agronomy

Article

A Data-Driven Dynamic Obstacle Avoidance Method for Liquid-Carrying Plant Protection UAVs

Shibbir Ahmed [1,2], Baijing Qiu [1,2,*], Chun-Wei Kong [3], Huang Xin [1,2], Fiaz Ahmad [1,2,4] and Jinlong Lin [1,2]

1 School of Agricultural Engineering, Jiangsu University, Zhenjiang 212013, China; 5103180326@stmail.ujs.edu.cn (S.A.); 2111816001@stmail.ujs.edu.cn (H.X.); fiazahmad@bzu.edu.pk (F.A.); rincy@jxau.edu.cn (J.L.)
2 Key Laboratory of Modern Agricultural Equipment and Technology, Ministry of Education, Jiangsu University, Zhenjiang 212013, China
3 School of Aerospace Engineering, University of Michigan, Ann Arbor, MI 1320, USA; chunwei@umich.edu
4 Department of Agricultural Engineering, Bahauddin Zakariya University, Multan 60800, Pakistan
* Correspondence: qbj@ujs.edu.cn

Abstract: Autonomous sprayer UAVs are one of the most used aerial machines in modern agriculture. During flight missions, some common narrow obstacles appear in the flying zone. These are non-detectable from satellite images and one of the biggest challenges for autonomous sprayer UAVs in farmland. This work introduces an obstacle avoidance architecture specifically for sprayer UAVs. This architecture has generality in the spraying UAV problem, and it reduces the reliance on the global mapping of farmland. This approach computes the avoiding path based on the onboard sensor fusion system in real-time. Moreover, it autonomously determines the transition of several maneuver states using the current spraying liquid data and the UAV dynamics data obtained by offline system identification. This approach accurately tracks the avoidance path for the nonlinear time-variant spraying UAV systems. To verify the performance of the approach, we performed multiple simulations with different spraying missions, and the method demonstrated a high spraying coverage of more than 98% while successfully avoiding all vertical obstacles. We also demonstrated the adaptability of our control architecture; the safe distance between the UAV and obstacles can be changed by specifying the value of a high-level parameter on the controller. The proposed method adds value to precision agriculture, reduces mission time, and maximizes the spraying area coverage.

Keywords: obstacle avoidance; plant protection UAV; precision agriculture; data-driven dynamic avoidance approach; spray coverage

check for updates

Citation: Ahmed, S.; Qiu, B.; Kong, C.-W.; Xin, H.; Ahmad, F.; Lin, J. A Data-Driven Dynamic Obstacle Avoidance Method for Liquid-Carrying Plant Protection UAVs. *Agronomy* **2022**, *12*, 873. https://doi.org/10.3390/agronomy12040873

Academic Editor: Francisco Manzano-Agugliaro

Received: 11 March 2022
Accepted: 27 March 2022
Published: 2 April 2022

1. Introduction

Recent trends for agricultural enhancement products are based on artificial intelligence, information technology, global positioning, geographic information systems, automated management systems, process control, robotics, and integrated precision [1]. As a part of high-tech precision agriculture, Unmanned Aerial Vehicles (UAV) are applied for crop monitoring and crop protection practices at a large scale, especially in advanced technological countries aiming for higher smart agriculture [2]. Research studies are actively conducted on Unmanned Aerial Vehicles' (UAVs) development and adaption [3,4] for agricultural practices with higher precision; they are mainly used for field mapping [5,6], plant stress detection [7,8], biomass estimation [9,10], weed management [11,12], inventory counting [13], chemical spraying [14–16], auxiliary pollination [17], etc. Of these, the most widely used application is spraying pesticides by UAVs for plant protection. Spraying pesticides on crops is a compulsory operation that is applied multiple times during crop lifetime. Starting from the Japanese Yamaha R50 1985 [18], the development of automated and intelligent agricultural pesticide sprayer UAVs is growing very fast. These automated sprayer UAVs are gaining more attention than manual flying UAVs because of their operational and environmental complexity [19]. They are expected to achieve some key points, including: (a) better spray deposition, (b) smarter obstacle avoidance, (c) more coverable spray path plan, etc. Intelligent obstacle avoidance for agricultural sprayer UAVs is a vital feature that increases the spray coverage, saves battery life, and avoids obstacles in a shorter time. A proper obstacle avoidance method specifically for sprayer UAVs can save time, increase pesticide deposition efficiency, and ensure UAV and human safety.

Previously, substantial research has been performed on obstacle avoidance for single and multiple obstacles. Chakravarthy and Ghose [20] developed a collision cone technique for single-obstacle avoidance. Another method called the velocity obstacle approach was presented for both static and moving obstacles by Fiorini and Shiller [21]. Chakravarthy and Ghose (2012) extended the collision cone method to detect obstacles in three-dimensional space. A radar-assisted collision guidance strategy (RACAGS) for low-altitude flying unmanned vehicles using the sliding circle algorithm was proposed by Kumar and Ghose [22]. The cross-entropy method has been used for obstacle detection and avoidance, also known as 'see and avoid' by Olivares-Mendez et al. [23]. They used a fuzzy controller to command the UAV and experimented with simulation. Another obstacle-avoidance control method that consists of acquiring a distance between UAVs and a front object in a flying direction of the UAV and controlling the flying altitude of the UAV according to the distance between the UAV and object was invented by Zou et al. [24]. In the same year, Zou [25] also developed another method to control obstacle avoidance for an unmanned aerial vehicle by obtaining the current craft body's attitude and position information. The UAV controls the detection apparatus direction to be in a preset direction according to the current attitude information of the UAV. Besides, plenty of research on multiple object detection and avoidance has been carried out previously. A Mixed-Integer Linear Program (MILP) approach was applied by Richards and How [26], using linear constrain of an approximate model of aircraft dynamics. A collision-free path using a visibility graph was planned by Lozano-Pérez and Wesley [27], where the obstacles consider the robot's dimensions. A modified Grossberg Neural Network (GNN) was proposed by Wang et al. [28] for multiple UAVs' cooperative formation. For a quadrotor, Park and Baek [29] presented a stereo vision-based obstacle collision avoidance method using an ellipsoidal bounding box and hierarchical clustering. Another image processing technique was used by Ferrick et al. [30], where a basic wall-following was combined with obstacle avoidance constructed from laser rangefinder sensor data using image processing in OpenCV and simplified occupancy grid maps. Naderhirn et al. [31] proposed a robust hybrid control for stationary obstacle avoidance in the presence of uncertain measurements, such as for a radar cone. Shanmugavel et al. [32] generated safe waypoints using Delaunay triangulation to isolate obstacles from one another. Many studies have been performed on collision avoidance between UAVs [33–37].

These different algorithms were formed concerning goal-point reaching and avoidance with shorter times and paths, as illustrated in Figure 1a–d. However, for automated agricultural sprayer UAVs, the objectives are particular [38,39] and different in obstacle avoidance, as illustrated in Figure 1e. The goal is to fly from one waypoint to the next waypoint to spray the maximum area in a shorter time. In between one waypoint and another waypoint, the UAV flies straight. If the UAV faces any obstacles between two waypoints, it needs to avoid that obstacle and return onto that straight path.

In an automatic operation, two categories of physical obstacles can appear in farmland. The ground station application collects the farmland's coordinates and larger obstacle images from satellite images and positioning [41]. Combining those data and the spray coverage width, the ground station makes a suitable spray mission, excluding those obstacles [42,43]. Sometimes they can be avoided by manually configuring the obstacle's placement by manual GPS positioning [44]. However, some relatively smaller/narrow size obstacles are undetectable from satellites, or precise position cannot be obtained from satellite data. These obstacles need to be avoided by local onboard obstacle avoidance systems. Examples of the second type of obstacle are electric poles, telegraph poles, grid line towers, scattered trees, windmills, pergolas, etc. Besides, another physical issue of agricultural sprayer UAVs is that they carry a continuously decreasing liquid tank, which changes the center of gravity of the tank [45]. Because of these phenomena, the UAV loses a massive amount of its overall weight during operation. This changing weight changes the flight controller's flight control parameters such as maximum pitch and roll degree. Figure 2 shows a basic sprayer UAV model, and the varying center of gravity has been illustrated. So, real-time obstacle avoidance with smart technology is an important feature that must include the flight technology of agricultural pesticide sprayer UAVs.

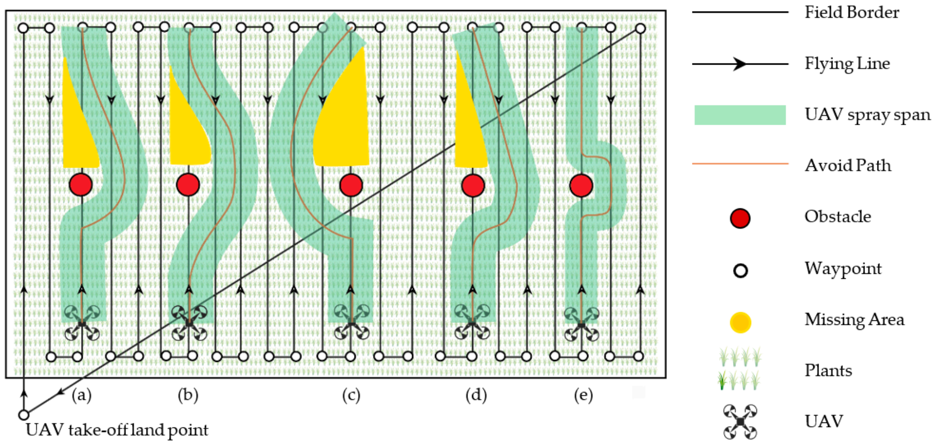

Figure 1. Traditional algorithms for sprayer UAV's operations and expected path: (**a**) sliding circle [22], (**b**) cross-entropy [23], (**c**) potential field [40], (**d**) collision cone [29], (**e**) expected path.

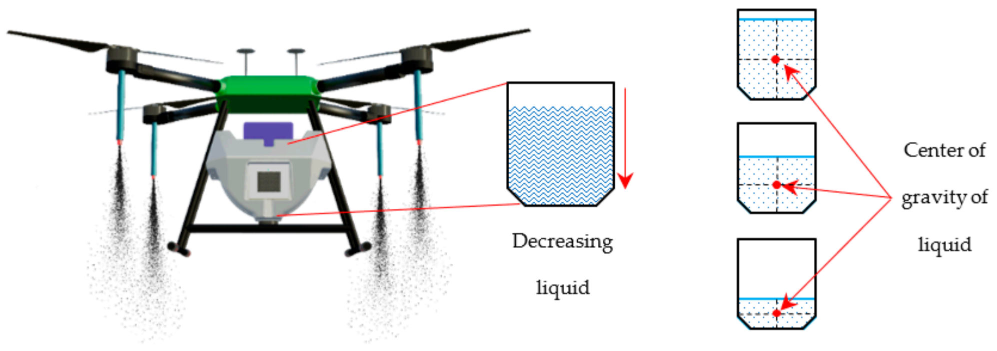

Figure 2. Agricultural sprayer UAV (quadrotor).

Considering these issues: (a) an agricultural liquid-carrying sprayer UAV must have a time saving effect, and the coverage must prioritize local or real-time obstacle avoidance systems; (b) the sprayer UAVs generated avoidance path must be out of the danger zone of the obstacle and should be as close as possible for maximum spray coverage; (c) the avoidance approach should vary depending on the speed and weight; and (d) the local obstacles need to be avoided by a local onboard sensing system, where a sensor fusion is also needed to detect the obstacle's front and side situation.

This study proposes a data-driven avoidance method targeting dynamic load-changing agricultural sprayer UAVs. Using the proposed method, agricultural UAVs can avoid obstacles in farmland with dynamic avoidance speed and fixed heading direction, saving time with safety concerns and covering a maximum area. The contributions of this study are summarized as follows. First, a new real-time obstacle avoidance method was developed for plant protection sprayer UAVs to avoid different static farmland obstacles. Second, a dynamic obstacle avoidance approach was created using the sprayer UAV's liquid level and the obstacle's orientation. Third, a suitable new sensor architecture was used for the new approach. Fourth, an independent local avoidance system was created with the ability to overdrive the pitch and roll control to let the mission controller keep the height and yaw static.

2. Material and Methods

2.1. Model

We created a primary physical model of an agriculture quadrotor drone in MATLAB Simulink to show the new algorithm's concept. The total body of the architectural model contained three essential parts: (a) the main body frame of the quadrotor; (b) the bottom-attached tank with changeable rigid weight; and (c) the suitable sensor architecture. We first made a quadcopter using 6-DOF (degrees of freedom) dynamic equations to make the entire model, where the frame generated thrust using a BLDC (brushless DC) motor. Then, we designed a PID-based velocity controller to control the flight maneuver. After that, we attached a model tank below the quadrotor frame, where the liquid inside the tank is simulated as a rigid body. That rigid body changes its mass and moment of inertia according to the liquid level. Lastly, we used a new multiple sensor architecture with the body frame to acquire obstacle knowledge from the environment. This overall control architecture is shown in Figure 3, and the detailed model is shown in the following subsections.

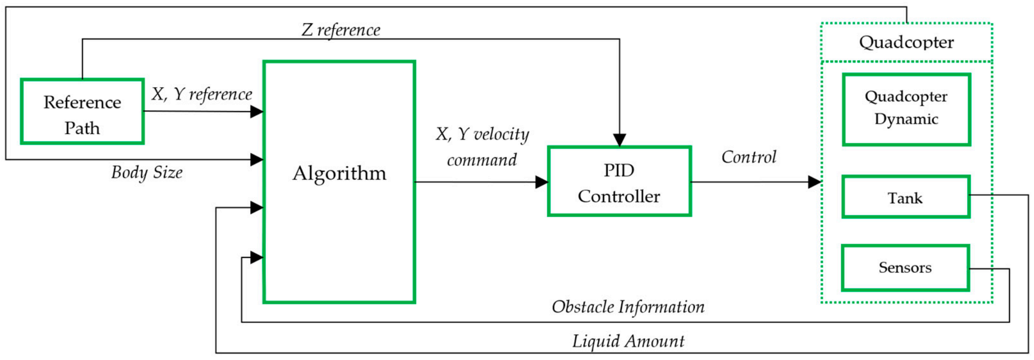

Figure 3. Operation drone control architecture.

2.1.1. Quadrotor and Tank Liquid Level Formation

We used a traditional rigid body quadcopter, using 6-DOF (degrees of freedom) dynamics governed by Newton's law and the Euler angles shown in Figure 4a. Four motors drive the quadcopter model at the four endpoints of each arm. The center of mass of the UAV is C_D and the length of the arm from the center to the outer end is L_{DA}. Motors 1, 2, 3, and 4 generate thrust, u_1, u_2, u_3, and u_4, respectively, and create torque τ_1, τ_2, τ_3 and τ_4, respectively. Equations (1) and (2) describe the 6-DOF dynamics of the quadcopter, where the total force is $\vec{F_b}$ and the total torque is $\vec{M_b}$. $\vec{V_b}$ and $\vec{W_b}$ are the velocity and angular rates on the body frame.

$$\vec{F_b} = M\left(\frac{d\vec{V_b}}{dt} + \vec{W_b} \times \vec{V_b}\right) \tag{1}$$

$$\vec{M_b} = I\left(\frac{d\vec{W_b}}{dt}\right) + \vec{W_b} \times \left(I\vec{W_b}\right) \tag{2}$$

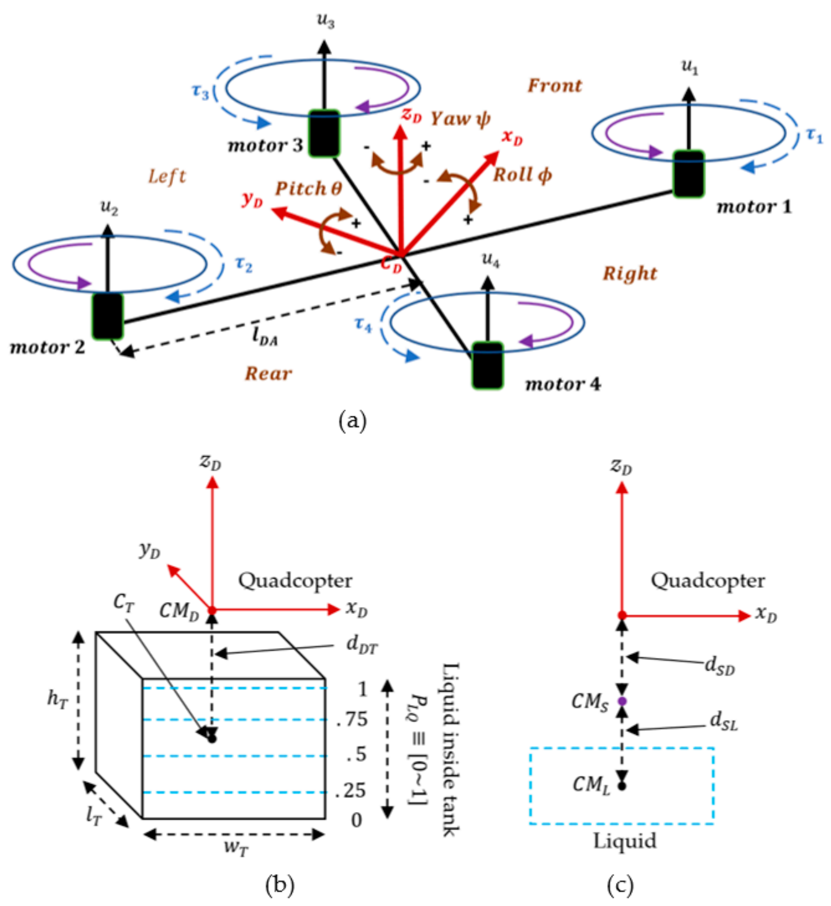

Figure 4. Quadrotor model schematic (**a**); liquid tank construction and liquid level assumption (**b**); different center of mass (**c**).

For the tank part, we added a rectangular-shaped empty body which is shown in Figure 4b, where C_T is the center of the tank. d_{DT} is the distance between the center of the quadcopter and the center of the tank. The placement of the tank is under the quadcopter. This study excluded the sloshing effect by constructing a lid that could be moved from bottom to top inside the tank. This lid allowed us to model the liquid as a rigid body. A range of 0 to 1 was set from bottom to top for the lid to define the liquid amount inside the tank. Equation (3) shows the liquid level assumption, and Figure 4b illustrates the liquid level assumption. It also shows the placement of the tank and quadrotor, where h_T, w_T, and l_T are the tank's height, width, and length. For simplification, we used $l_T = w_T$ here.

In Equation (4), the mass of liquid M_L depends on the level of the liquid, where M_{LQmax} is the maximum liquid mass and P_{LQ} is a nondimensional parameter that defines the amount of the liquid. The system's total mass will be the quadrotor and liquid's combined mass, stated in Equation (5). Equations (6)–(8) show that the tank liquid's moment of inertia concerning its principal axes depends on the tank liquid level, where I_{xL}, I_{yL}, and I_{zL} are the moment of inertia of x, y, and z axes, M is the mass of the entire system, and M_D is the mass of the UAV.

$$P_{LQ} = [0 \sim 1] \tag{3}$$

$$M_L = M_{LQmax} P_{LQ} \tag{4}$$

$$M = M_D + M_L \tag{5}$$

$$I_{xL} = \frac{1}{12} M_L \left[\left(h_T P_{LQ} \right)^2 + {w_T}^2 \right] \tag{6}$$

$$I_{yL} = \frac{1}{12} M_L \left[\left(h_T P_{LQ} \right)^2 + {l_T}^2 \right] \tag{7}$$

$$I_{zL} = \frac{1}{12} M_L \left[{l_T}^2 + {w_T}^2 \right] \tag{8}$$

When the tank is attached to the quadrotor, the system's center of mass and the moment of inertia change. In Equation (9), the center of the mass of the tank's liquid CM_L is shown, and the center of mass of the whole system CM_S is shown in Equation (10), where CM_D is the center of mass of the quadcopter. Both are illustrated in Figure 4c. Here, we defined two distance parameters, d_{SD} and d_{SL}, in Equations (11) and (12), where d_{SD} is the distance between the system's center of mass and the drone's center of mass, and d_{SL} is the distance between the system's center of mass and the tank liquid's center of mass. These two distances, d_{SD} and d_{SL} are illustrated in Figure 4c. The inertia of the total system is stated in Equations (13)–(15), where I_x, I_y, and I_z are the moment of inertia of the x, y, and z axes.

$$CM_L = -\left[d_{DT} - 0.5h_T + h_T - 0.5h_T P_{LQ} \right] \tag{9}$$

$$CM_S = \left(\frac{M_D CM_D + M_L CM_L}{M_D + M_L} \right) \tag{10}$$

$$d_{SD} = CM_S - CM_D \tag{11}$$

$$d_{SL} = CM_S - CM_L \tag{12}$$

$$I_x = I_{xD} + M_D d_{SD}^2 + I_{xL} + M_L d_{SL}^2 \tag{13}$$

$$I_y = I_{xD} + M_D d_{SD}^2 + I_{xL} + M_L d_{SL}^2 \tag{14}$$

$$I_z = I_{zD} + I_{zL} \tag{15}$$

2.1.2. Sensor Model

This system used one long-range wide-angle sensor and four single-point detection sensors facing different directions. This sensor architecture allowed the UAV to detect the obstacle's distance, width, and length, and avoid the obstacle using these parameters. A long-range millimeter-wave radar sensor and four single-point laser ranging sensors were used. In Figure 5, we show the sensors placements, where the long-range sensor S_0 is set up at the front middle, the single-point sensor S_1 is set up at the front left arm facing the front, the single-point sensor S_2 is set up at the back left arm facing left, the single-point sensor S_3 is set up at front right arm facing front, and single-point sensor S_4 is set up at the back right arm facing right. Details will be described in "Obstacle information" section.

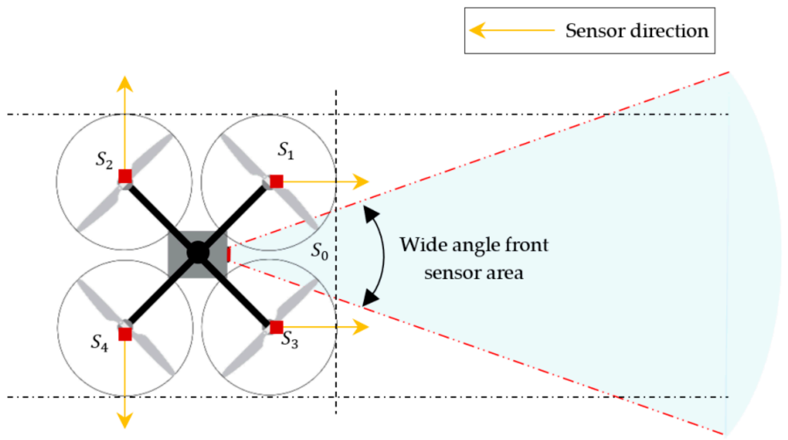

Figure 5. Sensor setup for quadcopter.

2.2. Path Planning Algorithm

From the information of related research work, sprayer UAVs, and obstacle data in farmland, we can assume that the obstacles in farmland are sparsely distributed. Here, we assume a common obstacle as a vertical object and simply formed. Following our data-driven concept, the data processing level converts primary data into decision-making datasets to perform final avoidance. This is the high-level path planning of the proposed obstacle avoidance procedure illustrated in Figure 6a. Using this process, a complete obstacle avoidance path shows the implementation of the path planning algorithm for a single obstacle in Figure 6b.

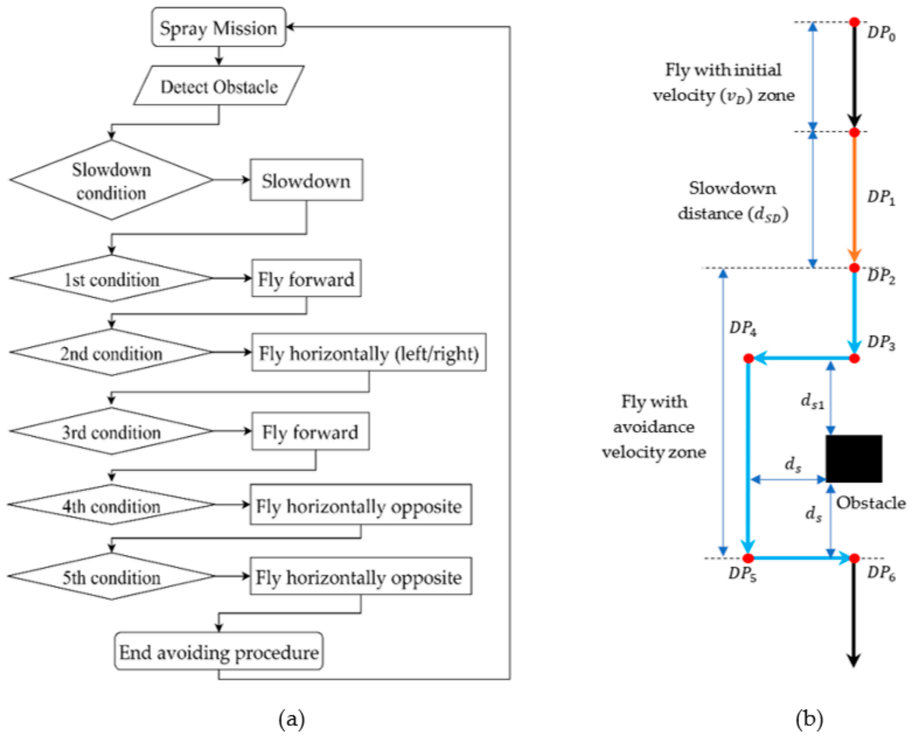

Figure 6. (**a**) High-level path planning flow. (**b**) UAV's positions in the perspective of an obstacle. DP_{0-6} are eventual drone positions.

2.2.1. Distance Definition

Safe Distance around the Obstacle

Different sprayer UAV sizes create different safety concerns in farmland. For example, a 20 L liquid-carrying UAV needs a more extensive, more substantial rotor span and body than a 5 L liquid-carrying UAV [46]. The larger wingspread creates a higher maneuver-reflex time. Concerning the issue, the safe distance should be dynamically related to the arm length of the UAV. The relation between the UAV's safe distance d_S and arm length l_{DA} is shown in Equation (16). A user-defined multiplier x is used to decide the safe distance, which is dependent on the size of the UAV. In our UAV model, we select $x = 2$, which means the safe distance is double that of the UAV's arm. This distance is flexible depending on the user's input.

$$d_S = xl_{DA} \tag{16}$$

Slowdown Distance

Large agricultural UAVs often fly in $2 \sim 6$ ms^{-1} [47]. However, the velocity is sometimes too high to avoid obstacles precisely and safely. If the UAV wants to go close enough, it cannot go with high velocity for spray coverage. A sudden stop at any point with high velocity and liquid load will create unexpected oscillation. The UAV needs to slow down gradually from initial flying velocity to avoidance velocity to perform a stable avoidance maneuver. This distance is named slowdown distance and illustrated in Figure 6. Here, DP_0 is the position at the very first moment that the UAV sees the obstacle and starts calculating the avoidance velocity and the slowdown distance. Then, from the position DP_1, the UAV starts slowing down and it reaches the avoidance velocity at position DP_2.

This slowdown distance varies depending on the initial velocity v_D and liquid level P_{LQ}. In our system, the slowdown distance is defined as d_{SD} and estimated by assuming a constant acceleration of motion, which is shown in Equation (17). From free body diagrams of a UAV [48] where total thrust is u_T, pitch angle is θ, u_{Tmax} is maximum thrust, g is the gravitational acceleration, and θ_{max} is maximum pitch angle:

$$u_{Tmax}\cos\theta_{max} - Mg = 0$$

$$u_{Tmax}\cos\theta_{max} = Mg$$

If the acceleration is a, from the free body diagram:

$$u_{Tmax}\sin\theta_{max} = Ma$$

So,

$$\tan\theta_{max} = \frac{a}{g}$$

$$\mathrm{a} = g\tan\theta_{max}$$

From Newton's law (the velocity of the drone is v_D and the deceleration distance d_{Ds}):

$$u_{Tmax}\sin\theta_{max} = Ma$$

$$d_{Ds} = \frac{{v_d}^2}{2a}$$

$$d_{SD} = \frac{{v_D}^2}{2g\tan\theta_{max}} \tag{17}$$

Because the trust has an upper bound, the UAV is limited to a certain tilt angle. Otherwise, the thrust is not enough to balance the mass, which causes a flight crash. Considering 3-degree force balance:

$$u_{Tmax}\cos^2\theta_{max} = Mg$$

$$\theta_{max} = \cos^{-1}\sqrt{\frac{Mg}{u_{Tmax}}}$$

Here, we introduce a safety factor named c_1 which is multiplied by the maximum tilt angle. The value of the safety factor is equal to the subtract value of the mass of the UAV and the mass of the UAV with the tank. Here M_D is the mass of the UAV, M is the mass of the whole system and $Thrust_{max}$ is the maximum thrust of the system created by four rotors. So, the maximum pitch angle for the system is:

$$\theta_{max} = c_1\cos^{-1}\sqrt{\frac{Mg}{u_{Tmax}}}$$

$$\theta_{max} = \left(\frac{M_D}{M}\right)\cos^{-1}\sqrt{\frac{Mg}{u_{Tmax}}} \tag{18}$$

By combining Equations (17) and (18) (where the slowdown distance is only a function of the initial velocity v_D and the nondimensional liquid parameter P_{LQ}):

$$d_{SD} = \frac{{v_D}^2}{2g\tan\theta_{max}}$$

$$d_{SD} = \frac{{v_D}^2}{2g\tan\left(\left(\frac{M_D}{M}\right)\cos^{-1}\frac{Mg}{u_{Tmax}}\right)}$$

$$d_{SD} = \frac{v_D{}^2}{2g\tan\left(\left(\frac{M_D}{M_D+M_L}\right)\cos^{-1}\sqrt{\frac{(M_D+M_L)g}{u_{Tmax}}}\right)}$$

$$d_{SD} = \frac{v_D{}^2}{2g\tan\left(\left(\frac{M_D}{M_D+\left(M_{LQmax}\cdot P_{LQ}\right)}\right)\cos^{-1}\sqrt{\frac{\left(M_D+\left(M_{LQmax}P_{LQ}\right)\right)g}{u_{Tmax}}}\right)} \tag{19}$$

Obstacle Information and Formation

Obstacle Information

From the view of the sprayer UAV, it defines the obstacle's position from two time segments. Firstly, when the obstacle appears in front of the drone, and secondly, when the drone is avoiding the obstacle. We used a millimeter-wave radar sensor as the primary obstacle detector, which is named S_0, and a single-point laser distance sensor for border detection, which is mentioned as S_1, S_2, S_3, and S_4. The basic advantage of a millimeter-wave radar is the capacity of object detection from a very long distance. Besides, it has an estimation capacity of the obstacle direction using its positive and negative angular field of view [49]. On the other hand, the single laser distance sensor has a very precise measurement capability which is suitable to detect the boundary of the obstacle [50]. In Figure 7a, we show the obstacle's positions at different moments in the perspective of the UAV. At position OP_0, the sensor S_0 detects the distance of the obstacle and calculates the distance d_0. The sensor S_0 can also detect whether the obstacle is on the right or left side of the UAV for the corner case. The detail of the corner case will be described in "Possible obstacle positions and first safe distance" section. Then, sensor S_1, S_2, S_3, and S_4 detect the obstacle at positions OP_1, OP_2, OP_3, and OP_4, and calculate the distances d_1, d_2, d_3, and d_4. These laser sensors collect two types of information for the system. One is the appearance of the obstacle, and another is the distance of the obstacle.

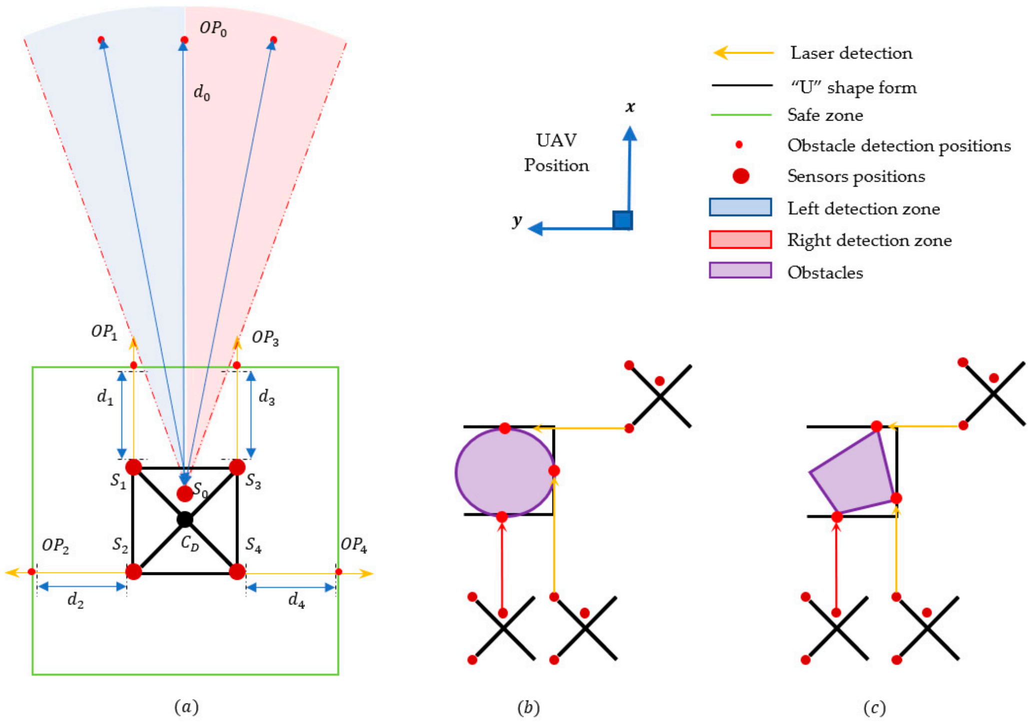

Figure 7. Obstacle positions in the perspective of different sensors. (**a**) Obstacle's positions in perspective of UAV, UAV's position rectangular; (**b**) shape formation for circular object; (**c**) shape formation for trapezium object.

Obstacle Formation

Figure 7a illustrates the possible positions for an obstacle considering the safety of the UAV, where outside of the green line is a safe zone. There are different shaped obstacles on farmland that need to be ideally avoided during a planned mission. The proposed algorithm's sensor architecture can automatically normalize any shaped obstacles into an absolute "U" shape using three detection marks. This transformation creates a simplified avoidance path that can be realized by shifting the velocity command in the x and y axes. Figure 7b,c show how the UAV's sensor architecture detects obstacles from different positions and extracts three major points.

First Safe Distance and Maneuver Direction

Possible Obstacle Positions and First Safe Distance

During the mission, the obstacle detection and avoidance procedure often face corner detection cases. Figure 8 shows different detection and direction cases while the UAV approaches the obstacle. Figure 8a describes the actual avoid region, the UAV's different approaching positions (p_1, p_2, p_3, p_4, p_5), and the safe distance d_s. It also shows a zone named D zone, where the system can detect the obstacle either from S_0, S_1, and S_3; or from S_0 and S_1; or from S_0 and S_3. Corner cases occur when the UAV is approaching inside the avoid region but outside the D zone. From the position p_1 to p_2, the UAV cannot detect the obstacle's edge via sensor S_3. From position p_4 to p_5, the UAV cannot detect the edge of the obstacle via sensor S_1. Figure 8b,c show some other corner cases and the data received from the sensor S_0 in these situations.

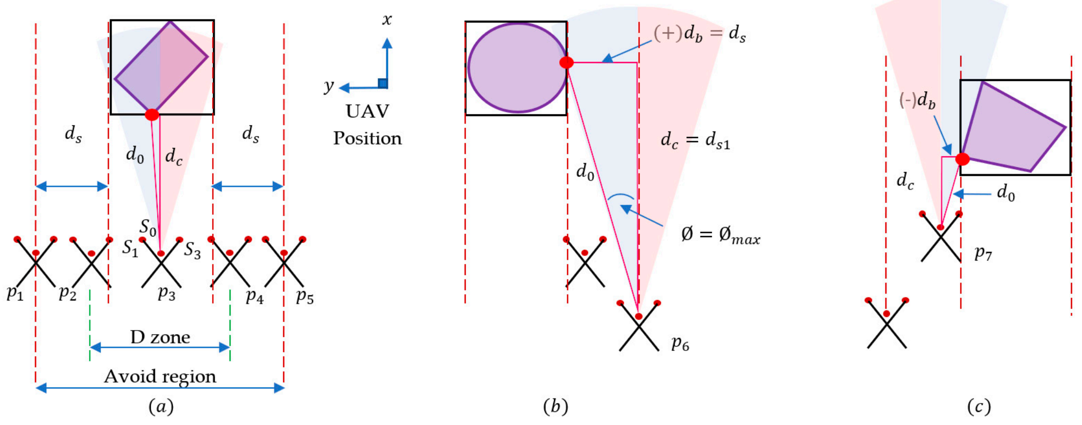

Figure 8. UAV's possible predicted positions and detection calculation. (**a**) possible UAV placements, (**b**) Left side position, (**c**) Right side critical position.

The relation of the data received from the center sensor S_0 is shown in Equations (20) and (21). The first safe distance d_{s1} is then defined as the trigger distance for the UAV to make correct avoidance in any possible approaching case, where the corner cases are the most crucial ones. This distance d_{s1}, should be determined appropriately. If d_{s1} is too large, the UAV will avoid too early and leave a large distance between the obstacle in any approaching case. However, if d_{s1} is too small in corner cases, the UAV will approach too close to the obstacle, lose the obstacle detection due to the angle limit of the center sensor S_0, and eventually decide not to avoid. Considering the most extreme situation of the corner cases, which is illustrated in Figure 8b, the value of the first trigger distance d_{s1} determined as a function of the maximum front sensor's angle $\varnothing_{max}$ and the safe distance d_S are shown in Equation (22). From Equation (22), a larger cone angle of the center sensor results in a smaller first safe distance for a specified safe distance, thus reducing the gap between the UAV and the obstacle.

$$d_b = d_0 \sin \varnothing \tag{20}$$

$$d_c = d_0 \cos \text{Ø} \tag{21}$$

$$d_{S1} = \frac{d_s}{\tan \text{Ø}_{max}} \tag{22}$$

Obstacle's Direction Detection

The UAV should choose the avoidance direction (left/right) according to the obstacle's position to create a safer and shorter path in the corner case. In Figure 8b,c, we show how it obtains the direction of the obstacle using Equation (20). If the value $d_b(+)$ is positive, the obstacle is on the left side, and the right way is safer to avoid. Besides, if the value is negative $d_b(-)$, the obstacle is on the right side, and the left is safer to avoid.

Algorithm Formation with State Machine

We form this algorithm for agricultural sprayer UAVs using the primary inputs and processed data. The inputs are the length of an arm l_{DA}, the mass of the UAV M_D, initial velocity v_D, liquid level P_{LQ}, front-middle sensor S_0's distance data d_0 and angle Ø, front-faced left-front sensor S_1's distance data d_1, left-faced left-back sensor S_2's distance data d_2, front-faced right-front sensor S_3's distance d_3, right-faced right-back sensor S_2's distance d_4 and drift distance d_{DR}. Processed inputs are slowdown distance d_{SD}, avoidance velocity v_A, trigger time t_{TR}, safe distance d_S, first safe distance d_{S1}, corner object's opposite length d_b, corner object's adjacent length d_c and obstacle's directional position $d_b(Negative)$ and $d_b(Positive)$. We used DFA (Deterministic Finite Automata) or the state machine mechanism to minimize the computational calculation and simplification of decisions. To use the mechanism, we constructed some states for the algorithm in Table 1, where the algorithm's state is described.

Table 1. State table of the algorithm.

State	Operation	Binary Condition	Logical Condition	Next State
q_0	Follow reference path	1	$d_0 \leq \ (d_{SD} + d_{S1})$	q_1
		0		q_0
q_1	Avoidance Velocity Fly ($x\text{-}vel\ = v_A$, $y\text{-}vel = 0$)	1	$d_0 \leq d_{S1}$	q_2
		0		q_1
q_2	Decide Avoiding Direction	1	$(d_1\ \&\ d_3 = max\ \&\&\ d_b \geq\ 0) \Vert d_3 \geq d_1$	q_3
		0	$(d_1\ \&\ d_3 = max\ \&\&\ d_b < 0) \Vert d_3 < d_1$	q_4
q_3	Fly right ($x\text{-}vel = 0$, $y\text{-}vel = v_A$) count time +	1	$S_1 \neq touched\ \&\&\ d_b \geq (l_{DA} \sin\alpha) \Vert S_1 = touched\ \&\&\ d_1 = Max$	q_5
		0		q_3
q_4	Fly Left ($x\text{-}vel = 0$, $y\text{-}vel = -v_A$) count time +	1	$S_3 \neq touched\ \&\&\ d_b \geq (l_{DA} \sin\alpha) \Vert S_3 = touched\ \&\&\ d_3 = Max$	q_5
		0		q_4
q_5	Continue $y\text{-}vel$, t_{TR} time	1	$finished\ flying\ t_{TR}\ time\ \&\&\ v_A < 0$	q_6
		0	$finished\ flying\ t_{TR}\ time\ \&\&\ v_A > 0$	q_7
q_6	Fly forward ($x\text{-}vel = v_A$, $y\text{-}vel = 0$)	1	$S_2 = touched\ \ \&\&\ \ d_2 = Max$	q_8
		0		q_6
q_7	Fly forward ($x\text{-}vel = v_A$, $y\text{-}vel = 0$)	1	$S_4 = touched\ \ \&\&\ \ d_4 = Max$	q_8
		0		q_7
q_8	Continue $x\text{-}vel, t_{TR}$ time	1	$finished\ flying\ t_{TR}\ time\ \&\&\ S_2 < S_4$	q_9
		0	$finished\ flying\ t_{TR}\ time\ \&\&\ S_2 > S_4$	q_{10}
q_9	Fly left ($x\text{-}vel = 0$, $y\text{-}vel = -v_A$) count time −	1	$counted\ time = 0$	q_0
		0		q_9
q_{10}	Fly right ($x\text{-}vel = 0$, $y\text{-}vel = v_A$) count time −	1	$counted\ time = 0$	q_0
		0		q_{10}

There are eleven states in this mechanism, which are q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, and q_{10}. State q_0 is the starting state of the process, which activates from the start of the mission. The front sensor is activated continuously with the whole process from this state. While the reading of the sensor S_0 is less or equal than the summation of slowdown distance d_{SD} and first safe distance d_{S1}, it switches the state from q_0 to q_1, otherwise q_0 remains active. The next state q_1 works to decrease or decelerate the velocity from the initial speed v_D to avoidance velocity v_A in the x-axis direction. While the distance d_0 is less or equal than the first safe distance d_{S1}, it switches the state to q_2.

In the state q_2, If the values of d_1 and d_3 are maximum or over buffer value, and the value of d_b is bigger or equal than 0, or the value d_3 is bigger or equal than d_1, it means the obstacle is at the left or absolute front side of the UAV, according to "Obstacle's direction detection" Section. In this condition, it switches the state to state q_3. On the other hand, if the values of d_1 and d_3 are maximum or over buffer value, and the value of d_b is less than 0; or if the value of d_3 is less than d_1, it means the obstacle is at the right side of the UAV, and it switches to the state q_4.

In the state q_3, the UAV stops flying in the x-axis direction and starts flying in y-axis direction positively using avoidance velocity v_A, which means flying the UAV bank right. Besides that, it begins counting the time of moving right. Here, while the drone is flying right, it detects when to fly forward when either of the two conditions occurs. The first condition is that the sensor S_1 is not touched, and the value of d_b is bigger or equal than the value of arm length l_{DA} multiplied with $\sin\alpha$, where $\alpha = 45°$ is defined in Figure 8. This corresponds to the situation where the drone has approached the obstacle in corner cases during q_1 state, shown in Figure 8b,c. The second condition is that the sensor S_1 has already touched the obstacle, and the S_1 sensor detects nothing in the forward direction, which means that it receives a maximum return value. This corresponds to the situation where the drone has approached the obstacle in the D zone during q_1 state, which is shown in Figure 8a. If either of these conditions are checked, state q_3 switches to state q_5; otherwise, it continues running state q_3. Similarly, in state q_4, the UAV stops flying in x-axis direction and starts flying in the y-axis direction negatively using avoidance velocity v_A, which means flying the UAV bank left. Besides that, it begins counting the time of moving left. In this state, if the sensor S_3 is not touched, and the value of d_b is bigger or equal than the value of arm length l_{DA} multiplied with $\sin\alpha$, or if the sensor S_3 has already touched the obstacle, and the S_3 sensor detects nothing in the forward direction, receiving a maximum return value, state q_4 switches to state q_5. Otherwise, it continues running state q_4. Now, q_5 is a single work process step; it flies the UAV t_{TR} time using the previous step's velocity. If it finishes flying t_{TR} time and the velocity is positive, it switches to the state q_6. Or, if the velocity is negative, it switches to the state q_7. In the state q_6, when sensor S_2 has touched the obstacle from the right side, and the length d_2 is equal to its maximum value, meaning that there is nothing on the left side, it switches to the state q_8. Similarly, in state q_7, when sensor S_4 has touched the obstacle from the left side, and d_4 is equal to its maximum value, meaning that there is nothing on the right side, it switches to the state q_8.

State q_8 is also a single work process, similar to step q_5. Here, the UAV flies t_{TR} time using the previous step's velocity. If it finishes flying t_{TR} time and the sensor S_2 has touched, which means the stored value is less than the value of S_4, it switches to the state q_9. Or, if the sensor S_4 has touched, which means the stored value is less than the value of S_2, it switches to the state q_{10}. State q_9 and q_{10} are both in a recovery state, and they run with a countdown timer. State q_9 flies the UAV in the left direction with avoidance velocity and state q_{10} flies the UAV in the right direction with avoidance velocity. When the countdown time is 0, which means the UAV has returned on the approximate straight line where it started the avoiding process, both states return to the state q_0.

Now, we describe the deterministic finite automaton M, which consists of a finite set of states Q shown in Equation (23), a finite set of input Σ shown in Equation (24), a transition function δ, an initial or start state q_0 shown in Equation (25), and a set of accept states F shown in Equation (26).

$$Q = \{q_0,\ q_1,\ q_2,\ q_3,\ q_4,\ q_5,\ q_6,\ q_7,\ q_8,\ q_9,\ q_{10}\} \tag{23}$$

$$\Sigma = \{1, 0\} \tag{24}$$

$$q_0 = \{q_0\} \tag{25}$$

$$F = \{q_0\} \tag{26}$$

2.3. Data-Driven Control

This section describes the algorithm's data-driven control method and system identification process. Generally, the avoidance system needs to identify some data by preflight system identification tests [51]; we are going to use an example dataset here to show the functionality of the system identification process. Later, all the simulation performances are completed using the same dataset in the simulation performance section. There are so many commercial liquid-carrying sprayer UAV models in operation in the marketplace, and many types of models are still under development [52]. We created a model in MATLAB R2020a software and performed simulations to collect the system identification data. This numerical simulation of the proposed method is performed by a quadrotor attached to a square-shaped tank setup and a suitable sensor setup shown in Figures 7 and 9. The constant parameters of the UAV, tank, one-millimeter-wave distance sensor [53], and single-point laser distance sensors [54] are summarized in Table 2.

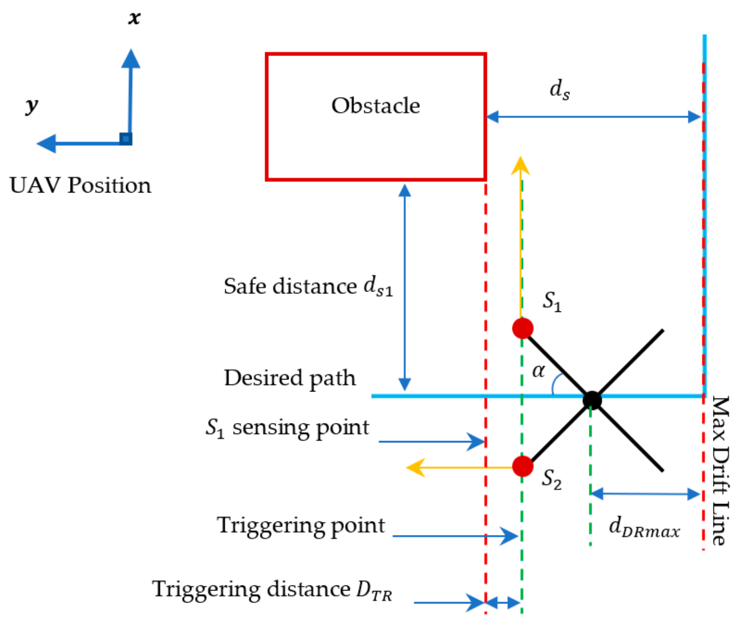

Figure 9. Max drift line and triggering point.

Table 2. Parameters of the tank-carrying quadrotor UAV system.

Quadrotor			Tank			Millimeter-wave Sensor		Laser Sensor	
Mass	M_D	20 kg	Mass	M_{Lmax}	32 kg	Frequency	10 Hz	Frequency	10 Hz
Arm Length	l_{DA}	50 cm	Size	l_T	0.4 m	FoV Hor.	50°	FoV	Single point
Moment of inertia	I_{xL}	$0.3933e^{-1}$ kgm^2		w_T	0.4 m	FoV Ver.	20°	Length	5 m
	I_{yL}	$0.3933e^{-1}$ kgm^2		h_T	0.2 m	Length	30 m		
	I_{zL}	$0.76e^{-1}$ kgm^2	Liquid level	P_{LQ}	$0 \sim 1$				
Gravity	g	9.8 ms^{-2}							
Max Trust	u_T	13.5 kg × 4							
The distance between the center of the quadrotor and the center of the tan k $d_{DT} = 0.4$ m									
Constant drift distance $d_{DR} = 0.59$									
Safe distance $d_s = 1$ m									

2.3.1. Drift Distance

Our path planning algorithm utilizes the velocity transition command to avoid obstacles, and it becomes critical to perform a precise avoidance maneuver around an obstacle if the drift distance cannot be well estimated. Drifting from the desired path is a common phenomenon for UAVs. To minimize this problem, drift distance d_{DR} is used as a primary input. The drift distance varies for different UAV systems' information such as vehicle dynamics, size, mass, configuration, PID gains, max pitch-roll angle, tank size, shape, placement, liquid amount, velocity, etc. The performance of drift distance can be analyzed and identified from both simulation and practical tests in the real world. Because all values for a specific UAV's physical parameters are fixed, this drift distance becomes only the function of the UAV's velocity and liquid level. Since we know the drift distance is the distance between the first time point of receiving stop command and the second time point of actual stop, we calculated the drift distance by testing different liquid levels and velocities in our simulation environment. The results are shown in Table 3. Here, drift distance is a function of different liquid levels and velocities.

Table 3. System identification data table of drift distance.

Drift Distance		**Liquid Level**				
		0	**0.5**	**0.75**	**0.9**	**1.0**
Velocity	0.50	0.26	0.27	0.25	0.27	0.30
	0.75	0.38	0.39	0.38	0.43	0.58
	1.00	0.51	0.52	0.52	0.67	0.94
	1.10	0.56	0.57	0.57	0.78	1.12
	1.25	0.63	0.65	0.71	0.95	1.41

2.3.2. Avoidance Velocity

If we use a large constant avoidance velocity, the drone will have the problem of drifting. On the other hand, if we use a small constant avoidance velocity to solve the drift problem, the time to finish the avoiding process will be too long. Thus, we use a dynamic avoiding velocity, which is related to the information of the sprayer UAV because the sprayer UAV continuously changes its weight, center of mass, and moment of inertia, as mentioned in Section 2.1.1. From this system identification result, we designed suitable avoidance velocity v_A for different liquid levels P_{LQ} based on the following two rules.

Rule 1: The drift distance should not be longer than d_{DRmax}, whose value is $[d_s - (L_{DA} \times \cos\delta)]$. This value is the limitation for extra skid caused by faster velocity. If the velocity creates a larger drift than this value it generates over drifting, even if we execute a stop command on the UAV. The d_{DRMax} range is illustrated in Figure 9.

Rule 2: Thus, we selected v_A from the system identification test. The d_{DR} becomes a constant, and we can obtain a function that fits the relation of liquid level and avoidance velocity. The relationship is shown in Equation (27):

$$v_A = f\{P_{LQ}\} \tag{27}$$

Now, by applying the two rules on the result of system identification, we select a constant drift distance $d_{DR} = 0.59$ m and summarize the avoidance velocity for our UAV model. For liquid levels 0, 0.10, 0.25, 0.50, 0.75, 0.90 and 1.0, the avoidance velocities are 0.75 ms^{-1}, 0.92 ms^{-1}, 1.10 ms^{-1}, 1.17 ms^{-1}, 1.17 ms^{-1}, 1.17 ms^{-1} and 1.17 ms^{-1}, respectively.

We then used the MATLAB curve fitting tool to fit the data. An exponential fit function shown in Equation (28) was used, and the four constants a, b, c, and d of this function were calculated. This formed the relational equation of avoidance velocity and liquid level mentioned in Equation (27).

$$f\{x\} = ae^{bx} + ce^{dx} \tag{28}$$

2.3.3. Triggering Time

In the algorithm, the avoidance process is performed by switching the avoidance velocity command in the x and y-axis. After sensing the border of the obstacle by the sensor S_1 and S_2, the algorithm calculates the time for shifting velocity command from one axis to another. Because we used Equation (27) to obtain the avoidance velocity as a function of water level, the drift distance is also constant for every water level. Therefore, trigger time to activate the next maneuver is derived from time. This trigger time between sensing and shifting command from x to y axes is named t_{TR}, and trigger time is named as D_{TR}, which is illustrated in Figure 9. We used the data of safe distance d_S, drift distance d_{DR}, avoidance velocity v_A, and quadcopter configuration angle α to calculate the triggering time interval t_{TR}. Here, by assuming constant velocity motion:

$$t_{TR} = \frac{d_{TR}}{v_A}$$

$$t_{TR} = \frac{[(d_S - l_{DA} \cdot \cos\alpha) - d_{DR}]}{v_A} \tag{29}$$

3. Numerical Simulation

To show the proposed data-driven dynamic obstacle avoidance algorithm, we performed multiple numerical simulations with MATLAB R2020a. This section first lists the specified constants in the numerical simulations and then presents and discusses the simulation results. All the simulations were performed using parameters from a data-driven control section (Section 2.3).

3.1. Performance Simulation and Results

To show the effectiveness and scalability of this method, we completed simulations in some segments. We tested with a single obstacle from the absolute front, carrying different loads to show how the avoidance results change in segment 1. Then, we tested a side case to verify the safety of the algorithm in segment 2. Then, we performed a coverage mission using random waypoints and random obstacles with lower density to simulate the practical spraying operation in the real world. Lastly, we showed the scalability of the proposed method by increasing the avoidance gap. The performance records are given below, and simulations codes are available from the online repository [55].

3.1.1. Test Segment 1 (Single-Obstacle Avoidance Tests)

In segment 1, we show an ideal case between the UAV and the obstacle. In this case, a circular parallelepiped centered at [0 0 3] with dimensions of 0.2 m $\times$ 0.2 m $\times$ 6 m was considered an obstacle. Then, we set the starting position as the takeoff position at $[-20\ 0\ 0]$, the start going forward point at $[-20 - 0\ 3]$, and the goal point at [20 0 3]. Finally, the velocity of the UAV was set as 3.5 ms$^{-1,}$ and the tank level was set as 0.1 or 10%.

The avoidance trajectory of the UAV using the proposed algorithm is shown in Figure 10. Figure 10a is the top view of the trajectory, where the system has successfully created an avoidance trajectory that looks rectangular, but the bank turn corners are round-shaped from the top view. On the other hand, in Figure 10b, we can see the orthographic view of the trajectory, where the trajectory has a fluctuation at the z-axis. The reason for the corner curves and z-axis fluctuation is that we tried to use the aerodynamics of the bank turn to have a quick movement. The black circle is an obstacle, such as a pole, inside the flying path.

In Figure 10c, sensor response and the flight command are shown. The algorithm reads the sensor data at every step and makes avoidance decisions as velocity commands in the x and y axes. The plots of d_c and d_b are the reading values of the sensor S_0. In d_c, the graph shows that the reading was 20 m from the beginning because the takeoff point started at 20 m away from the obstacle. Besides, there were three fluctuations of reading at the seconds 5, 5.9, and 9.1, and the reason for these fluctuations was the sudden Euler angle change of the UAV that also changed the sensor detection angle. The plots of d_1, d_2, d_3 and d_4 are readings of the single-point distance sensors S_1, S_2, S_3, and S_4, respectively. In this case, the UAV avoided the obstacle from the right side and detected the change in the values d_1 and d_2 of the left-sided sensors S_1 and S_2. Right-sided sensors S_3 and S_4 got no data because the right side was blank during the operation. From the plot of the x-velocity command, the sprayer UAV received a fly forward command at point a ($t = 5$), and the drone started flying towards the goal point. In the x-axis velocity and y-axis velocity graph, at point b ($t = 9.4$), the UAV started deceleration, and at point c ($t = 10.1$), it started maintaining avoidance velocity. From point c to d, it moved forward with avoidance velocity v_A, and from d ($t = 11.4$) it started moving right. At point e ($t = 12.9$), the sensor S_1 confirmed the crossing of the obstacle, and the UAV started moving forward again. At point f ($t = 15.4$), the sensor S_2 confirmed the obstacle as it passed, and it started moving left. Lastly, at point g ($t = 16.6$), it reached the path line. The total flight time was 25.2 s.

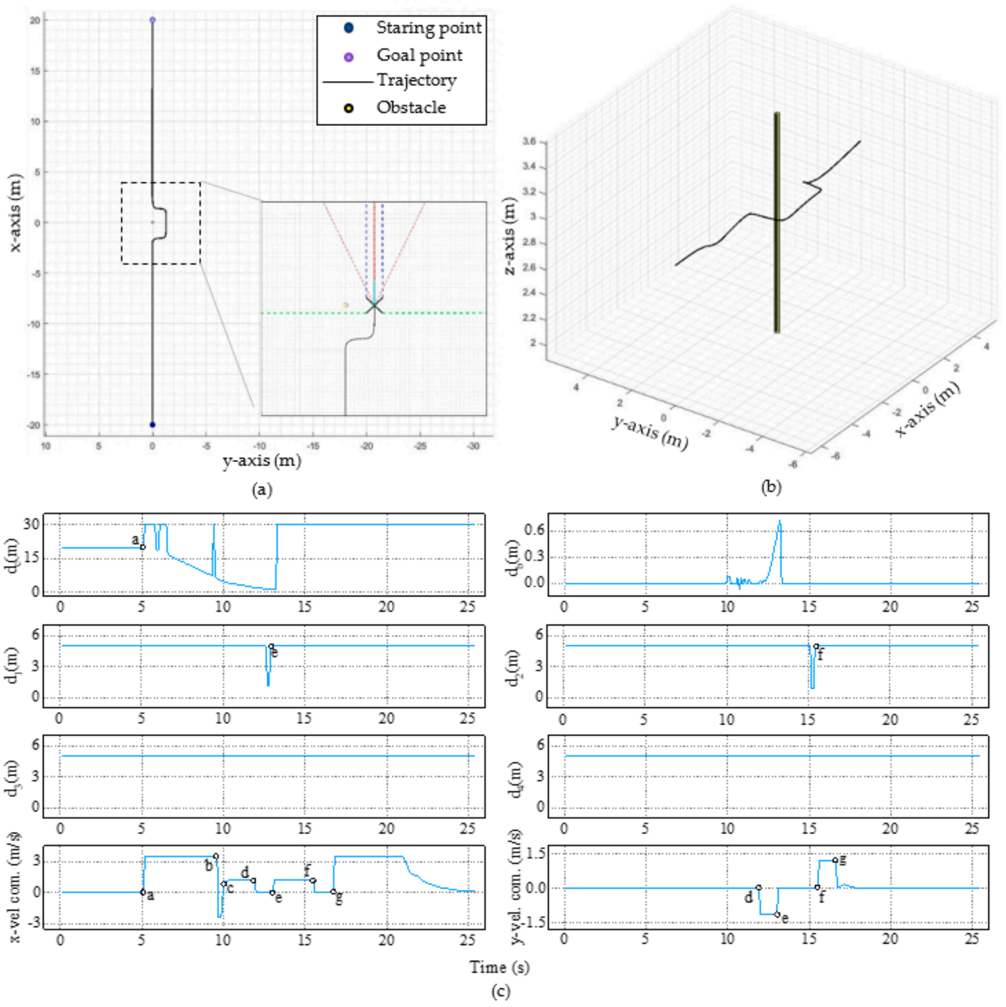

Figure 10. Trajectory of the UAV (**a**) from initial point to goal point with zoom, azimuth: −90°, elevation: 90°; (**b**) orthographic view, azimuth: −67°, elevation: 43°; (**c**) flight command and sensor readings. Sensor readings of d_b, d_c, d_1, d_2, d_3, d_4, velocity command on x-axis and y-axis.

To test the effectiveness of the dynamic algorithm, we did multiple simulations by changing the liquid level. We used liquid levels of 0.10, 0.25, 0.50, 0.75, 0.9 and 1.0 to compare the results. Figure 11 shows the detail of the flight data using six liquid levels. In Figure 11a, the distance changes on the *x*-, *y*- and *z*-axes are shown, where we can observe the position difference of the UAV. On the *x*-axis, every flight followed the same pattern through different flight times. However, on the *y*-axis, the UAV moved slightly further when it carried a lower liquid amount and moved slightly closer when it carried a higher liquid amount. Similarly, on the *z*-axis, with a lower liquid amount, the altitude fell lower than with a higher liquid amount. Figure 11b,c show the dynamic avoidance velocity and, therefore, different Euler angle changes for different liquid levels. The higher value of avoidance velocity and Euler angle for lower liquid load results in faster avoidance, leading to larger altitude drop. Lastly, Figure 11d shows the flight time difference for different liquid levels.

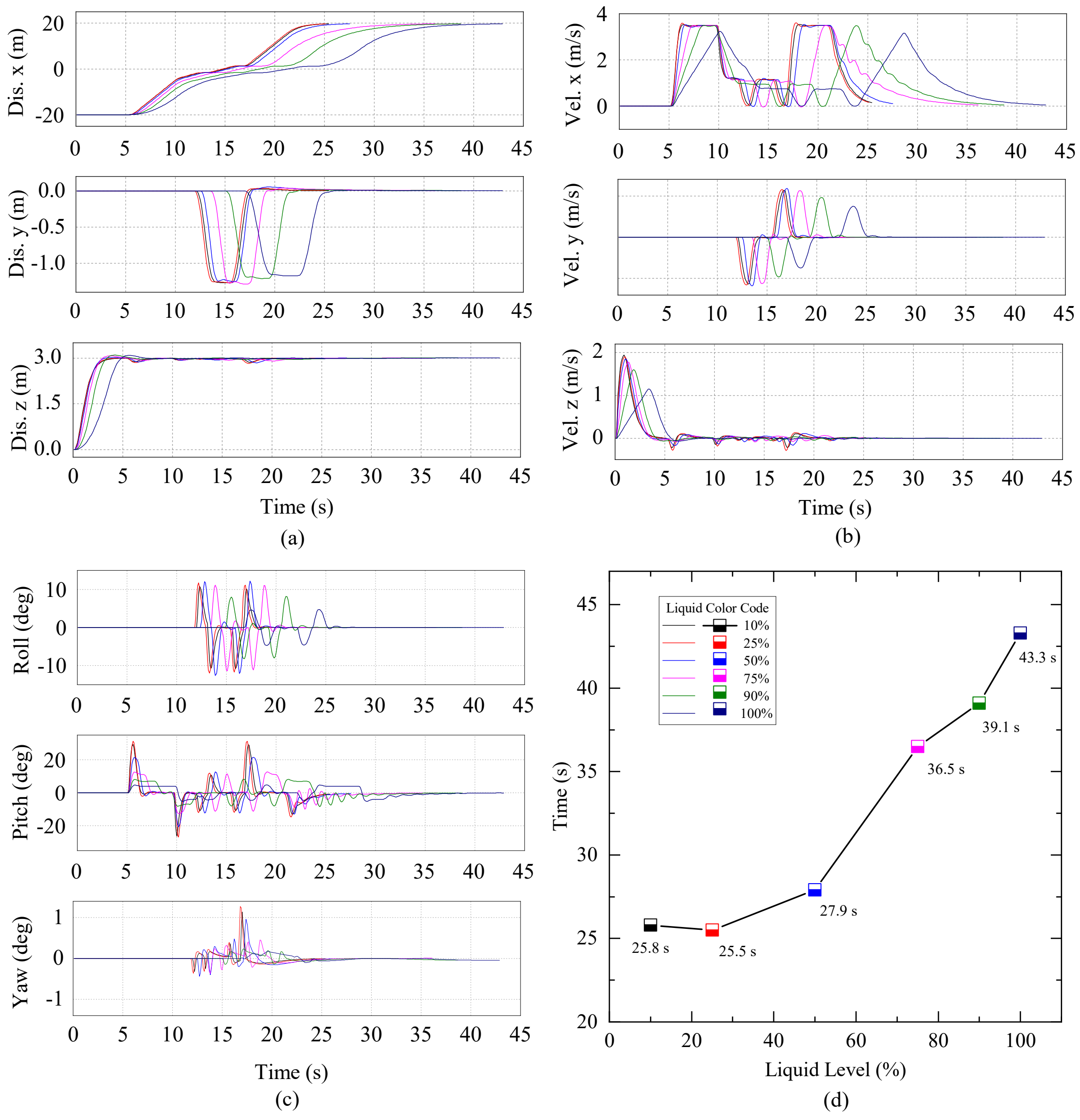

Figure 11. Flight history using different liquid levels. (**a**) *x*, *y*, *z* distance; (**b**) *x*, *y*, *z* velocity; (**c**) roll, pitch and yaw; (**d**) total flight times carrying different liquid level.

The summary of different flight data observations of six liquid levels is given in Table 4, and the path difference is shown in Figure 12. In Table 4, the first result shows the total time of flight for each liquid level, where we can see the lower liquid level took a shorter time than the higher liquid level. Then, the y-axis data show the displacement from the $y = 0$ lines, and the largest distance was 128.6 cm. In these tests, the ideal y-axis avoidance distance was calculated as 1.2 m or 120 cm, which was the sum of the radius of the obstacle, 0.2 m, and the safe distance, 1 m. After that, z-axis fall data showed the altitude fluctuation during the flight, affecting spray deposition. Here, the maximum altitude fall happened with a 25% liquid level. Lastly, we show the maximum and minimum roll angles for different liquid levels; this demonstrates how the algorithm controls the drone to avoid left and right. Figure 12 shows all the avoidance paths of the six liquid level tests. Despite the changing dynamics of the sprayer UAV due to different liquid loads, the proposed dynamic algorithm was verified for precision control since all the avoidance flight paths were very close to the ideal avoidance flight paths specified by the first safe distance and safe distance in Figure 8.

Table 4. Single-obstacle simulation result (ideal case).

Liquid Level	Travel Time	y-Max Move (cm)	z-Max Fall (cm)	Roll Max (deg)	
				Positive	Negative
0.10	25.8	127.1	13.5	10.8	10.8
0.25	25.5	126.8	17.5	11.7	11.9
0.50	27.9	125.2	15.6	12.2	12.6
0.75	36.5	128.6	9.3	11.1	11.4
0.90	39.1	121.1	5.0	8.2	8.2
1.00	43.3	117.4	2.5	4.7	4.7

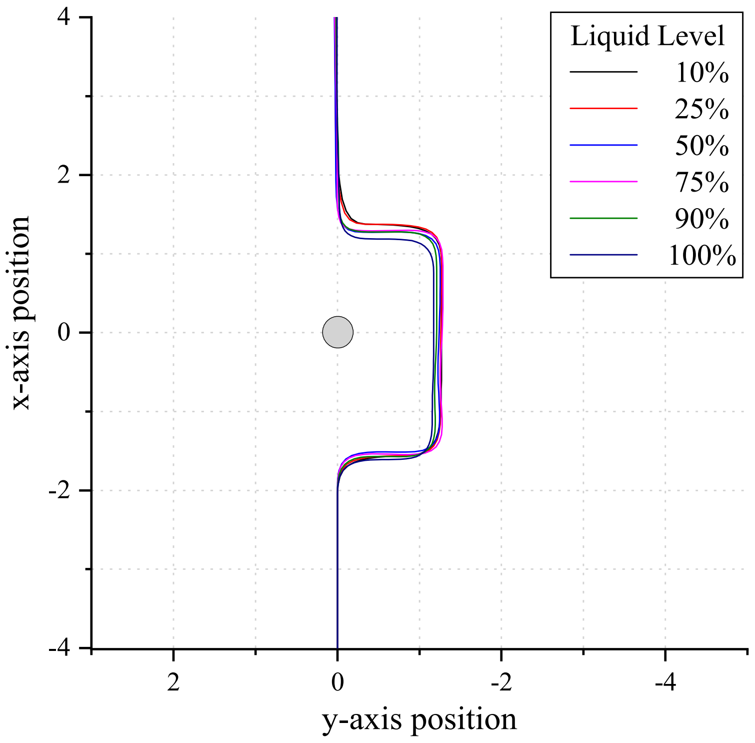

Figure 12. Different flight positions of UAVs carrying different liquid levels.

3.1.2. Test Segment 2 (Possible Obstacle Situations in the Farmland)

This section shows the dynamics of the algorithm where the other possible case can happen during the spraying operation. The first possible case of a single-obstacle test is the corner case described in Figure 8b,c. We used a second single-obstacle simulation to show the corner case avoidance maneuver.

In this test, the initial Y position of the drone was set in such a position so that it would approach the obstacle as the corner case. We set up a rectangular parallelepiped obstacle centered at [0 0 3] with dimensions of 1 m $\times$ 0.4 m $\times$ 6 m, and the tilt angle was $-45°$. The sprayer UAV carried a liquid level of 0.10 or 10%. The takeoff position was at [-20 1 0], the start going forward point was at [-20 1 3], and the goal point was at [20 1 3]. In Figure 13, we have shown the possible three approaches and the avoidance behavior, of where the UAV would approach the obstacle from the left, middle and right side. According to the algorithm's decision-making system described in "Obstacle's direction detection" section, Figure 13a shows that the right-appearing obstacle was avoided from the left direction. Figure 13b,c shows that the left-appearing obstacle was avoided in the right direction and the straight-middle-appearing obstacle was avoided from the left direction. This decision making confirms the shorter avoiding path, which is helpful for spray overlapping.

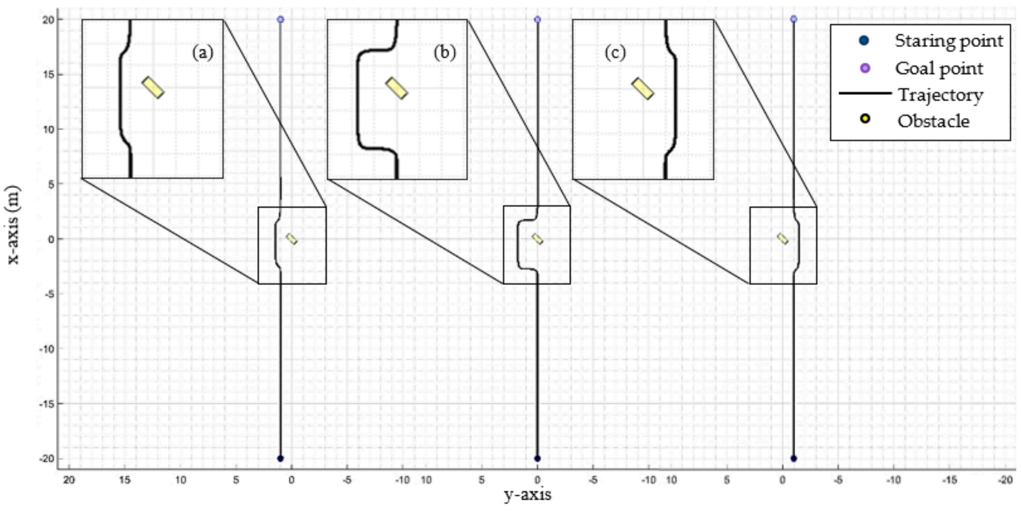

Figure 13. Flight trajectory of three cases avoidance. (**a**) Left case, (**b**) middle case, (**c**) right case.

3.1.3. Test Segment 3 (Spraying Mission Simulation and Performance)

The practical spraying mission was simulated by placing four different obstacles in the test field. Obstacle one was a circular parallelepiped obstacle that was smaller than the UAV, obstacle 2 was a square parallelepiped obstacle, obstacle 3 was a circular parallelepiped obstacle, and obstacle 4 was a rectangular tilted obstacle. Details of the obstacles are given in Table 5. We completed three mission sets with a different initial velocity and gap using this obstacle setup. The initial velocity and path gap were determined following the spray coverage parameters. For example, in the four-meter path gap mission set, the initial velocity was 5 ms^{-1}, the three-meter path gap mission set was performed using 3.5 ms^{-1} initial velocity, and the two-meter path gap mission set was performed using 2 ms^{-1} initial velocity. Besides, every mission set completed full flight using five different liquid level tests, 10%, 25%, 50%, 75%, and 100%, respectively. In the right section of Table 6, the waypoint positions are listed. The first column comprises the waypoint direction numbers, where the UAV follows from one to the next, respectively. The first waypoint was the takeoff point [-20 0 0], and after 5 s, the main flight started. Including the takeoff point, the four-meter path gap mission had nine waypoints, the three-meter path gap mission had eleven waypoints, and the two-meter path gap mission had fifteen waypoints.

Table 5. Obstacle and waypoint information in simulation.

Obstacle's Information					Mission Waypoint Coordinates (x, y, z)			
Obstacle	**Type**	**Size (m)**	**Location (x, y, z)**	**Rotation/Tilt**	**Direction $\Downarrow$**	**4-m Path Gap**	**3-m Path Gap**	**2-m Path Gap**
1	Circular	$0.2 \times 0.2 \times 6$	$[0 - 3\ 3]$	$[0\ 0\ 0]$	1	$[-20\ 0\ 0]$	$[-20\ 0\ 0]$	$[-20\ 0\ 0]$
2	Squire	$2 \times 2 \times 6$	$[10 - 3\ 3]$	$[0\ 0\ 0]$	2	$[-20\ 0\ 3]$	$[-20\ 0\ 3]$	$[-20\ 0\ 3]$
3	Circular	$0.75 \times 0.75 \times 6$	$[-10 - 9\ 3]$	$[0\ 0\ 0]$	3	$[20\ 0\ 3]$	$[20\ 0\ 3]$	$[20\ 0\ 3]$
4	Rectangular	$1 \times 0.4 \times 6$	$[15 - 9\ 3]$	$[0\ 0 - 45°]$	4	$[20 - 4\ 3]$	$[20 - 3\ 3]$	$[20 - 2\ 3]$
					5	$[-20 - 4\ 3]$	$[-20 - 3\ 3]$	$[-20 - 2\ 3]$
					6	$[-20 - 8\ 3]$	$[-20 - 6\ 3]$	$[-20 - 4\ 3]$
					7	$[20 - 8\ 3]$	$[20 - 6\ 3]$	$[20 - 4\ 3]$
					8	$[20 - 12\ 3]$	$[20 - 9\ 3]$	$[20 - 6\ 3]$
					9	$[-20 - 12\ 3]$	$[-20 - 9\ 3]$	$[-20 - 6\ 3]$
					10		$[-20 - 12\ 3]$	$[-20 - 8\ 3]$
					11		$[20 - 12\ 3]$	$[20 - 8\ 3]$
					12			$[20 - 10\ 3]$
					13			$[-20 - 10\ 3]$
					14			$[-20 - 12\ 3]$
					15			$[20 - 12\ 3]$

Table 6. Five liquid level mission simulation results.

Configuration	Liquid Level (%)	Mission Time (s)	Average z-Axis Fall (cm)	Actual Length (m)	Increased Length (m)	Area Coverage (%)
Path gap 4 m, Velocity 5 ms^{-1}, Path length 175 m, Total area 704 m^2	10	121.9	3.13	177.13	2.13	98.64
	25	124.2	4.09	177.67	2.67	98.58
	50	140.3	3.29	177.84	2.84	98.63
	75	198.1	1.72	178.12	3.12	98.9
	100	205.9	0.56	177.13	2.13	99.23
Path gap 3 m, Velocity 3.5 ms^{-1}, Path length 215 m, Total area 645 m^2	10	164.1	1.74	224.63	9.63	98.26
	25	165.2	2.30	225.03	10.03	98.22
	50	184.7	2.07	225.19	10.19	98.30
	75	254.6	1.18	225.00	10.00	98.55
	100	294.8	0.34	224.55	9.55	98.98
Path gap 2 m, Velocity 2.5 ms^{-1}, Path length 295 m, Total area 588 m^2	10	240.9	0.80	299.10	4.10	98.05
	25	243.3	1.13	299.68	4.69	98.08
	50	263.3	1.27	300.65	5.65	98.02
	75	338.8	0.95	301.22	6.22	98.13
	100	389.5	0.33	299.17	4.17	98.47

In total, 15 simulations were completed, and the results are combined and plotted in Figure 14. These figures include all five liquid level test results for each mission set. Figure 14a shows the flight paths of the four-meter path gap mission, Figure 14b shows the flight paths of the three-meter path gap mission, and Figure 14c shows the flight paths of the two-meter path gap mission. The yellow-colored blocks are the obstacles inside the mission path, and the dotted black lines are the boundaries of each spraying mission. These tests were completed following the mission operation of the plant protection UAV inside the plant field [42].

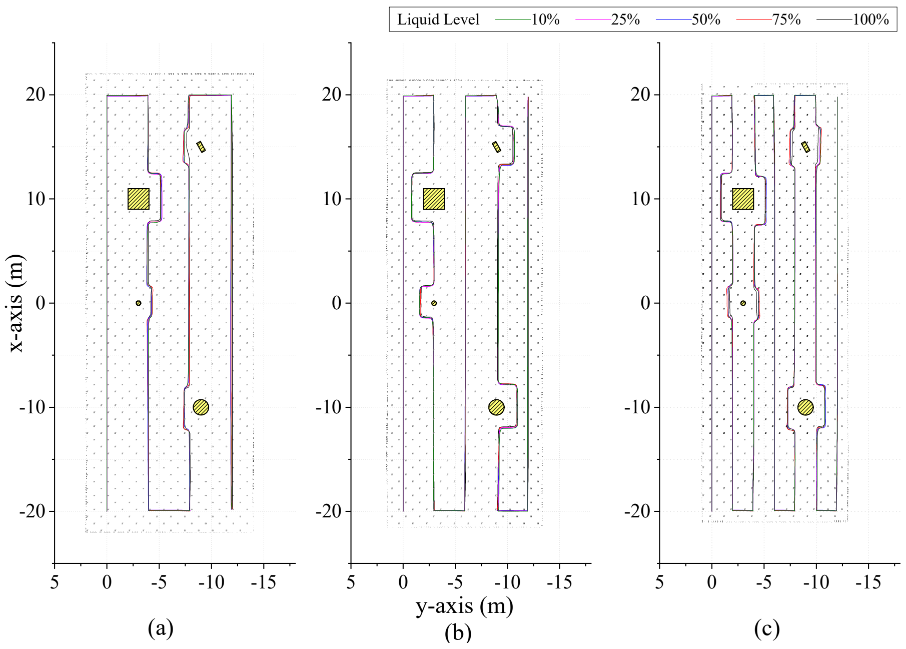

Figure 14. Different actions during different scenarios in farmland operation. (**a**) 4-m path gap, (**b**) 3-m path gap, (**c**) 2-m path gap.

In order to characterize how well our algorithm worked, we needed to know the spraying cover area after the UAV completed the mission. First, we calculated the spraying cover area with no obstacles in the field. This value was called the ideal coverage area A_{CI}, which can be calculated using Equation (30). For all the simulation tests, X_M was a constant 40 m, Y_M was a constant 10 m, and L_{SP} was the path gap of each mission set.

$$A_{CI} = (X_M + L_{SP})\cdot(Y_M + L_{SP}) \tag{30}$$

Then, we estimated the actual spraying coverage area A_C from the UAV's X and Y paths, which are shown in Figure 15. Finally, we characterized the performance of the algorithm in terms of the area coverage by calculating the cover area percentage A_{CP}. A higher value of A_{CP} means that the UAV can still cover the spraying area even though it was required to complete several avoidance maneuvers during the spraying mission. The cover area percentage is shown in Equation (31). Figure 15 illustrates the spraying coverage area A_C and the ideal coverage area A_{CI} by the 2-m path gap mission with 10% liquid load.

$$A_{CP} = 100\%\cdot\frac{A_C}{A_{CI}} \tag{31}$$

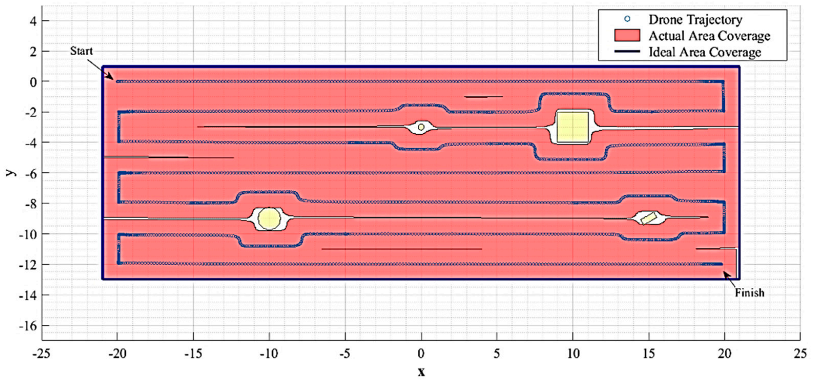

Figure 15. Spray coverage in 2-m path gap mission with 10% liquid load.

We also calculated the actual length of the path and the ideal length of the path to observe the extra distance travelled by the UAV. The ideal length of the four-meter path gap mission was 175 m, the ideal length of the three-meter path gap mission was 215 m, and the ideal length of the two-meter path gap mission was 295 m.

The summary of the simulation results using three sets of mission parameters is shown in Table 6. For each test, total mission time, average z-axis fall, which represents the altitude error during the mission, actual length, increased length from the ideal length, and lastly, the coverage area percentages are included. According to the simulation results, the algorithm can reduce flight time significantly while the liquid level is lower. We used the same PID controller for the altitude control for every mission flight. According to the average altitude decrease data during the mission, this PID is suitable for an initial fly velocity ranging from 2 ms^{-1} to 5 ms^{-1}. For a mission that requires higher initial velocity, the altitude error can be further reduced by fine-tuning the altitude PID controller. The flight length increases during medium-level liquid, and extended length depends on the obstacle and the UAV's positions during the mission. Due to the faster approach, while carrying lower liquid, the coverage area percentage is lower for low liquid levels, and the cover area percentage is higher when the fluid level is high. However, the cover area percentages of all the simulation tests have values ranging from 96.34% to 99.23%, demonstrating that this dynamic algorithm can reduce the impact of varying liquid load on the spraying coverage area.

3.1.4. Test Segment 4 (Scalability)

In this testing section, we will show the scalability and behavior changes depending on data. Previous test segments used a safety distance of one meter for all experiments. However, the farmland's environment may be affected by heavy wind, and in such cases we must increase the UAV's safety area. Here, we set the safety distance $d_s = 2$ m, and run the simulation on a three-meter path gap mission. We performed the simulation with 10%, 50% and 100% load to show the variance while avoiding, as shown in Figure 16. The area coverage was 95.41% for 10% load, 94.87% for 50% load, and 95.35% for 100% load, where the velocity was 3.5 ms^{-1}.

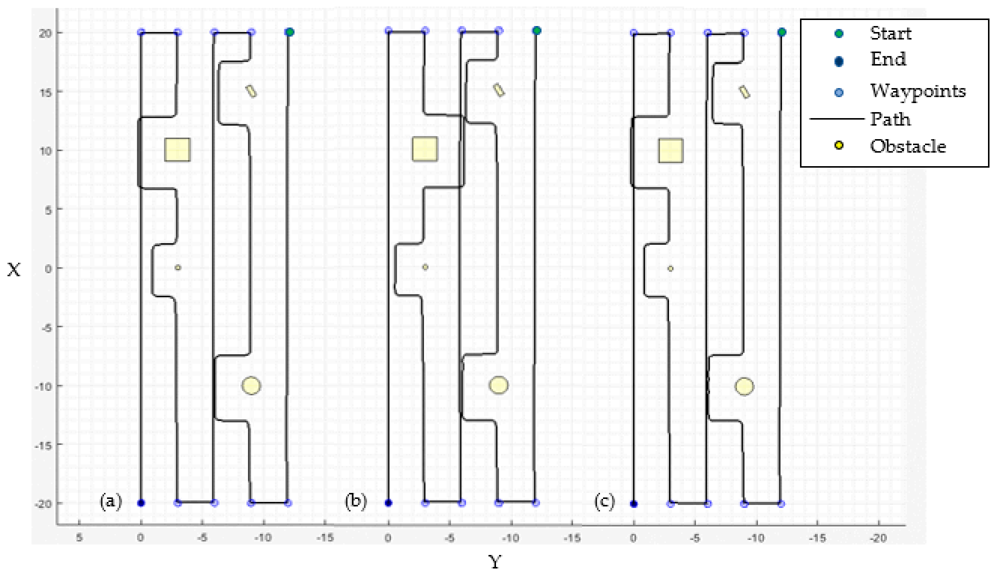

Figure 16. Area coverage in 3-m path gap mission with (**a**) 10% liquid load, (**b**) 50% liquid load, (**c**) 100% liquid load.

The algorithm was experimented with and validated by substantial numerical simulations to prove the concept. The liquid career quadcopter model was created using 6-DOF (degrees of freedom) dynamics and a rigid weight under the UAV as a tank load. In a real tank, fluids react differently during transportation, which may cause slosh force and influence flight data. Although we experimented with and developed a method to reduce repetitive slosh force [45], the liquid slosh force and multiple spray nozzle force still need to be considered when applying the proposed method.

4. Conclusions

A data-driven real-time obstacle avoidance algorithm and a sensor setup were proposed for agricultural plant protector UAVs, including liquid tanks. The method demanded system identification tests to acquire appropriate data for individual UAV models and the tank. The sensor setup used millimeter-wave radar and four single-point laser sensors. DFA, or the deterministic finite automata method, was used to form the algorithm. Since the pitch and roll axes had faster responses than the yaw axes for a normal quadcopter, the algorithm was designed to avoid obstacles by shifting the avoidance velocity command in the x and y axes instead of changing the yaw angle. The algorithm can avoid obstacles by shifting the avoidance velocity command in the x and y axes. The value of avoidance velocity was dynamic according to liquid level during mission flight. A single-obstacle simulation test was performed using a 20 kg UAV with a maximum 32 L tank capacity. The performance showed that, with 10%, 25%, 50%, 75%, 90% and 100% tank load, flight time took 25.8 s, 25.5 s, 27.9 s, 36.5 s, 39.1 s and 43.3 s, respectively. These results verified that the proposed algorithm achieved time-saving and precise obstacle avoidance objectives using real-time sensor data. The spraying mission simulation test showed that the proposed algorithm avoided common obstacles in farmland while maintaining a high cover area percentage. The 4-m, 3-m, and 2-m path gap missions covered a minimum of 98.58%, 98.22%, and 98.05, and a maximum of 99.23%, 98.98, and 98.47 farmland area, respectively. Lastly, increased safe distance performance showed the scalability of the algorithm. In the future, actual flight performance using an embedded system mounted on a quadcopter with the liquid tank will be performed to validate the proposed algorithm. This method will be implemented using multiple UAV models with different tank sizes to show the effectiveness of this data-driven approach. The spraying performance will be performed to show the spray coverage efficiency. This algorithm was created concerning spray coverage area, not spray overlapped area; variable rate spray control technology will also be used with this algorithm to optimize spray coverage. This method was developed to target vertical static obstacles in farmland. The limitations of this algorithm include horizontal hanging connection lines, flying objects, etc., which can be improved in the future.

Author Contributions: Conceptualization, S.A., B.Q.; methodology, S.A., B.Q. and C.-W.K.; validation, F.A.; formal analysis, C.-W.K., H.X.; investigation, F.A., J.L., H.X.; resources, H.X.; data curation, S.A., C.-W.K., J.L.; writing—original draft preparation, S.A.; writing—review and editing, S.A., B.Q., C.-W.K., J.L.; supervision, B.Q.; project administration, B.Q., F.A.; funding acquisition, B.Q.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation (no. 31971790), the Key Research and Development Program of Jiangsu province, China (no. BE2020328) and A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (no. PAPD-2018-87).

Data Availability Statement: Date is available in the manuscript.

Acknowledgments: The authors acknowledge the support of the School of Agricultural Engineering, Jiangsu University, China.

Conflicts of Interest: The authors declare no conflict of interest. And the funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Suprem, A.; Mahalik, N.; Kim, K. A review on application of technology systems, standards and interfaces for agriculture and food sector. *Comput. Stand. Interfaces* **2013**, *35*, 355–364. [CrossRef]
2. Mogili, U.R.; Deepak, B.B.V.L. Review on Application of Drone Systems in Precision Agriculture. *Procedia Comput. Sci.* **2018**, *133*, 502–509. [CrossRef]
3. Sanca, A.S.; Alsina, P.J.; Jés de Jesus, F.C. Dynamic modelling of a quadrotor aerial vehicle with nonlinear inputs. In Proceedings of the 2008 IEEE Latin American Robotic Symposium, Natal, Brazil, 29–30 October 2008; pp. 143–148. [CrossRef]
4. Ryll, M.; Bülthoff, H.; Giordano, P.R. A Novel Overactuated Quadrotor Unmanned Aerial Vehicle: Modeling, Control, and Experimental Validation. *IEEE Trans. Control Syst. Technol.* **2014**, *23*, 540–556. [CrossRef]
5. Marino, S.; Alvino, A. Detection of Spatial and Temporal Variability of Wheat Cultivars by High-Resolution Vegetation Indices. *Agronomy* **2019**, *9*, 226. [CrossRef]
6. Surový, P.; Ribeiro, N.A.; Panagiotidis, D. Estimation of positions and heights from UAV-sensed imagery in tree plantations in agrosilvopastoral systems. *Int. J. Remote Sens.* **2018**, *39*, 4786–4800. [CrossRef]
7. Cilia, C.; Panigada, C.; Rossini, M.; Meroni, M.; Busetto, L.; Amaducci, S.; Boschetti, M.; Picchi, V.; Colombo, R. Nitrogen Status Assessment for Variable Rate Fertilization in Maize through Hyperspectral Imagery. *Remote Sens.* **2014**, *6*, 6549–6565. [CrossRef]
8. Zaman-Allah, M.; Vergara, O.; Araus, J.L.; Tarekegne, A.; Magorokosho, C.; Zarco-Tejada, P.J.; Hornero, A.; Albà, A.H.; Das, B.; Craufurd, P.; et al. Unmanned aerial platform-based multi-spectral imaging for field phenotyping of maize. *Plant Methods* **2015**, *11*, 35. [CrossRef]
9. Chang, A.; Jung, J.; Maeda, M.M.; Landivar, J. Crop height monitoring with digital imagery from Unmanned Aerial System (UAS). *Comput. Electron. Agric.* **2017**, *141*, 232–237. [CrossRef]
10. Honkavaara, E.; Kaivosoja, J.; Mäkynen, J.; Pellikka, I.; Pesonen, L.; Saari, H.; Salo, H.; Hakala, T.; Marklelin, L.; Rosnell, T. Hyperspectral reflectance signatures and point clouds for precision agriculture by light weight uav imaging system. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *I-7*, 353–358. [CrossRef]
11. Pflanz, M.; Nordmeyer, H.; Schirrmann, M. Weed Mapping with UAS Imagery and a Bag of Visual Words Based Image Classifier. *Remote Sens.* **2018**, *10*, 1530. [CrossRef]
12. Rasmussen, J.; Nielsen, J.; Garciaruiz, F.; Christensen, S.; Streibig, J.C. Potential uses of small unmanned aircraft systems (UAS) in weed research. *Weed Res.* **2013**, *53*, 242–248. [CrossRef]
13. Rahnemoonfar, M.; Sheppard, C. Deep Count: Fruit Counting Based on Deep Simulated Learning. *Sensors* **2017**, *17*, 905. [CrossRef]
14. Lou, Z.; Xin, F.; Han, X.; Lan, Y.; Duan, T.; Fu, W. Effect of Unmanned Aerial Vehicle Flight Height on Droplet Distribution, Drift and Control of Cotton Aphids and Spider Mites. *Agronomy* **2018**, *8*, 187. [CrossRef]
15. Xiao, Q.; Xin, F.; Lou, Z.; Zhou, T.; Wang, G.; Han, X.; Lan, Y.; Fu, W. Effect of Aviation Spray Adjuvants on Defoliant Droplet Deposition and Cotton Defoliation Efficacy Sprayed by Unmanned Aerial Vehicles. *Agronomy* **2019**, *9*, 217. [CrossRef]
16. Ahmad, F.; Qiu, B.; Dong, X.; Ma, J.; Huang, X.; Ahmed, S.; Chandio, F.A. Effect of operational parameters of UAV sprayer on spray deposition pattern in target and off-target zones during outer field weed control application. *Comput. Electron. Agric.* **2020**, *172*, 105350. [CrossRef]
17. Liu, A.; Zhang, H.; Liao, C.; Zhang, Q.; Cenglin, X.; Juying, H.; Zhang, J.; Yan, H.; Li, J.; Xiwen, L. Effects of Supplementary Pollination by Single-rotor Agricultural Unmanned Aerial Vehicle in Hybrid Rice Seed Production. *Agric. Sci. Technol.* **2017**, *18*, 543–552.
18. Xiongkui, H.; Bonds, J.; Herbst, A.; Langenakens, J. Recent development of unmanned aerial vehicle for plant protection in East Asia. *Int. J. Agric. Biol. Eng.* **2017**, *10*, 18–30. [CrossRef]
19. Xue, X.; Lan, Y.; Sun, Z.; Chang, C.; Hoffmann, W.C. Develop an unmanned aerial vehicle based automatic aerial spraying system. *Comput. Electron. Agric.* **2016**, *128*, 58–66. [CrossRef]
20. Chakravarthy, A.; Ghose, D. Obstacle avoidance in a dynamic environment: A collision cone approach. *IEEE Trans. Syst. Man, Cybern.-Part A Syst. Hum.* **1998**, *28*, 562–574. [CrossRef]
21. Fiorini, P.; Shiller, Z. Motion Planning in Dynamic Environments Using Velocity Obstacles. *Int. J. Robot. Res.* **1998**, *17*, 760–772. [CrossRef]
22. Kumar, B.A.; Ghose, D. Radar-assisted collision avoidance/guidance strategy for planar flight. *IEEE Trans. Aerosp. Electron. Syst.* **2001**, *37*, 77–90. [CrossRef]
23. Olivares-Mendez, M.A.; Mejias, L.; Campoy, P.; Mellado-Bataller, I. Cross-Entropy Optimization for Scaling Factors of a Fuzzy Controller: A See-and-Avoid Approach for Unmanned Aerial Systems. *J. Intell. Robot. Syst.* **2012**, *69*, 189–205. [CrossRef]
24. Zou, Y.; Wang, C.; Wang, J.; Yan, G. Obstacle-Avoidance Control Method for Unmanned Aerial Vehicle (UAV), Flight Controller and Uav. U.S. Patent 20190271992A1, 5 September 2019.
25. Zou, Y. Method of Controlling Obstacle Avoidance for Unmanned Aerial Vehicle and Unmanned Aerial Vehicle. U.S. Patent 20190278303A1, 12 September 2019.
26. Richards, A.; How, J.P. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301), Anchorage, AK, USA, 8–10 May 2002; pp. 1936–1941. [CrossRef]
27. Lozano-Perez, T.; Wesley, M.A. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM* **1979**, *22*, 560–570. [CrossRef]

28. Wang, X.; Yadav, V.; Balakrishnan, S.N. Cooperative UAV Formation Flying with Obstacle/Collision Avoidance. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 672–679. [CrossRef]
29. Park, J.; Baek, H. Stereo vision based obstacle collision avoidance for a quadrotor using ellipsoidal bounding box and hierarchical clustering. *Aerosp. Sci. Technol.* **2020**, *103*, 105882. [CrossRef]
30. Ferrick, A.; Fish, J.; Venator, E.; Lee, G.S. UAV obstacle avoidance using image processing techniques. In Proceedings of the 2012 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), Woburn, MA, USA, 23–24 April 2012; pp. 73–78. [CrossRef]
31. Naderhirn, M.; Langthaler, P.; del Re, L. Robust Hybrid Control for Unknown Obstacle Avoidance. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Chicago, IL, USA, 10–13 August 2009; p. 6181. [CrossRef]
32. Shanmugavel, M.; Tsourdos, A.; White, B.A. Collision avoidance and path planning of multiple UAVs using flyable paths in 3D. In Proceedings of the 2010 15th International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland, 23–26 August 2010; pp. 218–222. [CrossRef]
33. Lalish, E.; Morgansen, K.A.; Tsukamaki, T. Decentralized reactive collision avoidance for multiple unicycle-type vehicles. In Proceedings of the 2008 American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 5055–5061. [CrossRef]
34. Smith, A.; Harmon, F. UAS collision avoidance algorithm minimizing impact on route surveillance. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Chicago, IL, USA, 10–13 August 2009; p. 6179. [CrossRef]
35. Cai, X.; De Queiroz, M. Adaptive Rigidity-Based Formation Control for Multirobotic Vehicles with Dynamics. *IEEE Trans. Control Syst. Technol.* **2014**, *23*, 389–396. [CrossRef]
36. Huang, S.; Teo, R.S.H.; Tan, K.K. Collision avoidance of multi unmanned aerial vehicles: A review. *Annu. Rev. Control* **2019**, *48*, 147–164. [CrossRef]
37. Huang, X.; Dong, X.; Ma, J.; Liu, K.; Ahmed, S.; Lin, J.; Qiu, B. The Improved A* Obstacle Avoidance Algorithm for the Plant Protection UAV with Millimeter Wave Radar and Monocular Camera Data Fusion. *Remote Sens.* **2021**, *13*, 3364. [CrossRef]
38. Oksanen, T.; Visala, A. Coverage path planning algorithms for agricultural field machines. *J. Field Robot.* **2009**, *26*, 651–668. [CrossRef]
39. Bochtis, D.D.; Sørensen, C.G.; Busato, P.; Berruto, R. Benefits from optimal route planning based on B-patterns. *Biosyst. Eng.* **2013**, *115*, 389–395. [CrossRef]
40. Fengbo, Y.; Xinyu, X.; Ling, Z.; Zhu, S. Numerical simulation and experimental verification on downwash air flow of six-rotor agricultural unmanned aerial vehicle in hover. *Int. J. Agric. Biol. Eng.* **2017**, *10*, 41–53. [CrossRef]
41. Bodur, M.; Mehrolhassani, M.B.A.M. Satellite Images-based Obstacle Recognition and Trajectory Generation for Agricultural Vehicles. *Int. J. Adv. Robot. Syst.* **2015**, *12*, 188. [CrossRef]
42. Ahmed, S.; Qiu, B.; Ahmad, F.; Kong, C.-W.; Xin, H. A State-of-the-Art Analysis of Obstacle Avoidance Methods from the Perspective of an Agricultural Sprayer UAV's Operation Scenario. *Agronomy* **2021**, *11*, 1069. [CrossRef]
43. Zhou, K.; Jensen, A.L.; Sørensen, C.G.; Busato, P.; Bothtis, D. Agricultural operations planning in fields with multiple obstacle areas. *Comput. Electron. Agric.* **2014**, *109*, 12–22. [CrossRef]
44. Wang, K.; Meng, Z.; Wang, L.; Wu, Z.; Wu, Z. Practical Obstacle Avoidance Path Planning for Agriculture UAVs. In *Advances and Trends in Artificial Intelligence. From Theory to Practice*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 196–203. [CrossRef]
45. Ahmed, S.; Xin, H.; Faheem, M.; Qiu, B. Stability Analysis of a Sprayer UAV with a Liquid Tank with Different Outer Shapes and Inner Structures. *Agriculture* **2022**, *12*, 379. [CrossRef]
46. Lan, Y.; Chen, S. Current status and trends of plant protection UAV and its spraying technology in China. *Int. J. Precis. Agric. Aviat.* **2018**, *1*, 1–9. [CrossRef]
47. Lan, Y.; Wang, G. Development Situation and Prospect of Plant Protection UAV in China. *Agric. Eng. Technol.* **2018**, *38*, 17–27. [CrossRef]
48. Quan, Q. *Introduction to Multicopter Design and Control*, 1st ed.; Springer: Singapore, 2017; pp. 99–120.
49. Xiang, J.; Zhang, M. *Millimeter-Wave Radar and Its Applications*; National Defense Industry Press: Beijing, China, 2005.
50. Donges, A.; Noll, R. *Laser Measurement Technology*; Springer: Berlin/Heidelberg, Germany, 2015. [CrossRef]
51. Antonucci, F.; Armano, M.; Audley, H.; Auger, G.; Benedetti, M.; Binetruy, P.; Bogenstahl, J.; Bortoluzzi, D.; Bosetti, P.; Brandt, N.; et al. The LISA Pathfinder mission. *Class. Quantum Gravity* **2012**, *29*, 124014. [CrossRef]
52. Rahman, M.; Fan, S.; Zhang, Y.; Chen, L. A Comparative Study on Application of Unmanned Aerial Vehicle Systems in Agriculture. *Agriculture* **2021**, *11*, 22. [CrossRef]
53. Rankin, G.; Tirkel, A.; Leukhin, A. Millimeter wave array for UAV imaging MIMO radar. In Proceedings of the 2015 16th International Radar Symposium (IRS), Dresden, Germany, 24–26 June 2015; pp. 499–504. [CrossRef]
54. Pham, D.D.; Suh, Y.S. Remote length measurement system using a single point laser distance sensor and an inertial measurement unit. *Comput. Stand. Interfaces* **2016**, *50*, 153–159. [CrossRef]
55. Ahmed, S. Dynamic_OA_for_Sprayer_UAV. Available online: https://github.com/shibbir7ahmed/Dynamic_OA_for_Sprayer_UAV (accessed on 26 March 2022).