World Academy of Science, Engineering and Technology
International Journal of Aerospace and Mechanical Engineering
Vol:17, No:1, 2023

Image-Based UAV Vertical Distance and Velocity Estimation Algorithm during the Vertical Landing Phase Using Low-Resolution Images

Seyed-Yaser Nabavi-Chashmi, Davood Asadi, Karim Ahmadi, Eren Demir

***Abstract*—**The landing phase of a UAV is very critical as there are many uncertainties in this phase, which can easily entail a hard landing or even a crash. In this paper, the estimation of relative distance and velocity to the ground, as one of the most important processes during the landing phase, is studied. Using accurate measurement sensors as an alternative approach can be very expensive for sensors like LIDAR, or with a limited operational range, for sensors like ultrasonic sensors. Additionally, absolute positioning systems like GPS or IMU cannot provide distance to the ground independently. The focus of this paper is to determine whether we can measure the relative distance and velocity of UAV and ground in the landing phase using just low-resolution images taken by a monocular camera. The Lucas-Konda feature detection technique is employed to extract the most suitable feature in a series of images taken during the UAV landing. Two different approaches based on Extended Kalman Filters (EKF) have been proposed, and their performance in estimation of the relative distance and velocity are compared. The first approach uses the kinematics of the UAV as the process and the calculated optical flow as the measurement. On the other hand, the second approach uses the feature's projection on the camera plane (pixel position) as the measurement while employing both the kinematics of the UAV and the dynamics of variation of projected point as the process to estimate both relative distance and relative velocity. To verify the results, a sequence of low-quality images taken by a camera that is moving on a specifically developed testbed has been used to compare the performance of the proposed algorithm. The case studies show that the quality of images results in considerable noise, which reduces the performance of the first approach. On the other hand, using the projected feature position is much less sensitive to the noise and estimates the distance and velocity with relatively high accuracy. This approach also can be used to predict the future projected feature position, which can drastically decrease the computational workload, as an important criterion for real-time applications.

***Keywords*—**Automatic landing, multirotor, nonlinear control, parameters estimation, optical flow.

I. Introduction

THE current trend to use fully automated UAVs in urban areas in applications like surveillance, delivery service, and air taxi requires to consider safety issues seriously, especially in failure scenarios. There is an increasing requirement for the reliability, safety, and fault tolerances of UAVs as any minor fault may have catastrophic results [1]. In these scenarios, it may be required to have an emergency landing in an unprepared landing position while it is not accurately known. One of the key parameters which should be measured/estimated for a safe landing is the relative distance and velocity between the UAV and ground. In a general perspective, distance estimation can be looked at as a specific form of localization and positioning of the UAV which is generally performed using sensors like LIDAR, GPS, IMU, Ultrasonic, and camera. There are different estimation mechanisms for positioning and localization, each uses specific sensors and differs in terms of accuracy, operational range, reliability, cost, weight, power consumption, and computational burden. Some of these sensors used for UAV positioning in literature and their qualitative comparison are presented in Table I. Fusing GPS and vision information is a classical solution to Simultaneous Localization And Mapping (SLAM) during the flight [2] and using an ultrasonic sensor [3] is a common exercise in relative distance estimation during the landing. For example, [4] applied two LIDAR sensors to analyze the movement in a maritime application to reduce the dependency on the positioning system on GPS signals. Considering the weight and power limitations, the applicability of such methods is in doubt. Currently, vision sensors as powerful, lightweight, and accurate tools are used widely as a rich source of information [5] allowing the UAV to react to the actual scene [6].

One of the applications of image sensors in the landing phase is landing site detection and navigating the UAV to the landing site. Some researchers used images through the categorization techniques [7] or feature detection techniques [8] to detect a suitable site and then employ sensors like IMU and GPS for the automatic landing of the UAV. However, landing on an unknown landing site or in GPS denied environments is a challenge that these researches cannot address. To tackle the reliance on GPS signals, image data were used in researches like [9] and [10] to estimate the relative position/attitude of the UAV to a known landing site, which is marked with pre-determined features. Similarly, [11] fused the visual and inertial information obtained from the camera and IMU, using a Kalman filter, to estimate the relative velocity and position of the UAV and a ship to predict the touchdown point. However,

* This research is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under the 3501 program with project number [120M793].

Seyed-Yaser Nabavi-Chashmi and Karim Ahnmadi are researchers with the Adana Alparslan Science and Technology University, Adana, Turkey (e-mail: synabavi@atu.edu.tr, kadastgerdi@atu.edu.tr).

Davood Asadi is an assistant professor at Adana Alparslan Science and Technology University, Adana, Turkey (corresponding author, phone: +90-553-83220246; e-mail: dasadihendoustani@atu.edu.tr).

Eren Demir is BSc student at Adana Alparslan Science and Technology University, Adana, Turkey (e-mail: daerendemir@gmail.com).

because of the reliance of these methods on known characteristics of the landing site, they are not suitable for failure scenarios in which landing on an unprepared or even unknown site may be required.

Some other researchers used image sensors to analyze the physical condition of the site and its safety issues instead of estimating the relative position of the drone and the ground. In this category, some tried to reconstruct the area using image information to evaluate height maps [12], [13]. Having the 3D reconstructed map of the landing site can be used to analyze the suitability of the site for the landing of the UAV. However, because of the computational burden of these methods, they cannot be used for online estimation of relative distance and velocity. In some other researches, in this category, Convolutional Neural Network (CNN) is used by [14] to detect the landing site and avoid the congested landing areas. The focus of these researches is safety, and relative position and velocities are not their concern.

TABLE I
COMPARISON BETWEEN DIFFERENT SENSORS FOR DISTANCE ESTIMATION

	Accuracy	Range	Reliability	Cost	Weight	Power	Computation	Duration
GPS	A	E	A	A	E	A	A	E
Inertial Navigation System	E	E	E	X	A	A	X	P
Pressure Altimeter	P	E	E	P	P	E	E	E
Laser Range Finder	E	P	E	P	P	A	E	A
Ultrasonic sensor	E	X	A	E	E	E	E	E
Vision camera	A	A	A	E	E	A	P	E

X: Very Poor, P: Poor, A: Average, E: Excellent

The other application of image sensors during the landing phase is the guidance and control of the UAV. Landing control is a problem that is tackled by some researches using approaches like sliding mode control [15] or robust approaches [16], however controllers typically rely on relative height and velocity. To solve this problem some researchers [19]-[21] used optical flow as a measure of landing quality. The relative position and velocity between the UAV and ground can also be estimated by tracking these features in sequencing images. Using optical flow for UAVs' guidance and control is a bio-inspired approach, utilized by many researchers [22]-[24] in different phases of UAV operation. Optical flow can be interpreted as the distribution of apparent velocities of movement of brightness patterns in an image which is the result of the relative motion of objects and the viewer [17]. A sequence of the images taken by a monocular camera can be used to calculate the optical flow. The images of this sequence should be used through a feature detection algorithm like the Lukas-Kanade method to deter some features on the ground. The position of these features in the camera plane is used as the measurements to calculate the optical flow of the detected features. References [18] and [19] apply the optical flow method to provide a soft and smooth landing for a UAV while [20] examines optical flow to avoid ground during the cruise flight of UAVs. On the other hand, [3], [21] and [22] focused on the estimation of the position and velocity of a quadcopter in the landing phase.

The problem of simultaneous height estimation and vertical flight control has been studied by [23]. Accordingly, an adaptive proportional feedback controller is examined, while its parameters were adjusted using the vertical distance estimates provided by a flow divergence estimator. A variation of the NDI method is proposed in [24] to deal with the nonlinearities of optical flow control in which the model parameters were estimated during flight. Despite the investigation of the control algorithm, there is no discussion or results about the optical flow estimation accuracy in this paper. Since the proposed controller is robust to uncertainties [25], [26], having a smooth landing does not necessarily mean that the system is identified properly.

In this paper, the problem of estimation of distance and velocity of a quadcopter in the landing phase using the vision sensors is studied. To make the proposed suitable for scenarios that the height of the UAV is high, low-quality images have been used in the testbed. For the estimation process, two approaches have been developed. In the first approach, called Optic-Flow Based approach (OFBA), the optic flow equation has been considered as the measurement equation while the kinematics equations of the UAV have been considered as the process equation. In the second approach, called Feature-Position Based approach (FPBA), the detected feature position in the camera plane has been considered as the measurement and both the optic-flow equation and the Kinematics equations of the UAV have been considered as the process equations. Both approaches use EKF for estimation while the first one estimates just distance and velocity while the second approach estimates pixel positions as well as the distance and the velocity.

The presented formulation for the dynamics of UAV and optic flow includes all translational and rotational movements. However, considering that the landing phase is the main problem of this research, it is assumed in case studies that the angular rates of the UAV are negligible and the UAV just moves downward. A specific testbed has been developed to perform the experimental tests. The studies show that the first approach is always observable while in the second approach, in specific situations, the observability matrix is not full rank. The results also show that the second approach is much better than the first approach in terms of accuracy and sensitivity to the measurement noises. As the EKF can be employed to predict the future, the second approach enables us to have a prediction about the future feature position and so apply the image processing techniques to the neighborhood of that point. This prediction can be used to reduce the computational load or to

use more accurate feature detection techniques.

II. Estimation Process and Formulation

A. Optical Flow Formulation

In this section, the dynamic of the variation of the projection of the detected feature in the camera plane is presented. To formulate the optical flow, the schematic presented in Fig. 1 is used. The body frame attached to the UAV is presented by $OX_BY_BZ_B$ where the camera is mounted in its z-direction. Assuming that the position of the detected feature in $OX_BY_BZ_B$ is $\vec{P}^B = [X\ Y\ Z\ ]^T$ (superscript B indicates the body coordinate system which) then its position in the camera plane ($\vec{P}_c = [x\ y\ f\ ]^T$) can be calculated as:

$$\vec{P}_c = \frac{f}{Z}\vec{P} \tag{1}$$

where f is the focal length of the camera. For brevity, this position will be called pixel position in the following. The time derivative of (1) can be written as:

$$\dot{\vec{P}}_c = \frac{f\dot{\vec{P}}}{Z} - \frac{f\vec{P}\dot{Z}}{Z^2} = \frac{f\dot{\vec{P}}}{Z} - \frac{\vec{P}_c\dot{Z}}{Z} \tag{2}$$

Considering the kinematic equations and assuming that the velocity of the quadcopter stated in the body frame, the velocity of a fixed point on the inertia frame can be stated in the body coordinate system as:

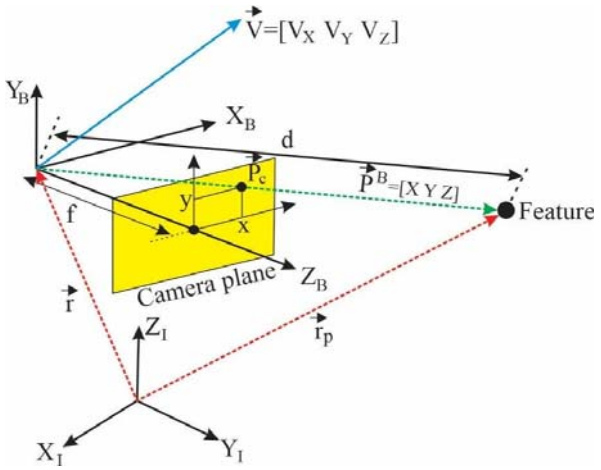


Fig. 1 Projection of the detected feature in the camera plane

$$\dot{\vec{P}}^B = -\dot{\vec{r}}^B - \vec{\omega}^B \times \vec{P}^B \tag{3}$$

Replacing $\vec{\omega}^B = [\omega_x\ \ \omega_y\ \ \omega_z]^T$ and $\dot{\vec{r}}^B = [V_x\ \ V_y\ \ V_z]^T$ in (3), and using (2), the optic flow equations can be presented as:

$$\dot{x} = -\frac{fV_x}{Z} - \omega_y f + \omega_z y + \frac{V_z x}{Z} + \frac{1}{f}\left(\omega_x xy - \omega_y x^2\right) \tag{4}$$

$$\dot{y} = -\frac{fV_y}{Z} - \omega_z x + \omega_x f + \frac{V_z y}{Z} + \frac{\omega_x y^2}{f} - \frac{\omega_y xy}{f} \tag{5}$$

This equation can be rearranged in the matrix form of:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \frac{1}{Z}\begin{bmatrix}-f & 0 & x\\ 0 & -f & y\end{bmatrix}\begin{Bmatrix}V_x\\ V_y\\ V_z\end{Bmatrix} + \begin{bmatrix}\frac{xy}{f} & -\left(f+\frac{x^2}{f}\right) & y\\ \left(f+\frac{y^2}{f}\right) & -\frac{xy}{f} & -x\end{bmatrix}\begin{Bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{Bmatrix} \tag{6}$$

For the UAV in the landing phase, it is assumed that $\vec{\omega} = 0$ and $V_x = V_y = 0$ in this case the dynamic is simplified as:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}\frac{xV_z}{Z}\\ \frac{yV_z}{Z}\end{bmatrix} \tag{7}$$

B. UAV Dynamics Formulation

To consider the dynamics of the UAV in the estimation process the kinematic equations can be used to relate the position, velocity, and acceleration of the UAV.

$$\begin{Bmatrix}\dot{\vec{P}}\\ \dot{\vec{V}}\end{Bmatrix}^I = \begin{bmatrix}0_{3\times3} & I_{3\times3}\\ 0_{3\times3} & 0_{3\times3}\end{bmatrix}\begin{Bmatrix}\vec{P}\\ \vec{V}\end{Bmatrix} + \begin{Bmatrix}0_{3\times1}\\ \vec{a}^I\end{Bmatrix} \tag{8}$$

while the velocity term ($\dot{\vec{r}}^B$) appears in (6) is in $OX_BY_BZ_B$ the position and velocity in (8) are in the inertia coordinate system. So, it is necessary to transform the position from the body frame to the inertia frame. To transform the position and velocity into the body frame, the Euler angles can be used through a convenient transformation matrix T_I^B which is in the form of:

$$\boldsymbol{T}_I^B = \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta)\\ s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & s(\phi)c(\theta)\\ c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) & c(\phi)c(\theta)\end{bmatrix} \tag{9}$$

where, $\phi, \theta,$ and ψ are the Euler angles also $c(x)$ and $s(x)$ stand for $cos\ cos\ (x)$ and $sin\ sin\ (x)$ respectively. Without loss of generality, the Euler angles can be considered as small values in the landing phase ($\varphi = \epsilon_\varphi, \theta = \epsilon_\theta$ and $\psi = \epsilon_\psi$) so ignoring the second-order terms, the transformation matrix can be presented as:

$$\boldsymbol{T}_I^B \approx \begin{bmatrix}1 & \epsilon_\psi & -\epsilon_\theta\\ -\epsilon_\psi & 1 & \epsilon_\phi\\ \epsilon_\theta & -\epsilon_\phi & 1\end{bmatrix}$$

C. Estimation Formulation Approaches

Two different estimation approaches have been applied in this research. In the first approach, (7) is used as the process equation to estimate $\vec{\alpha} = [Z\ V_z\ ]^T$ as the state vector. In this approach, the pixel position ($[x\ y\ ]$) is used to calculate the velocity of the pixel ($[\dot{x}\ \dot{y}\ ]^T$). To calculate the velocity, variation of the position of the pixel position in the sequence of images, taken with a specific imaging rate, is used.

$$\dot{x} = \frac{x(t+Ts)-x(t)}{T_s} \qquad \dot{y} = \frac{y(t+Ts)-y(t)}{T_s} \tag{10}$$

where, $x(t)$ and $y(t)$ are the measurements of pixel positions in consecutive images and T_s is the time difference between the

images. Based on (7) the measurement equation can be written:

$$\frac{V_z}{Z} = \frac{\dot{x}}{x} = \frac{\dot{y}}{y} \tag{11}$$

For a UAV with full rotational and translational movements, (6) should be used instead of (11), which makes the measurement equation more nonlinear.

In the second approach, the pixel position is also added to the state vector ($\vec{\alpha}$) to reduce the effects of image noises or low image qualities. So, in this approach, the state vector is $\vec{\alpha} = [x\ y\ Z\ V_z\ ]^T$ and (4), (5), and (7) are considered as the process equations. As the pixel position is calculated using the Lukas-Kanade algorithm, the measurement equation can be written as:

$$\vec{z} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \vec{\alpha} \tag{12}$$

This equation is completely linear and for a UAV with full rotational and translational movements, it will not change. Equations (10)-(12) show that vertical distance and velocity can be estimated using only the measurement of optical flow if the UAV angular velocity vector is kept to zero. It enables the UAV to measure its distance and velocity from the ground without the need for any other measurement.

D. Feature Detection Algorithm

To detect the feature the Lukas-Kanade algorithm has been employed. Lucas-Kanade Feature Tracker is a widely used feature tracking algorithm in computer vision because of its low computational cost and low iteration time. The following assumptions are considered in feature detection procedure:

1- The images contain textured objects with smooth changes
2- The tracked features are relatively close to each other between two consecutive images.

The algorithm tracks the given feature between any two consecutive images by monitoring the pixel intensity changes in the neighborhood of the feature in the first image. So, the feature tracker needs to calculate the pixel intensity at the location of a feature coordinates (x, y) in the first image. It should also calculate the rate of change of pixel intensity relative to both x and y directions (I_x, I_y). Then the tracker calculates the pixel intensity at the same location (x, y) in the second image. If the pixel intensity difference between two images, is (x, y), then,

$$-I_t(x,y) = I_x(x,y)\,.u + I_y(x,y)\,.v \tag{13}$$

where u and v are the movements of the feature in x-direction and y-direction between two images. Because a single-pixel cannot hold enough information N neighbors of the pixel are also used. Equation (13) for n neighbors can be written as:

$$\begin{aligned} -I_t(x_1,y_1) &= I_x(x_1,y_1)\,.u + I_y(x_1,y_1)\,.v \\ -I_t(x_2,y_2) &= I_x(x_2,y_2)\,.u + I_y(x_2,y_2)\,.v \\ -I_t(x_3,y_3) &= I_x(x_3,y_3)\,.u + I_y(x_3,y_3)\,.v \\ -I_t(x_n,y_n) &= I_x(x_n,y_n)\,.u + I_y(x_n,y_n)\,.v \end{aligned} \tag{14}$$

Equation (14) can be written as matrix form:

$$\Phi = \begin{bmatrix} I_x(x_1,y_1) & I_y(x_1,y_1) \\ I_x(x_2,y_2) & I_y(x_2,y_2) \\ \vdots & \vdots \\ I_x(x_n,y_n) & I_y(x_n,y_n) \end{bmatrix} \vec{\beta} = \begin{pmatrix} u \\ v \end{pmatrix} \vec{U} = \begin{bmatrix} -I_t(x_1,y_1) \\ -I_t(x_2,y_2) \\ \vdots \\ -I_t(x_n,y_n) \end{bmatrix} \tag{15}$$

Then this system of equations (15) with n equations and two unknowns can be solved using conventional algorithms like Least Square (LS) method as:

$$\vec{\beta} = (\Phi^T\Phi)^{-1}\Phi^T\vec{U} \tag{16}$$

E. Extended Kalman Filter

To overcome the nonlinearity of the measurement equation in the first approach and the nonlinearity of the process equation in the second approach EKF is used in this research to estimate the states of the system [27]. In this section, the EKF process and the related formulation are presented first and then it is specifically applied to the developed equations in the previous sections. We assume a process governed by nonlinear stochastic difference equations of:

$$\vec{x}_k = f(\vec{x}_{k-1}, \vec{u}_k, w_{k-1}) \tag{17}$$

and with the measurements:

$$\vec{z}_k = h(x_k, v_k) \tag{18}$$

where, $\vec{x}_k$ is the state vector, $\vec{u}_k$ is the control vector, w_k and v_k are measurement process and measurement noises respectively. They are assumed to be independent, white noise, and with normal probability distributions.

$$p(w) = N(0, Q), \quad p(v) = N(0, R) \tag{19}$$

where Q is the process noise covariance matrix and R is the measurement noise covariance matrix. Linearizing (17) and (18) concerning states and measurements results in:

$$\vec{x}_k \approx \tilde{x}_k + A(\vec{x}_{k-1} - \hat{x}_{k-1}) + W\vec{w}_{k-1} \tag{20}$$

$$\vec{z}_k \approx \tilde{z}_k + H(\vec{x}_k - \tilde{x}_k) + V\vec{v}_k \tag{21}$$

In these equations $\tilde{x}_k$ and $\tilde{z}_k$ are the approximate state and measurement vectors from (17) and (18) and $\hat{x}_k$ is an a posteriori estimate of the state at step k. The matrices A, W, H, and V are Jacobian of f and h. The EKF process is an iterative process where each step begins with receiving a new measurement and has the following steps:

Step1. Project the state ahead using (20).

Step2. Project the error covariance ahead using:

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \tag{22}$$

Step3. Compute the Kalman gain using

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \tag{23}$$

Step4. Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - h(\hat{x}_k^-, 0)\right) \tag{24}$$

Step5. Update the error covariance

$$P_k = (I - K_k H_k)P_k^- \tag{25}$$

In the first approach, the state vector is $\vec{\alpha} = [Z\ V_z\ ]^T$ and measurement vector is $\vec{z} = \frac{V_z}{Z}$ so the Jacobian matrices in this case are:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad H = \begin{bmatrix} -\frac{V_z}{Z^2} & \frac{1}{Z} \end{bmatrix} \tag{26}$$

The observability matrix for the states in this approach is:

$$M = \begin{bmatrix} -\frac{V_z}{Z^2} & \frac{1}{Z} \\ 0 & -\frac{V_z}{Z^2} \end{bmatrix} \tag{27}$$

This equation shows that the observability matrix is always of rank 2.

In the second approach, $\vec{\alpha} = [x\ y\ Z\ V_z\ ]^T$and measurement vector is $\vec{z} = [x\ y\ ]^T$. To use (7), the Jacobian matrices can be derived as:

$$A = \begin{bmatrix} -\frac{V_z}{Z} & 0 & \frac{xV_z}{Z^2} & -\frac{x}{Z} \\ 0 & -\frac{V_z}{Z} & \frac{yV_z}{Z^2} & -\frac{y}{Z} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{28}$$

To analyze the observability of this approach the matrices A and H can be rewritten as:

$$A = \begin{bmatrix} p\boldsymbol{I} & uv^T \\ \boldsymbol{O} & \boldsymbol{C} \end{bmatrix} \qquad H = [\boldsymbol{I} \quad \boldsymbol{O}] \tag{29}$$

where $\boldsymbol{I} = I_{2\times2}$ and $\boldsymbol{O} = 0_{2\times2}$ and $\boldsymbol{C} = [0\ 1\ 0\ 0\ ]$

$$p = -\frac{V_z}{Z} \qquad \xi = [x\ y\ ]^T \qquad \eta = \left[\frac{V_z}{Z^2} \quad -\frac{1}{Z}\right] \tag{30}$$

Using these definitions, the observability matrix in this case is:

$$M = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ p\boldsymbol{I} & \xi\eta^T \\ p^2\boldsymbol{I} & (p\boldsymbol{I} + \boldsymbol{C})\xi\eta^T \\ p^3\boldsymbol{I} & (p\boldsymbol{I} + \boldsymbol{C})\xi\eta^T\boldsymbol{C} + p^2\boldsymbol{I}\xi\eta^T \end{bmatrix} \tag{31}$$

Studying the two first columns of (31) this system is always observable if: $\frac{V_z}{Z} \neq 1$, $x \neq y$, $V_z \neq 0$ and $Z \neq 0$.

Equations (26) and (27) will be implemented in the next section to a testbed to verify the proposed algorithm and compare the performance of the two approaches.

III. Case Studies

To analyze the two approaches presented in the previous section, a specific testbed has been designed as presented in Fig. 2. The testbed uses an Arduino board to convert the desired trajectory to the appropriate commands and then used a CNC shield to derive the stepper motor to accurately move the imaging mechanism (including a Raspberry pi board and its camera) in the imaging direction. A test has been defined with parameters presented in Table II.

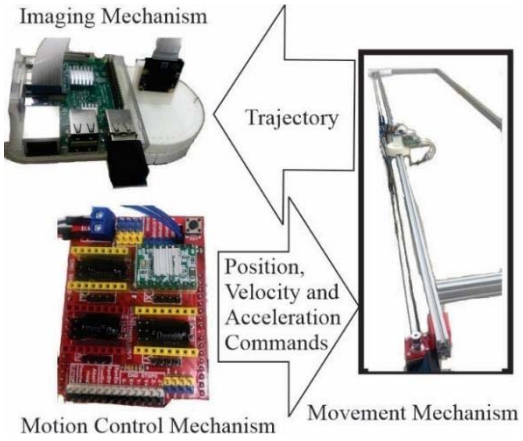


Fig. 2 Test Facility for Accurate Positioning of Camera

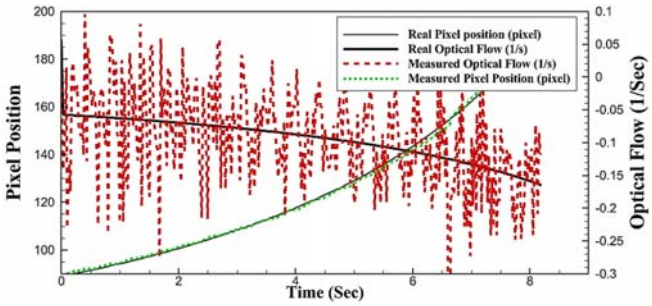


Fig. 3 Measured feature positions and related optical flow

TABLE II
Parameters of the Model

Parameter	Symbol	Unit	Value
Initial distance	d_0	m	2.04
Final distance	d_f	m	0.5
Initial velocity	V_0	m/s	0.117
Final velocity	V_f	m/s	0.167
Feature distance	X_f	m	0.2
Camera focal length	f	mm	0.304
Imaging sampling time	T_s	$1/s$	1/40
Image resolution	---	pixel	384*288

In Fig. 3 the position of the feature measured by the camera is presented by the green dashed curve and compared to its real value (solid thin black curve). It can be seen that the measurements follow the real values. However, the time derivative of the measured feature points (red dashed curve), which appears in (10) and (11), makes the calculated optic flow very noisy. These measurements have been used in both

approaches, to study their performance in the estimation of vertical distance and velocity. The filter parameters are presented in Table III.

TABLE III
FILTER PARAMETERS

Parameter	Symbol	Unit	Value
Initial covariance	P_0	--	100
Process noise	ω_n	--	0.00001
Measurement noise	v_n	--	0.05
Init. distance estimation	$\hat{d}_0$	m	3.0
Init. velocity estimation	$\hat{v}_0$	m/s	0.1
Init. pix pos. estimation	$\hat{x}_0$	m	

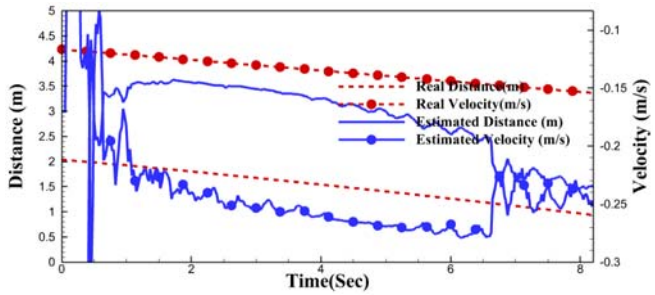
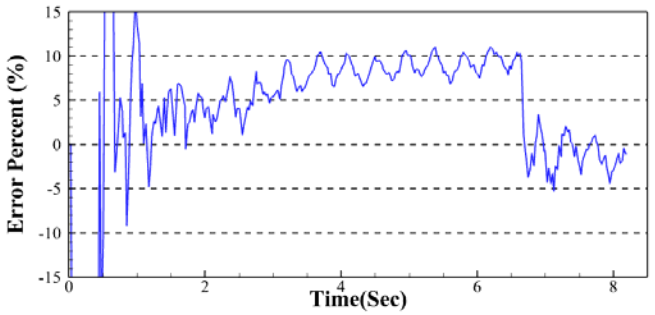
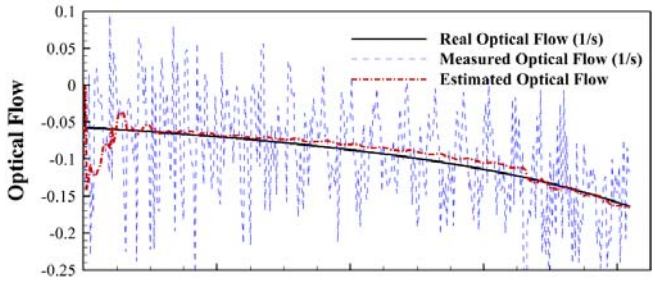


Fig. 4 Comparison of distance and velocity with their estimated values for the first approach



(a)



(b)

Fig. 5 Comparison of (a) estimation error and (b) real, measured, and estimated optical flows (first approach)

The results of applying the first approach to the measured data are presented in Fig. 4. In this figure, the solid lines are for estimated values while the dashed lines are for real values. Comparing the distances, presented on the left axis of the image shows that there is about 50 cm distance error after 8 seconds of estimation. Comparing the symbolled lines, it can be seen also that there is about 10 cm/sec velocity estimation error. However, Fig. 5 (a) shows that the estimation of optical flow is close to its real value with less than 5% error after 7 seconds. Fig. 5 (b) shows that although the estimated distance and velocity have considerable errors, their ratio (as optical flow) is relatively accurate. So, this approach can be suitable for applications like optical flow control (for example [24]), but if the vertical distance and velocity values are required, this approach is not suitable.

To estimate the vertical distance and velocity instead of their ratio, the second estimation approach is proposed. In the second approach, the camera's feature position is applied instead of the optical flow. The estimated distance and velocity are presented in Fig. 6. It can be seen that using the pixel position as the measurement parameter increases the distance and velocity estimations' accuracy greatly such that the final distance error is less than 10 cm and the final velocity error is less than 0.3 cm/s. The estimated pixel position is also compared with its real and measured values and it can be seen that the noises which are resulted from the low image qualities are removed and its real values are tracked accurately. The comparison between real, measured, and estimated pixel positions presented in Fig. 7 shows that the estimated pixel position is in accordance with its real value. Fig. 8 (b) shows that the resulting optical flow is accurate with a final error of less than 2% error which makes this approach suitable for applications that use optical flow control as the guidance strategy.

Fig. 9 presents the vertical velocity estimation using different approaches. Optical flow measurement and the pixel position measurement have been compared with the real velocity. Accordingly, pixel position measurement has better performance.

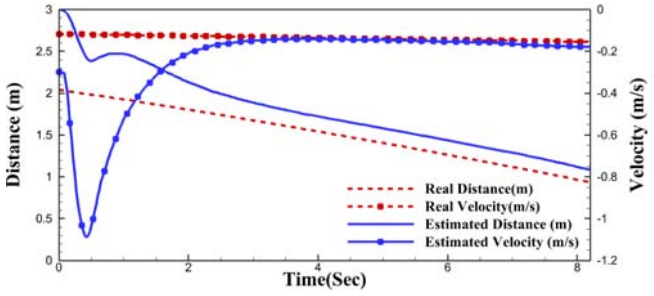


Fig. 6 Comparison of distance and velocity with their estimated values for the second approach

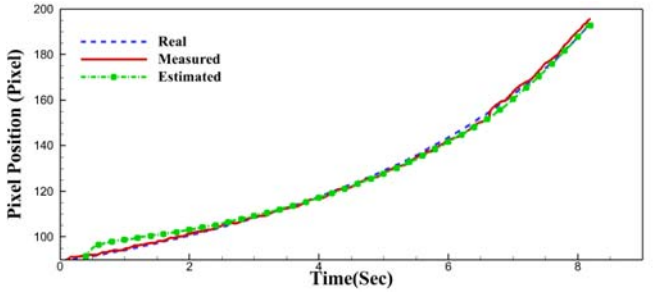
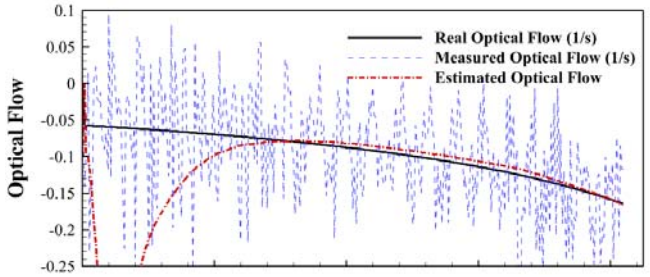
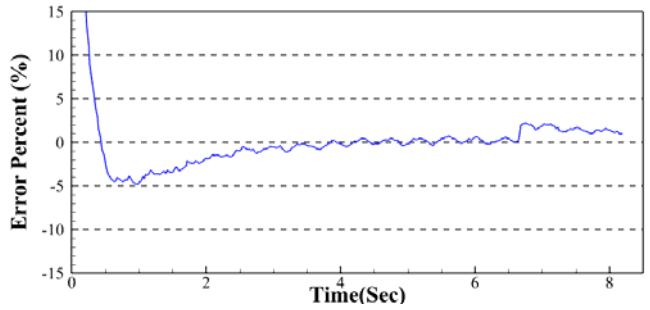


Fig. 7 Comparison of estimated pixel position with its measured, and real values for the second approach

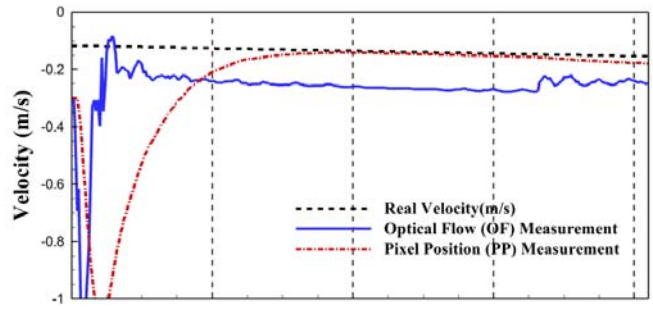


(a)

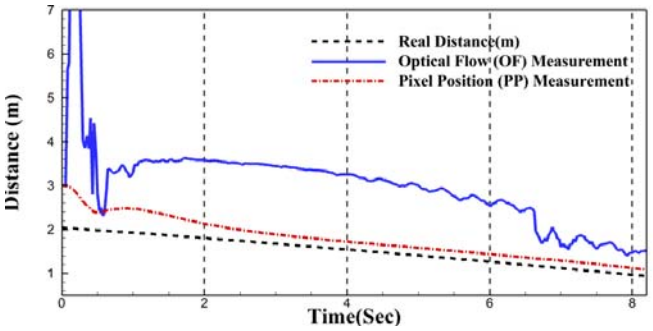


(b)

Fig. 8 Comparison of real, measured, and estimated optical flows and estimation error (second approach)



(a)



(b)

Fig. 9 Comparison of different approaches in the estimation of vertical velocity

IV. Conclusion

The main focus of this paper was to find how we can use image data to estimate the distance and velocity of a UAV during its landing. Our formulations showed that these parameters can be estimated generally by fusion of the camera and IMU data. However, in the landing phase, assuming low lateral movements and rotations, the image data are enough for the estimation of the relative distance and velocity to the ground. Two formulations have been proposed to estimate the states of the system (distance and velocity). The first one converts the detected feature position measurements into the optical flow and uses it as the measurement. While the other algorithm uses the feature position directly to estimate the states of the system. A testbed is designed and implemented to perform the tests with appropriate measurements of the states of the system. The evaluations show that:

- Theoretically, it is possible to estimate both the distance and velocity of UAV using only image data in the landing phase.
- The measurement noise is the main challenge for the estimation process when using optical flow to estimate the relative distance and velocity of the UAV, especially when the quality of the images is low.
- Using optical flow as the measured parameter makes the optical flow smoothed which is suitable for control optical flow control. However, the estimations of vertical distance and velocity are erroneous with respect to the other approach.
- Using pixel positions as the measured parameters requires some more calculations but estimations of vertical distance and velocity and optical flow are all more accurate with respect to the first approach.
- The second approach provides a good estimation of feature position. So, this approach can be used to predict the future points of the feature position.
- In summary, it is preferable to use feature positions as the measurement as it provides more accurate results and it also can be used for decreasing the computational burden by providing a prediction about the future point of the pixel position.

For further researches, it is planned to improve the algorithms to include all translational and rotational movements of the camera. Application of the algorithms on a real UAV and analyzing the algorithms is another interesting work to determine the applicability of developed algorithms in real scenarios.

Acknowledgment

This research is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under 3501 program, with project number [120M793].

References

[1] Asadi, D., Ahmadi, K., Nabavi-chashmi, S., and Ö., T. "Controlability of Multi-Rotors under Motor Fault Effect." Artıbilim: Adana Alparslan Türkeş Bilim ve Teknoloji Üniversitesi Fen Bilimleri Dergisi, Vol. 4, No. 2, 2021, pp. 24–43.

[2] G. Fink, M. Franke, A. F. Lynch, K. Röbenack, and B. Godbolt, "Visual Inertial SLAM: Application to Unmanned Aerial Vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1965–1970, 2017, doi: https://doi.org/10.1016/j.ifacol.2017.08.162.

[3] H. W. Ho and Q. P. Chu, "Automatic Landing System of a Quadrotor UAV Using Visual Servoing," *EuroGNC*, pp. 1264–1283, 2013, (Online).

Available: https://aerospace-europe.eu/media/books/delft-0046.pdf.

[4] S. Abujoub, J. McPhee, C. Westin, and R. A. Irani, "Unmanned Aerial Vehicle Landing on Maritime Vessels using Signal Prediction of the Ship Motion," *Ocean. 2018 MTS/IEEE Charleston, Ocean 2018*, 2019, doi: 10.1109/OCEANS.2018.8604820.

[5] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, "A survey on vision-based UAV navigation," *Geo-spatial Inf. Sci.*, vol. 21, no. 1, pp. 21–32, Jan. 2018, doi: 10.1080/10095020.2017.1420509.

[6] Z. Tang, R. Cunha, T. Hamel, and C. Silvestre, "Aircraft Landing Using Dynamic Two-Dimensional Image-Based Guidance Control," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 5, pp. 2104–2117, 2019, doi: 10.1109/TAES.2018.2881354.

[7] A. Yol, B. Delabarre, A. Dame, J. Dartois, and E. Marchand, "Vision-based absolute localization for unmanned aerial vehicles," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3429–3434, doi: 10.1109/IROS.2014.6943040.

[8] T. Baca *et al.*, "Autonomous landing on a moving vehicle with an unmanned aerial vehicle," *J. F. Robot.*, vol. 36, no. 5, pp. 874–891, 2019, doi: 10.1002/rob.21858.

[9] A. Borowczyk, D. T. Nguyen, A. Phu-Van Nguyen, D. Q. Nguyen, D. Saussié, and J. Le Ny, "Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10488–10494, 2017, doi: 10.1016/j.ifacol.2017.08.1980.

[10] A. Borowczyk, D. T. Nguyen, A. P. Van Nguyen, D. Q. Nguyen, D. Saussié, and J. Le Ny, "Autonomous landing of a quadcopter on a high-speed ground vehicle," *J. Guid. Control. Dyn.*, vol. 40, no. 9, pp. 2373–2380, 2017, doi: 10.2514/1.G002703.

[11] Y. Meng, W. Wang, H. Han, and J. Ban, "A visual/inertial integrated landing guidance method for UAV landing on the ship," *Aerosp. Sci. Technol.*, vol. 85, pp. 474–480, 2019, doi: 10.1016/j.ast.2018.12.030.

[12] T. Yang, P. Li, H. Zhang, J. Li, and Z. Li, "Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments," *Electron.*, vol. 7, no. 5, 2018, doi: 10.3390/electronics7050073.

[13] M. K. Mittal, R. Mohan, W. Burgard, and A. Valada, "Vision-Based Autonomous UAV Navigation and Landing for Urban Search and Rescue," *ArXiv*, vol. abs/1906.0, 2019.

[14] C. Symeonidis, E. Kakaletsis, I. Mademlis, N. Nikolaidis, A. Tefas, and I. Pitas, "Vision-based UAV Safe Landing exploiting Lightweight Deep Neural Networks," *ACM Int. Conf. Proceeding Ser.*, pp. 13–19, 2021, doi: 10.1145/3447587.3447590.

[15] S. Lee *et al.*, "Sliding Mode Guidance and Control for UAV Carrier Landing," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 2, pp. 951–966, 2019, doi: 10.1109/TAES.2018.2867259.

[16] D. Cabecinhas, R. Naldi, C. Silvestre, R. Cunha, and L. Marconi, "Robust Landing and Sliding Maneuver Hybrid Controller for a Quadrotor Vehicle," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 400–412, 2016, doi: 10.1109/TCST.2015.2454445.

[17] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1–3, pp. 185–203, 1981, doi: 10.1016/0004-3702(81)90024-2.

[18] G. De Croon, D. Alazard, and D. Izzo, "Controlling spacecraft landings with constantly and exponentially decreasing time-to-contact," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 2, pp. 1241–1252, 2015, doi: 10.1109/TAES.2014.130135.

[19] H. W. Ho, C. De Wagter, B. D. W. Remes, and G. C. H. E. de Croon, "Optical-flow based self-supervised learning of obstacle appearance applied to MAV landing," *Rob. Auton. Syst.*, vol. 100, pp. 78–94, 2018, doi: https://doi.org/10.1016/j.robot.2017.10.004.

[20] F. Ruffier and N. Franceschini, "Aerial robot piloted in steep relief by optic flow sensors," *2008 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 1266–1273, 2008, doi: 10.1109/IROS.2008.4651089.

[21] H. W. Cheng, T. L. Chen, and C. H. Tien, "Motion estimation by hybrid optical flow technology for UAV landing in an unvisited area," *Sensors (Switzerland)*, vol. 19, no. 6, pp. 1–13, 2019, doi: 10.3390/s19061380.

[22] A. Miller, B. Miller, A. Popov, and K. Stepanyan, "UAV Landing Based on the Optical Flow Videonavigation," *Sensors*, vol. 19, no. 6. 2019, doi: 10.3390/s19061351.

[23] H. W. Ho, G. C. H. E. De Croon, E. Van Kampen, Q. P. Chu, and M. Mulder, "Adaptive Gain Control Strategy for Constant Optical Flow Divergence Landing," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 508–516, 2018, doi: 10.1109/TRO.2018.2817418.

[24] Y. Zhou, H. W. Ho, and Q. Chu, "Extended incremental nonlinear dynamic inversion for optical flow control of micro air vehicles," *Aerosp. Sci. Technol.*, vol. 116, p. 106889, 2021, doi: 10.1016/j.ast.2021.106889.

[25] Asadi, D., and Ahmadi, K. "Nonlinear Robust Adaptive Control of an Airplane with Structural Damage." Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, Vol. 234, No. 14, 2020, pp. 2076–2088.

[26] Asadi, D., Ahmadi, K., and Nabavi, S. Y. "Fault-Tolerant Trajectory Tracking Control of a Quadcopter in Presence of a Motor Fault." International J. of Aeronautical and Space Sciences, 2021. https://doi.org/10.1007/s42405-021-00412-9.

[27] Karim Ahmadi, Davood Asadi, Seyed-Yaser Nabavi-Chashmi, Onder Tutsoy, Modified adaptive discrete-time incremental nonlinear dynamic inversion control for quad-rotors in the presence of motor faults, Mechanical Systems and Signal Processing, Volume 188, 2023, doi.org/10.1016/j.ymssp.2022.109989.