Autonomous Landing Solution of Low-Cost Quadrotor on a Moving Platform

Article *in* Robotics and Autonomous Systems · September 2019

DOI: 10.1016/j.robot.2019.05.004

6 authors, including:

Jin Wu
The Hong Kong University of Science and Technology
48 PUBLICATIONS **157** CITATIONS
SEE PROFILE

Jianan Wang
Beijing Institute of Technology
37 PUBLICATIONS **236** CITATIONS
SEE PROFILE

Chunyan Wang
Beijing Institute of Technology
30 PUBLICATIONS **180** CITATIONS
SEE PROFILE

Robotics and Autonomous Systems 119 (2019) 64–76

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Autonomous landing solution of low-cost quadrotor on a moving platform

Yuhua Qi [a], Jiaqi Jiang [a], Jin Wu [b], Jianan Wang [a,*], Chunyan Wang [a], Jiayuan Shan [a]

[a] School of Aerospace Engineering, Beijing Institute of Technology, Beijing, China
[b] University of Electronic Science and Technology of China, Chengdu, China

HIGHLIGHTS

- A novel landing pad composed of patterns of different sizes is presented.
- The detection method for a low-cost quadrotor is considered.
- A backstepping approach is presented to attenuate the influence of the external disturbance.

ARTICLE INFO

Article history:
Received 25 January 2019
Received in revised form 7 May 2019
Accepted 15 May 2019
Available online 31 May 2019

Keywords:
Autonomous landing
Low-cost quadrotor
Backstepping method

ABSTRACT

In this paper, a complete solution for autonomous landing of a low-cost quadrotor on a moving platform is presented. First, the dynamic model of the quadrotor is described and simplified for landing task. Second, a novel landing pad associated with detection algorithm is designed for robust detection by a low-cost monocular camera. In order to deal with mirror effect and occasional misidentification, a 3D point cluster algorithm for relative position estimation is proposed. Third, using the backstepping approach, an adaptive position controller is designed to calculate the desired attitude for attitude loop. The overall stability of the position loop is proven by a Lyapunov method. Meanwhile, some landing strategies are presented to achieve a safe and stable landing task. Finally, the low-cost system architecture of the quadrotor and the experiment results are both demonstrated to showcase the effectiveness of the proposed method.

1. Introduction

Nowadays, UAVs receive great attentions from worldwide researchers and commercial companies. UAVs, especially quadrotors which can vertical take off and land (VTOL), are an ideal flight platform for many indoor and outdoor applications including but not limited to aerial surveillance, agricultural irrigation, pipeline inspection and object transporting [1–4]. As an important part of a flight task, autonomous landing is a fundamental yet challenging problem of quadrotors. During a flight task, it is the last but still critical step for the entire flight task.

Due to the importance of the autonomous landing task, several research communities have studied the problem in the past few decades. Representative works are as follows: The authors in [5] presented a real-time computer vision system for a rotor-craft UAV to land onto a known landing target which simplifies the vision computation process. [6] presented an onboard monocular vision system for autonomous landing on a typical landing pad which consists of the letter 'H' surrounded by a circle. 6 DOF pose of the quadrotor relative to the landing pad can be determined from images using projective geometry. [7] presented a nonlinear controller for a UAV using the measurement of average optical flow on a spherical camera along with the inertial measurement unit (IMU) data to land on a moving platform. In [8], an image-based visual servo controller for the landing problem of a VTOL vehicle was presented. The image of the centroid for a collection of landmarks is used as position measurement and translational optical flow is used as velocity measurement for the autonomous landing. The authors in [9] designed a novel landing pad for the situation that parts of the target are not visible and a vision based detection algorithm for estimation of the 3D position of the UAV relative to the landing pad. [10] presented an integrated control demonstration in which a rotary wing UAV landed on a moving platform using vision-based localization of the target vehicle.

To-date, many vision-based algorithms are developed for autonomous landing of UAVs. Published works investigated the landing on static, moving as well as unknown sites. These algorithms usually require pre-known marker to be attached on the landing sites. In [11], the detection of the marker is based on the color detection algorithm with morphological filter. In [12], a pad

* Corresponding author.
E-mail address: wangjianan@ieee.org (J. Wang).

https://doi.org/10.1016/j.robot.2019.05.004
0921-8890/

design aiming for possible detection from high as well as small altitudes was presented. The pattern comprised one red square and one blue circle, which can increase the robustness and range of detection. [6] presented a landing pad which consists of the letter 'H' inside a black circle. The 5 DOF pose was calculated from the elliptic projection of the circle. The remaining geometric ambiguity was resolved after incorporating the estimation of the gravity vector using the vehicle's inertial measurement unit. Respectively, Yang presented an algorithm which did not need a pre-known marker, the landing site was defined by a template image and identified using the matching of detected ORB features. The relative pose was estimated by matching the landing site features to the PTAM map points. But it consequently loses its accuracy in the case of the absence of the map [11].

Most of the papers above adopted global shutter camera as the landing detection sensor [6,8,11], which is even more expensive than the total cost of a quadrotor UAV. Using a low-cost camera will lead to rolling shutter effect [13] and imaging noise because of the rolling shutter and low resolution. In the marker detection, it will decrease the detection accuracy and recall rate. In addition, it will cause mirror effect [14] when we use iterative PnP methods to calculate the relative pose between the camera and marker. However, autonomous landing is a general requirement for quadrotor including low-cost quadrotor. In real applications, there are significant needs to decrease the cost and enhance the practicability of the algorithm.

Aiming at the actual application and low-cost autonomous system, this paper presents a quadrotor system capable of landing on a moving platform using the low-cost onboard vision system. A low-cost monocular camera is employed to detect the landing pad. To solve the negative influence caused by the low-cost camera, vision algorithm and position control law are improved to enhance the landing task robustness and stability. Moreover, some landing strategies are presented to decrease the negative influence caused by low-cost system. The contributions of this paper can be summarized as follows:

(i) A novel landing pad composed of patterns of different sizes is presented, which can be detected from high as well as very low altitudes. A 3D point cluster algorithm is presented to eliminate wrong estimated poses caused by occasional misidentification and mirror effect of planar landing pad.
(ii) Backstepping approach is adopted to design an adaptive position controller which can not only track the reference state but also attenuate the influence of the external disturbance.
(iii) A low-cost and open-source based system architecture is proposed.

The rest of this paper is organized as follows. Section 2 presents the fundamental equations of the dynamic model considered in this work. In Section 3, the landing pad design and the detection algorithm are given. Section 4 presents a position control law and the proof of the controller. Section 5 presents some landing strategies. Experiments setup and practical experiments results are given in Section 7. Finally, Section 8 concludes this paper.

2. Dynamic model

Consider a quadrotor vehicle represented by a rigid body of mass m and of moment of inertia J, as shown in Fig. 1. To describe the motion of the quadrotor, two reference frame are introduced: an world inertial reference $\sum_w$ and a body-fixed frame $\sum_b$ attached to the quadrotor at the mass center. $\boldsymbol{p} = [x, y, z]^T$ and $\boldsymbol{v} = [\dot{x}, \dot{y}, \dot{z}]^T$ are, respectively, the position and linear velocity

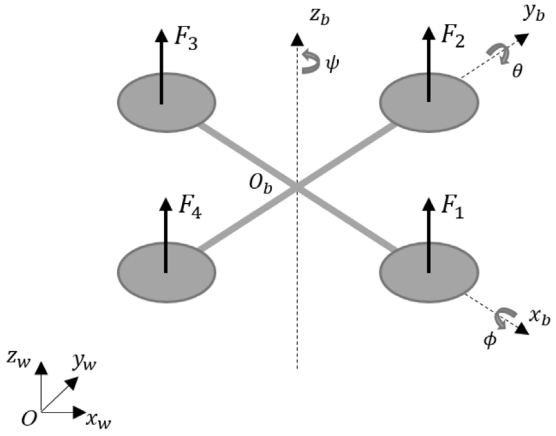


Fig. 1. The quadrotor with corresponding frames.

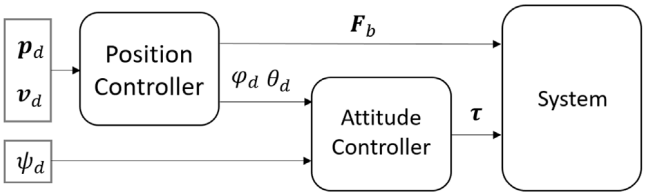


Fig. 2. Hierarchical control scheme of quadrotor.

of the quadrotor in $\sum_w$. With the roll/pitch/yaw angles of the quadrotor represented by $\boldsymbol{\phi} = [\varphi, \theta, \psi]^T$, the rotation matrix R_b from $\sum_b$ to $\sum_w$ is given as

$$R_b = \begin{bmatrix} c_\theta c_\psi - s_\varphi s_\theta s_\psi & -c_\varphi s_\psi & s_\theta c_\psi + s_\varphi c_\theta s_\psi \\ c_\theta s_\psi + s_\varphi s_\theta c_\psi & c_\varphi c_\psi & s_\theta s_\psi - s_\varphi c_\theta c_\psi \\ -c_\varphi s_\theta & s_\varphi & c_\varphi c_\theta \end{bmatrix}, \tag{1}$$

where c_γ and s_γ denote, respectively, $\cos\gamma$ and $\sin\gamma$.

In the frame system as shown in Fig. 1. The equations of motion for the quadrotor can be written as:

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \tag{2}$$

$$m\dot{\boldsymbol{v}} = R_b \boldsymbol{F}_b - mg\boldsymbol{e}_3 + \boldsymbol{\Delta}_t, \tag{3}$$

$$J\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times J\boldsymbol{\omega} + \boldsymbol{\tau}, \tag{4}$$

where $\boldsymbol{F}_b = [0, 0, f_b]^T$ is the force applied to the vehicle expressed in $\sum_b$, g is the gravity constant, $\boldsymbol{\tau}$ is the torque applied to the vehicle, $\boldsymbol{\Delta}_t = \left[\Delta_x, \Delta_y, \Delta_z\right]^T$ is the external disturbance in the translational dynamics and $\boldsymbol{w}$ is the angular velocity of the quadrotor. We further assume that the external disturbance are small time varying and bounded, that is $\|\boldsymbol{\Delta}_t\| \leq \delta_t$ where δ_t is known constant.

In most autopilot, a hierarchical control scheme as shown in Fig. 2 is used for the quadrotor control. In our case, we only consider the translation dynamics. For the orientation dynamics of (4), a high-gain controller is used to ensure that attitude $\boldsymbol{\phi}$ converges to the desired attitude $\boldsymbol{\phi}_d$. The translational dynamics of the quadrotor are shown in (2) (3) and also can be represented as

$$m\ddot{x} = f_x + \Delta_x, \tag{5}$$

$$m\ddot{y} = f_y + \Delta_y, \tag{6}$$

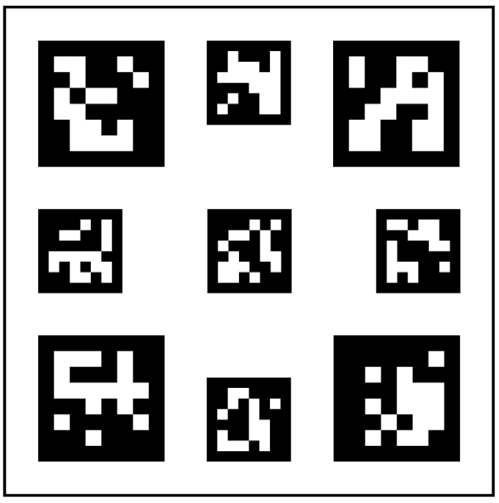


Fig. 3. Landing pad.

$$m\ddot{z} = f_z - mg + \Delta_z, \tag{7}$$

where $\boldsymbol{F}_w = \left[f_x, f_y, f_z\right]^T$ is the force applied to quadrotor expressed in world inertial frame. It can be written as

$$f_x = f_b\left(\sin\theta\cos\psi + \sin\varphi\cos\theta\sin\psi\right), \tag{8}$$

$$f_y = f_b\left(\sin\theta\sin\psi - \sin\varphi\cos\theta\cos\psi\right), \tag{9}$$

$$f_z = f_b\cos\varphi\cos\theta. \tag{10}$$

3. Detection method and landing pad design

In this section, the landing pad design and the onboard vision system for providing the relative pose between the quadrotor and the landing platform will be investigated.

3.1. Landing pad design & detection

To solve the negative influence caused by the low-cost camera, we design the landing pad as shown in Fig. 3. The landing pad consists of different size ArUco markers, which can make it can be detected by camera from high as well as low altitudes and decreases the possibility of loss of tracking target. The bigger patterns can be detected from high altitudes as shown in Fig. 4(a)(b). When the quadrotor is close to the platform during landing, these patterns gradually leave the camera Field of View (FOV), while the smaller ones could still be detected as shown in Fig. 4(c)(d). This configuration is designed to achieve the goals of keeping the desired target within the camera FOV and reducing measurement noise, which can compensate the influence of the low-cost monocular camera.

Comparing to single marker, multiple markers can increase the recall rate. The landing pad can be detected when at least one of the makers is detected. If the recall rate for each marker is p_i, the recall rate for multiple markers is $1 - \sum_i (1 - p_i)$. In addition, multiple markers can increase the detection accuracy. Because it can reduce the measurement noise to average the position of multiple markers. The estimation precision and recall rate are tested using single marker and our landing pad and the result is shown in Table 1.

Table 1
Single marker vs. Multiple markers.

	Recall rate	Estimation precision
Single marker	99.1%	3.68 cm
Multiple markers	99.9%	3.12 cm

Table 2
Recall rate for different size markers at different distance.

	0.2 m	0.5 m	1 m	3 m	5 m	7 m
5 cm	100%	97.2%	88.4%	7.2%	0%	0%
10 cm	99.2%	99.1%	97.2%	89.3%	31.6%	18.7%
15 cm	96.2%	95.2%	99.1%	99.5%	90.2%	45.8%
20 cm	37.4%	90.2%	97.8%	99.7%	99.1%	84.1%
30 cm	0%	89.7%	95.1%	100%	100%	100%

Table 3
Estimation precision for different size markers at different distance.

	0.2 m	0.5 m	1 m	3 m	5 m	7 m
5 cm	2.73 cm	4.83 cm	9.84 cm	63.25 cm	–	–
10 cm	1.82 cm	3.29 cm	4.47 cm	12.61 cm	49.15 cm	80.36 cm
15 cm	1.64 cm	3.07 cm	3.68 cm	8.38 cm	15.26 cm	46.71 cm
20 cm	1.23 cm	2.38 cm	2.94 cm	6.77 cm	9.54 cm	18.63 cm
30 cm	–	1.72 cm	2.28 cm	4.49 cm	7.35 cm	10.83 cm

Moreover, it is easy to modify detection range by changing the pattern size in terms of different application scenarios. In our case, the side length of the landing pad is 60 cm. The side length of big and small marker is 15 cm and 10 cm respectively. To choose a suitable landing pad, we need consider estimation precision and recall rate in different relative distance. Recall rate is defined as

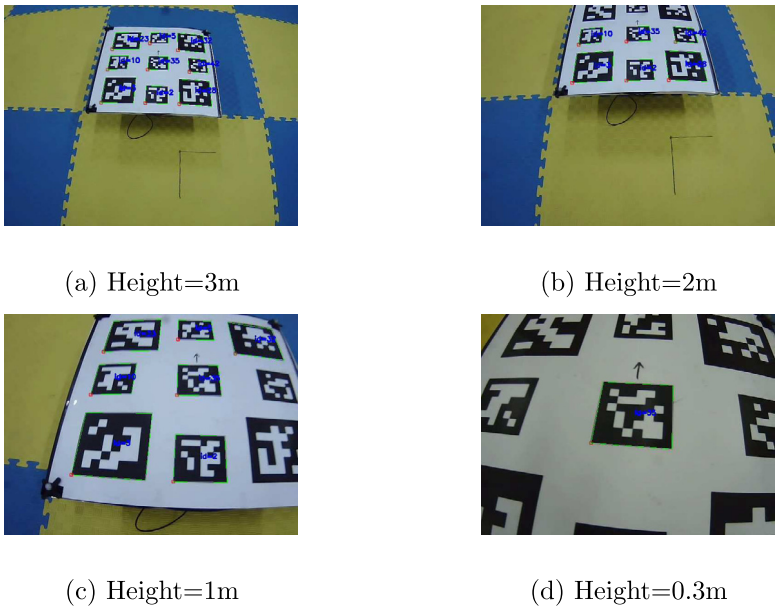
$$Recall = \frac{TP}{TP + FP}. \tag{11}$$

where true positive (TP) is expressed as the total number of markers detected and decoded successfully, false positive (FP) as the total number of markers detected and decoded unsuccessfully.

To simulation the camera vibration in real application, we let the quadrotor hover in different altitudes and we test the recall rate and estimation precision for different size makers with 1500 images in each altitude. The result is shown in Tables 2 and 3.

From the table, we can find the estimation precision and recall rate is decided by the size of marker and the relative distance. In Table 1, generally, the recall rate is higher when the relative distance is smaller. However, when the relative distance is too small, some big marker cannot be detected. In Table 2, the estimation precision is better when the size of marker is bigger. However, the big marker will leave the camera FOV because of the maneuver of quadrotor. The choice of marker size has a trade-off between the recall rate and estimation precision.

The ArUco library [15] is chosen for the design of the landing pad and the onboard vision system. Its detection pipeline is more effective and faster than that of the similar AprilTag library [16], which is suited for our low-cost on-board computer. The marker detection process is composed of two main steps: the detection of marker candidates and codification analysis. In order to detect square shapes which are candidates, adaptive thresholding and contour extraction in combination with additional extra filtering are used in the detection process. The objective of the second step is to analyze inner codification. Firstly, perspective transformation is used to obtain the marker in canonical form. Then, aiming at the separation of white and black sections, Otsu thresholding method is used. Finally, bit by bit analysis is performed in order to determine whether the marker belongs to a specific marker dictionary.



(a) Height=3m (b) Height=2m

(c) Height=1m (d) Height=0.3m

Fig. 4. Detection at different heights.

The detection algorithm provides the identity of all detected markers, with four corners of each marker in the image. Using the real size of the marker, it is sufficient to estimate its full relative pose of the landing pad in absolute scale.

3.2. Relative pose estimation

The problem of estimating the pose of a calibrated camera using n reference 3D points and their corresponding 2D projections is commonly referred as Perspective-n-Point (PnP) problem. Regarding the ArUco markers, non-iterative methods like Efficient Perspective-n-Point (EPnP) are not considered, because they are not robust in planar case. The homography method for planar targets [17] followed by non-linear optimization using the well-known Levenberg–Marquardt (LM) algorithm to minimize reprojection error is selected to estimate the transformation matrix T_{MC} from camera frame to marker frame. Then the relative attitude between quadrotor and landing pad can be calculated with T_{MC} and the quadrotor position in marker frame can be calculated as follows:

$$\begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix} = T_{MC} \begin{bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{bmatrix}, \tag{12}$$

where C denotes camera frame and M denotes marker frame.

It is more convenient to estimate the optic central position relative to the static markers than to estimate the marker position relative to the quadrotor. However, the planar surface of the landing pad and the limitation of the camera make mirror effect and occasional misidentification of some of the detected markers happen more frequently. So the method of just weighting the average of raw estimated pose could not provide accurate result.

Remark 1. In this paper, the cost function of PnP method is the sum of squared distances between the observed marker corners and the projected marker corners in image frame. The optimization problem is to find such a pose that minimizes reprojection error. However, the cost function of PnP has two distinct minima which would lead to significant unstable results when all marker corners lie on a plane. Compared to global shutter cameras, rolling shutter cameras provide more noise which will influence the initial pose of the optimization problem. If the initial pose is closer to the local minima, a wrong relative pose will be solved. Because the wrong estimated pose and the true pose look like in two different sides of a mirror, we call it mirror effect.

To solve this issue, a 3D points cluster algorithm is implemented which does not need to define the amount of the clusters. Firstly, the error between two estimated positions $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ are defined in (13).

$$error_{ij} = \|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2 \tag{13}$$

Then, a threshold is defined for checking whether two positions are in the same cluster. After checking all estimated positions, the cluster with the largest amount of data is the correct detection cluster. At last, weighted average of correct detection data is the optimal relative position and yaw angle between quadrotor and landing pad. Particularly, if more than one clusters contain the largest amount of data, then we choose the one which is the nearest to the last correct position as the correct detection cluster. The 3D points cluster algorithm is shown in Alg. 1.

Fig. 5 shows the relative position in x-axis using raw detection results and 3D point cluster algorithm respectively. In Fig. 5, it can be found that the position estimation result are inaccurate using raw detection results. By the contrary, the erroneous position estimations are completely removed after apply by the 3D point cluster algorithm. Fig. 6 shows the same improvement in the relative yaw estimation. The ground truth is measured by OptiTrack system.

To verify the robustness of the onboard vision system, let the quadrotor move from 0.5 m to 2.5 m in marker's x-axis at a constant altitude of 2 m. The mean error and standard deviation of the estimation result are shown in Table 4

4. Control law

The control component of the landing task includes a tracking problem and a backstepping position control law, which will be introduced in this section.

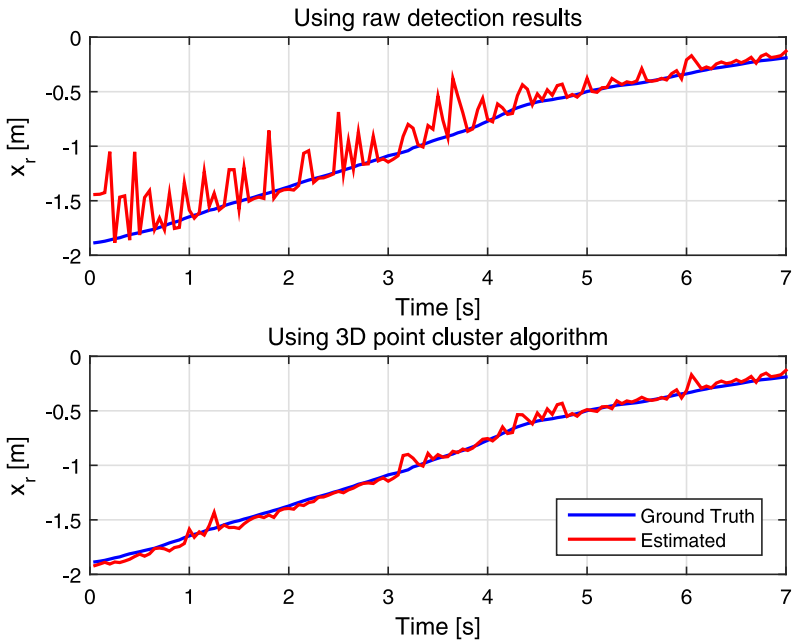


Fig. 5. Position estimation result.

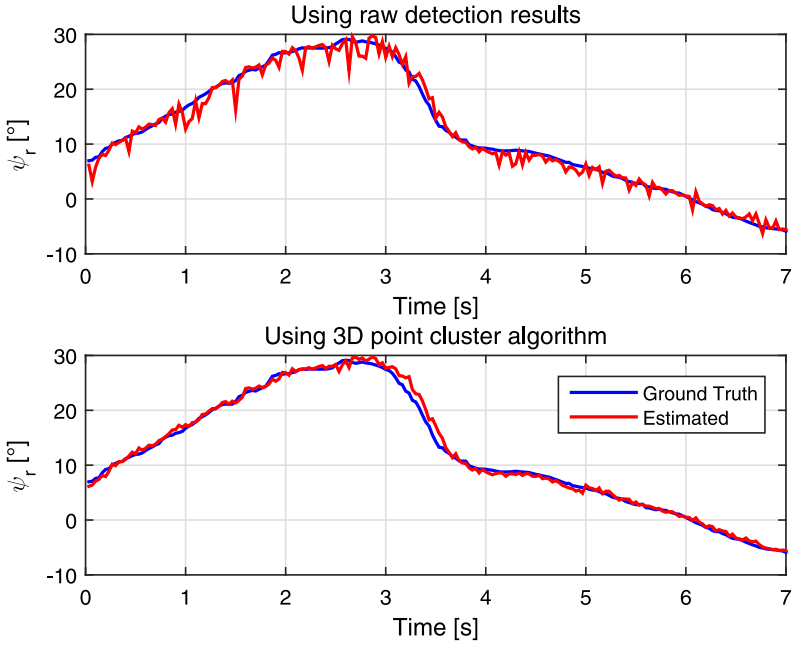


Fig. 6. Yaw estimation result.

Table 4
Estimation result at different distance.

Distance in x-axis (m)	0.5	1	1.5	2	2.5
Mean error (cm)	3.71	4.02	4.25	4.34	7.66
Standard deviation (cm)	4.13	4.53	4.84	5.37	9.12

4.1. Problem statement

As shown in Fig. 7, we assume that the landing pad is initially in the FOV of camera by ignoring the searching procedure.

From the onboard vision system, the position of the quadrotor in landing pad frame $\boldsymbol{p}^l_{uav} = \left[x^l_{uav}, y^l_{uav}, z^l_{uav}\right]^T$ and relative yaw angle $\Delta\psi$ can be measured. We exploit the knowledge of the quadrotor's pose in the world inertial frame $\sum_w$ to calculate the position of the landing pad in $\sum_w$. Due to the camera is set in front of the quadrotor, we can get

$$x^w_l = x - x^l_{uav}\cos\Delta\psi + y^l_{uav}\sin\Delta\psi, \tag{14}$$

$$y^w_l = y - x^l_{uav}\sin\Delta\psi - y^l_{uav}\cos\Delta\psi, \tag{15}$$

$$z^w_l = z - z^l_{uav}, \tag{16}$$

where $\left[x^w_l, y^w_l, z^w_l\right]^T$ is the position of landing pad in world inertial frame $\sum_w$. Take the time derivative of (14), (15) and (16) and

Algorithm 1 3D points cluster algorithm

Input: D: a set of 3D points to be clustered; Yaw: a set of yaw angle corresponding to 3D points; p_{last}: last optimal points centroid; *threshold*: distance threshold;
Output: optimal points centroid c^*; optimal yaw yaw^*;

```
k=0(number of clusters);
N = {n_1, n_2..., n_k}(set of clusters number);
C = {c_1, c_2..., c_k}(set of cluster centroids);
Y = {y_1, y_2..., y_k}(set of average yaw);
for all p_i ∈ D do
    yaw_i ∈ Yaw corresponding to p_i;
    if k = 0 then
        k ← 1; n_1 ← 1;
        c_1 ← p_i; y_1 ← yaw_i;
    else
        found ← false;
        for j ← 1 to k do
            if ||p_i − c_j||_2 < threshold then
                update(c_j, y_j);
                n_j ← n_j + 1;
                found ← true;
                break;
            end if
        end for
        if found = false then
            k ← k + 1; n_k ← 1;
            c_k ← p_i; y_k ← yaw_i;
        end if
    end if
end for
n_max = max(N);
for all i such that (i ≤ k) and (n_i = n_max) do
    index ← arg min_i ||c_i − p_last||_2;
end for
c* ← c_index; yaw* ← y_index;
```

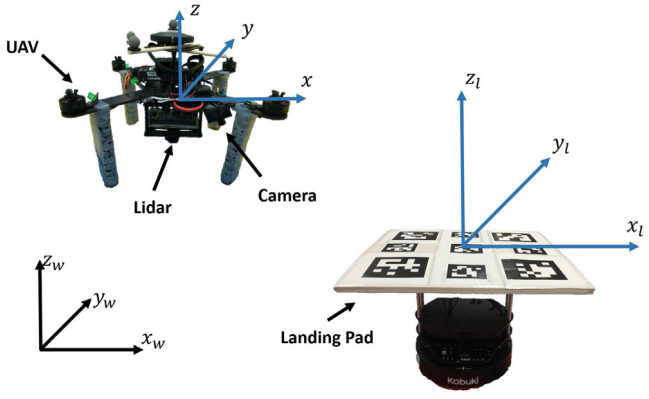


Fig. 7. Quadrotor and landing pad.

we can get

$$\dot{x}_l^w = \dot{x} - \dot{x}_{uav}^l \cos\Delta\psi + \dot{y}_{uav}^l \sin\Delta\psi + x_{uav}^l \Delta\dot{\psi} \sin\Delta\psi + y_{uav}^l \Delta\dot{\psi} \cos\Delta\psi, \tag{17}$$

$$\dot{y}_l^w = \dot{y} - \dot{x}_{uav}^l \sin\Delta\psi - \dot{y}_{uav}^l \cos\Delta\psi - x_{uav}^l \Delta\dot{\psi} \cos\Delta\psi + y_{uav}^l \Delta\dot{\psi} \sin\Delta\psi, \tag{18}$$

$$\dot{z}_l^w = \dot{z} - \dot{z}_{uav}^l. \tag{19}$$

The reference state of the position loop can be set as $\boldsymbol{p}_d = \left[x_l^w, y_l^w, z_l^w\right]^T$ and $\boldsymbol{v}_d = \left[\dot{x}_l^w, \dot{y}_l^w, \dot{z}_l^w\right]^T$. A tracking control law is presented in next subsection ensuring that $\boldsymbol{p} \to \boldsymbol{p}_d$ and $\boldsymbol{v} \to \boldsymbol{v}_d$.

4.2. Control law

The main objective of this section is to design a backstepping position controller of quadrotor ensuring that the position $\boldsymbol{p}$ and velocity $\boldsymbol{v}$ tracks the desired position $\boldsymbol{p}_d$ and velocity $\boldsymbol{v}_d$ respectively. To enhance tracking of the position controller, an adaptive rule is described to account of the external disturbance.

Let the position error $\boldsymbol{z}_1 = \boldsymbol{p} - \boldsymbol{p}_d$ and $\dot{\boldsymbol{z}}_1 = \boldsymbol{v} - \boldsymbol{v}_d$. To apply the standard backstepping procedure, introduce the new state variable

$$\boldsymbol{z}_2 = \boldsymbol{v} - \boldsymbol{\alpha}, \tag{20}$$

where $\boldsymbol{\alpha}$ is a virtual control input. It would stabilize $\boldsymbol{z}_1$ to zero and is given by

$$\boldsymbol{\alpha} = -\boldsymbol{K}_1 \boldsymbol{z}_1 + \boldsymbol{v}_d, \tag{21}$$

where $\boldsymbol{K}_1 > 0$ is a positive diagonal matrix. From (20) and (21), the velocity tracking error can be written as

$$\dot{\boldsymbol{z}}_1 = -\boldsymbol{K}_1 \boldsymbol{z}_1 + \boldsymbol{z}_2. \tag{22}$$

Take the time derivative of (20) and using the equation of (5), (6) and (7) results in

$$\dot{\boldsymbol{z}}_2 = \dot{\boldsymbol{v}} - \dot{\boldsymbol{v}}_d + \boldsymbol{K}_1(\boldsymbol{v} - \boldsymbol{v}_d), \tag{23}$$

$$m\dot{\boldsymbol{z}}_2 = -mg\boldsymbol{e}_3 + \boldsymbol{F}_w + \boldsymbol{\Delta}_t - m\dot{\boldsymbol{v}}_d + m\boldsymbol{K}_1(\boldsymbol{v} - \boldsymbol{v}_d). \tag{24}$$

Considering a Lyapunov function

$$\boldsymbol{V}\left(\boldsymbol{z}_1, \boldsymbol{z}_2, \tilde{\boldsymbol{\Delta}}_t\right) = \frac{1}{2}\boldsymbol{z}_1^T\boldsymbol{z}_1 + \frac{1}{2}m\boldsymbol{z}_2^T\boldsymbol{z}_2 + \frac{1}{2}\tilde{\boldsymbol{\Delta}}_t^T\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\Delta}}_t, \tag{25}$$

where $\boldsymbol{\Gamma} > 0$ is a positive diagonal matrix, and $\tilde{\boldsymbol{\Delta}}_t = \hat{\boldsymbol{\Delta}}_t - \boldsymbol{\Delta}_t$ is the external disturbance estimation error and $\hat{\boldsymbol{\Delta}}_t$ is an estimate of the external disturbance $\boldsymbol{\Delta}_t$. The time derivative of $\boldsymbol{V}\left(\boldsymbol{z}_1, \boldsymbol{z}_2, \tilde{\boldsymbol{\Delta}}_t\right)$ is

$$\begin{aligned}
&\dot{\boldsymbol{V}}\left(\boldsymbol{z}_1, \boldsymbol{z}_2, \tilde{\boldsymbol{\Delta}}_t\right) \\
&= \boldsymbol{z}_1^T\dot{\boldsymbol{z}}_1 + m\boldsymbol{z}_2^T\dot{\boldsymbol{z}}_2 + \tilde{\boldsymbol{\Delta}}_t^T\boldsymbol{\Gamma}^{-1}\dot{\hat{\boldsymbol{\Delta}}}_t \\
&= \boldsymbol{z}_1^T(\boldsymbol{v} - \boldsymbol{v}_d) + m\boldsymbol{z}_2^T\dot{\boldsymbol{z}}_2 + \tilde{\boldsymbol{\Delta}}_t^T\boldsymbol{\Gamma}^{-1}\dot{\hat{\boldsymbol{\Delta}}}_t \\
&= \boldsymbol{z}_1^T(\boldsymbol{z}_2 + \boldsymbol{\alpha} - \boldsymbol{v}_d) + m\boldsymbol{z}_2^T\dot{\boldsymbol{z}}_2 + \tilde{\boldsymbol{\Delta}}_t^T\boldsymbol{\Gamma}^{-1}\dot{\hat{\boldsymbol{\Delta}}}_t \\
&= \boldsymbol{z}_1^T\boldsymbol{z}_2 - \boldsymbol{z}_1^T\boldsymbol{K}_1\boldsymbol{z}_1 + \tilde{\boldsymbol{\Delta}}_t^T\boldsymbol{\Gamma}^{-1}\dot{\hat{\boldsymbol{\Delta}}}_t \\
&+ \boldsymbol{z}_2^T(-mg\boldsymbol{e}_3 + \boldsymbol{F}_w + \boldsymbol{\Delta}_t - m\dot{\boldsymbol{v}}_d + m\boldsymbol{K}_1(\boldsymbol{v} - \boldsymbol{v}_d)) \\
&= -\boldsymbol{z}_1^T\boldsymbol{K}_1\boldsymbol{z}_1 + \tilde{\boldsymbol{\Delta}}_t^T\boldsymbol{\Gamma}^{-1}\dot{\hat{\boldsymbol{\Delta}}}_t \\
&+ \boldsymbol{z}_2^T(\boldsymbol{z}_1 - mg\boldsymbol{e}_3 + \boldsymbol{F}_w + \boldsymbol{\Delta}_t - m\dot{\boldsymbol{v}}_d + m\boldsymbol{K}_1(\boldsymbol{v} - \boldsymbol{v}_d)).
\end{aligned} \tag{26}$$

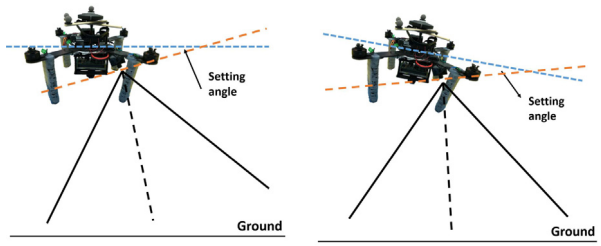
The stabilization of z_1 and z_2 can be obtained by constructing the control input and external disturbance estimate law as

$$\boldsymbol{F}_w = -\boldsymbol{z}_1 + mg\boldsymbol{e}_3 - \hat{\boldsymbol{\Delta}}_t + m\dot{\boldsymbol{v}}_d - m\boldsymbol{K}_1(\boldsymbol{v} - \boldsymbol{v}_d) - \boldsymbol{K}_2\boldsymbol{z}_2, \tag{27}$$

$$\dot{\hat{\boldsymbol{\Delta}}}_t = \boldsymbol{\Gamma}\boldsymbol{z}_2, \tag{28}$$

where $\boldsymbol{K}_2$ is a diagonal positive matrix. Substituting (27) and (28) to (26), we obtain

$$\dot{\boldsymbol{V}}\left(\boldsymbol{z}_1, \boldsymbol{z}_2, \tilde{\boldsymbol{\Delta}}_t\right) = -\boldsymbol{z}_1^T\boldsymbol{K}_1\boldsymbol{z}_1 - \boldsymbol{z}_2^T\boldsymbol{K}_2\boldsymbol{z}_2 \le 0. \tag{29}$$



(a) Hovering (b) Moving forward.

Fig. 8. Setting angle.

From (29), it is obvious to conclude that $\dot{\boldsymbol{V}}\left(\boldsymbol{z}_1, \boldsymbol{z}_2, \tilde{\boldsymbol{\Delta}}_t\right)$ is negative semi-definite. With (23), (27) and (28), closed-loop system (3) can be written as the following matrix form

$$\begin{bmatrix} \dot{\boldsymbol{z}}_1 \\ \dot{\boldsymbol{z}}_2 \end{bmatrix} = \begin{bmatrix} -\boldsymbol{K}_1 & \boldsymbol{I} \\ -\frac{1}{m}\boldsymbol{I} & -\frac{1}{m}\boldsymbol{K}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{m}\boldsymbol{I} \end{bmatrix} \tilde{\boldsymbol{\Delta}}_t, \tag{30}$$

$$\dot{\tilde{\boldsymbol{\Delta}}} = \begin{bmatrix} 0^T & \frac{1}{m}\boldsymbol{\Gamma} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_1 \\ m\boldsymbol{z}_2 \end{bmatrix}. \tag{31}$$

In conclusion, the equilibrium of system (30) and (31) is uniformly asymptotically stable, which implies that $\boldsymbol{p} \rightarrow \boldsymbol{p}_d$ and $\boldsymbol{v} \rightarrow \boldsymbol{v}_d$.

With (8), (9) and (10), the reference state for the roll and pitch angles can be computed as

$$\phi_d = \arctan\left(\frac{f_x \cos\psi + f_y \sin\psi}{f_z}\right), \tag{32}$$

$$\theta_d = \arcsin\left(\frac{f_x \sin\psi - f_y \cos\psi}{f_b}\right). \tag{33}$$

The relative yaw angle $\Delta\psi$ between the quadrotor and landing pad can be obtained from the onboard vision system and the desired value of yaw angle can be computed as

$$\psi_d = \psi + \Delta\psi. \tag{34}$$

5. Landing strategies

In our quadrotor system, gimbal camera is not adopted because that it is expensive and will decrease the payload ability of the system. As we know that quadrotor changes its attitude to achieve translation movement. Towards a fixed camera, aggressive maneuvers may make the landing pad out of the camera FOV, which can in turn cause failure of the detection (especially in the case of a moving target). To avoid this situation, we setup the camera in front of the quadrotor with a setting angle. We assume that the landing pad is in front of the quadrotor. When the quadrotor moves forward, the landing pad is still in the camera FOV as shown in Fig. 8.

According to the aforementioned arguments, a landing scheme is proposed and the state machine diagram is given in Fig. 9.

At the beginning, the quadrotor searches for the landing platform. When more than two correct tags are detected by onboard vision system, the quadrotor starts to track the landing pad. Generally, the camera can detect more markers in a higher altitude. When the tracking of the pad is lost (no marker is detected), the quadrotor increases its altitude slowly, until the landing pad is redetected. We use the relative position error between the

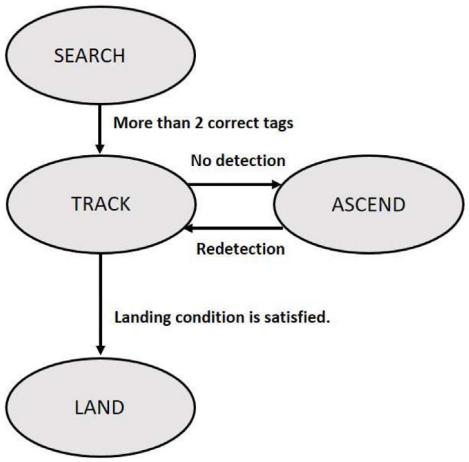


Fig. 9. State machine for landing.

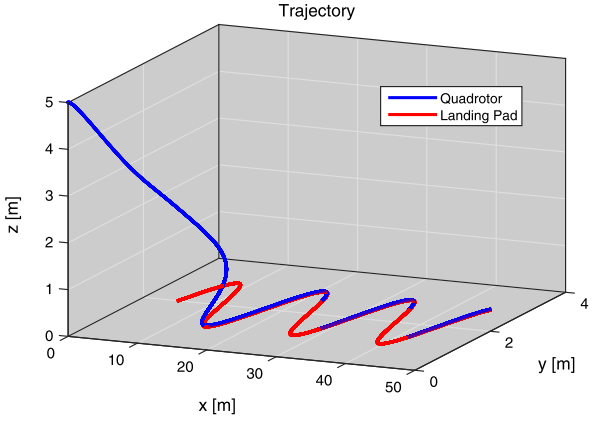


Fig. 10. Trajectory of quadrotor (blue line) and landing platform (red line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

quadrotor and landing pad $\boldsymbol{p}_r = \boldsymbol{p} - \boldsymbol{p}_l$ as the landing criteria. The landing condition is satisfied when

$$\boldsymbol{p}_r = \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} \leq \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \end{bmatrix}, \tag{35}$$

where δ_x, δ_y and δ_z are positive constants. If the landing condition (35) is satisfied, the motors will be turned off directly and then quadrotor will land on the landing pad quickly.

Because the accuracy of depth estimated by onboard vision system is limited, a facing downward single-point lidar is used to measure the accuracy vertical distance z_r between the quadrotor and the landing pad. It helps to determine the time to turn off motors and avoid the collision. Due to the ground effect, the stabilization of the quadrotor becomes extremely difficult when approaching the landing pad. Hence, the quadrotor will fly above the minimum altitude of the quadrotor $z_{\min} < \delta_z$, which can reduce the impact of the ground effect.

6. Simulation results

In this section, simulation results are shown to evaluate the efficiency of the proposed landing controller (27)–(28) on top of a high-gain inner-loop attitude controller. The presented simulations consider landing problem of the quadrotor on a moving platform.

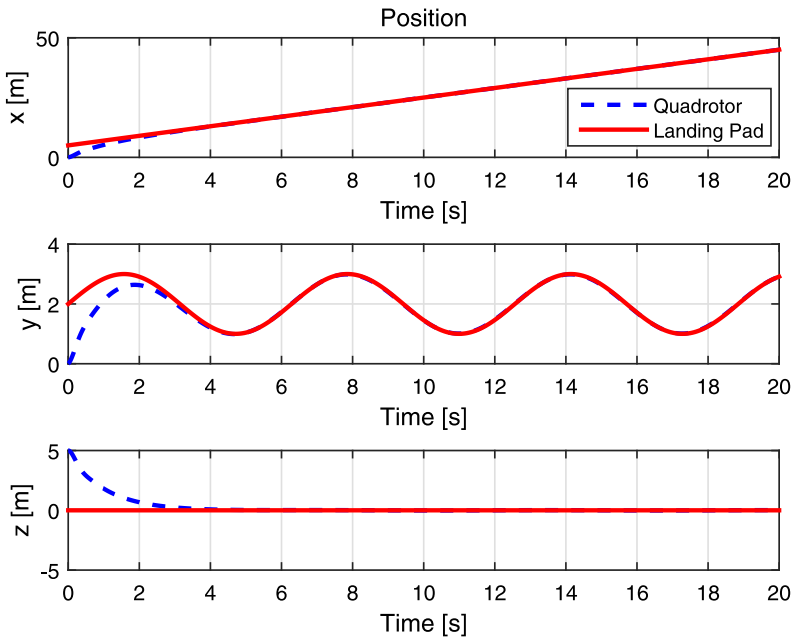


Fig. 11. Position of the quadrotor (dashed line) and position of the landing pad (solid line).

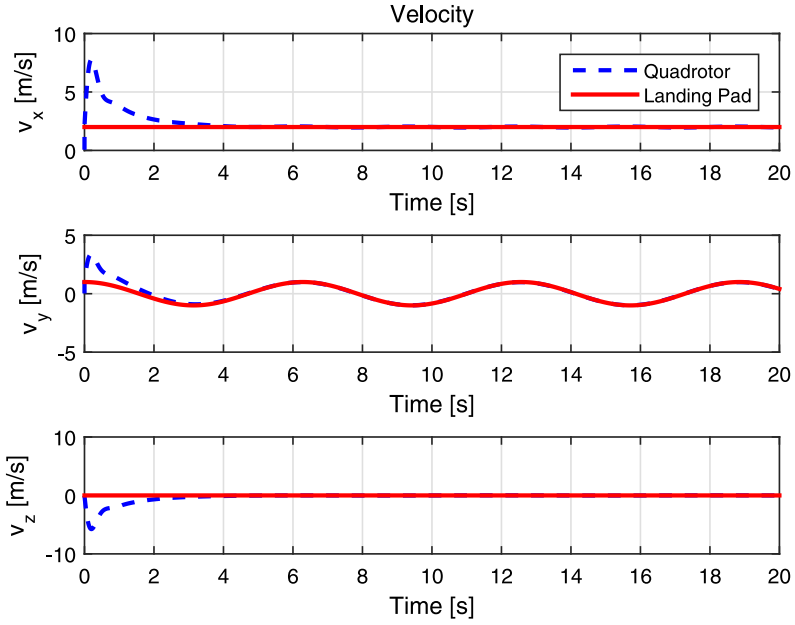


Fig. 12. Velocity of the quadrotor (dashed line) and velocity of the landing pad (solid line).

The mass of the vehicle is chosen $m = 1.1$ kg. It corresponds to the physical mass of the quadrotor that is used for experiments. At the begging, the quadrotor hover in the air and the initial position of the quadrotor is $\boldsymbol{p} = [0, 0, 5]$. The trajectory of landing platform is $\boldsymbol{p}_l = [5 + 2t, 2 + sin(t), 0]$. The control gain is set to $K_1 = diag\begin{pmatrix}1 & 1 & 1\end{pmatrix}$, $K_2 = diag\begin{pmatrix}10 & 10 & 10\end{pmatrix}$ and $\Gamma = diag\begin{pmatrix}90 & 90 & 90\end{pmatrix}$. The external disturbance are chosen as $\boldsymbol{\Delta}_t = [2 + 2sin(2t), 1 + 2cos(t), -3 + 2sin(t)]$. The simulation results are shown from Figs. 10 to 11. From these plots, it can be concluded that the numerical results illustrate the landing controller successful tracking performance.

Fig. 10 shows the trajectory of the quadrotor(blue line) and landing platform(red line) in 3-D space. Figs. 11 and 12 depicts the position and the velocity of the quadrotor(dashed line) and landing pad(solid line) with time. It can be seen that the position and velocity of quadrotor can converges to the position and velocity of the landing pad. Fig. 13 depicts the external disturbance(solid line) and its estimation(dashed line). It can be seen that the estimation can converges to the external disturbance.

7. Experiment results

In this section, the proposed approach for autonomous landing is validated with two scenarios. This section presents the experimental setup and results of experiments including landing on a moving platform including linear movement and rotational movement. Then, analysis on the experimental results will be presented .

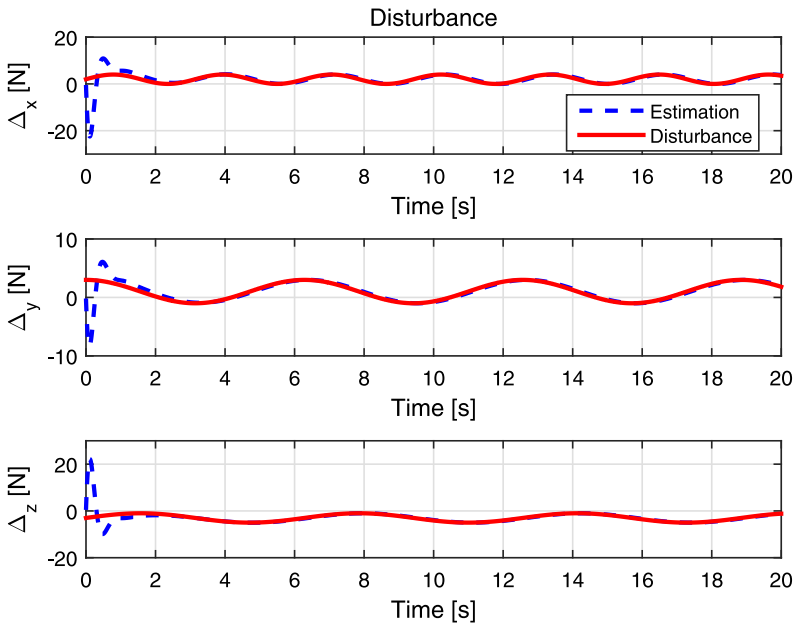


Fig. 13. External disturbance (solid line) and its estimation (dashed line).

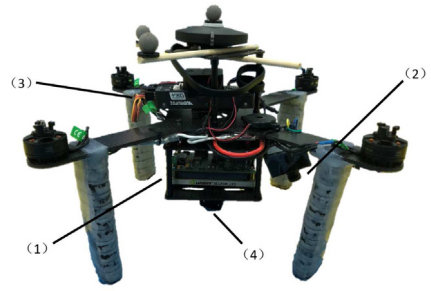


Fig. 14. Quadrotor and its components. (1) The onboard computer running our algorithms. (2) The monocular camera used for detection. (3) The PX4 autopilot. (4) The lidar used to measure altitude.

7.1. Experimental setup

The quadrotor used in this study is shown in Fig. 14. The overall system architecture, used to perform the experiments for validation of autonomous landing, is shown in Fig. 15. The quadrotor is developed based on the off-the-shelf frame. It weighs about 1.1 kg with battery included and the arm length form the center of mass to each motor is 20 cm. An open source autopilot hardware (Pixhawk [18]) and autopilot software (PX4 [19]) was adopted. The onboard computer (Nvidia TX1 [20]) with a ARM processor running at 1.7 GHz runs three major software components:

(1) interface with camera and lidar, detection algorithm and relative pose estimation;
(2) computation of control inputs and state machine;
(3) interface with flight controller, sending control inputs to the inner-loop controller [21].

In our experiment, the position of the quadrotor is measured by a motion capture system (OptiTrack [22]). Using the motion capture system can reduce the negative influence caused by inaccurate navigation system and we can focus on the negative influence caused by low-cost camera. Those values are computed at the ground computer and transmitted to the onboard computer via a Wi-Fi communication by vrpn-clinet-ros package [23]. The

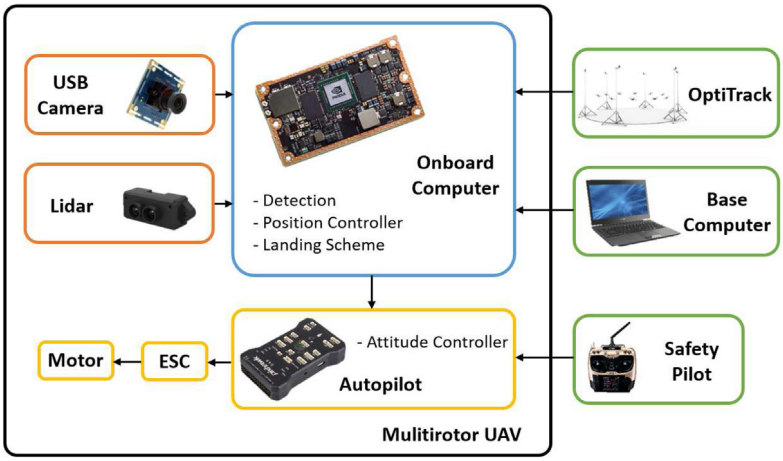


Fig. 15. Experiments setup.

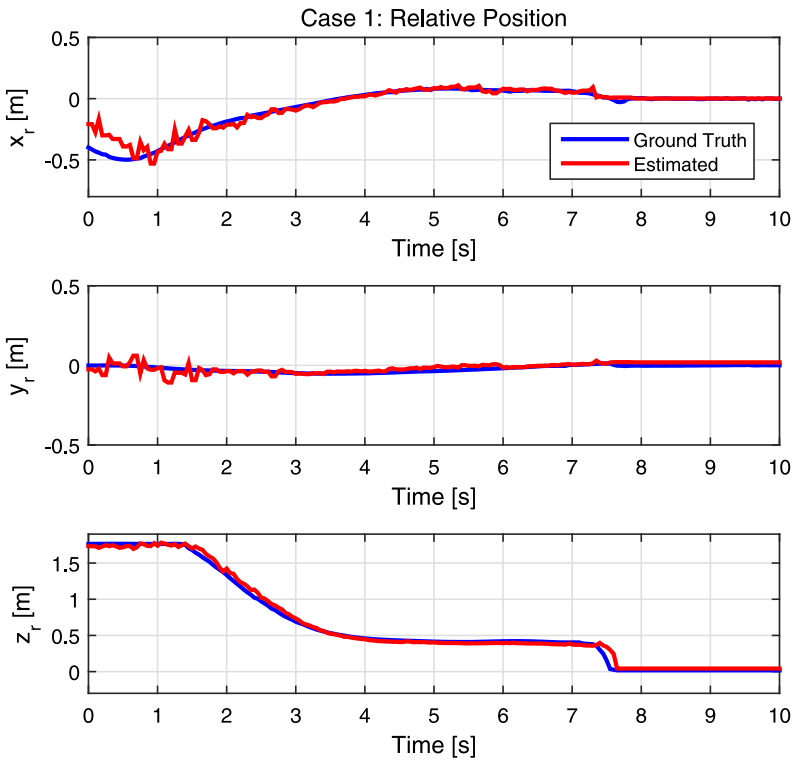


Fig. 16. Case 1: Relative position measured by Opti-Track (blue line) and onboard vision system (red line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

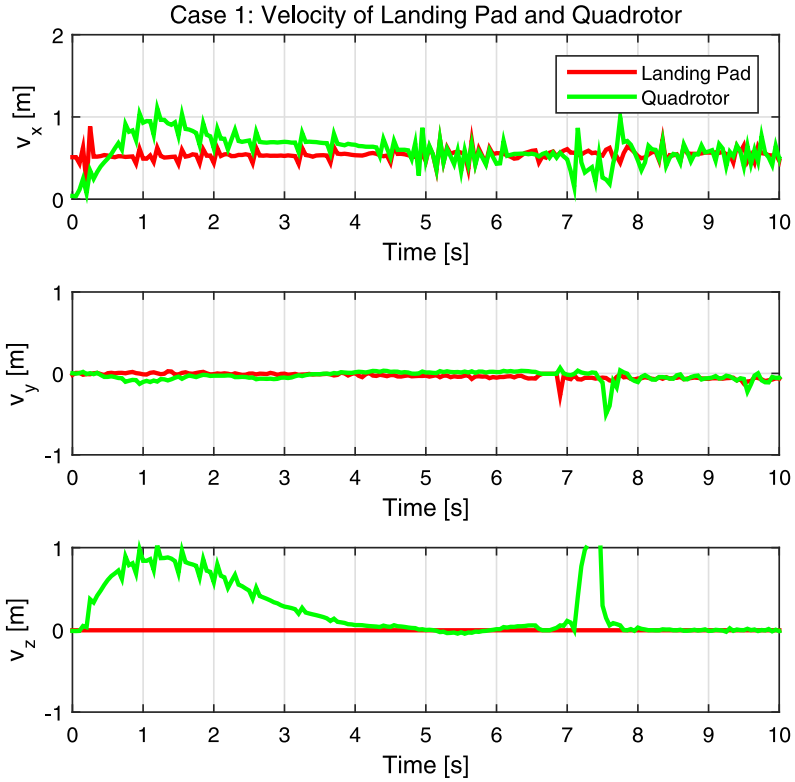


Fig. 17. Case 1: Velocities of quadrotor (green line) and landing pad (red line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

landing pad is fixed on the Turtle Bot [24] which can achieve linear and rotational movement. To detect the landing pad, a monocular camera is fixed in front of the quadrotor with a setting angle. The camera has about 90° FOV and the resolution of the

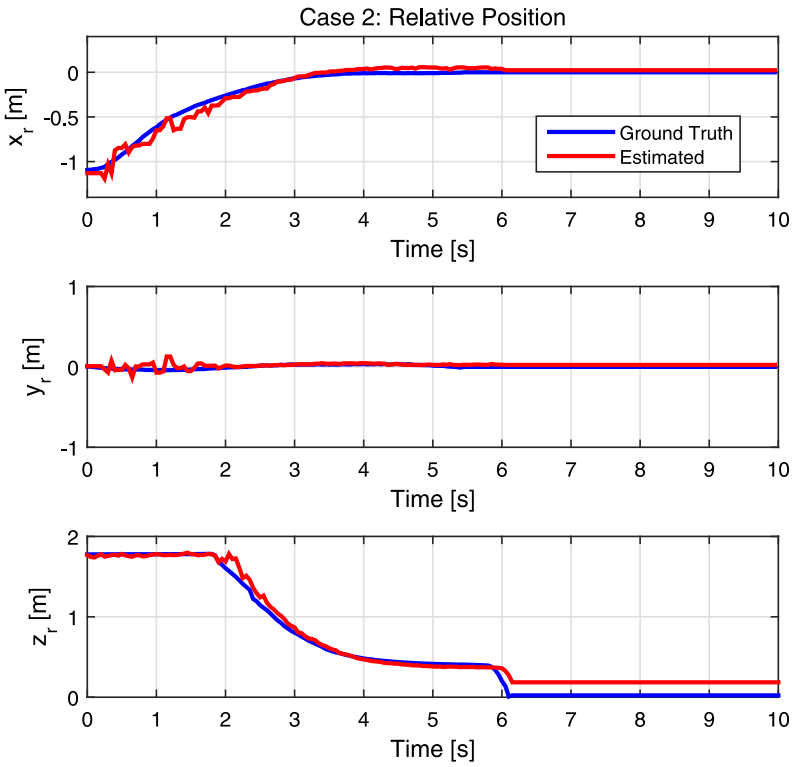


Fig. 18. Case 2: Relative position measured by Opti-Track (blue line) and onboard vision system (red line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

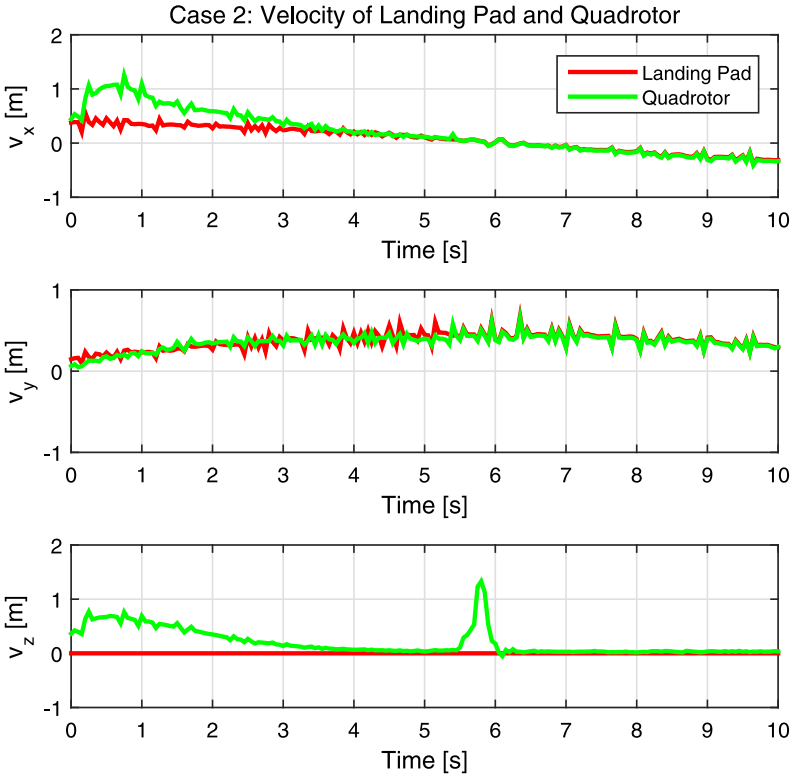


Fig. 19. Case 2: Velocities of quadrotor (green line) and landing pad (red line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Total cost of the quadrotor.

Hardware	Price (USD)
Nvidia TX1	250
Pixhawk	60
Camera	20
Lidar	15
Frame	10
Other components	10
Total cost	365

image is 640*480 with 100 fps. The total cost of our quadrotor is shown in Table 5. The average price of global shutter camera without lens in [6,8,11] is 340 USD.

The high price of Nvidia TX1 is mainly on the GPU, which is not used in our system. In addition, its CPU computing power is far beyond the amount of calculation in our task. For our task (including image processing and control input calculation), the CPU utilization of Nvidia TX1 is only 20% as shown in bellowing figure. And It can be seen that our task can be achieved using other low-cost onboard computer, e.g. Raspberry Pi, Odroid XU4.

7.2. Experiment results: Case 1

In Case 1, this section presents the autonomous landing scenario on a linear moving landing pad. For brevity, the landing platform moves on a straight line alone x axes at 0.5 m/s. The experiment results are presented in Figs. 16 and 17. We choose the same control gains as simulation case. For the landing criteria, we choose $\delta_x = 5$ cm, $\delta_y = 5$ cm , $\delta_z = 40$ cm and $z_{\min} = 30$ cm. Fig. 16 depicts the relative position between the quadrotor and landing pad $\boldsymbol{p}_r = [x_r, y_r, z_r]^T$. The red line represents the estimated value from onboard vision system and blue line represents the ground truth from OptiTrack. It can be seen that the estimation results coincide with ground truth and the relative position x_r, y_r, z_r converges to zero. Fig. 17 shows the time history of the velocities of quadrotor (green line) and landing pad (red line). It can be seen that the quadrotor's horizontal velocity tracks the platform's steadily and the vertical velocity converges to zero. When the landing criteria is satisfied, the quadrotor will drop on the landing pad. Therefore, there is a sudden change at 7.2 s in z axes.

7.3. Experiment results: Case 2

In Case 2, this section presents the autonomous landing scenario on a circular moving landing pad. The landing moves with a linear velocity at 0.5 m/s and rotational velocity at 0.2 rad/s. We choose the same control gains and landing criteria as Case 1. The experimental results are shown from Figs. 18 to 20. Fig. 18 depicts the relative position between the quadrotor and landing pad measured by OptiTrack (blue line) and onboard vision system (red line) respectively. Note that estimated value of z_r at the end of the mission is not zero, because the landing platform is out of the FOV of vision system when they are close enough. Fig. 19 shows the time history of the velocities of quadrotor (green line) and landing pad (red line). Fig. 20 shows the evolution of the yaw angle of the quadrotor (green line) and landing pad (red line). It can be seen that the yaw angle of the quadrotor tracks the yaw angle of the landing pad steadily.

8. Conclusion

In this paper, we present a low-cost vision-based quadrotor system capable of autonomous landing on a moving platform. For the autonomous landing task, the dynamic model of the quadrotor is simplified. A novel landing pad is designed for robust detection, ensuring detectability from high as well as low altitude. A 3D point cluster algorithm for pose estimation is presented to solve the problem of mirror effect and occasional misidentification. Based on the backstepping approach, an adaptive position controller was designed to enhance the landing task robustness and stability. Some landing strategies are presented to decrease the negative influence caused by low-cost system. In future work, we plan to improve the estimation accuracy of vision system and test our control law in outdoor environment.

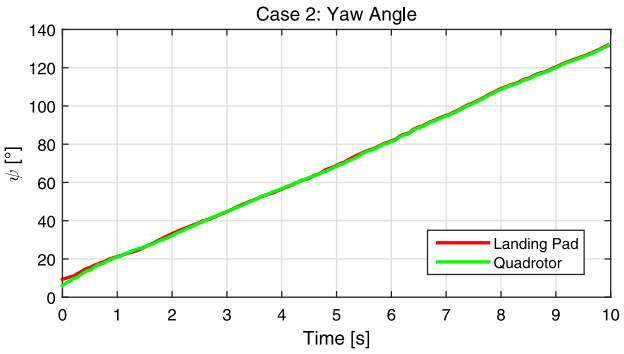


Fig. 20. Case 2: Yaw angle of quadrotor (green line) and landing pad (red line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Supplementary data

The video of the experiments can be found in https://v.youku.com/v_show/id_XMzc3ODM3MTQ4OA==.html?firsttime=2 or https://youtu.be/EaPGIvZqMnE

Acknowledgments

This work was supported by the National Nature Science Foundation of China under Grant No.61873031.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.robot.2019.05.004.

References

[1] R. Mahony, V. Kumar, P. Corke, Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor, IEEE Robot. Autom. Mag. 19 (3) (2012) 20–32.
[2] Y. Qi, J. Wang, J. Shan, Aerial cooperative transporting and assembling control using multiple quadrotor–manipulator systems, Internat. J. Systems Sci. (2017) 1–14.
[3] T. Cieslewski, E. Kaufmann, D. Scaramuzza, Rapid exploration with multi-rotors: A frontier selection method for high speed flight, in: Ieee/Rsj International Conference on Intelligent Robots and Systems, 2017.
[4] H. Lee, H. Kim, Constraint-based cooperative control of multiple aerial manipulators for handling an unknown payload, IEEE Trans. Ind. Inf. 13 (6) (2017) 2780–2790.
[5] C.S. Sharp, O. Shakernia, S.S. Sastry, A vision system for landing an unmanned aerial vehicle, in: IEEE International Conference on Robotics and Automation, 2001. Proceedings, 2003, pp. 1720–1727 vol.2.
[6] S. Yang, S.A. Scherer, A. Zell, An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle, J. Intell. Robot. Syst. 69 (1–4) (2013) 499–515.
[7] T. Hamel, R. Mahony, Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow, IEEE Trans. Robot. 28 (1) (2012) 77–89.
[8] P. Serra, R. Cunha, T. Hamel, D. Cabecinhas, C. Silvestre, Landing of a quadrotor on a moving target using dynamic image-based visual servo control, IEEE Trans. Robot. PP (99) (2016) 1–12.

[9] S. Lange, N. Sunderhauf, P. Protzel, A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments, in: International Conference on Advanced Robotics, 2009, pp. 1–6.
[10] T.S. Richardson, C.G. Jones, A. Likhoded, E. Sparks, A. Jordan, I. Cowling, S. Willcox, Automated vision-based recovery of a rotary wing unmanned aerial vehicle onto a moving platform, J. Field Robotics 30 (5) (2013) 667–684.
[11] H. Lee, S. Jung, D.H. Shim, Vision-based UAV landing on the moving vehicle, in: International Conference on Unmanned Aircraft Systems, 2016, pp. 1–7.
[12] X. Chen, S.K. Phang, M. Shan, B.M. Chen, System integration of a vision-guided UAV for autonomous landing on moving platform, in: IEEE International Conference on Control and Automation, 2016, pp. 761–766.
[13] M. Grundmann, V. Kwatra, D. Castro, I. Essa, CaLibration-free rolling shutter removal, in: 2012 IEEE International Conference on Computational Photography (ICCP), IEEE, 2012, pp. 1–8.
[14] G. Schweighofer, A. Pinz, Robust pose estimation from a planar target, IEEE transactions on pattern analysis and machine intelligence 28 (12) (2006) 2024–2030.
[15] S. Garrido-Jurado, R. Mu?oz-Salinas, F.J. Madrid-Cuevas, M.J. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recognit. 47 (6) (2014) 2280–2292.
[16] E. Olson, Apriltag: A robust and flexible visual fiducial system, in: IEEE International Conference on Robotics and Automation, 2011, pp. 3400–3407.
[17] S. Malik, Robust 2d tracking for real-time augmented reality, in: Proceedings of Vision Interface Calgary Alberta Canada, 2002.
[18] Pixhawk, http://www.pixhawk.com.
[19] PX4, https://github.com/px4/Firmware.
[20] Nvidia TX1, https://developer.nvidia.com/embedded/buy/jetson-tx1.
[21] Mavros, https://github.com/mavlink/mavros.
[22] OptiTrack, http://www.optitrack.com.
[23] vrpn-client-ros, https://github.com/ros-drivers/vrpn-client-ros.
[24] Turtle Bot, http://www.turtlebot.com.

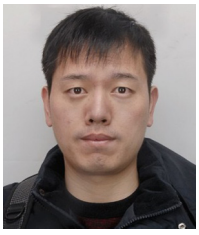
Yuhua Qi received his B.S. degree from Beijing Institute of Technology, Beijing China, in 2010. He is currently a Ph.D. candidate at Beijing Institute of Technology. His research interests include cooperative control and autonomous UAV systems.

Jiaqi Jiang received his B.S. degree from the Beijing Institute of Technology (BIT) in 2016. He is currently pursuing the master's degree with School of Aerospace Engineering, BIT. His research interests span over target detection, target tracking, autonomous systems, and multi-agent system.

Jin Wu was born in May, 1994 in Jiangsu Province, China. He received the B.S. degree in automation engineering from University of Electronic Science and Technology of China, Chengdu, China.

His research interests include inertial navigation, optimal filtering, control theory and robot vision. He has published over 20 technical papers on academic journals e.g. IEEE Transactions, IEEE Journals, IET Journals and etc.

Jianan Wang is currently an Associate Professor in the School of Aerospace Engineering at Beijing Institute of Technology, Beijing, China. He received his B.S. and M.S. in Control Science and Engineering from the Beijing Jiaotong University and Beijing Institute of Technology, Beijing, China, in 2004 and 2007, respectively. He received his Ph.D. in Aerospace Engineering at Mississippi State University, Starkville, MS, USA in 2011. His research interests include cooperative control of multiple dynamic systems, UAV formation control, obstacle/collision avoidance, trustworthy networked system, and estimation of sensor networks. He is a senior member of IEEE and AIAA.



Chunyan Wang received the M.Sc. degree in electrical and electronic engineering from the University of Greenwich, London, U.K., in 2012, and the Ph.D. degree in control systems from the University of Manchester, Manchester, U.K., in 2016.

He was a Research Associate with the School of Electrical and Electronic Engineering, the University of Manchester from 2016 to 2018. He is currently an Associate Professor with the School of Aerospace Engineering, Beijing Institute of Technology (BIT). His current research interests include cooperative control, robotics, and control of UAVs.

Jiayuan Shan received the B.S. degree from Huazhong University of Science and Technology in 1988, and the M.S. and Ph.D. degrees from Beijing Institute of Technology, in 1991 and 1999, respectively. He is currently a Professor at Beijing Institute of Technology.His research interests include guidance, navigation and control of the aircraft and hardware-in-the loop simulation. He is the Director of Department of Flight Vehicles Control and the Deputy Director of Flight Dynamics and Control Key Laboratory of Ministry of Education.