

Article

Simultaneous Obstacles Avoidance and Robust Autonomous Landing of a UAV on a Moving Vehicle

Jinglong Guo ^{1,†}, Xin Dong ^{1,†}, Yang Gao ¹, Daochun Li ¹ and Zhan Tu ^{1,2,*} ¹ School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China² Institute of Unmanned System, Beihang University, Beijing 100191, China

* Correspondence: zhantu@buaa.edu.cn

† These authors contributed equally to this work.

Abstract: For unmanned aerial vehicles (UAVs), landing on a moving vehicle robustly is an open challenge, especially under cluttered surroundings with the presence of unknown obstacles. Those undesired environmental factors could induce collisions and thus affect flight safety significantly. Currently, there are few solutions to address such a challenge. In this paper, we propose a systematic autonomous landing scheme that enables the robust autonomous landing performance of a quadrotor UAV. The proposed scheme integrates target detection, state estimation, trajectory planning, and landing control. The position and attitude information of the target ground vehicle and the test quadrotor are estimated by the onboard vision system and GPS. In order to detect landing markers at different altitudes, a particular landing pad with an Apriltag bundle is implemented. As a typical aerial–terrestrial cooperation system, the trajectory planner of the quadrotor updates continuously to avoid obstacles via real-time sensing and re-planning. A finite state machine is used to label the current flight status and triggers the control laws correspondingly. The effectiveness of the proposed method has been validated in a high-fidelity simulator with environmental obstacles.

Keywords: autonomous landing; obstacle avoidance; path planning; flight control



Citation: Guo, J.; Dong, X.; Gao, Y.; Li, D.; Tu, Z. Simultaneous Obstacles Avoidance and Robust Autonomous Landing of a UAV on a Moving Vehicle. *Electronics* **2022**, *11*, 3110. <https://doi.org/10.3390/electronics11193110>

Academic Editors: Hongbin Deng, Jifeng Guo and Chengchao Bai

Received: 22 August 2022

Accepted: 24 September 2022

Published: 28 September 2022

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



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, unmanned aerial vehicles (UAVs) are receiving extensive attention from academia and industry around the world. To date, they have been used in search and rescue, precision agriculture, logistics transportation, and aerial photographing [1–4]. On the other hand, the insufficient payload and endurance of current UAV systems have already limited the development of applications. In order to address such common issues of UAVs and to broaden application scenarios, in recent years, several attempts focusing on the cooperation of UAVs and unmanned ground vehicles (UGVs) have been conducted. Throughout the whole aerial–terrestrial collaborative process, the landing of UAVs on UGVs is one of the key steps.

In order to achieve an autonomous landing, UAVs need to obtain the state feedback of the ground vehicle, which requires appropriate sensors. GPS is a widely used positioning sensor. For example, Refs. [5–7] introduced aircraft that use GPS information for precise guidance and landing. However, the low accuracy and low sampling frequency of GPS affect the landing accuracy of UAVs significantly. Most importantly, GPS cannot work properly in indoor conditions. On the other hand, autonomous landing research on UAVs based on airborne cameras has made great progress in recent years. The authors of [8] described a vision-based algorithm to control a VTOL UAV while tracking and landing on a mobile platform. Additionally, Ref. [9] demonstrated an airborne monocular vision system for autonomous landing on a typical landing pad, with the identification mark consisting of the letter “H” surrounded by a circle. The authors of [10] combined the model predictive control, vision-based localization, and extended Kalman filters for path

tracking, navigation, and guidance to enable micro UAVs to land autonomously on moving platforms. Based on the aforementioned studies, vision sensors usually demonstrate high accuracy but are constrained by the detection range and limited camera-view field. In the case of autonomous landing, vision sensors and GPS can be integrated as complementary sensory systems. For example, GPS can be used to guide drones to roughly approach the target out of view of the camera. Then, visual sensors can take over the guidance until the UGV and landing pad can be seen.

With reliable sensor perception, the presence of environmental disturbances, including obstacles and wind gusts, should be carefully considered in landing trajectory planning since they threaten flight safety. In [11], the UAV avoided certain collisions (e.g., with the ground) when automatically landing on a moving platform. In [12], obstacles were undertaken by adjusting the altitude obtained from the elevation model of the front area. Nevertheless, in the above landing scenarios, only known obstacles in the environment are considered. The methods discussed above may not be able to be adopted in the real world directly. For instance, if tall buildings block the view of the target UGV, then the UAV would be stuck or out of control. Consequently, these methods require the runtime of the desired landing trajectory to be updated to guarantee obstacle avoidance and landing accuracy.

In this work, we propose a systematic scheme for UAVs to perform autonomous landing on a moving UGV and simultaneously bypass unexpected obstacles. Such a method can also handle wind gusts during the approaching and landing. With the continuous perspective from GPS and onboard vision sensors, our tested UAV continuously updates UGV tracking and the autonomous landing trajectory to avoid obstacles while approaching the target UGV. As the UAV enters a safe landing area without surrounding obstacles, the fine-tuned flight controller handles the landing accuracy. For validation, the proposed scheme for UAVs to avoid obstacles and land on a UGV is simulated and evaluated in the simulator.

The main contributions of this paper are summarized as follows:

- (1). A systemic landing scheme integrated with the trajectory planning algorithm is proposed for a quadrotor to autonomously land on a moving UGV with unknown environmental obstacles and disturbances.
- (2). The proposed scheme demonstrates successful landing performance in high-fidelity simulated flights under several extreme environmental disturbances, covering the cases of obstacles blocking the view during approaching and wind gust disruption at both approaching and landing stages.
- (3). A comparative study is conducted between the proposed method and a state-of-the-art motion planning algorithm [13] to validate the robustness and reliability of the proposed landing scheme.

The rest of this paper is organized as follows: In Section 2, the system developed to complete the scheme is overviewed. Section 3 focuses on the detection method and landing pad. Section 4 introduces the trajectory planning and control methods, followed by the results in Section 5. Finally, conclusions will be made in Section 6.

2. System Overview

This paper proposes a complete scheme for UAVs landing on a moving UGV with unknown obstacles in the environment. The quadrotor, UGV, and main coordinate frames used in the scheme are shown in Figure 1. The UAV is at a distance from the target UGV at the initial moment. At this stage, the UAV needs to avoid obstacles while approaching the UGV. After the drone moves closer to the UGV, the onboard vision system begins to work to guide the drone to land precisely.

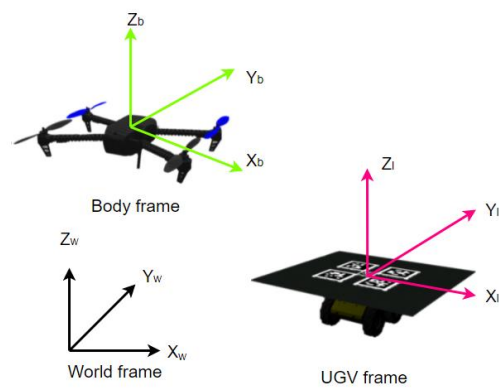


Figure 1. System overview and main coordinate frames.

2.1. UAV Dynamic Model

The mathematical model of the quadrotor dynamics is presented. The model in this work follows a similar model presented in [14,15]. The quadrotor is treated as a rigid body, and the model is generated using the Euler–Newton method.

$$F_i = kr_i^2 \quad (1)$$

$$M_i = \tau r_i^2 \quad (2)$$

where r_i is the angular speed of the rotor, F_i is the force, and M_i is the moment.

The state of the system is given by the position (x, y, z) , orientation (Φ, θ, Ψ) , velocity $(\dot{x}, \dot{y}, \dot{z})$, and angular velocity (p, q, r) :

$$\mathbf{x} = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, p, q, r]^T \quad (3)$$

The qualitative dynamics equation of the quadrotor is as follows:

$$\sum F = m \frac{d}{dt} V = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \quad (4)$$

According to the moment of momentum theorem, we obtain the following formula:

$$\sum M = \frac{d}{dt}(H) = \frac{d}{dt}(I\omega) \quad (5)$$

where H is the angular momentum, and I is the inertia matrix. Due to the symmetry of the quadrotor model, the inertia matrix can be expressed as:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (6)$$

The angular rate of the quadrotor can be expressed as:

$$\begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} = I \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \omega \times H \quad (7)$$

U_2 , U_3 , and U_4 are the moment related to the roll, pitch, and yaw, respectively.

The force and moments of the system can be expressed in matrix form as:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} k & k & k & k \\ 0 & kL & 0 & -kL \\ -kL & 0 & kL & 0 \\ \tau & -\tau & \tau & -\tau \end{bmatrix} \begin{bmatrix} r_1^2 \\ r_2^2 \\ r_3^2 \\ r_4^2 \end{bmatrix} \quad (8)$$

where L is the distance between the axis of rotation of the rotors and the center of gravity of the quadrotor; U_1 is the net body thrust.

2.2. Finite State Machine

For the smooth implementation of the landing scheme, we designed a finite state machine (FSM) to determine the action of the quadrotor in a three-dimensional environment with unknown obstacles. FSM contains four states: hovering, tracking and avoiding obstacles, landing, and disarmed. The states and the respective transitions are depicted in Figure 2.

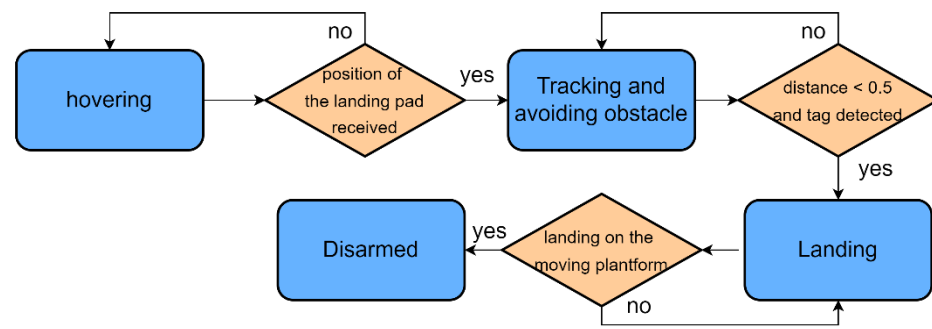


Figure 2. Finite state machine of the landing scheme.

Hovering: The hovering state includes the takeoff and loitering of the drone. At this stage, the quadrotor waits to receive the position of the landing pad and then transitions to the next state.

Tracking and avoiding obstacles: After receiving the position of the landing pad, a collision-free trajectory from the quadrotor to the UGV is planned to avoid the obstacles, and the drone begins to approach the UGV.

Landing: After the horizontal distance between the UAV and the mobile UGV is less than 0.5 m, we believe that the UAV then enters a safe area without obstacles. At this stage, the UAV will be guided by a visual system.

Disarmed: After the drone lands on the moving UGV, the blades of the drone will stop rotating, and the landing mission is complete.

3. Detection Method and Landing Pad

This section introduces the simulated GPS and the vision system that we used to obtain the position and attitude of the landing pad. Platform state estimation method: Extended Kalman filtering (EKF) and the landing pad used in the vision inspection system are also described.

3.1. Moving Platform State Estimation

In this paper, we assume that the UGV is not within the camera's field of view initially. At this stage, simulated GPS is used to guide the drone, and the motion state of the mobile UGV is updated by the EKF.

EKF, which takes linearization errors into account to improve the estimation performance of nonlinear systems, is widely used in the tracking and prediction of moving objects [16–18]. The first process of EKF is to predict the position, then to weight the

predicted position and the observed position to update it in a certain way, and to finally output the processed result. The state vector of the moving UGV is

$$S_p = [P_x, P_y, V, \varphi] \quad (9)$$

where P_x and P_y are the positions in the x and y directions, V is the velocity of the UGV, and φ is the orientation angle. Since the UGV moves on flat ground throughout the experiment, the position of the target UGV in the z direction is not considered.

Moving UGVs can be modeled as

$$\dot{S}_p(t) = f_p(S_p(t)) + \omega(t) \quad (10)$$

$$M_p = [P_x, P_y, \varphi] \quad (11)$$

where $\omega(t)$ is white Gaussian noise used to add in the ground truth value of the landing pad obtained in Gazebo to simulate the real-world GPS signal. M_p is the measurement vector, and the Kalman filter processes the measurement value to obtain the estimated state. This article uses the simulated GPS and visual system to estimate the status of the moving UGV. When the distance between the quadrotor and landing pad is over 2 m, the position is obtained from the simulated GPS. At this stage, the heading angle is calculated from the motion of the platform.

3.2. Visual Detection System

The accurate position of the landing pad in the Global coordinate system is estimated by an onboard downward monocular camera when the drone is within a certain distance of the UGV. In order to obtain the precise pose of the target at different heights, we designed a landing pad that contains five Apriltags in the TAG36H11 family, as shown in Figure 3. AprilTags are a specific type of fiducial marker, consisting of a black square with a white foreground that has been generated in a particular pattern.

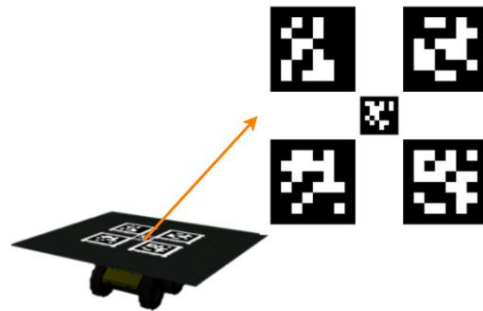


Figure 3. Landing pad of the visual detection system.

The landing pad, which consists of AprilTag markers of different sizes, can be divided into two parts: The first part positioned in the center of the marker is a standalone tag with a side length of 0.15 m. The second part comprises tag bundles composed of four different markers around the center, and the length of each side is 0.4 m, which enables the drone to approach the mobile UGV, even if parts of the landing pad are occluded.

At the beginning of the landing stage, the markers in the tag bundles are detected to estimate the pose. According to [19], for a marker with a side length of 0.15 m, when the distance from the UAV is within 1 m, the estimation precision is within 0.4 m. Therefore, after the distance between the UAV and the landing pad is less than 1 m, the standalone tag will be used to guide the drone.

4. Trajectory Planning and Control Law

4.1. Trajectory Planning

In the tracking and avoiding obstacles stage, we used an onboard depth camera to model the obstacles and to then plan a feasible trajectory for the UAV. Initially, a B-spline curve that does not consider obstacles was generated, and then the A* algorithm was used to generate a collision-free trajectory on the line segment that passes through obstacles on the curve to guide the curve to away from obstacles.

The B-spline function, which is short for basis spline, is used to create smooth curves and is controlled by a number of points. The research and application of B-spline are mainly in the fields of trajectory planning, trajectory tracking, and path optimization.

B-spline is a linear combination of the primary curves. The primary function of the curve is described as follows [20]:

$$P_{i,1}(t) = \begin{cases} 1, & \text{for } \tau_i \leq t < \tau_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

$$P_{i,n}(t) = \frac{t - \tau_i}{\tau_{i+n-1} - \tau_i} P_{i,n-1}(t) + \frac{\tau_{i+n} - t}{\tau_{i+n} - \tau_{i+1}} P_{i+1,n-1} \quad (13)$$

With $n + 1$ control points Q_i , the expression for a B-spline parametric curve of degree n is

$$B(t) = \sum_{i=1}^{n+1} P_{i,n}(t) Q_i \quad (14)$$

The B-spline function has several properties:

- (1) Convex hull property. The curve lies in the convex hull of the control points and can be easily adjusted by changing the position of the points.
- (2) By adding control points, the curve can be changed locally without affecting the overall shape.
- (3) The B-spline's primary functions of degree n can be expressed by the linear combinations of B-splines of lower order.

Since B-spline has the property of its k th derivative still being the B-spline, the velocity V_i , acceleration A_i , and jerk J_i of the control points can be expressed as

$$V_i = \frac{Q_{i+1} - Q_i}{\Delta t}, \quad A_i = \frac{V_{i+1} - V_i}{\Delta t}, \quad J_i = \frac{A_{i+1} - A_i}{\Delta t} \quad (15)$$

where Q_i is the control points required for the B-spline, and Δt is the time interval between the control points and is independent of the B-spline.

The A* algorithm uses a combination of heuristic searching and the shortest path searching [21]. Its cost function can be expressed as

$$f(n) = h(n) + g(n) \quad (16)$$

where $h(n)$ is the cost from the initial state to state n , and $g(n)$ is the estimated distance of the current state to the goal state.

In order to make the quadrotor land on the moving platform successfully, the constraints of the endpoint of the trajectory are designed as follows:

$$\begin{bmatrix} P_x^{end} \\ P_y^{end} \\ P_z^{end} \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (17)$$

$$\begin{bmatrix} V_x^{end} \\ V_y^{end} \\ V_z^{end} \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \\ 0 \end{bmatrix} \quad (18)$$

where $P^{end} = [P_x^{end}, P_y^{end}, P_z^{end}]^T$ is the position of the endpoint, $P = [P_x, P_y, P_z]^T$ is the position of the central location of the landing pad, $V^{end} = [V_x^{end}, V_y^{end}, V_z^{end}]^T$ is the velocity of the endpoint, and V_x and V_y are the velocity of the landing pad in x and y directions, respectively.

In order to plan the trajectory in real-time, we used the Euclidean Signed Distance Field (ESDF)-free method proposed by [13] to avoid the obstacles.

4.2. Control Law

A controller similar to [14] is designed to ensure the quadrotor land on the platform smoothly. The desired net force U_1 is computed as follows:

$$U_1 = -K_p e_p - K_v e_v + m\ddot{r} \quad (19)$$

$$r = [x, y, z]^T \quad (20)$$

where e_p and e_v are the errors on the position and velocity between the quadrotor and the landing pad, K_p, K_v are positive definite gain matrices and r is the position of the quadrotor.

Our controller is defined in $SO(3)$ space and computes the angle error using the small angle assumption.

The desired moments related to the roll, pitch and yaw $[U_2, U_3, U_4]^T$ are expressed as:

$$[U_2, U_3, U_4]^T = K_\omega e_\omega \quad (21)$$

where e_ω is the error on the actual and desired orientation.

The attitude obtained from the AprilTags is represented by quaternions, which avoid the gimbal lock phenomenon. However, what we desired is the Euler angle, so the quaternions need to be converted.

$$\begin{bmatrix} \varphi \\ \theta \\ \Psi \end{bmatrix} = \begin{bmatrix} \arctan \frac{2(wx+yz)}{1-2(x^2+y^2)} \\ \arcsin(2(wy-zx)) \\ \arctan \frac{2(wz-zx)}{1-2(y^2+z^2)} \end{bmatrix} \quad (22)$$

$[\varphi, \theta, \psi]$ are the Euler angles in the body coordinate system, and $[w, x, y, z]$ are the quaternions.

5. Simulation Results

This section presents the simulation results of the quadrotor avoiding unknown obstacles and landing on a moving UGV in an environment with unknown obstacles. We used a depth camera at the nose of the drone to model the obstacles and a downward monocular camera to detect the landing pad.

We first used the method proposed in [13] to plan the landing trajectory, and the results are shown in Figure 4. The UGV moves along the x direction at a speed of 0.8 m/s. The quadrotor follows the UGV about 0.5 m behind, making it impossible for the drone to land on it, meaning that the landing mission has failed.

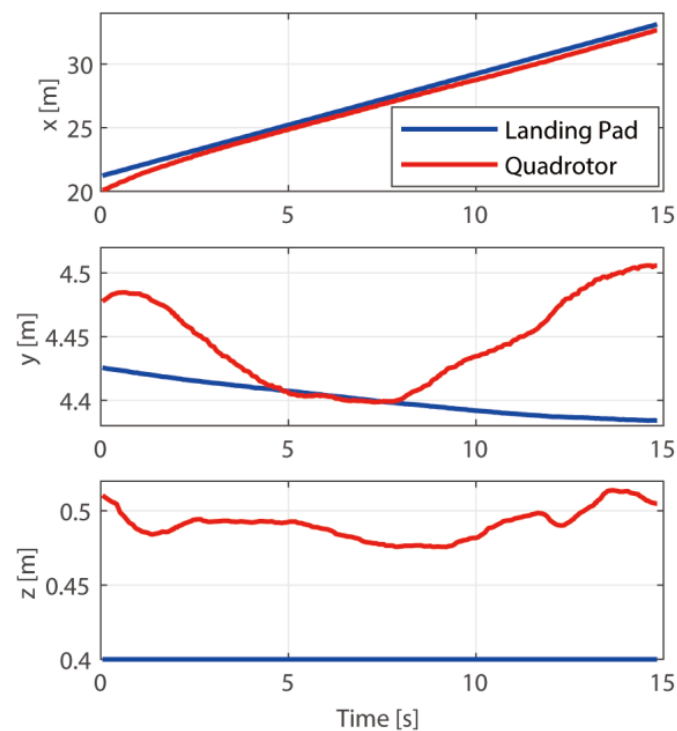


Figure 4. Trajectory of quadrotor and landing pad.

Then, we planned the landing trajectory using the method proposed in this paper. As shown in Figure 5, the drone initially hovers at 2 m in the z-direction, and the UGV is between buildings. A collision-free trajectory needs to be planned to ensure the drone's safety.

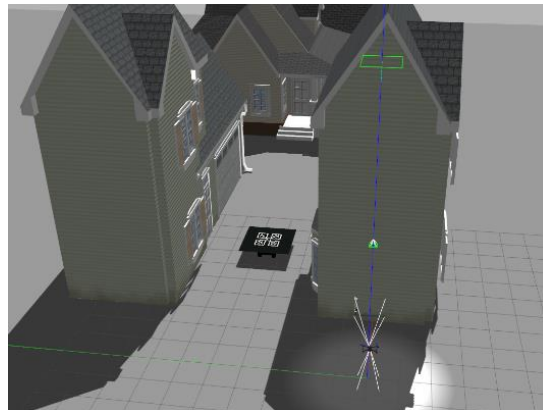


Figure 5. Gazebo simulation environment.

The local trajectory is visualized in Rviz, as shown in Figure 6. The height of the obstacles in the figure can be seen intuitively through colors: red represents the lowest, while purple represents the highest. It can be seen in Figure 6 that the algorithm used here can model the obstacles accurately to plan a collision-free trajectory between the UAV and the moving UGV.

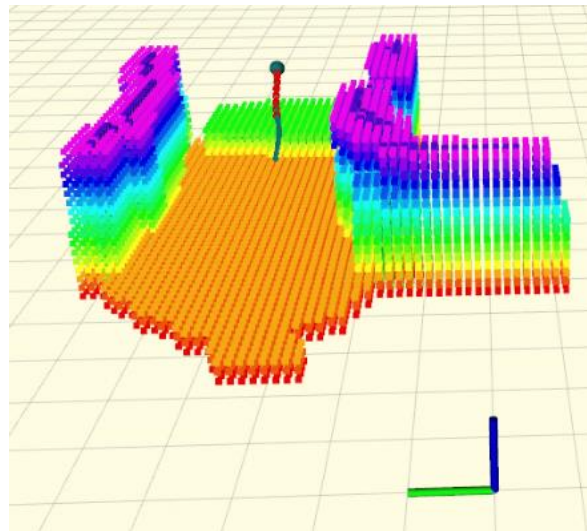


Figure 6. Visualization of local trajectory planning.

We verified the feasibility of our proposed landing scheme in two cases. The first case is that the UGV takes a straight–turn–straight route between buildings. The second case is that the UGV moves in a circular motion. The quadcopter successfully landed on the mobile platform in both scenarios.

In case 1, the UGV moves forward at a linear speed of 0.8 m/s. As shown in Figure 7, the quadrotor is able to land on the UGV as the Gaussian noise increases. The drone first avoids the building in front of it and approaches the UGV. The initial position error between the quadrotor and the landing pad is 7 m in the x direction, 6.5 m in the y direction, and 2 m in the z direction. Figure 8 shows the quadrotor approaching the UGV in the x and y directions after 10 s and landing on it in 26 s. Figure 9 shows the position error near the landing point. The position of the landing pad is provided by the simulated GPS at the beginning, and the UAV is guided by the vision system after 15 s.

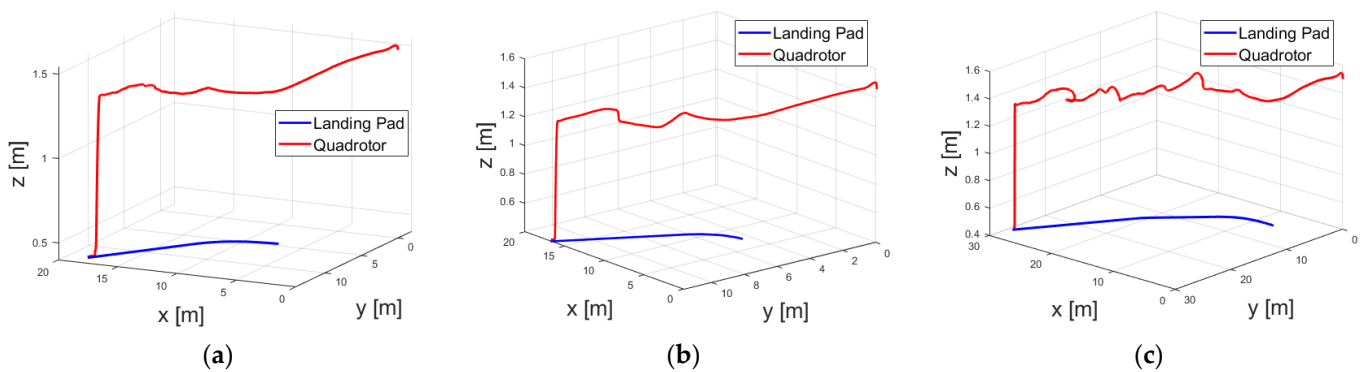


Figure 7. Trajectory of quadrotor and landing pad in case 1. (a) Trajectory with Gaussian noise with a standard deviation $\sigma = 0.01$. (b) Trajectory with Gaussian noise with a standard deviation $\sigma = 0.5$. (c) Trajectory with Gaussian noise with a standard deviation of $\sigma = 1.0$.

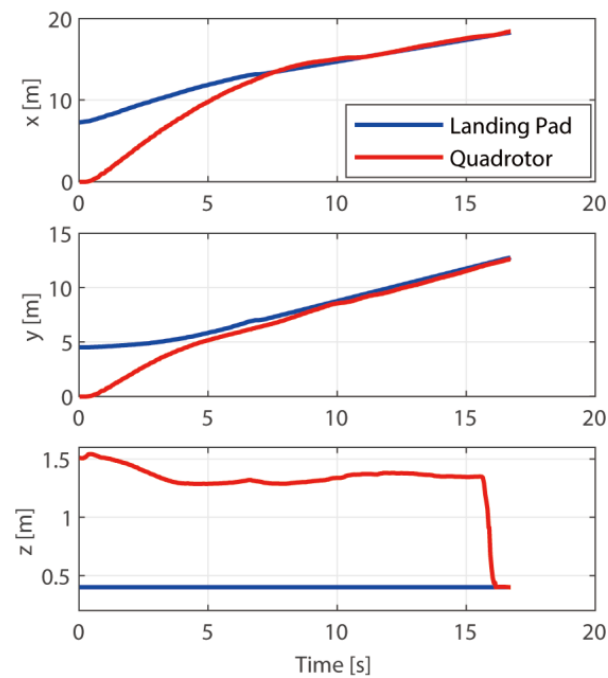


Figure 8. Position of quadrotor and landing pad with $\sigma = 0.01$.

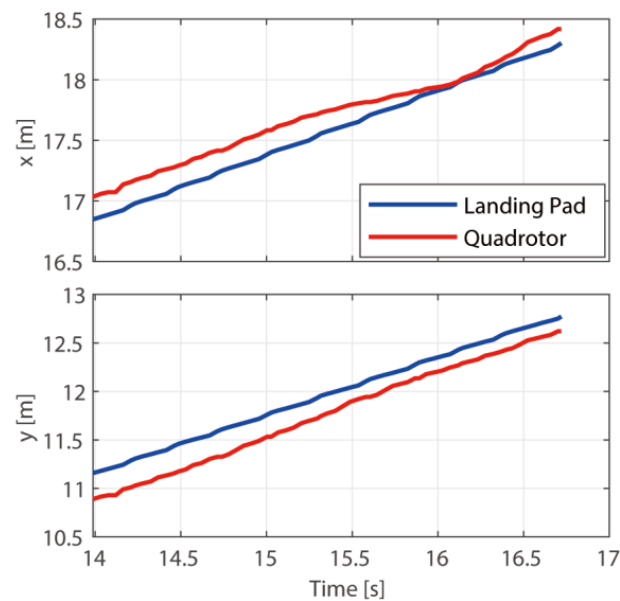


Figure 9. Position of quadrotor and landing pad near the landing point with $\sigma = 0.01$.

In case 2, the UGV performs circular motion with a linear velocity of 0.8 m/s and an angular velocity of 0.15 rad/s between buildings. As shown in Figure 10, the quadrotor is able to land on the UGV as the Gaussian noise increases. It can be seen that the drone can follow the UGV while avoiding obstacles. As shown in Figure 11, the quadrotor approaches the UGV in the x and y directions after 15 s and lands on it in 28 s. Figure 12 shows the position error near the landing point.

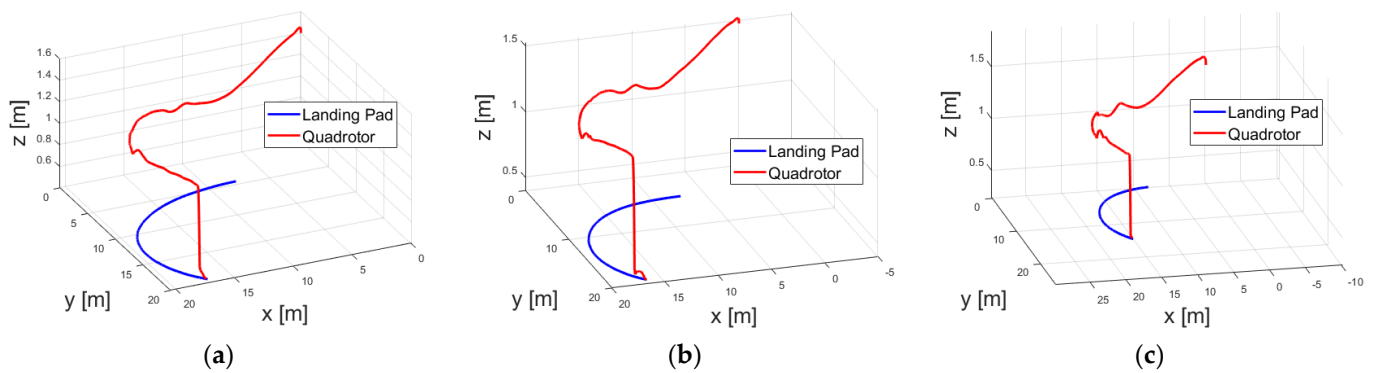


Figure 10. Trajectory of quadrotor in case 2. (a) Trajectory with Gaussian noise with a standard deviation of 0.01. (b) Trajectory with Gaussian noise with a standard deviation of 0.5. (c) Trajectory with Gaussian noise with a standard deviation of 1.0.

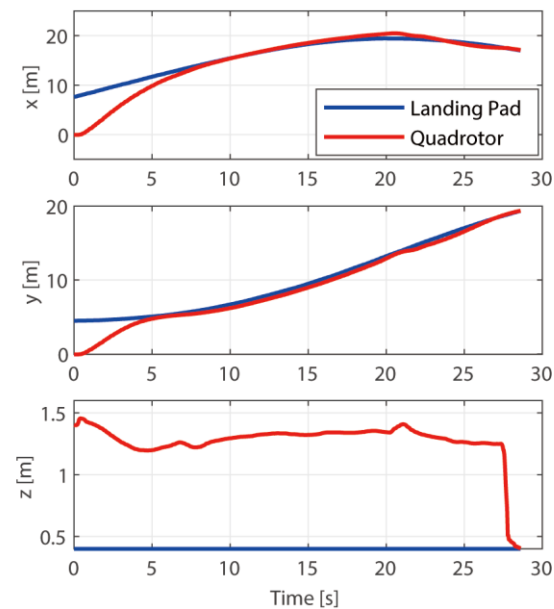


Figure 11. Position of quadrotor and landing pad in case 2.

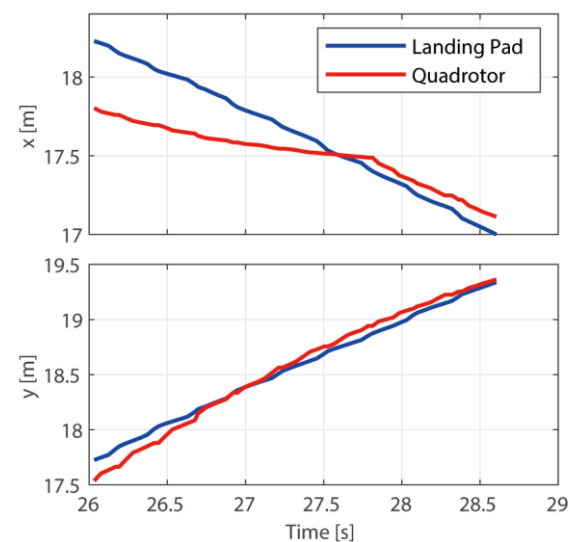


Figure 12. Position of quadrotor and landing pad near the landing point.

We test the robustness of the proposed algorithm under different wind field conditions. The average wind velocity we set is 4–11 m/s, and the maximum wind speed is 6–15 m/s. As shown in Figure 13, as the wind speed increases, the oscillation of the quadrotor trajectory becomes larger. As shown in Table 1, we calculate the position error between the UAV and the moving UGV in the x and y directions within 5 s before landing. The position error increases as the wind speed increases, but the safe landing of the quadrotor is still guaranteed.

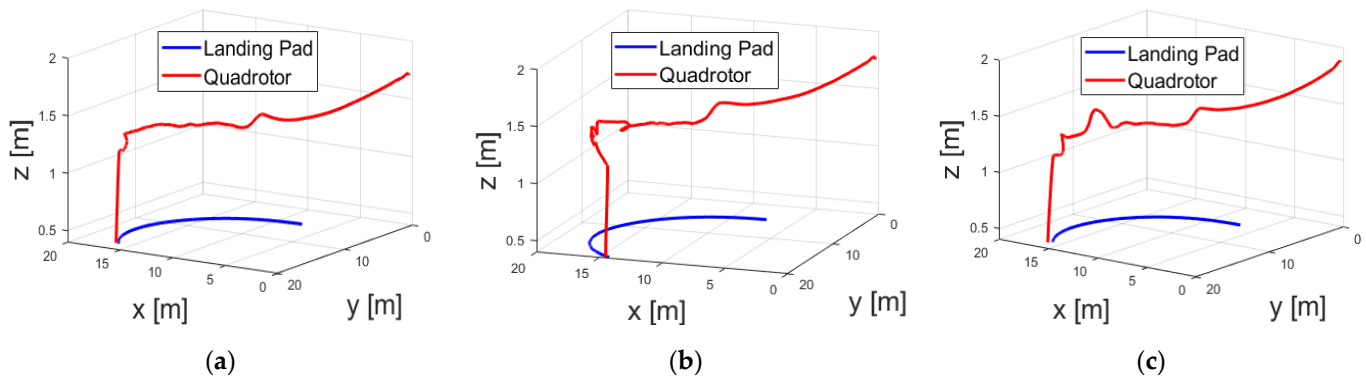


Figure 13. Trajectory of quadrotor in the wind field. (a) Trajectory with a mean wind velocity of 4 m/s and maximum wind velocity of 7 m/s. (b) Trajectory with a mean wind velocity of 7 m/s and maximum wind velocity of 11 m/s. (c) Trajectory with a mean wind velocity of 11 m/s and maximum wind velocity of 15 m/s.

Table 1. Average position error in x direction and y direction within 5 s before landing.

Mean Wind Velocity	Position Error in x Direction(m)	Position Error in y Direction(m)
0.3	0.226	0.263
4	0.241	0.340
7	0.380	0.427
11	0.394	0.489

6. Conclusions

In this work, a systematic landing scheme with a trajectory planning algorithm for UAVs to autonomously land on a moving UGV was proposed. The proposed scheme can effectively address the unknown environmental disturbances during both UAV approaching and landing stages. A novel landing pad with a corresponding tracking algorithm was designed to track the landing platform in real-time. A trajectory planning algorithm was adopted to guide the drone when landing on a moving platform with environmental disturbances. Based on the simulated flight tests, compared to state-of-the-art motion planning algorithms, the proposed method demonstrates overall better landing robustness and precision, which illustrates the outperforms of the proposed framework. Future work will mainly focus on the implementation and experimental tests of the proposed scheme.

Author Contributions: Conceptualization, J.G.; methodology, Z.T.; software, Y.G. and X.D.; validation, Y.G.; formal analysis, J.G. and X.D.; investigation, J.G., D.L. and Z.T.; data curation, Z.T.; writing—original draft preparation, J.G.; writing—review and editing, X.D. and Z.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available upon request.

Acknowledgments: We thank Shiyu Zuo (AVIC Xi'an Flight Automatic Control Research Institute) for providing valuable brainstorming and discussion on the simulation environment setup.

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

References

1. Mahony, R.; Kumar, V.; Corke, P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robot. Autom. Mag.* **2012**, *19*, 20–32. [\[CrossRef\]](#)
2. Qi, Y.; Wang, J.; Shan, J. Aerial cooperative transporting and assembling control using multiple quadrotor–manipulator systems. *Int. J. Syst. Sci.* **2018**, *49*, 662–676. [\[CrossRef\]](#)
3. Cieslewski, T.; Kaufmann, E.; Scaramuzza, D. Rapid exploration with multi-rotors: A frontier selection method for high speed flight. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada, 24–28 September 2017; pp. 2135–2142.
4. Lee, H.; Kim, H.J. Constraint-based cooperative control of multiple aerial manipulators for handling an unknown payload. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2780–2790. [\[CrossRef\]](#)
5. Guo, Y.; Guo, J.; Liu, C.; Xiong, H.; Chai, L.; He, D. Precision landing test and simulation of the agricultural UAV on apron. *Sensors* **2020**, *20*, 3369. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Cho, A.; Kim, J.; Lee, S.; Choi, S.; Lee, B.; Kim, B.; Kee, C. Fully automatic taxiing, takeoff and landing of a UAV using a single-antenna GPS receiver only. In Proceedings of the 2007 International Conference on Control, Automation and Systems, Guangzhou, China, 30 May–1 June 2007; pp. 821–825.
7. Wang, F.; Liu, P.; Zhao, S.; Chen, B.M.; Phang, S.K.; Lai, S.; Lee, T.H. Development of an unmanned helicopter for vertical replenishment. *Unmanned Syst.* **2015**, *3*, 63–87. [\[CrossRef\]](#)
8. Lee, D.; Ryan, T.; Kim, H.J. Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing. In Proceedings of the 2012 IEEE International Conference on Robotics And automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 971–976.
9. Yang, S.; Scherer, S.A.; Zell, A. An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *J. Intell. Robot. Syst.* **2013**, *69*, 499–515. [\[CrossRef\]](#)
10. 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), Nagoya, Japan, 23–25 March 2020; pp. 771–780.
11. Falanga, D.; Zanchettin, A.; Simovic, A.; Delmerico, J.; Scaramuzza, D. Vision-based autonomous quadrotor landing on a moving platform. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; pp. 200–207.
12. Kyristsis, S.; Antonopoulos, A.; Chaniakakis, T.; Stefanakis, E.; Linardos, C.; Tripolitsiotis, A.; Partsinevelos, P. Towards autonomous modular UAV missions: The detection, geo-location and landing paradigm. *Sensors* **2016**, *16*, 1844. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Zhou, X.; Wang, Z.; Ye, H.; Xu, C.; Gao, F. Ego-planner: An esdf-free gradient-based local planner for quadrotors. *IEEE Robot. Autom. Lett.* **2020**, *6*, 478–485. [\[CrossRef\]](#)
14. Kose, O.; Oktay, T. Simultaneous quadrotor autopilot system and collective morphing system design. *Aircr. Eng. Aerosp. Technol.* **2020**, *92*, 1093–1100. [\[CrossRef\]](#)
15. Mellinger, D.; Kumar, V. Minimum snap trajectory generation and control for quadrotors. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2520–2525.
16. 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. [\[CrossRef\]](#)
17. Kwon, W.; Park, J.H.; Lee, M.; Her, J.; Kim, S.H.; Seo, J.W. Robust autonomous navigation of unmanned aerial vehicles (UAVs) for warehouses’ inventory application. *IEEE Robot. Autom. Lett.* **2019**, *5*, 243–249. [\[CrossRef\]](#)
18. Pinto, M.F.; Coelho, F.O.; De Souza, J.P.; Melo, A.G.; Marcato, A.L.; Urdiales, C. EKF design for online trajectory prediction of a moving object detected onboard of a uav. In Proceedings of the 2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO), Ponta Delgada, Azores, Italy, 4–6 June 2018; pp. 407–412.
19. Qi, Y.; Jiang, J.; Wu, J.; Wang, J.; Wang, C.; Shan, J. Autonomous landing solution of low-cost quadrotor on a moving platform. *Robot. Auton. Syst.* **2019**, *119*, 64–76. [\[CrossRef\]](#)
20. Stoican, F.; Prodan, I.; Popescu, D.; Ichim, L. Constrained trajectory generation for uav systems using a b-spline parametrization. In Proceedings of the 2017 25th Mediterranean Conference on Control and Automation (MED), Valletta, Malta, 3–6 July 2017; pp. 613–618.
21. Duchoň, F.; Babinec, A.; Kajan, M.; Beňo, P.; Florek, M.; Fico, T.; Jurišica, L. Path planning with modified a star algorithm for a mobile robot. *Procedia Eng.* **2014**, *96*, 59–69. [\[CrossRef\]](#)