approach used in this work falls under the second class. An LQR algorithm is applied for generating the position, velocity, and acceleration reference trajectories. As mentioned in Section 1, we are inspired by the work of Sanyal in [18]. The main differences between our approach and Sanyal's are:

1. Our approach uses the snap of the quadrotor as the input control instead of the crackle. We believe that going up to the snap rather than the crackle assures to obtain trajectories with less curvatures. Besides, this would guarantee both continuity and smoothness of the reference trajectories at the way-points.
2. In our approach, the output vector is not constructed only from the position and the acceleration at the way-points only. The vehicle velocity is added. The aim is to (1) constrain the velocity at the way-points and (2) respect flying in corridors.
3. Our approach assumes an online generation of the quadrotor attitude with the use of a pointing direction and the IMU vector measurements. Hence, the proposed Kalman method in [18] is avoided.

The system states $x(t)$ are given by:

$$x(t) = [b(t) \quad \dot{b}(t) \quad \ddot{b}(t) \quad \ddot{\ddot{b}}(t)]^T \quad (2)$$

where $b(t)$ is the position of the quadrotor in the 3D environment at instant t . The output $y(t)$ is constructed such that the desired output y_i at the way-points is given by:

$$y(t) = [b_i \quad v_i \quad a_i]^T \quad (3)$$

where b_i , v_i and a_i are the position, the velocity and the acceleration constraints of the vehicle at the way-points, respectively.

The cost function for the position trajectory that is to be minimized is expressed as

$$J = \sum_{i=1}^N (y(t_i) - y_i)^T S (y(t_i) - y_i) + \int_{t_0}^{t_N} \frac{1}{2} (x^T Q x + u^T R u) dt \quad (4)$$

Subjected to the constraint equations:

$$\begin{cases} \dot{x}(t) = Ax + Bu \\ y(t) = Cx(t) \end{cases} \quad (5)$$

where $A = \begin{bmatrix} 0_{9 \times 3} & I_{9 \times 9} \\ 0_{3 \times 3} & 0_{3 \times 9} \end{bmatrix}$, $B = \begin{bmatrix} 0_{9 \times 3} \\ I_{3 \times 3} \end{bmatrix}$, $C = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}$. $I_{n \times n}$ is the identity matrix of dimension n , t_0 is the starting time, t_N is the final time and the control input u is the snap. Let the Hamiltonian H be expressed as

$$H = \frac{1}{2} x^T Q x + \frac{1}{2} u^T R u + \lambda^T (Ax + Bu) \quad (6)$$

The conditions of optimality expressed in H are given below:

$$\begin{cases} \dot{x}(t) = \frac{\partial H}{\partial \lambda} = Ax + Bu \\ \dot{\lambda}(t) = -\frac{\partial H}{\partial x} = -Qx - A^T \lambda \\ 0 = \frac{\partial H}{\partial u} = Ru + B^T \lambda \\ 0 = \frac{1}{2} \frac{\partial (y(t_i) - y_i)^T S (y(t_i) - y_i)}{\partial x(t_i)} + \lambda(t_i^+) - \lambda(t_i^-) \end{cases} \quad (7)$$

where t_i^+ is the time instant t_i when approached from times $t > t_i$ and t_i^- is the time instant t_i when approached from times $t < t_i$. Putting $\lambda(t) = P(t)x(t) + \eta(t)$ then the governing equations for minimizing the cost J are:

$$\dot{P} = -PA - A^T P - Q + PBR^{-1}B^T P \quad (8)$$

$$\dot{\eta} = (-A^T + PBR^{-1}B^T P)\eta \quad (9)$$

$$\dot{x} = (A^T + BR^{-1}B^T P)x - BR^{-1}B^T \eta \quad (10)$$

$$u = -R^{-1}B^T(Px + \eta) \quad (11)$$

While the boundary condition $\forall i \in W/\{N\}$ is

$$P(t_i^-) = P(t_i^+) + C^T S C \quad (12)$$

$$\eta(t_i^-) = P(t_i^+) - C^T S y_i \quad (13)$$

At $i = N$, $P(t_N) = C^T S C$ and $\eta(t_N) = -C^T S y_i$. The Equations (8) and (9) are solved backward in time starting at $t = t_N$ and updating the boundary conditions at every instant $t = t_i, i < N$ as in Equations (12) and (13). The solutions obtained can be replaced in Equations (10) and (11) to get the vector state $x(t)$ and $u(t)$ at every instant $t \in [t_0, t_N]$, respectively.

2.3.3. Online Trajectory Re-planning

To cope with unmodeled obstacles, a lower trajectory re-planner module is required. The aim of this module is to run in several milliseconds to ensure the safety of quadrotor and keep it close to the global path while simultaneously avoiding unpredicted obstacles. The existing methods for quadrotor online trajectory re-planning can be mostly categorized as polynomials methods, sampling-based methods, and optimization methods [36]. Polynomials and splines are mostly suitable for re-planning since they are computationally efficient. However, they cannot provide time optimal policies, since polynomials are smooth by their own nature. Sampling-based methods such as RRT (Rapidly-exploring Random Tree) construct road maps from sampling free space and searching a graph map space. This can be achieved by computing edge cost and heuristic functions [37]. Such methods are inherently non-smooth and often do not satisfy the quadrotor's dynamics constraints. Trajectory optimization methods take use of two types of cost functions: a smoothing function and a collision avoidance function. The sum of the two cost functions can be minimized then using optimizations approaches such as gradient-descent, Gauss-Newton by computing the gradient or the Jacobian of each cost function [38]. This method can be useful when dealing with cluttered environments. However, it suffers from the local minimum problem.

In our case, replanning is performed using an online Improved APF method [39]. This is widely used because its model is simple and elegant, and it is applicable for real-time implementation. Firstly, an attractive potential field is constructed at the target using the following expression:

$$U_{att}(P) = \frac{1}{2} k_{att} \cdot d^2(P, P_g) \quad (14)$$

where $d(P, P_g) = P_g - P$ is the Euclidean distance between the quadrotor's position and the target position. k_{att} is the attractive potential field constant.

The attractive force of the quadrotor in the attractive potential field is the negative gradient of U_{att} :

$$F_{att}(P) = -\nabla U_{att}(P) = k_{att} \cdot d(P, P_g) \quad (15)$$

Secondly, an improved repulsive potential field is constructed around the obstacles and can be defined as:

$$U_{rep}(P) = \begin{cases} \frac{1}{2}(\frac{1}{d(P,P_o)} - \frac{1}{d_o})^2 d^n(P, P_g) & d(P, P_o) \leq d_o \\ 0 & d(P, P_o) > d_o \end{cases} \quad (16)$$

where P_o is the position of the obstacle, d_o is influence range of the repulsive potential field and n is an arbitrary real number which is greater than zero (for our case $n = 2$).

The repulsive potential field is called improved since the distance correction factor $d^n(P, P_g)$ is added to the conventional repulsive potential field known in literature. This creates a balance between the two kind of forces especially in case where a rapid increase in the repulsive force occurs. In this manner, the repulsive force can be decreased gradually when the quadrotor is adjacent to the target. Most important of all, this force ensures that the overall potential field at the target is the global minimum. Thus, the improved repulsive force function is obtained as follows

$$F_{rep}(P) = -\nabla U_{rep}(P) \begin{cases} F_{rep1}(P) + F_{rep2}(P) & d(P, P_o) \leq d_o \\ 0 & d(P, P_o) > d_o \end{cases} \quad (17)$$

where F_{rep1} and F_{rep2} are expressed as

$$F_{rep1}(P) = k_{rep}(\frac{1}{d(P, P_o)} - \frac{1}{d_o}) \frac{d^n(P, P_g)}{d^2(P, P_o)} \quad (18)$$

$$F_{rep2}(P) = \frac{n}{2} k_{rep}(\frac{1}{d(P, P_o)} - \frac{1}{d_o})^2 d^{n-1}(P, P_g) \quad (19)$$

The quadrotor moves toward the target in the joint action of the resultant force model (See Figure 4).

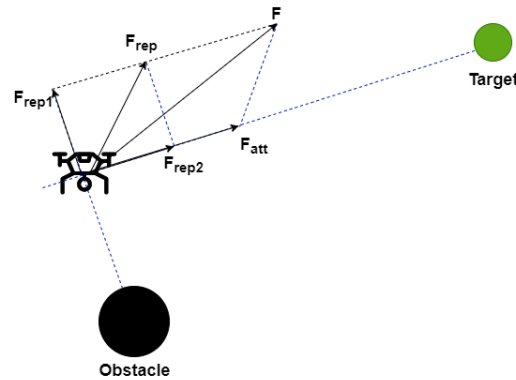


Figure 4. The improved resultant for model applied on the quadrotor.

Figure 5 depicts a flowchart that explains the whole re-planning process. P_c refers to the current position of the quadrotor.

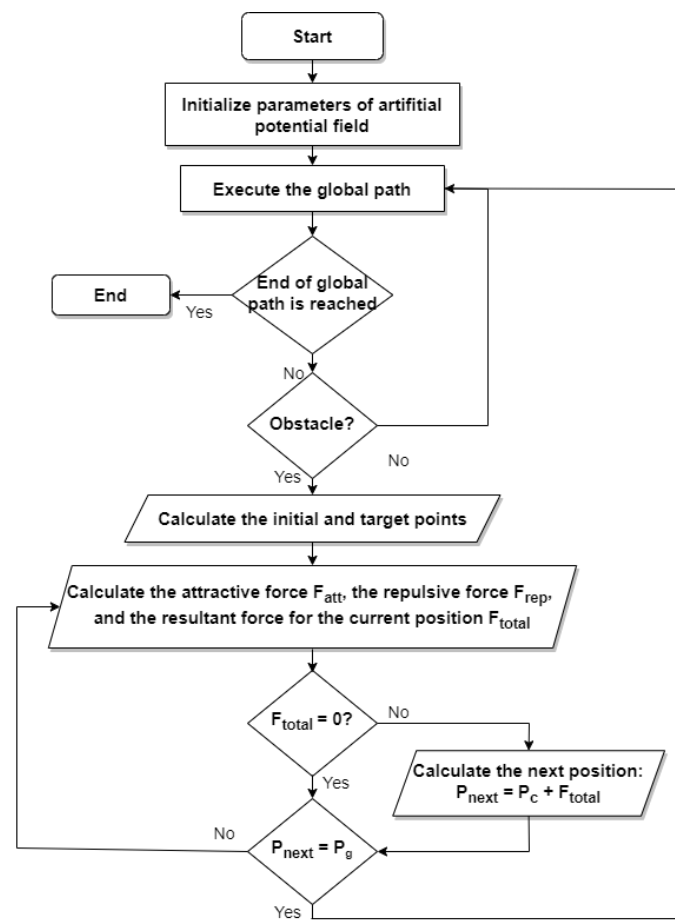


Figure 5. The flowchart of the proposed trajectory re-planning method based on the improved APF.

2.4. Control and Trajectory Tracking

Once the optimal trajectory is generated as previously described, it is necessary to design a control strategy to track the reference trajectory. Many research studies have been conducted to construct such strategies for UAVs. Linear control systems such as Proportional-Derivative controller or LQRs are widely used to improve the equilibrium stability properties [40,41]. Nonlinear trajectory tracking techniques like backstepping [42] and sliding mode [43] are developed. However, most of these strategies exhibit singularities when dealing with complex rotational maneuvers. To avoid these singularities, geometric controllers are applied on dynamic systems evolving on nonlinear manifolds that are not globally identified with Euclidean spaces [19,44]. In this work, a geometric controller strategy, that uses the quadrotor dynamics expressed globally on the Special Euclidean $SE(3)$ configuration manifold, is constructed to track predefined trajectories.

2.4.1. Quadrotor Dynamics Model

A proper quadrotor mathematical model must be defined for designing a control system. In order to derive such a model, a set of coordinates systems for specifying the position, velocity, forces, and moments acting on the vehicle must be defined. Let the inertial frame be the surface of the earth and the body frame be fixed on the quadrotor rigid body as shown in Figure 6. $\{\vec{b}_1, \vec{b}_2, \vec{b}_3\}$ are the unit vectors along the body-fixed frame axis. $\{\vec{a}_1, \vec{a}_2, \vec{a}_3\}$ are the unit vectors along the inertial frame axis.

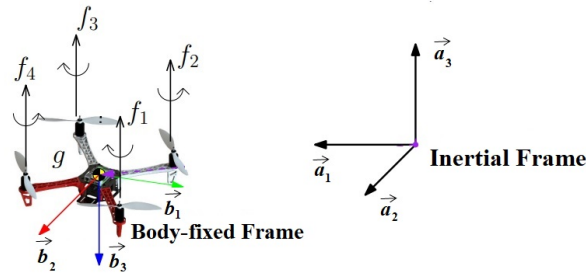


Figure 6. Quadrotor inertial and body-fixed frames.

Using the Newton-Euler equations, the equation of motion of a quadrotor is derived. The model obtained is sufficiently reliable for simulating and controlling the vehicle behavior. It is given by

$$\begin{cases} \dot{b} = v \\ m\dot{v} = fRe_3 - mge_3 \\ \dot{R} = R[\Omega]_{\times} \\ J\dot{\Omega} = -\Omega \times J\Omega + \tau \end{cases} \quad (20)$$

where $b \in \mathbb{R}^3$ denote the position, $v \in \mathbb{R}^3$ is the translational velocity in the inertial frame, m is the mass, f is the thrust magnitude, $e_3 = [0, 0, 1]^T$, g is the gravity, J is the moment of inertia, Ω is the angular velocity expressed in the body-fixed frame and τ is the external torque applied on the vehicle.

2.4.2. Geometric Control Using Measurement Vectors

As stated in Section 1, we are inspired by the work in [19] to design a geometric controller for tracking the generated trajectory. However instead of using a heading direction and a thrust direction, inertial measurement vectors that are provided by a sensor are used to construct the vehicle attitude. In this way, new definitions of the errors between the real vectors and the desired ones are used (See Equations (28)–(30)). These errors are injected directly in the control law. Besides, the notion of a pointing direction is introduced instead of the heading direction. The controller structure is shown in Figure 7.

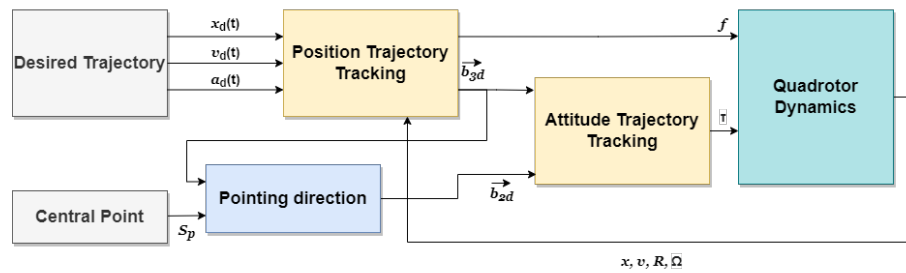


Figure 7. The structure of the geometric controller.

The quadrotor translational dynamics is controlled using the total thrust fRe_3 . The magnitude of the total thrust f is directly controlled and its direction Re_3 is along the third body-fixed axis b_3 . Hence, to obtain stabilized translational motion along a desired trajectory, the total thrust f and a desired direction of b_{3d} are selected. A direction is required to complete the degrees of freedom of the desired attitude $R_d \in SO(3)$. Thus, a pointing direction vector s^c , which has to be corrected online using the desired direction b_{3d} , is chosen. The desired attitude is then obtained as $R_d = [b_{2d} \times b_{3d}, b_{2d}, b_{3d}]$ where $b_{2d} = s^c$. This desired attitude is followed by the control moment τ .

Section 2.3.2 results in the optimal desired position trajectories $x_d(t)$. The pointing directions at every instant $t \in [t_0, t_f]$ are

$$s(t) = \frac{S_p - x_d(t)}{\|S_p - x_d(t)\|} \quad \forall t \in [t_0, t_f] \quad (21)$$

The pointing directions can be corrected using the vector $b_{3d}(t)$ as shown in the expression below

$$s^c(t) = \frac{s(t) - (s(t)^T b_{3d}(t)) b_{3d}(t)}{\|s(t) - (s(t)^T b_{3d}(t)) b_{3d}(t)\|} \quad (22)$$

where $s^c(t) = b_{2d}(t)$, and $s^c(t) \perp b_{3d}(t)$. The vector $b_{3d}(t)$ is expressed as

$$b_{3d}(t) = \frac{-k_x e_x - k_v e_v + m(g e_3 + a(t))}{\|-k_x e_x - k_v e_v + m(g e_3 + a(t))\|} \quad (23)$$

where k_x, k_v are some positive constants, e_x and e_v are the tracking errors for the position $x(t)$ and the velocity $v(t)$, respectively. They are expressed as

$$e_x = x - x_d \quad (24)$$

$$e_v = v - v_d \quad (25)$$

The vector $b_{1d}(t)$ can be produced as

$$b_{1d}(t) = b_{2d}(t) \times b_{3d}(t) \quad (26)$$

We assume that the quadrotor's attitude is unknown for measurements (unavailable for feedback). We assume also that it is equipped with sensors that provide unfiltered vector measurements (in the body-fixed frame). Hence, the only variables available are the vector measurements which are denoted by b_i . We propose the following control law

$$\tau = \Omega \times J\Omega + J\dot{\Omega}_d + Jz_\rho - J e_{\Omega 1} - J e_{\Omega 2} \quad (27)$$

where z_ρ , $e_{\Omega 1}$ and $e_{\Omega 2}$ are the errors and given by

$$z_\rho = \sum_{i=1}^k \rho_i S(b_{id}) b_i \quad (28)$$

where $\rho_i > 0$ and k is the number of measured vectors in the body frame ($k = 2$ in our case). $S(\cdot)$ is the hat mapping transforms an vector to a skew-symmetric matrix: $S(\cdot) : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$.

$$e_{\Omega 1} = \Omega - \Omega_d \quad (29)$$

$$e_{\Omega 2} = \Omega_d \times \Omega \quad (30)$$

The desired angular velocity is given by

$$\Omega_d = Vex([b_{1d}, b_{2d}, b_{3d}]^T [\dot{b}_{1d}, \dot{b}_{2d}, \dot{b}_{3d}]) \quad (31)$$

with $Vex(\cdot) : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$.

The proposed control law τ guarantees Almost Global Asymptotic Stability (AGAS) of the body attitude and angular velocity to their desired values. The proof is shown in the subsection below.

2.4.3. Stability Analysis of the Proposed Control Law

The lemma in [45] is the key to the proof of almost asymptotic stability of the control law. Using the lemma, we can write

$$z_\rho \equiv \sum_{i=1}^k \rho_i S(b_{id}) b_i = -2R_d^T (q_0^e I - S(q^e)) W_\rho q^e, \quad (32)$$

where $W_\rho = -\sum_{i=1}^k \rho_i S(r_i)^2$ is a symmetric positive definite matrix according to the lemma. The tracking error $R^e = R R_d^T$ of the attitude corresponds to the unit quaternion errors

$$Q^e = Q \odot Q_d^{-1} \equiv (q_0^e, q^e) \quad (33)$$

where $Q \in \mathbb{S}^3 = \{Q = (q_0, q) \in \mathbb{R} \times \mathbb{R}^3 | q_0^2 + q^T q = 1\}$. Given the attitude dynamics in (20) and the control law (27), if we consider $\bar{\omega} = R_d e_{\Omega 1}$, where $e_{\Omega 1}$ is given in Equation (29), the closed loop dynamics are obtained as

$$\dot{Q}^e = \begin{bmatrix} \dot{q}_0^e \\ \dot{q}^e \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}(q^e)^T \bar{\omega} \\ \frac{1}{2}(q_0^e I + S(q^e)) \bar{\omega} \end{bmatrix} \quad (34)$$

$$\dot{\bar{\omega}} = -\alpha \bar{\omega} - 2(q_0^e I - S(q^e)) W_\rho q^e \quad (35)$$

Note that these dynamics are autonomous. Defining $x = (Q^e, \bar{\omega})$ in the state space $\mathcal{X} := \mathbb{S}^3 \times \mathbb{R}^3$, the above dynamics can be written in the form

$$\dot{x} = f(x) \quad (36)$$

We can also note that (35) can be written as

$$\dot{\bar{\omega}} = -\alpha \bar{\omega} + R_d z_\rho \quad (37)$$

Theorem 1. Consider the rigid body dynamics (20) with the control law (27) resulting in the closed loop attitude dynamics given by (34) and (35). Then, under assumptions of the lemma, the trajectories of (34) and (35) converges to the following subsets $\mathbb{S}^3 \times \mathbb{R}^3$, given by $\Theta_1 = (\pm 1, 0, 0)$ and $\Theta_2 = \{(0, \pm v_1, 0), (0, \pm v_2, 0), (0, \pm v_3, 0)\}$, where v_i ($i = 1, 2, 3$) are unit eigenvectors of W_ρ .

- The equilibrium set Θ_1 is asymptotically stable with the domain of attraction containing $\Phi = \{x := (Q^e, \bar{\omega}) \in \mathbb{S}^3 \times \mathbb{R}^3 | x^T P x < c\}$ with $P = \text{diag}(0, 2W_\rho, \frac{1}{2})$ and $c < 2\lambda_{\min}(W_\rho)$ and $\lambda_{\min}(\cdot)$ is the smallest eigenvalue of (\cdot)
- The equilibria defined by the set Θ_2 are unstable and Θ_1 is almost globally asymptotically stable.

Proof. Let us propose the Lyapunov function candidate

$$V = 2(q^e)^T W_\rho q^e + \frac{1}{2} \bar{\omega}^T \bar{\omega} \quad (38)$$

With W_ρ symmetric positive definite. The time derivative of Equation (38) in view of (34) and (35) is expressed as

$$\dot{V} = 2(q^e)^T W_\rho (q_0^e I + S(q^e)) \bar{\omega} + \bar{\omega}^T (-\alpha \bar{\omega} - 2(q_0^e I - S(q^e)) W_\rho q^e) \quad (39)$$

$$\dot{V} = 2\bar{\omega}^T (q_0^e I - S(q^e)) W_\rho q^e + \bar{\omega}^T (-\alpha \bar{\omega} - 2(q_0^e I - S(q^e)) W_\rho q^e) \quad (40)$$

After simplification, we obtain

$$\dot{V} = -\alpha \bar{\omega}^T \bar{\omega} \quad (41)$$

It can be noticed in (41) that $\dot{V} \leq 0$. In addition to the same equation, we have $\dot{V} = 0$ only $\bar{\omega} = 0$. Using the Equation (37) and $\bar{\omega} = 0$, we get $z_\rho = 0$. According to the lemma,

this leads to the equilibrium sets ($q_0^e = \pm 1, q^e = 0, \tilde{\omega} = 0$) or ($q_0^e = 0, q^e = \pm v_i, \tilde{\omega} = 0$) which corresponds to Θ_1 and Θ_2 , respectively. It remains to show that Θ_2 is unstable to complete the proof.

Let us define $\delta \equiv (q^e)^T \tilde{\omega}$, and consider the dynamics of q_0^e and δ around ($q_0^e = 0, \tilde{\omega} = 0$) which corresponds to ($q_0^e = 0, \delta = 0$)

$$\dot{q}_0^e = -\frac{1}{2}\delta \quad (42)$$

$$\dot{\delta} = -\alpha\delta - 2\eta q_0^e \quad (43)$$

where η is an eigenvalue of W_ρ . Hence, the system expressed by (42) and (43) can be written as follows

$$\dot{X} = \begin{bmatrix} 0 & -\frac{1}{2} \\ -\alpha & -2\eta \end{bmatrix} X = AX \quad (44)$$

With $X = [q_0^e \quad \delta]^T$ and $A = \begin{bmatrix} 0 & -\frac{1}{2} \\ -\alpha & -2\eta \end{bmatrix}$. The two eigenvalues of A are real and of opposite sign. The characteristic equation of A is given by $P(\lambda) = \lambda^2 + 2\eta\lambda - \frac{\alpha}{2}$. Hence, the roots are $\lambda_{1,2} = -\eta \pm \sqrt{\eta^2 + \frac{\alpha}{2}}$. Since the equilibrium $(q_0^e, \delta) = (0, 0)$ is unstable, it can be concluded that Θ_2 is unstable.

Given the proofs above and according to the Krasowsky-La Salle theorem, the equilibrium is almost globally asymptotically stable in this case. Hence, the proposed control law (27) guarantees AGAS. \square

For the translational dynamics stability analysis, we can refer to [19].

3. Results

In this section, we present simulation results to validate the effectiveness of the proposed method. All simulation experiments were performed by Matlab on an HP laptop with a 2.90 GHz Intel® i7-10700 CPU and 32 Gb of RAM. The simulation have been carried out using a quadrotor with a mass of $m = 4.34$ kg, and a moment of inertia of $J = \text{diag}(0.820, 0.0845, 0.1377)$ kg m². The quadrotor started at $t_0 = 0$ s at rest from a given position b_0 i.e. $x(t_0) = [b_0, 0_{3 \times 1}, 0_{3 \times 1}, 0_{3 \times 1}]^T$ with an initial attitude of simply I_3 . Therefore, the vehicle's initial pose was given by

$$g_0 = \begin{bmatrix} I_3 & b_0 \\ 0 & 1 \end{bmatrix} \quad (45)$$

3.1. In-Field Environment Setup

Vineyards terrains were selected to be our experimental environments. As stated before, those terrains are known by their steep slopes and limited room for maneuvering. We wanted the quadrotor to fly at relatively constant altitude between the vines and perform remote sensing. Since aerial imagery data was not available to build both real-life OGMs and DEMs, inflated artificial (generated) maps were used. The maps were hand-drawn using black and white colors, then converted to binary occupancy maps using Matlab. Two vineyard terrain scenarios were considered (See Figure 8). The first is a flat rectangle. The second is a tilted irregular hexagon.

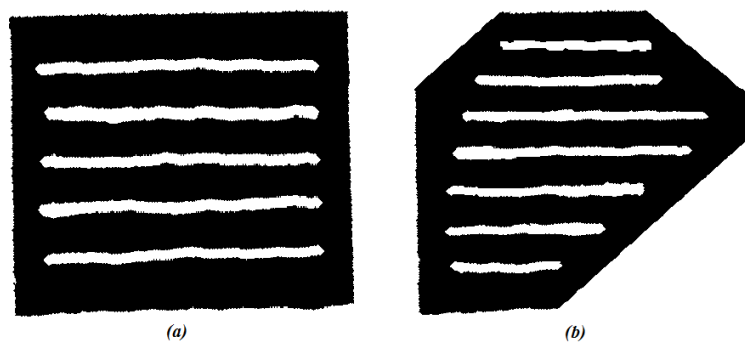


Figure 8. The vineyards scenarios in 2D: (a) flat rectangle, (b). tilted hexagon.

3.2. Coverage Path Finding

The binary occupancy maps above were fed to Matlab where the ISOA coverage path planning algorithm was run. As a result, a back and forth boustrophedon path was generated and way-points were extracted as explained in Section 2.2, as shown in Figure 9.

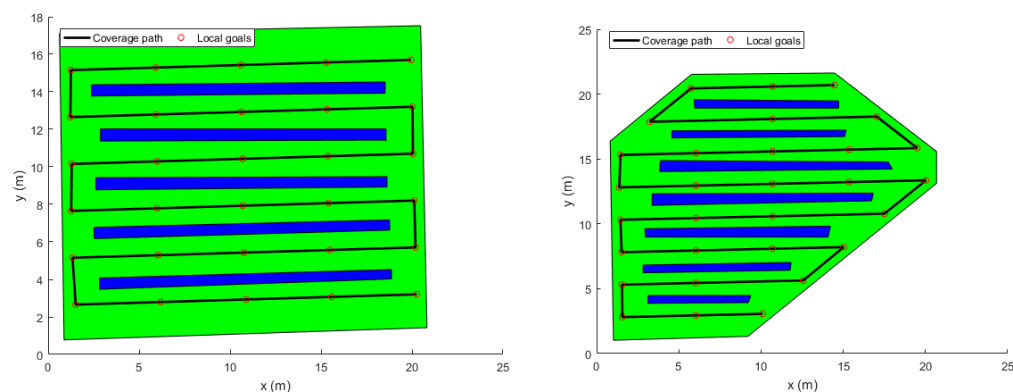


Figure 9. The generated coverage paths and way-points.

3.3. Trajectory Generation

The extracted way-points were not fed directly to the trajectory generator since they contained geometric information only. Velocity and acceleration constraints were added to the way-points to keep the quadrotor flying in a corridor between the vines. To do so, a velocity of $v = 1$ m/s was chosen. Besides, the time between the way-points was set to be 4 s. This information was used by the LQR trajectory generator. The resulting position trajectories of both fields, maneuvering around the way-points, are shown in Figures 10 and 11.

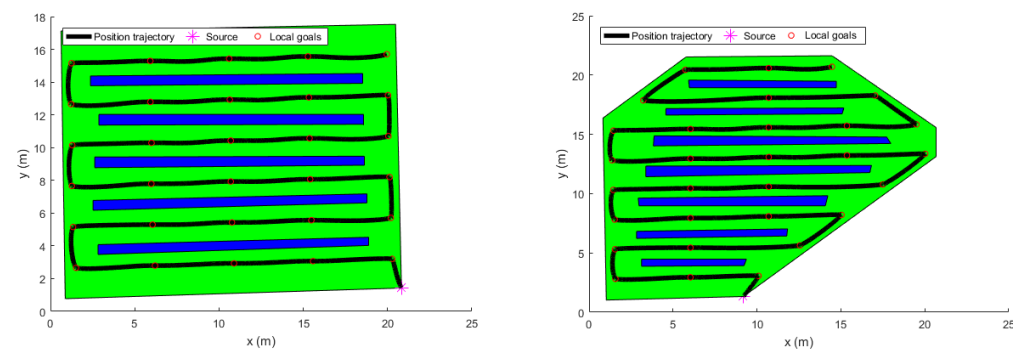


Figure 10. The generated optimal trajectories (2D View).

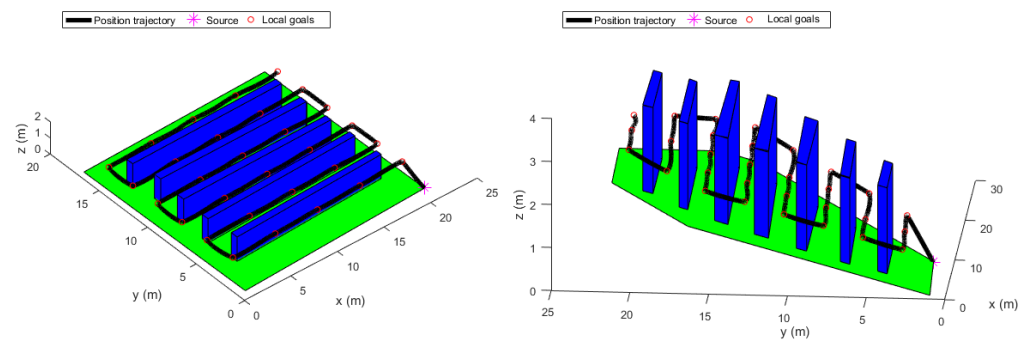


Figure 11. The generated optimal trajectories (3D View).

3.4. Trajectory Tracking

In order to test the proposed trajectory tracking strategy, the flight position, velocity and acceleration information generated from the previous step were fed to the trajectory tracker. The pointing direction was chosen to be as expressed (21). Without loss of generality, S_p was taken as the polygon centroid.

The control law in (27) contains the measurement vectors b_i and Ω . These vectors were considered to be noisy. A zero mean white noise with variance of 5×10^{-4} was added. The resulting noisy measurement vectors are shown in Figures 12 and 13. Figures 14 and 15 show the tracking results of the generated flight position trajectories. Figures 16–18 depict both the pointing and the thrust directions during the whole flights.

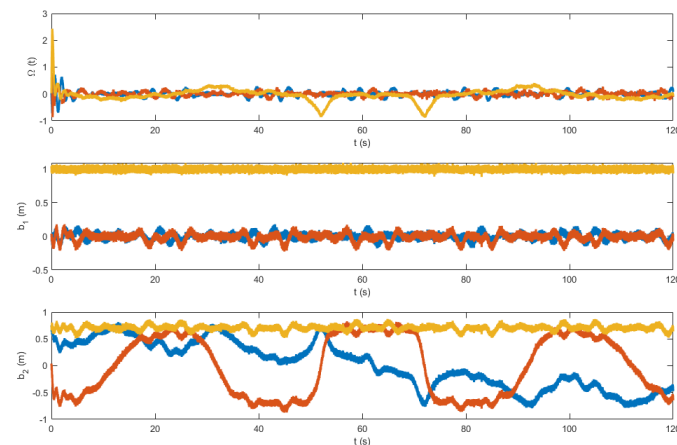


Figure 12. The vector measurements of the flight in terrain 1.

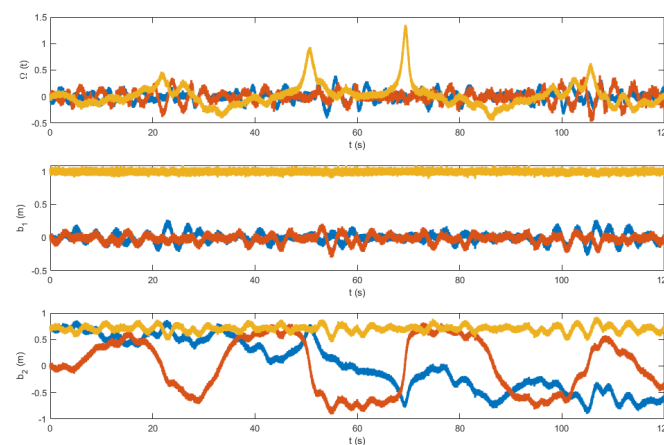


Figure 13. The vector measurements of the flight in terrain 2.

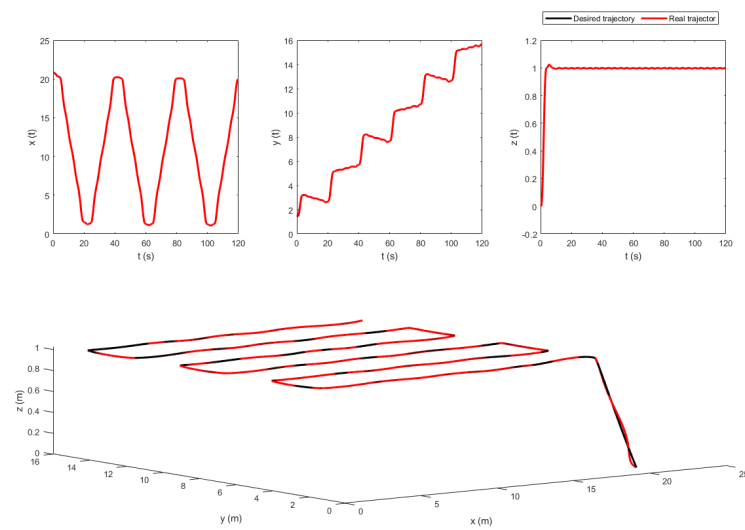


Figure 14. The tracking results of the flight in terrain 1.

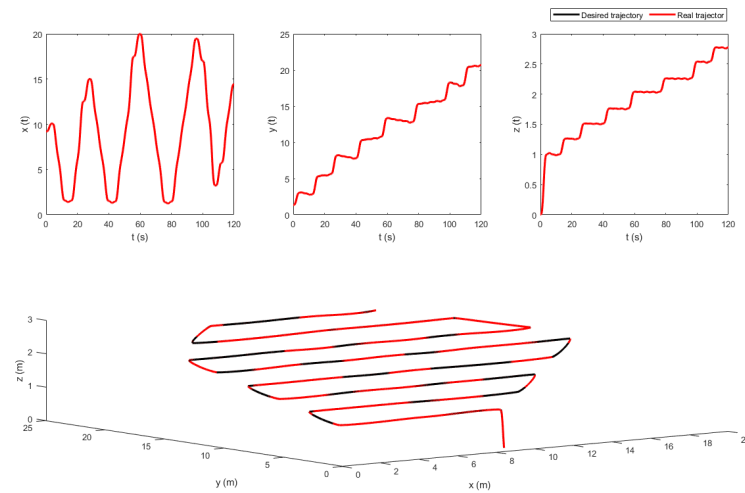


Figure 15. The tracking results of the flight in terrain 2.

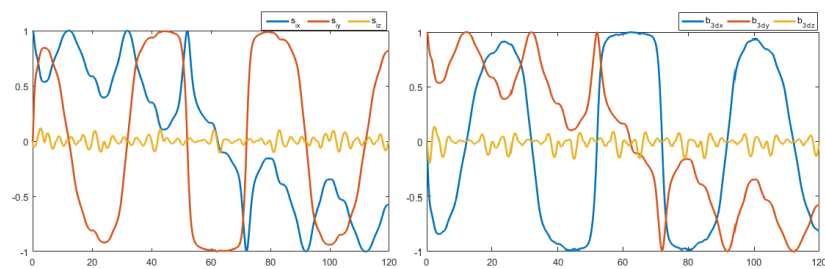


Figure 16. The pointing and thrust directions of the flight in terrain 1.

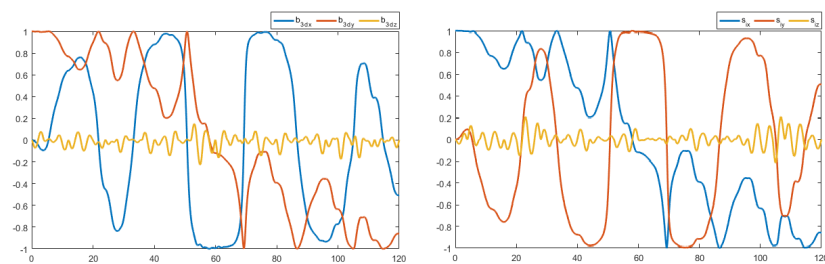


Figure 17. The pointing and thrust directions of the flight in terrain 2.

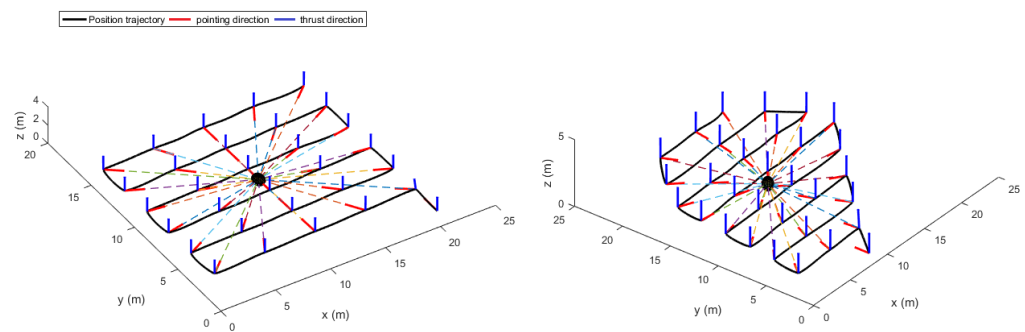


Figure 18. The pointing and the thrust direction vectors.

3.5. Trajectory Replanning

To evaluate the performance of the improved APF re-planning method, the first terrain was selected. Firstly, the quadrotor was assumed to be a particle. Secondly for sake of simplicity, we assumed no uncertainties in the simulated measured ranges between the vehicle and the obstacles. Besides, we supposed that the obstacles were in form of cylinders of equal sizes, located in the free environment (between the vines) and fixed at constant coordinates. The influence range of a single obstacle was fixed at $d_{det} = 2.1$ m. Figure 19 shows the trajectory generated by the proposed method.

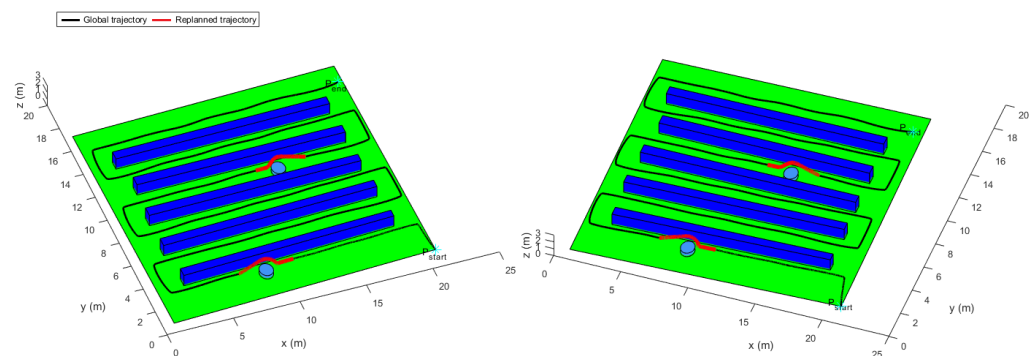


Figure 19. The global position and the replanned trajectories.

4. Discussion

In this paper, a complete solution for the GNC problem is proposed, which can provide stable and efficient navigation service for UAV autonomous flight mission in agricultural fields. As anticipated, the solution is based on a combination of existing algorithms. In path finding, Occupancy Grid Maps (OGMs) that maintain the environment information obtained from the terrain mapping process are adopted. Based on these maps, collision-free geometric coverage paths are generated according to the Iterative Structured Orientation algorithm (ISOA). In trajectory optimization, this paper proposes a Linear Quadratic Regulator (LQR) to generate minimum snap trajectories that satisfy position, velocity and acceleration constraints at the extracted local goals. In trajectory re-planning, an Improved Artificial Potential Field (APF) algorithm is used to enhance the safety of the trajectories when unknown obstacles are detected. In the trajectory tracking, a control law, which generates the attitude trajectories online using raw vector measurements, is implemented. The control law is used also to track the generated position and attitude trajectories while simultaneously pointing towards predefined direction along the whole flight mission.

In order to verify the effectiveness of the proposed method, a series of simulation experiments is conducted (See Section 3). As it can be seen in Figure 9, the ISOA generates efficiently complete optimal paths from a starting location to a final location. The back-and-forth generated path contains sharp turns which may contradict with the dynamics of the quadrotor. Hence, smoothing it represents a crucial step before sending any command to

the vehicle. However, the main advantage of such algorithm is that it makes use of OGMs. These offer a powerful technique for representing the agricultural field environments and can be build from off-the-shelf sensors.

Figure 10 shows the position trajectory obtained from the LQR trajectory generator. It can clearly be seen that the continuous and smooth position trajectory maneuvers around the extracted waypoints successfully. This is achieved thanks to the LQR weighting diagonal terms S , Q and R . The choice of these terms is essential and plays a significant role in smoothing the trajectory features. However, choosing appropriate weighting terms is a challenging task.

The generated flight trajectories were tracked and analyzed. After flying 120 s, the quadrotor succeeded to track the desired trajectories despite the presence of white noise in the IMU. The comparison between the actual position/attitude trajectory (blue) and the desired position/attitude trajectory (red) is depicted in Figures 14 and 15, showing that the two trajectories are identical, hence highlighting the effectiveness of the proposed control scheme to properly track the desired trajectories while fulfilling the pointing direction constraints not only at the local goals but at every position on the trajectory, as illustrated in Figure 18. In addition, a quantitative analysis of the two position trajectories along the x -, y - and z - axes is shown in Figures 20 and 21 below. The absolute value of the maximum error, after stability is reached, can be calculated to be: $\sqrt{(0.0010)^2 + (-0.0026)^2 + (-0.0001)^2} = 0.0028\text{m}$ for the first terrain and $\sqrt{(0.0018)^2 + (-0.0037)^2 + (-0.0003)^2} = 0.0041\text{m}$ for the second one. As it can be noticed, the errors are in the order of 10^{-3}m . Such errors are suitable relative to the size of the terrains. Hence, this data validates the efficiency of the proposed tracking algorithm.

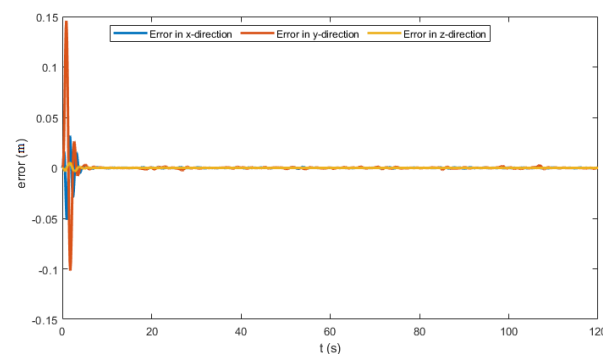


Figure 20. The position error profile for terrain 1.

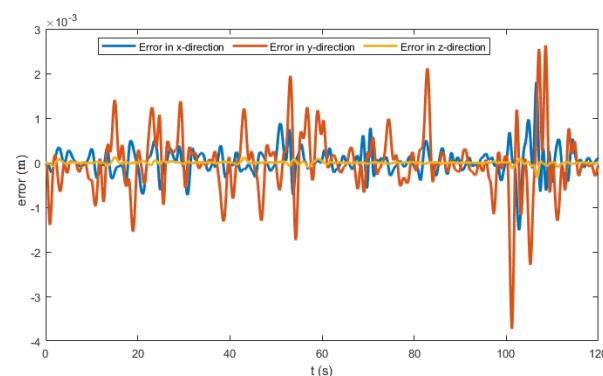


Figure 21. The position error profile for terrain 2.

Figure 19 shows the selected vineyard terrain where static obstacles were added between the vine rows to obstruct the movement of the quadrotor in its global trajectory. The figure illustrates the quadrotor's avoidance trajectory near the static obstacles. Note that the blue trajectory represents the desired re-planned trajectory obtained from the

improved APF algorithm and the red trajectory represents the global trajectory obtained from the LQR trajectory generator. At every time step, the trajectory re-planner takes the position information of the global trajectory. Once the obstacle is sensed, it takes the position computed by the improved APF. It can be observed that the obstacle avoidance was performed vertically. This would prevent collision with the vine rows if the avoidance was done horizontally. In addition, it can be noticed that the relative distance maneuver remains greater than collision threshold that represents a physical collision. Above all, one can say that the quadrotor completes the avoidance mission safely.

However, although the combination of the algorithms has offered a complete and efficient solution for navigation in agricultural terrains, there are some issues that need to be improved. Since the selected algorithms are used in phases sequentially except the trajectory re-planning phase (See Figure 1), the output of a particular phase depends on the output of the previous one. Unreasonable outputs in a particular phase lead to intractable outputs in the next one. Consequently, the quality of the flight trajectories will be poor. In addition, the solution computational time depends on the dimensions of the mapped terrains. The more expansive the terrains are, the larger the computational time.

5. Conclusions

In this paper, an efficient guidance, navigation and control method for UAVs to complete 3D coverage flight missions in mapped agricultural fields is proposed. The method is mainly improved in four modules: coverage path searching, trajectory generation, trajectory re-planning, and trajectory tracking. In the coverage path searching module, a safe geometric coverage guiding path is generated on the artificial occupancy grid maps using the Iterative Structured Orientation method. The generated coverage path is optimized and smoothed using the trajectory generation module. The Linear Quadratic Regulation approach uses predefined velocity and acceleration constrained way-points from the former path to generate global minimum snap position trajectory with appropriate smoothness. In the trajectory re-planning module, the UAV modifies its global trajectory and the real-time local trajectory is generated to avoid the unpredicted obstacles. This can be achieved by the use of the Improved Artificial Potential Field method. The last module builds the geometric tracking controller on the Special Euclidean group. The aim of this module is tracking the predefined trajectory while simultaneously pointing toward the predetermined direction using the vector measurement provided by the vehicle inertial unit. Several numerical simulation results demonstrated the validity of the proposed GNC method. The presented approach offers an advantage to propose an innovative solution in the direction of autonomous navigation in agricultural fields since it is designed such that it complies with the peculiar features of the fields and the UAV. The approach also provides suitable performances in terms of optimal and efficient generation of trajectories and optimal tracking abilities for fully autonomous UAV. However, susceptibility to weather conditions like rain, fog and dust and maneuverability in winds and turbulence represent the main limitations of the approach. This latter becomes more challenging in large and unstructured agricultural environments where the vehicle localization and endurance issues are inherited.

Our future work will focus on extending the the proposed method to deal with external disturbances and to adapt to dynamic environments. Besides, although we carried out an extensive effectiveness evaluation of the method, the experiments were only carried out in simulation. Hence, the implementation and experimental validation of the method as presented herein is an important step towards this end.

Author Contributions: Conceptualization, A.M.; Formal analysis, A.M., A.B., A.C.-B., A.E.H. and B.C.; Funding acquisition, A.B. and A.C.-B.; Methodology, A.M.; Project administration, A.B. and A.C.-B.; Software, A.M., A.B., A.E.H. and B.C.; Supervision, A.B., A.C.-B., A.E.H. and B.C.; Validation, A.M.; Visualization, A.M.; Writing—original draft, A.M.; Writing—review & editing, A.B. and A.E.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work is supported by the Partenariats Hubert Curien (PHC) Tassili and the Direction Générale de la Recherche Scientifique et du Développement Technologique (DGRSDT), Algeria.

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

References

1. Santos, L.C.; Santos, F.N.; Pires, E.S.; Valente, A.; Costa, P.; Magalhães, S. Path planning for ground robots in agriculture: A short review. In Proceedings of the 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Ponta Delgada, Portugal, 15–17 April 2020; pp. 61–66.
2. McBratney, A.; Whelan, B.; Ancev, T.; Bouma, J. Future directions of precision agriculture. *Precis. Agric.* **2005**, *6*, 7–23. [\[CrossRef\]](#)
3. Mulla, D.J. Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosyst. Eng.* **2013**, *114*, 358–371. [\[CrossRef\]](#)
4. Roldán, J.J.; del Cerro, J.; Garzón-Ramos, D.; Garcia-Aunon, P.; Garzón, M.; De León, J.; Barrientos, A. Robots in agriculture: State of art and practical experiences. In *Service Robots*; IntechOpen: Rijeka, Croatia, 2018; pp. 67–90.
5. Jensen, M.F.; Nørremark, M.; Busato, P.; Sørensen, C.G.; Bochtis, D. Coverage planning for capacitated field operations, Part I: Task decomposition. *Biosyst. Eng.* **2015**, *139*, 136–148. [\[CrossRef\]](#)
6. Hameed, I.A.F.A. Intelligent behavior of autonomous vehicles in outdoor environment. *Tech. Rep. Mechan. Eng.* **2012**, *1*.
7. Ly, O.; Gimbert, H.; Passault, G.; Baron, G. A fully autonomous robot for putting posts for trellising vineyard with centimetric accuracy. In Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions, Vila Real, Portugal, 8–10 April 2015; pp. 44–49.
8. Gao, M.; Lu, T.F. Image processing and analysis for autonomous grapevine pruning. In Proceedings of the International Conference on Mechatronics and Automation, Luoyang, China, 5–28 June 2006; pp. 922–927.
9. Riggio, G.; Fantuzzi, C.; Secchi, C. A low-cost navigation strategy for yield estimation in vineyards. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2200–2205.
10. Thayer, T.C.; Vougioukas, S.; Goldberg, K.; Carpin, S. Routing algorithms for robot assisted precision irrigation. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2221–2228.
11. Contente, O.; Lau, N.; Morgado, F.; Morais, R. A path planning application for a mountain vineyard autonomous robot. In *Robot 2015: Second Iberian Robotics Conference*; Springer: Cham, Switzerland, 2016; pp. 347–358.
12. Santos, L.; Ferraz, N.; dos Santos, F.N.; Mendes, J.; Morais, R.; Costa, P.; Reis, R. Path planning aware of soil compaction for steep slope vineyards. In Proceedings of the 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Torres Vedras, Portugal, 25–27 April 2018; pp. 250–255.
13. Zoto, J.; Musci, M.A.; Khaliq, A.; Chiaberge, M.; Aicardi, I. Automatic path planning for unmanned ground vehicle using uav imagery. In Proceedings of the International Conference on Robotics in Alpe-Adria Danube Region, Kaiserslautern, Germany, 19–21 June 2019; Springer: Cham, Switzerland, 2019; pp. 223–230.
14. Valente, J.; Barrientos Cruz, A.; Cerro Giner, J.D.; Sanz Muñoz, D. A waypoint-based mission planner for a farmland coverage with an aerial robot—a precision farming tool. In Proceedings of the Precision Agriculture 2011, Praga, Czechia, 11–14 July 2011; pp. 427–436.
15. Valente, J.; Sanz, D.; Del Cerro, J.; Barrientos, A.; de Frutos, M.Á. Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields. *Precis. Agric.* **2013**, *14*, 115–132. [\[CrossRef\]](#)
16. Mammarella, M.; Comba, L.; Biglia, A.; Dabbene, F.; Gay, P. Cooperative agricultural operations of aerial and ground unmanned vehicles. In Proceedings of the 2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Trento, Italy, 4–6 November 2020; pp. 224–229.
17. Mammarella, M.; Comba, L.; Biglia, A.; Dabbene, F.; Gay, P. Cooperation of unmanned systems for agricultural applications: A case study in a vineyard. *Biosyst. Eng.* **2021**, *223*, 81–102. [\[CrossRef\]](#)
18. Dhullipalla, M.H.; Hamrah, R.; Warier, R.R.; Sanyal, A.K. Trajectory Generation on se (3) for an Underactuated Vehicle With Pointing Direction Constraints. In Proceedings of the American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019; pp. 1930–1935.
19. Lee, T.; Leok, M.; McClamroch, N.H. Geometric tracking control of a quadrotor UAV on SE (3). In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 15–17 December 2010; pp. 5420–5425.
20. Stachniss, C. *Robotic Mapping and Exploration*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 55.
21. Lee, S.J.; Cho, D.W.; Chung, W.K.; Lim, J.H.; Kang, C.U. Feature based map building using sparse sonar data. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 1648–1652.
22. Tarutoko, Y.; Kobayashi, K.; Watanabe, K. Topological map generation based on Delaunay triangulation for mobile robot. In Proceedings of the 2006 SICE-ICASE International Joint Conference, Busan, Korea, 18–21 October 2006; pp. 492–496.

23. Cabreira, T.M.; Brisolara, L.B.; Paulo, R.F., Jr. Survey on coverage path planning with unmanned aerial vehicles. *Drones* **2019**, *3*, 4. [\[CrossRef\]](#)
24. Mokrane, A.; Choukchou-Braham, A.; Cherki, B. Survey on Coverage path planning of autonomous marsupial systems for supporting fruit counting process. In Proceedings of the 2020 International Conference on Electrical Engineering (ICEE), Istanbul, Turkey, 25–27 September 2020; pp. 1–6.
25. Horvath, E.; Pozna, C.; Precup, R.E. Robot coverage path planning based on iterative structured orientation. *Acta Polytech. Hung.* **2018**, *15*, 231–249.
26. Hoffmann, G.; Waslander, S.; Tomlin, C. Quadrotor helicopter trajectory tracking control. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, HI, USA, 18–21 August 2008; p. 7410.
27. Cowling, I.D.; Yakimenko, O.A.; Whidborne, J.F.; Cooke, A.K. A prototype of an autonomous controller for a quadrotor UAV. In Proceedings of the European Control Conference (ECC), Kos, Greece, 2–5 July 2007; pp. 4001–4008.
28. Bouktir, Y.; Haddad, M.; Chettibi, T. Trajectory planning for a quadrotor helicopter. In Proceedings of the 16th Mediterranean Conference on Control and Automation, Ajaccio, Corsica, France, 25–27 June 2008; pp. 1258–1263.
29. Mellinger, D.; Kumar, V. Minimum snap trajectory generation and control for quadrotors. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2520–2525.
30. Richter, C.; Bry, A.; Roy, N. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In Proceedings of the Robotics Research, Singapore, 16–19 December 2016; pp. 649–666.
31. Hehn, M.; D’Andrea, R. Real-time trajectory generation for quadcopters. *IEEE Trans. Robot.* **2015**, *31*, 877–892. [\[CrossRef\]](#)
32. Bouffard, P.; Aswani, A.; Tomlin, C. Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results. In Proceedings of the IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14–18 May 2012; pp. 279–284.
33. Hehn, M.; Ritz, R.; D’Andrea, R. Performance benchmarking of quadrotor systems using time-optimal control. *Auton. Robots* **2012**, *33*, 69–88. [\[CrossRef\]](#)
34. Geisert, M.; Mansard, N. Trajectory generation for quadrotor based systems using numerical optimal control. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 2958–2964.
35. Sencer, B.; Ishizaki, K. Smooth polynomial interpolation for point-to-point trajectories with vibration avoidance. In Proceedings of the IECON 2015—41st Annual Conference of the IEEE Industrial Electronics Society, Yokohama, Japan, 9–12 November 2015; pp. 2070–2075.
36. Romero, A.; Penicka, R.; Scaramuzza, D. Time-Optimal Online Replanning for Agile Quadrotor Flight. *arXiv* **2022**, arXiv:2203.09839.
37. Peng, X.Z.; Lin, H.Y.; Dai, J.M. Path planning and obstacle avoidance for vision guided quadrotor UAV navigation. In Proceedings of the 12th IEEE International Conference on Control and Automation (ICCA), Kathmandu, Nepal, 1–3 June 2016; pp. 984–989.
38. Oleynikova, H.; Burri, M.; Taylor, Z.; Nieto, J.; Siegwart, R.; Galceran, E. Continuous-time trajectory optimization for online uav re-planning. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 5332–5339.
39. Fan, X.; Guo, Y.; Liu, H.; Wei, B.; Lyu, W. Improved artificial potential field method applied for AUV path planning. *Math. Prob. Eng.* **2020**, *2020*, 6523158. [\[CrossRef\]](#)
40. Thusoo, R.; Jain, S.; Bangia, S. PID Control of a Quadrotor. In *Advances in Communication and Computational Technology*; Springer: Singapore, 2021; pp. 633–645.
41. Martins, L.; Cardeira, C.; Oliveira, P. Linear quadratic regulator for trajectory tracking of a quadrotor. *IFAC-PapersOnLine* **2019**, *52*, 176–181. [\[CrossRef\]](#)
42. Das, A.; Lewis, F.; Subbarao, K. Backstepping approach for controlling a quadrotor using Lagrange form dynamics. *J. Intell. Robot. Syst.* **2019**, *56*, 127–151. [\[CrossRef\]](#)
43. Xu, R.; Ozguner, U. Sliding mode control of a quadrotor helicopter. In Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, 13–15 December 2006; pp. 4957–4962.
44. Goodarzi, F.; Lee, D.; Lee, T. Geometric nonlinear PID control of a quadrotor UAV on SE (3). In Proceedings of the European Control Conference (ECC), Zürich, Switzerland, 17–19 July 2013; pp. 3845–3850.
45. Tayebi, A.; Roberts, A.; Benallegue, A. Inertial vector measurements based velocity-free attitude stabilization. *IEEE Trans. Autom. Control* **2013**, *58*, 2893–2898. [\[CrossRef\]](#)