

## Article

# Vision-Based UAV Landing with Guaranteed Reliability in Adverse Environment

Zijian Ge <sup>1</sup>, Jingjing Jiang <sup>1,\*</sup> , Ewan Pugh <sup>2</sup>, Ben Marshall <sup>1</sup>, Yunda Yan <sup>3</sup>  and Liang Sun <sup>4</sup> 

<sup>1</sup> Department of Aeronautical and Automotive Engineering, Loughborough University, Leicester LE11 3TU, UK

<sup>2</sup> Fuel and Landing Gear Flight Test Analysis, Airbus Operations Limited, Bristol BS34 7PA, UK

<sup>3</sup> School of Engineering and Sustainable Development, De Montfort University, Leicester LE1 9BH, UK

<sup>4</sup> School of Intelligence Science and Technology, University of Science and Technology Beijing, Beijing 100081, China

\* Correspondence: j.jiang2@lboro.ac.uk

**Abstract:** Safe and accurate landing is crucial for Unmanned Aerial Vehicles (UAVs). However, it is a challenging task, especially when the altitude of the landing target is different from the ground and when the UAV is working in adverse environments, such as coasts where winds are usually strong and changing rapidly. UAVs controlled by traditional landing algorithms are unable to deal with sudden large disturbances, such as gusts, during the landing process. In this paper, a reliable vision-based landing strategy is proposed for UAV autonomous landing on a multi-level platform mounted on an Unmanned Ground Vehicle (UGV). With the proposed landing strategy, visual detection can be retrieved even with strong gusts and the UAV is able to achieve robust landing accuracy in a challenging platform with complex ground effects. The effectiveness of the landing algorithm is verified through real-world flight tests. Experimental results in farm fields demonstrate the proposed method's accuracy and robustness to external disturbances (e.g., wind gusts).

**Keywords:** UAV; autonomous landing; AprilTags



**Citation:** Ge, Z.; Jiang, J.; Pugh, E.; Marshall, B.; Yan, Y.; Sun, L. Vision-Based UAV Landing with Guaranteed Reliability in Adverse Environment. *Electronics* **2023**, *12*, 967. <https://doi.org/10.3390/electronics12040967>

Academic Editor: Shiho Kim

Received: 10 January 2023

Revised: 6 February 2023

Accepted: 13 February 2023

Published: 15 February 2023



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

## 1. Introduction

Due to the flexibility and generally decreasing cost of the Unmanned Aerial Vehicle (UAV) systems, UAVs have become more prevalent recently and have been widely used in a variety of applications, such as agriculture monitoring [1], terrain mapping [2–4], and assisting human rescue [5,6]. In particular, for precision agriculture, according to the Association for Unmanned Vehicle Systems International (AUVSI), 80% of UAVs will be deployed for agriculture purposes shortly, as the investment for the agriculture sector has increased by 80% in the last five years, aims to achieve productivity growth of 70% by 2050 [7,8].

UAV missions normally consist of five phases: take off, climb, cruise, descent, and landing [9]. One of the most challenging phases of an intelligent UAV is landing. It was found that most drone crash accidents occur in the landing stage, as a cushion of air is generated underneath the drone while it is landing. Any minor operation error could easily tip the drone over and result in a catastrophic crash [10]. The situation is even worse and more risky in an environment with large but unpredictable wind disturbances.

Typically, a landing control system requires sensors to provide position information. A commonly equipped global position system (GPS) is largely used for the autonomous navigation of UAVs. However, the GPS signal may be unavailable in some remote areas. In addition, its measurements are inaccurate (with errors of 2–5 m) [9]. Even though some techniques have been applied to improve the UAVs' localization performance such as the sensor fusion method in [11], the estimated positions still cannot meet the requirements of precise landing. Hence, close-range sensors, such as cameras and Lidars, are used

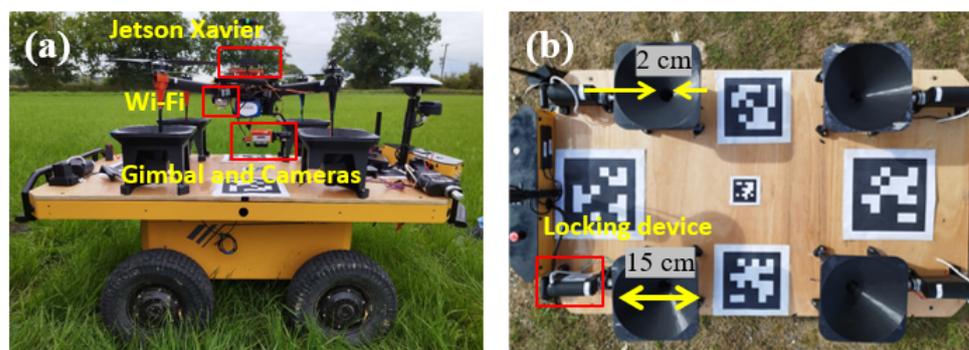
to provide accurate position measurements. There is plenty of research focusing on the vision-based UAV autonomous landing, such as using a Pan-Tilt-Based Visual Servoing (PTBVS) system in GPS-denied environments for auto-landing [12] and introducing feature-based image matching strategy to find the natural landmarks which are then be used for UAV navigation and landing [13]. Visual algorithms based on AprilTag detection via a monocular camera can also be applied to locate the UAV [14]. A more comprehensive survey focuses on the vision-based autonomous landing of the UAV is well-studied in [15].

With the support of the onboard processor, the above-mentioned methods can guarantee the real-time accuracy of the UAV's localization. Subsequently, different control methods, such as PID controllers, are developed based on the measurement of altitude between the UAV and the ground platform [12,14,16,17]. Another group of widely used control techniques in autonomous landing is optimal control methods, such as Model Predictive Control (MPC) [18–20] and Linear Quadratic Regulator (LQR) control [21]. In addition, with the rapid evolution of machine learning techniques, learning-based landing algorithms are also implemented to conduct the auto-landing task. In [22], a Neural Network (NN) based controller combined with Reinforcement Learning (RL) and Proximal Policy Optimization (PPO) is proposed aiming to optimally control the drone approach to the landing target. A Deep Deterministic Policy Gradient (DDPG) algorithm based on the idea of Deep Q-Learning is proposed for correcting UAV's landing maneuver [23]. In deep learning, a Region-Based Convolutional Neural Network (R-CNN) feature matching algorithm is used to output the decision of landing based on the calculated area safety level [24]. Similarly, CNN model in [25] is used to classify the landing environments.

Most of the existing research assumes that the landing environment is certain, without considering strong winds or gusts. However, our UAV landing system is mainly designed for deploying in farms that are open fields and subject to large wind disturbances. Therefore, one of the main challenges for our landing system is how to deal with such disturbances, particularly consistent winds and gust winds. These factors are vital to the reliable and safe operation of the UAV, hence cannot be ignored when designing the control algorithm. To deal with external wind disturbances, many motion control algorithms are investigated. For example, a control law combined with the analyzed wind Computational Fluid Dynamics (CFD) model is designed to stabilize the drone's hover position [26]. An ultrasonic wind sensor was mounted on the drone to measure the real-time wind parameters that can be further utilized as feedback to the controller [27]. Alternatively, wind disturbances to the UAV can be estimated via the nonlinear Disturbance Observer (DOB) which can then be compensated by Disturbance Observer-Based Control (DOBC) designs [28–30]. Whereas, there are limited studies for vision-based landing systems, focusing on scenarios when the vision is lost during the landing due to wind disturbances which results in no feedback signal to the controller. Hence, a high-level landing strategy that can guarantee and retrieve visual detection is necessary for UAV's safe landing in adverse environments.

Another challenge in our work is introduced by the unique structure of the landing platform used for UAV docking shown in Figure 1. As can be seen from Figure 1b, the docking device consists of four small conical structures and the size of which is highlighted. Mechanical locking devices are designed and placed towards the bottom of each cone to lock each leg of the UAV so that the UAV is held firmly by the UGV even if the UGV is moving. This is to make sure that the UAV can be charged safely on the UGV when it is not stationary. Different from most other works related to auto-landing, such as the works presented in [12,31,32], it can be clearly seen that the landing platform adopted in this work for the UAV does not have a flat surface. If one leg is outside of the docking cone, a fatal crash may occur. This is also the reason why the task has high requirements on the precision of the landing task. Additionally, our landing platform contains multi-level heights which introduces much more complicated ground effects compared with landing a drone on the ground. More specifically, there are three different altitudes for different surfaces: ground, UGV's surface, and the top plates of the cones. Hence the turbulence generated by each surface is different and is coupled, resulting in an anomalous fountain flow under the UAV

while it hovers over the UGV at a close range. Therefore, the designed controller must be robust enough to overcome this nonlinear effect during the final landing stage.



**Figure 1.** The structure and hardware configuration of the designed platform: (a) Side view of the UAV docked on the UGVs; (b) Top view of the layout of the docking platform with its critical dimensions.

In this paper, the UAV's autonomous landing problem on the safety-critical platform described above was solved by using an effective controller combined with a vision-based target detection algorithm. In particular, the contributions of this work, compared with the state of the art, can be summarized as follows:

- The developed safety strategy can adjust the altitude of the UAV and recover visual detection (if it is lost in open fields) in the presence of strong wind disturbances. Such a mechanism is integrated into the controller.
- A robust visual-based controller, using the April tags as the landing mark, is designed to achieve the *high-precision* landing on a customized *multi-altitude* landing platform.
- The proposed system's performance was evaluated and validated through 20 real-world experimental scenarios and the results show the proposed strategy is effective and reliable even in adverse landing conditions.

The rest of this paper is organized as follows. Section 2 introduces the details of the practical configuration of the developed landing system. The proposed safe critical strategy is explained in Section 3. The results for simulations and real experimental tests with several case studies are presented in Section 4, demonstrating the effectiveness of the proposed landing mechanism. Finally, the conclusion is drawn in Section 5.

## 2. UAV-UGV Auto-Landing System

The developed UAV vision-based target detection and safe landing system are built with the following components: a Pixhawk flight controller, a Jetson Xavier, a USB webcam, and a gimbal. Additional devices, such as GPS, WiFi, and Lora, are used to establish communication among UAV, UGV, and the ground station. Figures 1a and 2 show the hardware layout of the overall system.

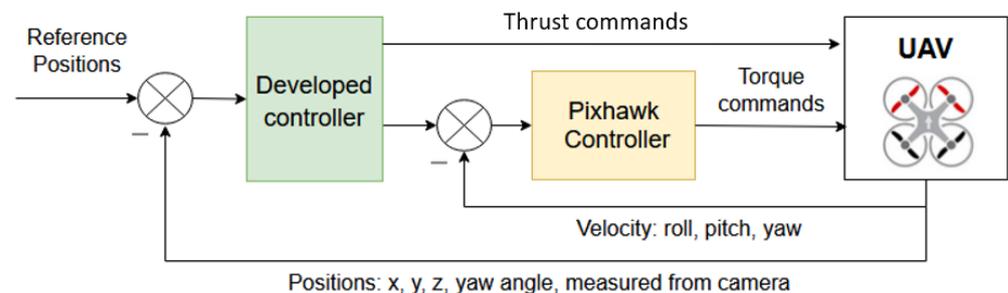
### 2.1. Flight Controller

PX4 is a high-performance, open-source software designed for autonomous or semi-autonomous UAVs, acting as a brain of a UAV, running inside the PixHawk controller (hardware), which can stabilize the UAV's attitudes and enable its autonomous flying. It can also be regarded as the inner-Loop controller, adjusting four motors' speeds, and maneuvering the UAV to its reference attitudes (roll, pitch, yaw) which are provided by the outer-loop controller. The Pixhawk controller has a 32-bit ARM Cortex M4 core with FPU which has a 168 Mhz clock frequency. It is also integrated with Multiple sensors: MPU6000 as the main accelerometer, and MEAS MS5611 as the barometer. A block diagram of the working mechanism of the flight controller is shown in Figure 3. To achieve the requested control accuracy for the further developed control method, the parameters of Pixhawk (K, P, and I of the PID controller) were calibrated via outdoor tests according to

the kinetic characteristics (e.g., weight distributions and inertia of the propellers) of the UAV used in this paper. All the minor gain tuning and calibration work are supported by the QGroundControl software running under a Linux-based operating system: Ubuntu.



**Figure 2.** The UAV and its hardware layout.



**Figure 3.** UAV system control structure.

### 2.2. Jetson Xavier

The Jetson Xavier is a low-power, small-size onboard computer carried with 8 core CPU (Carmel ARM v8.2) and 512-core Volta GPU with 64 Tensor Cores, running the Linux operating system. It has the same capability as a desktop computer and supports any programs written in python and C++. Thanks to its fast executing time, and high computational ability, in our work, Jetson Xavier is able to run the proposed control algorithm, and process images captured from the camera in real time. The camera is connected to Jetson via a USB port which continuously transfers images to it. Further, the outputs are fed to the Pixhawk flight controller to generate the control commands.

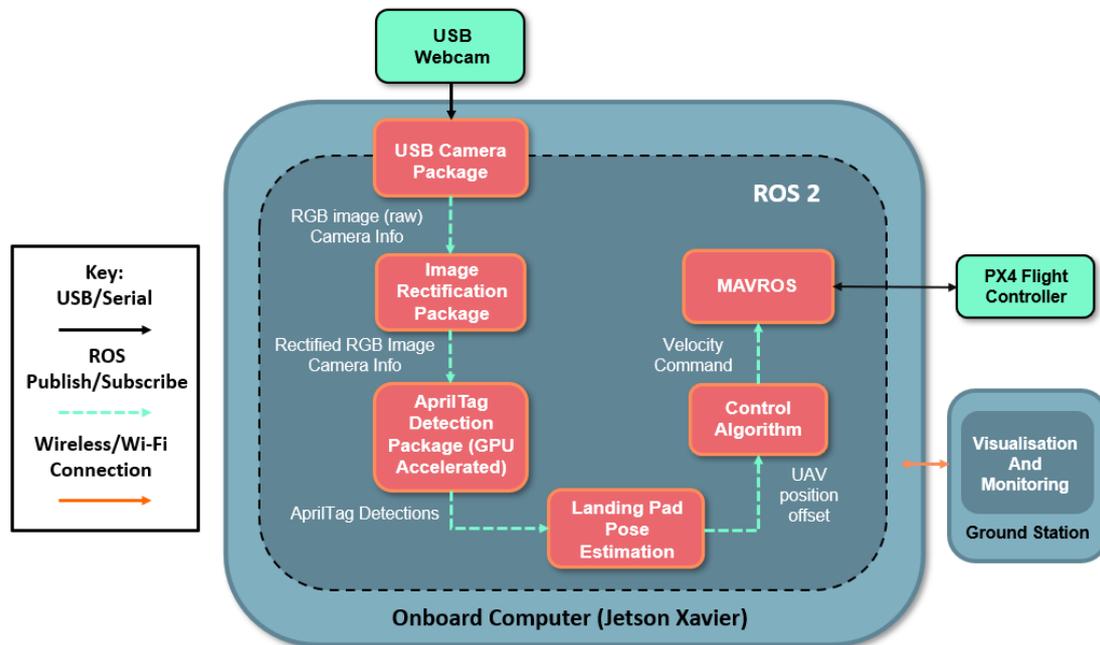
### 2.3. Camera and Gimbal

The camera selected in our landing system is a USB 3.0 webcam with 90 fps frame rate,  $512 \times 512$  resolution, 4.5 W power assumption and it has a FOV of  $150^\circ$ . It directly connects to the Jetson Xavier. It is held by the Mio gimbal, an ultra-lightweight 3-axis gimbal that is specially designed for small-size visual devices, and ability to quickly initialize up to 3 s. Gtune: a diagnosis and calibration software designed for the gimbal is used to balance the gimbal position with the mounted camera. The use of the gimbal ensures that the optical axis is always perpendicular to the ground.

## 3. Vision-Based UAV Autonomous Safe Landing Control

To ensure the reliability of the UAV autonomous landing on the customized platform, two algorithms were developed: (1) the target detection algorithm and (2) a precise and robust vision-based auto-landing algorithm. The architecture of the overall landing system is illustrated in Figure 4. Those red blocks represent all the packages on the UAV's onboard computer, while bright green blocks are those external devices connected to the onboard

computer. Communications between those packages are supported by ROS topics. Captured images by the webcam are the key inputs of the whole system, and the generated reference velocities are the outputs that are fed to the PX4.

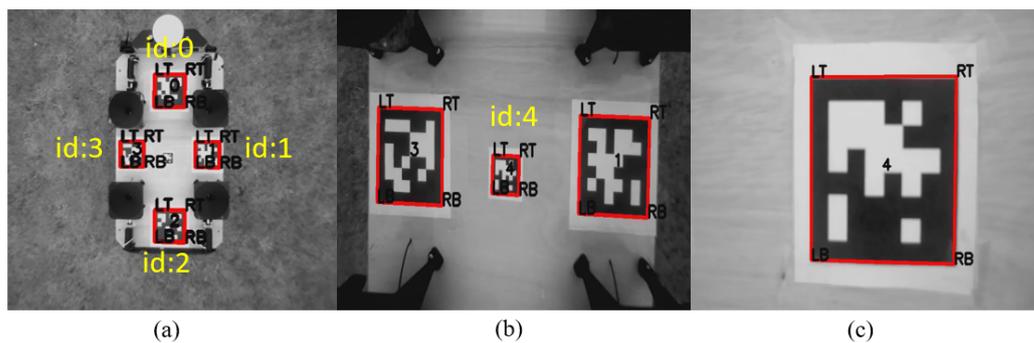


**Figure 4.** Architecture of the UAV auto-landing system.

### 3.1. Target Detection Algorithm

The goal of using vision-based target detection is to obtain the relative positions between the UAV and the center of the landing platform. In our work, we select the AprilTag visual position algorithm owing to its high detective accuracy and capability of supporting long-range vision, uneven light conditions, and low camera resolution. Generally, the detection process of the AprilTag algorithm can be divided into three steps [33]. First, the lines in the image are detected and the gradient direction and the magnitude of each pixel are calculated. Next, pixels with similar gradient directions and magnitudes are grouped based on the clustering method proposed in [34]. Once completed the clustering, the quads in the image can be searched using a recursive depth-prior approach. The first layer treats all line segments as actual line segments. From the second layer to the fourth one are used to find the segment that is the most adjacent to the end of the starting line. The threshold value for defining adjacent lines affects the robustness of the segmentation errors. After obtaining those detected lines and quads, they can be further identified by comparing them in the pre-defined tags library. Eventually, the relative positions between the tags and the camera can be calculated by the method discussed in [33].

The landing platform in our work is shown in Figure 5. There are five tags attached to the UGV and each tag has a label. Due to the resolution limitation of the camera, if the tag size is too small, UAV may not able to identify the target at a far range. However, a small-size tag is essential for close-range detection as the tag must stay within the field of view (FOV) of the camera. Therefore, to ensure the camera can always see the target, four large side tags (labeled as 0, 1, 2, 3) and one small tag (labeled as 4) are used to calibrate the UAV's position relative to the docking cones. When the UAV hovers at a high altitude, the four large side tags are used, while when the UAV is close to the platform, the small tag is primarily used for close-range detection. The detailed pose estimation algorithm of the UAV is described in Algorithm 1.



**Figure 5.** Detection and identification of the AprilTags at different altitudes during the UAV’s auto-landing process: (a) Acquired image at high altitude; (b) Acquired image at medium altitude; (c) Acquired image at final landing moment.

---

**Algorithm 1:** Pose Estimation

---

```

Input :Detected Tags  $id$ 
          Calculated tags’ relative positions:
           $x_{id}, y_{id}, z_{id}$ , orientation  $yaw_{id}$ , saved in list  $Q$ 
Output: Estimated UAV positions from the center of the landing pad:
           $x_{off}, y_{off}, z_{off}, yaw_{off}$ ;
1 while list  $Q$  of the detected  $id$  numbers is not empty do
2   if  $length(Q) \geq 2$  then
3      $(x_{off}, y_{off}, z_{off}, yaw_{off}) \leftarrow FindCenter(Q)$ ;
4   end
5   else if only  $id\ 4 \in Q$  then
6      $(x_{off}, y_{off}, z_{off}, yaw_{off}) \leftarrow (x_4, y_4, z_4, yaw_4)$ ;
7   end
8   else if only one side tag’s  $id$  in the  $Q$  then
9      $(x_{off}, y_{off}, z_{off}, yaw_{off}) \leftarrow Correction(id)$ ;
10  end
11 end

```

---

**Remark 1.**  $FindCenter(Q)$  and  $Correction(id)$  are two functions to calculate the relative positions between the UAV and the center of the landing pad and generate the offsets between one side tag and the center tag, respectively.

**3.2. Safe Landing Control**

A classical PID controller integrated with a visual-guaranteed mechanism is developed in our work to protect the landing algorithm against wind disturbances in the open farm fields. The outputs of the PID controller are the velocity commands which are sent to the PX4 flight controller based on the position offsets obtained from the visual positioning algorithm. There are four channels required for landing control:  $x, y, z$ , and yaw angle in the local inertial frame. Since the camera is stabilized by the gimbal, there are no rotations in pitch and roll ( $x - z, y - z$  planes align with the local frame). Hence, only the  $x - y$  plane of the yaw angle needs to be converted from the body coordinate to the local inertial coordinate. This can be achieved by

$$P^1 = R_0^1 \cdot P^0, \quad R_0^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \tag{1}$$

where  $P^1$  is the local frame,  $P^0$  is the body frame and  $R_0^1$  denotes the rotation matrix with rotation angle  $\alpha$ .

As the position offset derived from the camera is proportional to the UAV speed, the rates of position offsets are directly taken as the speed inputs of the PID controller, yielding the output of the PID controller as

$$V = k_p \cdot e_t + k_i \cdot \int_0^t e_t dt + k_d \cdot (e_t - e_{t-1}), \tag{2}$$

where  $k_p, k_i, k_d$ , are the proportional, integral, and differential gain, respectively, and  $e_t$  is the error at the current time step, while  $e_{t-1}$  is the error from the last time step. Based on the tuning results from testing, the gains of the controller for different channels are selected and shown in Table 1.

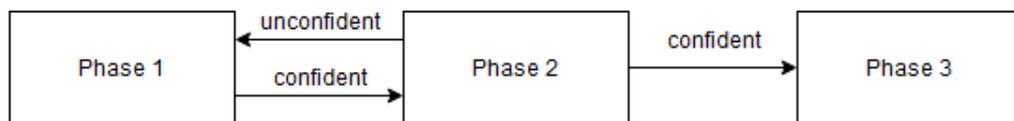
**Table 1.** PID gains for speed control of roll ( $x$ ), pitch ( $y$ ), yaw ( $\theta$ ), and thrust ( $z$ ).

Channel	$k_p$	$k_i$	$k_d$
$x$	0.55	0.02	0.15
$y$	0.55	0.02	0.15
$z$	0.4	0.03	0.1
$\theta$	0.5	0.01	0.1

In general, the autonomous landing in our case is separated into three phases:

- Phase 1: Triggered by the valid tag detection from the camera, the UAV hovers 2.5 m over the center of the UGV, and exchanges signals with the UGV by LoRa to check if the cones of the platform are open for landing;
- Phase 2: Triggered by the signal sent from the UGV when attitude errors are within a pre-defined range, the UAV hovers 0.5 m over the center of the UGV, for final calibration before landing;
- Phase 3: Triggered by the confidence level at Phase 2, the UAV cuts off the thrust, and lands on the platform.

**Remark 2.** Phase 2 will be conducted if and only if Phase 1 has been completed; while Phase 3 will be conducted if and only if both Phase 1 and Phase 2 have been finished. However, it is possible that the phase transmitted from 2 to 1, rather than 3, when the confidence level is not sufficiently large, as shown in Figure 6.



**Figure 6.** UAV confident-related phase switch.

The reliability and accuracy of the UAV’s safe landing mainly depend on the control performance in Phase 2 because it is the final calibration before fully landing. Moreover, any small deviations could be amplified during this stage as it is close to the surface of the docking device. To minimize the impacts of any external disturbances that occur in Phase 2, a robust mechanism that guarantees the detection of tags (even if strong gusts appear) is integrated into the control strategy, the details of which are explained in Algorithm 2. At a high level, there are two visual recovery methods in two different situations. In the first situation, if the camera can only see one of those side tags (with the id number belonging to  $\{0, 1, 2, 3\}$ ), then the controller uses the distance offset (saved in each tag) between the side tag and the center tag to correct the UAV’s position deviation, aiming to retrieve the visual detection of the center tag (see Lines 21–23). In the second situation, when the camera completely loses tag detection, the UAV will increase its altitude to expand the

scale of FOV and recover the tag detection. Once the UAV successfully detects the tags, the UAV will switch to Landing Phase 1 and attempt to land from the initial position again (see Lines 24–27). Additionally, the confidence level in Phase 2 is a time-dependent value derived from the TimeHolding function which outputs the holding time of the position errors and aims to keep the UAV within the required range for high-precision landing (i.e.,  $|x_{error}| < 0.07$  m,  $|y_{error}| < 0.07$  m,  $|z_{error}| < 0.1$  m,  $|yaw_{error}| < 5^\circ$ ). Whereas, if the elapsed time in Phase 2 exceeds 10 s, it will back to Landing Phase 1. Accordingly, the confidence level and the maximum elapsed time in Phase 2 directly determine if UAV's positions are sufficiently accurate for moving to Phase 3 (see Lines 13–16). For our design, the UAV must maintain its position errors within the restricted range for at least 2 s before descending.

---

**Algorithm 2:** Safe landing algorithm
 

---

**Input** : Estimated position offsets:  $x_{off}, y_{off}, z_{off}, yaw_{off}$ ;  
**Output**: Published velocity commands;

```

1 Initialize: Landing Phase = 1;
2 while true do
3   errors ← offsets – GetReference(Landing Phase);
4   if Landing Phase = 1 then
5     | published velocity ← controller(errors);
6     | if  $x_{error} < 0.15, y_{error} < 0.15, yaw_{error} < 5^\circ$  then
7     | | Landing Phase = 2;
8     | end
9   end
10  else if Landing Phase = 2 then
11    | if target id:4 detected then
12    | | published velocity ← controller(errors);
13    | | confidence level ← TimeHolding( $x_{error}, y_{error}, z_{error}, yaw_{error}, 0.07, 0.07,$ 
14    | | | 0.1,  $5^\circ$ );
15    | | if confidence level > 2 s then
16    | | | Landing Phase = 3
17    | | end
18    | | else if elapsed time in Landing Phase 2 > 10 s then
19    | | | Landing Phase = 1
20    | | end
21    | end
22    | else if only single side tag detected then
23    | | published velocity ← controller(errors);
24    | end
25    | else
26    | | Landing Phase = 1;
27    | | increase altitude: published velocity ← trust commands;
28    | end
29  end
30  else if Landing Phase = 3 then
31    | published velocity ← controller(errors);
32  end

```

---

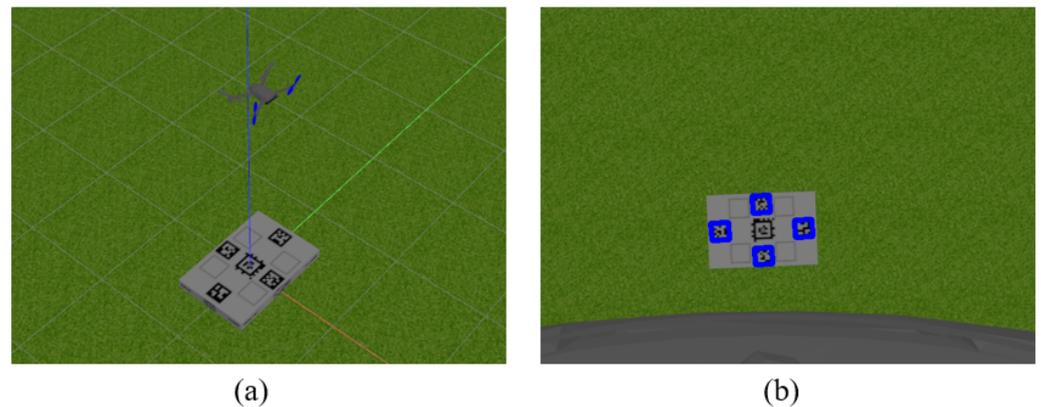
**Remark 3.** *GetReference (Landing Phase)* is a function that outputs the reference positions according to the corresponding Landing Phase.

### 4. Experiments and Results

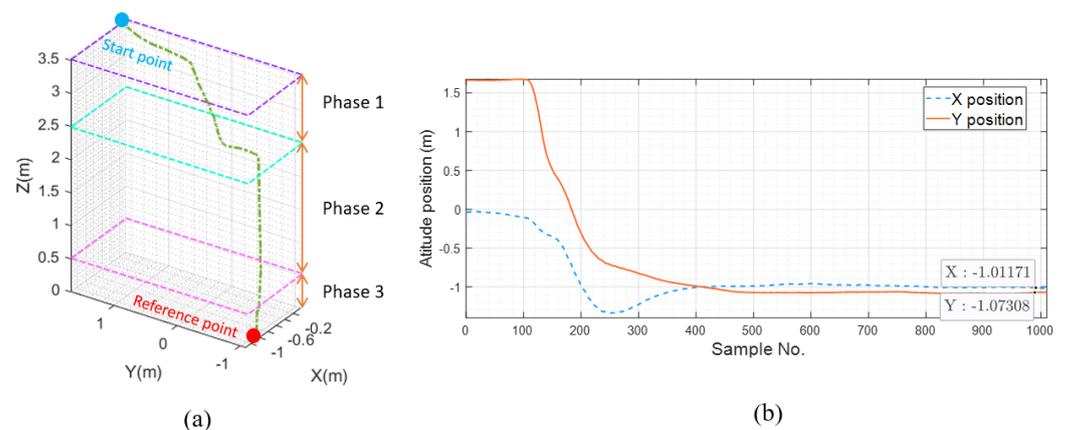
Before real tests in the agriculture fields, the basic landing performance of the proposed system in a static environment was first verified in simulations in Gazebo. Then, field tests were conducted in different locations in the UK with different wind conditions.

#### 4.1. Simulations

In the simulation, UAV landing is controlled by a python script built in ROS2. A full-scale mockup of the UGV landing platform is created and placed on the ground in the virtual environment as shown in Figure 7a. In the simulation, the drone automatically takes off near the landing pad using QGroundControl software. After the tags are detected when it reaches a certain height (maximum altitude for valid detection: 4 m), it is then switched to the auto-landing mode. Due to the complexity of the wind model, there is no wind introduced in the simulation. However, the overall performance of the designed landing system in no wind conditions can be still verified. In Figure 8, it can be seen that the overall landing trajectories consist of three phases, and the landing error in the steady state along  $x$  and  $y$  axes are controlled around 0.01 m, and 0.07 m, respectively, which stays within the safe margin of the designed docking device (i.e., 0.15 m  $\times$  0.15 m).



**Figure 7.** UAV auto-landing in the simulation environment: (a) Environment setting with full-scale platform mockup; (b) Detected tags from the camera.



**Figure 8.** (a) Autonomous landing trajectories of the UAV in the simulation; (b) Time histories of  $x, y$  positions of the UAV with reference point coordinates  $(-1, -1)$ .

#### 4.2. Field Testing

To further verify the effectiveness of the proposed UAV landing system in the agriculture fields, the UAV landing on the UGV was evaluated in both windy and calm conditions. For the experimental results presented here, a laptop is used as a ground station remotely

connect to the onboard computer of the UAV via WiFi, and save the flight data as the ROS bags. All the field testing setups follow the same procedure: A pilot manually takes off the UAV and flies it close to the UGV, hovering over 4~5 m. If the tags are detected, then switch the UAV to the off-board mode using QGround control, where the proposed landing system starts taking control of the drone. There are two types of data recording the flight trajectories in our experiments: raw GPS data and relative positions from the camera. The benefit of using the data recorded by the camera is the high accuracy as it is directly taken as the control feedback. However, the recording will be terminated if tag detection is lost. For the raw GPS data, due to the bad precision, it was only used as a rough flight trajectory to check the UAV's performance when the visual positions are unavailable. It is also worth noticing that for the safety concern, before the landing test on the real UGV, multiple real experiments were also conducted in an open area on the campus of Loughborough University using a full-scale mock-up as the landing pad which was similar to the one used in the simulation to refine the landing method. During the field tests in different farms, a total of three cases were observed and they can be described and defined as follows:

- Case 1 is an instance of “no disturbance”, where the UAV lands in a relatively calm environment (the effect of the breeze can be ignored). The results indicate that the proposed method works for a “no disturbance” scenario.
- Case 2 is an instance of “consistent disturbance”, where the UAV lands against the consistent wind. The results indicate that the proposed method works for a “consistent disturbance” scenario.
- Case 3 is an instance of “pulse disturbance”, while the strong wind gust unexpectedly emerges during the UAV landing. The results indicate that the proposed method works for the “pulse disturbance” scenario.

#### 4.2.1. Case 1: UAV Landing without Disturbances

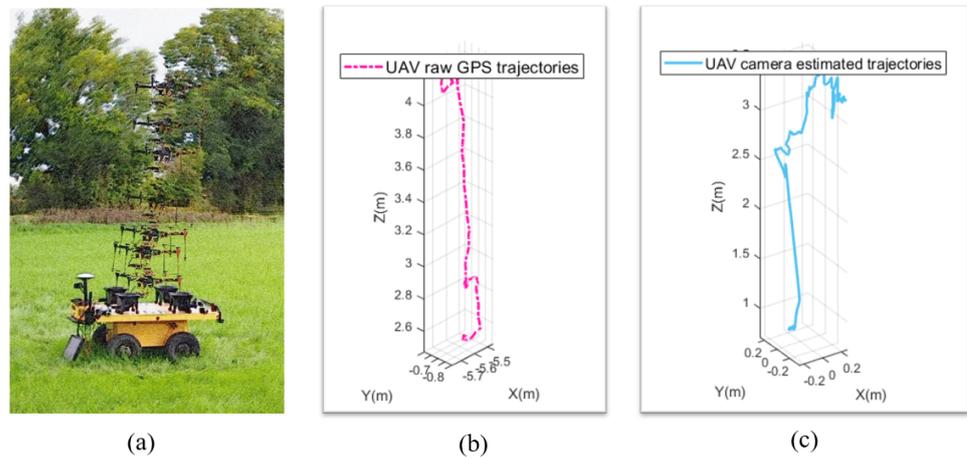
As emphasized earlier in this paper, due to the multi-level docking platform, the autonomous landing algorithm of the UAV needs to be accurate and robust. To slip the UAV's four legs into the docking device during the final landing, UAV attitude deviations must stay within the edge of the cones, i.e., 15 cm × 15 cm shown in Figure 1b. When the UAV moves to Landing Phase 3, it will cut off the thrust and descends due to gravity. Nevertheless, even during the final few inches descending before reaches to the safe docking margin, UAV's legs may still accidentally drop outside the margin leading to a body flip potentially caused by any minor disturbances and the asymmetric weight distribution of the body. Hence, as it is shown in Algorithm 2, the UAV can be guaranteed successfully land if and only if the UAV stays in the *safe zone* (0.07 m for both  $x$  and  $y$  axes).

A total of 20 landing tests in different fields in the UK were conducted, among all field tests in different farms without wind disturbances (15 landings), the landing success rate of the designed system reaches 100%. Figure 9 intuitively demonstrates the flight data of the basic auto-landing without strong wind disturbances (Please be aware that we did not measure the wind speed, but felt calm during all these 15 tests) recorded in different data types. Note that there are significant differences (around 0.5 m–1.5 m) between GPS recordings and estimated positions from the camera due to the poor accuracy of raw GPS measurements.

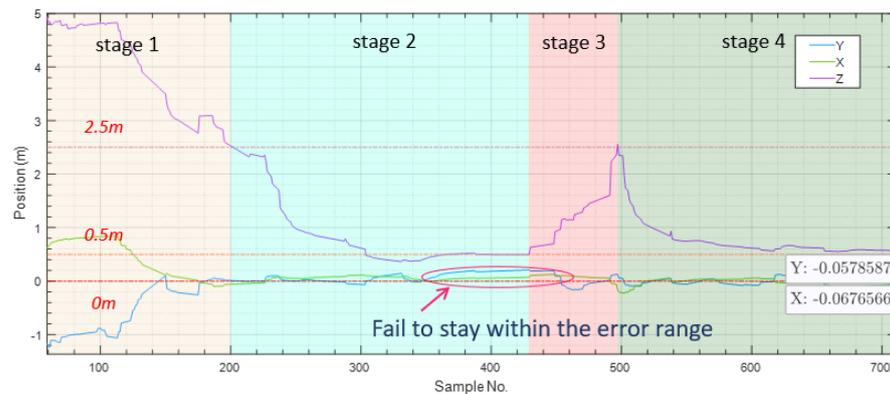
#### 4.2.2. Case 2: UAV Landing Subject to Consistent Disturbance

The second case shown in Figure 10 records the results derived from the vision algorithm when the proposed auto-landing system is performed in Deepdale Farm which is located on the east coast of the UK. As the testing field is near the seaside, there are sea breezes intermittently occurring during the landing process. In this case, the first time when the UAV moves from Landing Phase 1 (i.e., stage 1) to Landing Phase 2 (i.e., stage 2), it struggles against the consistent disturbance and stabilizes its attitude. However, it still fails to calibrate the attitudes within the safe range caused by wind disturbances within

the pre-set time threshold. Then it switches back to Landing Phase 1 and climbs to 2.5 m over the UGV when the restricted time is out (i.e., stage 3). In stage 4, the UAV moves to Landing Phase 2 again and this time control errors are staying in the required range (0.06 m and 0.07 m for  $x$  and  $y$  axis, respectively) until the UAV feels “confident” for the final landing. Note that the camera data is unable to be recorded for Landing Phase 3 as the power is off.



**Figure 9.** (a) Screenshots of a typical landing test conducted in Crippings Farm; (b) Raw GPS data of the landing. (c) Estimated positions from the visual positioning algorithm.

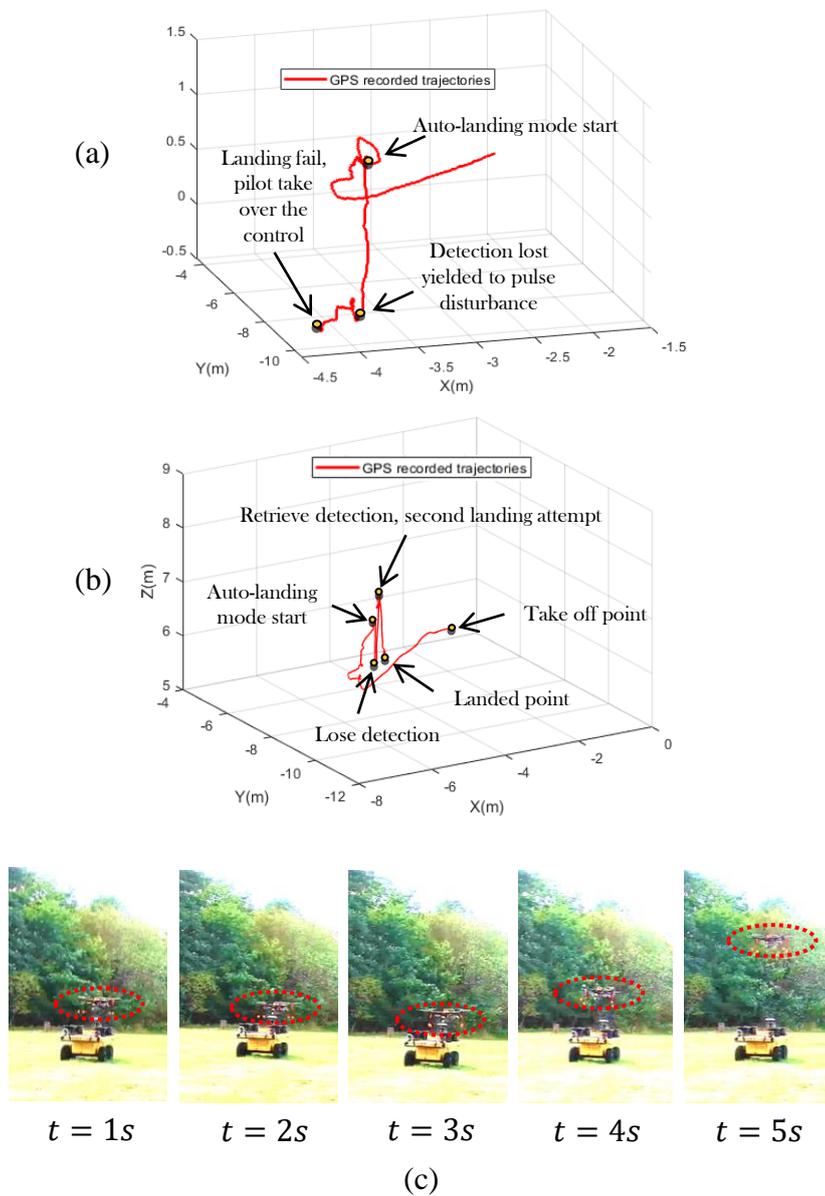


**Figure 10.** Estimated positions from the camera, UAV auto-landing in the field with mild winds.

4.2.3. Case 3: UAV Landing Subject to Pulse Disturbance

In the final case, ignoring the drifting errors from the raw GPS data, an extreme scenario was observed, shown in Figure 11b,c, when a test is performed on a farm in Chelmsford located in the south of the UK. A strong wind gust emerged when the UAV did the final calibration, forcing the UAV to move aggressively away from the center of the landing pad resulting in no tag detection. At that moment, the UAV reacted quickly and increased its altitude to recover the visual detection and eventually safely landed in the second attempt.

On the contrary, a similar scenario was observed and the controller without the proposed safety strategy was tested, the result of which is shown in Figure 11a. When the UAV suffered an unexpected “pulse disturbance” at the final landing, it failed to maintain its position and was blown away from the landing pad since it lost tag detection and could not generate any control feedback. In the end, the pilot manually took over the control of the UAV before any potential crash happened.



**Figure 11.** Comparison of the UAV landing in extreme scenarios with and without the integrated safety strategy (a) Raw GPS flight data of UAV landing without safety strategy subject to a wind gust; (b) Raw GPS flight data of UAV landing encounters strong wind gusts with integrated safety strategy; (c) UAV vision retrieve action: wind gust emerges at  $t = 1$  s, UAV is blown away and loses tag detection at  $t = 3$  s, then  $t = 4$  s~ $5$  s, the drone increases thrust to retrieve the tag detection and perform the landing again.

### 5. Conclusions and Discussion

In this paper, a solution is presented to UAV auto-landing problem on a customized multi-level platform in windy scenarios for agriculture-related applications. The control algorithm combined with the visual positioning algorithm achieves a high landing accuracy on narrow-size docking devices with complex ground effects. Moreover, the designed control strategy can adjust and retrieve the visual detection at the final landing step even in the presence of strong and unknown wind disturbances.

To verify the performance of our system, tests in the simulation were carried out in Gazebo to prove the validity and feasibility of the proposed landing system in a basic landing scenario neglecting weather conditions. Moreover, the UAV and UGV have been

taken to different farms in the UK to conduct real experiments. According to the experimental results, the landing was smooth and accurate with a 100% landing success rate within our 20 field tests in three different fields (since the weather condition is beyond control, system performance is unknown for some other extreme scenarios). Few scenarios of landing encountering wind disturbances were spotted in our tests. As expected, in all special cases with (consistent or pulse) disturbances, the UAV could always adjust or recover from those detrimental situations benefiting from the visual retrieve function and the restricted landing confidence, which has proved the reliability of our accurate and robust landing strategy in adverse environments. Future work can be focused on the accurate modeling of wind disturbances and improving the control performance based on quantified wind information.

**Author Contributions:** Conceptualization and methodology, J.J., E.P. and Z.G.; validation, Z.G., E.P. and B.M.; writing—original draft preparation, Z.G.; writing—review and editing, J.J., Y.Y., L.S. and B.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by Innovate UK under Grant 10004402.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Wei, Z.; Han, Y.; Li, M.; Yang, K.; Yang, Y.; Luo, Y.; Ong, S.H. A small UAV based multi-temporal image registration for dynamic agricultural terrace monitoring. *Remote Sens.* **2017**, *9*, 904. [[CrossRef](#)]
2. Siebert, S.; Teizer, J. Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system. *Autom. Constr.* **2014**, *41*, 1–14. [[CrossRef](#)]
3. Lewicka, O.; Specht, M.; Stateczny, A.; Specht, C.; Brčić, D.; Jugović, A.; Widźgowski, S.; Wiśniewska, M. Analysis of GNSS, Hydroacoustic and Optoelectronic Data Integration Methods Used in Hydrography. *Sensors* **2021**, *21*, 7831. [[CrossRef](#)]
4. Nevalainen, O.; Honkavaara, E.; Tuominen, S.; Viljanen, N.; Hakala, T.; Yu, X.; Hyypä, J.; Saari, H.; Pölonen, I.; Imai, N.N.; et al. Individual tree detection and classification with UAV-based photogrammetric point clouds and hyperspectral imaging. *Remote Sens.* **2017**, *9*, 185. [[CrossRef](#)]
5. Rudol, P.; Doherty, P. Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery. In Proceedings of the 2008 IEEE Aerospace Conference, Big Sky, MT, USA, 1–8 March 2008; pp. 1–8.
6. Molina, P.; Colomina, I.; Victoria, T.; Skaloud, J.; Kornus, W.; Prades, R.; Aguilera, C. Searching lost people with UAVs: The system and results of the close-search project. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *39*, 441–446. [[CrossRef](#)]
7. Radoglou-Grammatikis, P.; Sarigiannidis, P.; Lagkas, T.; Moscholios, I. A compilation of UAV applications for precision agriculture. *Comput. Netw.* **2020**, *172*, 107148. [[CrossRef](#)]
8. Sylvester, G. *E-Agriculture in Action: Drones for Agriculture*; Food and Agriculture Organization of the United Nations and International Telecommunication Union: Rome, Italy, 2018.
9. Gautam, A.; Sujit, P.; Saripalli, S. A survey of autonomous landing techniques for UAVs. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 1210–1218.
10. Talha, M.; Asghar, F.; Rohan, A.; Rabah, M.; Kim, S.H. Fuzzy Logic-Based Robust and Autonomous Safe Landing for UAV Quadcopter. *Arab. J. Sci. Eng.* **2019**, *44*, 2627–2639. [[CrossRef](#)]
11. Nemra, A.; Aouf, N. Robust INS/GPS sensor fusion for UAV localization using SDRE nonlinear filtering. *IEEE Sens. J.* **2010**, *10*, 789–798. [[CrossRef](#)]
12. Chen, C.; Chen, S.; Hu, G.; Chen, B.; Chen, P.; Su, K. An auto-landing strategy based on pan-tilt based visual servoing for unmanned aerial vehicle in GNSS-denied environments. *Aerosp. Sci. Technol.* **2021**, *116*, 106891. [[CrossRef](#)]
13. Cesetti, A.; Frontoni, E.; Mancini, A.; Zingaretti, P.; Longhi, S. A vision-based guidance system for UAV navigation and safe landing using natural landmarks. *J. Intell. Robot. Syst.* **2010**, *57*, 233–257. [[CrossRef](#)]
14. Li, Z.; Chen, Y.; Lu, H.; Wu, H.; Cheng, L. UAV autonomous landing technology based on apriltags vision positioning algorithm. In Proceedings of the Chinese Control Conference, CCC, Guangzhou, China, 27–30 July 2019; pp. 8148–8153. [[CrossRef](#)]
15. Xin, L.; Tang, Z.; Gai, W.; Liu, H. Vision-Based Autonomous Landing for the UAV: A Review. *Aerospace* **2022**, *9*, 634. [[CrossRef](#)]
16. Shim, T.; Bang, H. Autonomous landing of UAV using vision based approach and PID controller based outer loop. In Proceedings of the 2018 18th International Conference on Control, Automation and Systems (ICCAS), PyeongChang, Republic of Korea, 17–20 October 2018; pp. 876–879.

17. Sharma, A.; Barve, A. Controlling of quad-rotor uav using pid controller and fuzzy logic controller. *Int. J. Electr. Electron. Comput. Eng.* **2012**, *1*, 38–41.
18. Mohammadi, A.; Feng, Y.; Zhang, C.; Rawashdeh, S.; Baek, S. Vision-based autonomous landing using an MPC-controlled micro UAV on a moving platform. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 771–780.
19. Feng, Y.; Zhang, C.; Baek, S.; Rawashdeh, S.; Mohammadi, A. Autonomous landing of a UAV on a moving platform using model predictive control. *Drones* **2018**, *2*, 34. [[CrossRef](#)]
20. Baca, T.; Stepan, P.; Spurny, V.; Hert, D.; Penicka, R.; Saska, M.; Thomas, J.; Loianno, G.; Kumar, V. Autonomous landing on a moving vehicle with an unmanned aerial vehicle. *J. Field Robot.* **2019**, *36*, 874–891. [[CrossRef](#)]
21. Setyawan, G.E.; Kurniawan, W.; Gaol, A.C.L. Linear quadratic regulator controller (LQR) for AR. Drone’s safe landing. In Proceedings of the 2019 International Conference on Sustainable Information Engineering and Technology (SIET), Lombok, Indonesia, 28–30 September 2019; pp. 228–233.
22. Pionidis, M.; Aristodemou, P.; Theocharides, T. Towards a Fully Autonomous UAV Controller for Moving Platform Detection and Landing. In Proceedings of the 2022 35th International Conference on VLSI Design and 2022 21st International Conference on Embedded Systems (VLSID), Bangalore, India, 26 February–2 March 2022; pp. 180–185.
23. Rodriguez-Ramos, A.; Sampedro, C.; Bavle, H.; De La Puente, P.; Campoy, P. A deep reinforcement learning strategy for UAV autonomous landing on a moving platform. *J. Intell. Robot. Syst.* **2019**, *93*, 351–366. [[CrossRef](#)]
24. Lee, M.F.R.; Nugroho, A.; Le, T.T.; Bastida, S.N. Landing area recognition using deep learning for unammaned aerial vehicles. In Proceedings of the 2020 International Conference on Advanced Robotics and Intelligent Systems (ARIS), Taipei, Taiwan, 19–21 August 2020; pp. 1–6.
25. Bektash, O.; Naundrup, J.J.; la Cour-Harbo, A. Analyzing visual imagery for emergency drone landing on unknown environments. *Int. J. Micro Air Veh.* **2022**, *14*, 17568293221106492. [[CrossRef](#)]
26. Krishnakumar, R.; Rasheed, A.M.; Kumar, K.S. Enhanced hover control of quad tilt frame UAV under windy conditions. *Int. J. Adv. Robot. Syst.* **2015**, *12*, 146. [[CrossRef](#)]
27. Scicluna, L.; Sant, T.; Farrugia, R.N. Investigation of wind flow conditions on the flight endurance of UAVs in hovering flight: A preliminary study. In Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering, St. Julian’s, Malta, 3–6 November 2019; Volume 59353, p. V001T01A037.
28. Yang, J.; Liu, C.; Coombes, M.; Yan, Y.; Chen, W.H. Optimal path following for small fixed-wing UAVs under wind disturbances. *IEEE Trans. Contro. Syst. Technol.* **2020**, *29*, 996–1008. [[CrossRef](#)]
29. Yan, Y.; Yang, J.; Liu, C.; Coombes, M.; Li, S.; Chen, W.H. On the actuator dynamics of dynamic control allocation for a small fixed-wing UAV with direct lift control. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 984–991. [[CrossRef](#)]
30. Yan, Y.; Liu, C.; Oh, H.; Chen, W.H. Dual-layer optimization-based control allocation for a fixed-wing UAV. *Aerosp. Sci. Technol.* **2021**, *119*, 107184. [[CrossRef](#)]
31. Wu, L.; Wang, C.; Zhang, P.; Wei, C. Deep Reinforcement Learning with Corrective Feedback for Autonomous UAV Landing on a Mobile Platform. *Drones* **2022**, *6*, 238. [[CrossRef](#)]
32. Rabah, M.; Rohan, A.; Talha, M.; Nam, K.H.; Kim, S.H. Autonomous Vision-based Target Detection and Safe Landing for UAV. *Int. J. Control Autom. Syst.* **2018**, *16*, 3013–3025. [[CrossRef](#)]
33. Olson, E. AprilTag: A robust and flexible visual fiducial system. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3400–3407.
34. Felzenszwalb, P.F.; Huttenlocher, D.P. Efficient graph-based image segmentation. *Int. J. Comput. Vis.* **2004**, *59*, 167–181. [[CrossRef](#)]

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