

Autonomous landing of rotary-wing aerial vehicles by image-based visual servoing in GPS-denied environments

Rafik Mebarki, Vincenzo Lippiello, and Bruno Siciliano

Abstract—In this paper we propose a new control method for quadrotor autonomous landing on a visual target without linear velocity measurements. Only onboard sensing is exploited, such that only the images of the landing pad from a down-looking camera, along with data from an Inertial Measurement Unit's gyro, are used. The control system consists of an image-based nonlinear observer that estimates online the linear velocity of the vehicle and a backstepping image-based controller that generates attitude, and thrust setpoints to the quadrotor autopilot. Both observer and controller share the same feedback information: spherical visual features. Therefore no further image elaboration is needed for the estimation. This, along with the fact that only simple computations on low- and constant-dimension arrays are involved, makes the proposed solution computationally cheap. Real-hardware experiments on a quadrotor are carried out to verify the validity of the proposed control system.

I. INTRODUCTION

Being able to reach remote regions in an unconstrained fashion, without the need of roads, and in relatively short deployment time, Unmanned Aerial Vehicles (UAVs) are receiving an increasing interest from the research community. This period represents the dawn of UAVs in industry, such that we are witnessing a continual raise of the number of companies investing in the field. Particularly, Vertical Take-off and Landing (VTOL) UAVs feature the hovering capability, which yields them appealing if not the unique candidates for numerous potential applications.

After deployment in the air, a UAV needs likewise to be capable of landing autonomously [1], for different purposes such as maintenance and especially battery recharge. Under GPS coverage, the transmitted 3D position can be used to automatically control the motions of the UAV towards the landing pad, as presented in [2]. However, GPS relies on an external source (satellite) of information and, thus, is not passive, is unreliable at low altitudes, suffers from signal cut, and does not work properly in urban and cluttered areas. On the other hand, vision is passive, provides rich information with pixel order resolution, involves light-weight, and low-cost hardware requiring furthermore light computational resources. Modern technology furthermore affords increasing

The research leading to these results has been supported by ARCAS and SHERPA collaborative projects, which have received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements ICT-287617 and ICT-600958, respectively. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

The authors are with PRISMA Lab, Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Naples, Italy, {rafik.mebarki, vincenzo.lippiello, and bruno.siciliano}@unina.it

vision streaming rates. In addition, since it is unlikely that the pad (exact) location is a priori known, and such a location can furthermore change in time in case the pad is mobile, vision becomes necessary to guide the vehicle towards it and with enhanced precision in an autonomous fashion.

The issue of autonomous landing based on vision thus triggered a number of research works. In [2], a visual controller for helicopter autonomous landing is presented. GPS is used to provide the vehicle position and velocity as control feedback, while vision is used to locate the landing location. In [3], an image-based visual-servoing technique is proposed for autonomous landing of fixed-wing UAVs. The linear velocity of the craft needs, similarly, to be known. In [4], elevated horizontal sites (providing vantage points) for landing are detected by applying homography on the images from a down-looking onboard camera. The vehicle is then guided to the final position by fusing visual SLAM (VSLAM), sonar altimeter, and IMU. Similarly, in [5], landing sites are detected after terrain reconstruction by fusing images from a monocular camera and IMU data in a VSLAM framework. Also based on VSLAM, in [6] the helipad is detected and the UAV is servoed for landing. A prior knowledge on the pad size is used to infer the relative elevation from it. In [7] and [8], safe terrain for autonomous landing is detected from monocular onboard camera. In [9], a visual pad marker is designed for autonomous landing. A combination of onboard vision, IMU, optical flow, and sonar is employed. Work [10] presents vision-based autonomous landing on a coded tag. In [11], a nonlinear controller is fed with optical-flow from a down-looking camera. A survey of UAVs autonomous landing is reported in [12].

The work we present in this paper is close to [2], except that we do not rely on any external source, as GPS, but exploit instead only onboard sensing. In the control system we propose, the same image of the pad from the onboard camera to be used for visual servoing is also used to estimate online the linear velocity of the vehicle. The proposed solution does not require that the environment be textured enough, as do those based on optical flow. It only needs that the landing pad be described by a set of visual features. In this work, only four visual features have been used (see Fig. 1). They consist in landmarks affixed to the landing pad in a form of a rectangle. No prior knowledge on the target geometry or the landmarks is leveraged. The proposed method is thus general in the sense that the pad does not necessitate to be engineered, since only few visual features, like corners, describing the target will be sufficient.

The remainder of the paper is organised as follows. The

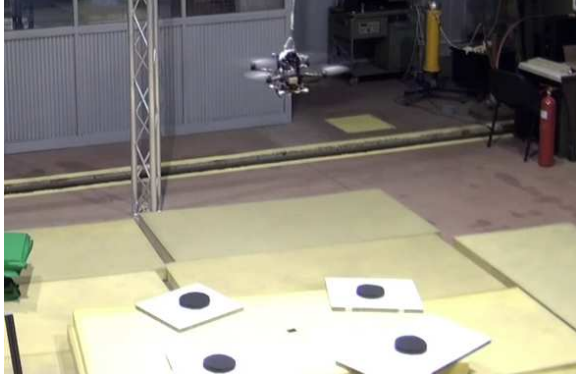


Fig. 1. Picture of the Pelican quadrotor employed in the reported experiments. The landing pad (target) is also shown and is characterised by four roughly-circular black landmarks. No prior knowledge on the pad geometry or on the landmarks is used. A down-looking video camera is attached to the bottom of the vehicle for visual servoing.

employed spherical visual feature is briefly introduced in Sect. II. Section III presents the velocity measurements-free control system to endow the UAV with the capability of reaching and hovering centered on the pad, while Section IV shows how to exploit this for autonomous landing. Section V reports experimental validation on an ASCTEC Pelican quadrotor, before concluding in Sect. VI.

II. VISUAL INFORMATION

Due to the fact that a quadrotor is an under-actuated system, care needs to be taken when selecting a feedback visual feature. In [13] it has been shown that spherical visual features are amenable from a control design stand point. Such features indeed preserve the passivity property and, thus, allow designing a motion controller independently of the angular velocity.

Let the landing pad be characterised by a set of n points, such that P_i denotes the i -th point (see Fig. 2). In general real environments, such points could correspond to visual corners or markers as displayed in Figs. 1 and 3(a) for instance. Let $\{R_s\}$ be a Cartesian frame attached to the onboard camera (see Fig. 2). Let $s_i \in \mathbb{R}^3$ correspond to the spherical projection of P_i on the normal sphere centered on $\{R_s\}$. The unnormalised spherical centroid of the scene can then be expressed as follows:

$$q = \sum_{i=1}^n s_i, \quad (1)$$

such that considering ${}^sP_i = (X_i \ Y_i \ Z_i)^\top \in \mathbb{R}^3$ as the 3D coordinates of P_i in camera frame $\{R_s\}$, s_i can be expressed as $s_i = {}^sP_i / |{}^sP_i|$, where $|\cdot|$ denotes the Euclidean norm. Likewise, by considering $p_i = (x_i \ y_i)^\top \in \mathbb{R}^2$ as the perspective projection (see Fig. 2) of P_i , the spherical feature s_i can also be solely expressed in terms of p_i as $s_i = \frac{\bar{p}_i}{|\bar{p}_i|}$ such that $\bar{p}_i = (x_i \ y_i \ 1)^\top = (p_i^\top \ 1)^\top \in \mathbb{R}^3$.

The spherical visual feature proposed in [13] writes as follows:

$$\delta = q - q^*, \quad (2)$$

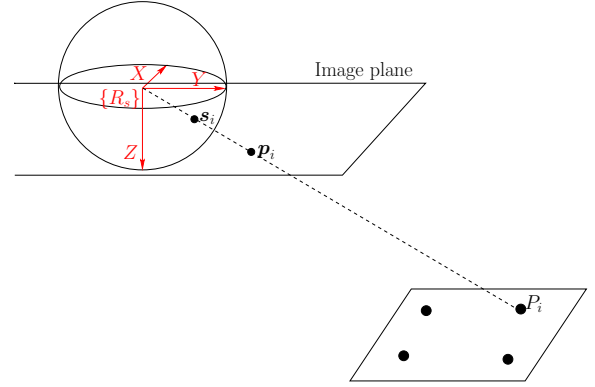


Fig. 2. Spherical s_i and perspective p_i image coordinates of 3D point P_i .

where q and q^* represent the actual and desired spherical centroid, respectively; q is computed according to (1) while q^* is set to $\mathbf{R}^\top b$, such that $b \in \mathbb{R}^3$ is a known desired direction of the visual feature in inertial frame $\{R_i\}$. Rotation matrix $\mathbf{R}$ describes the orientation of $\{R_s\}$ with respect to $\{R_i\}$. In order to improve the conditioning induced by the above feedback visual feature, another version is proposed in [14] as follows:

$$\delta_1 = q_0^* \times q + \lambda q_0^* q_0^{*\top} \delta, \quad (3)$$

where $q_0^* = q^* / |q^*| \in \mathbb{R}^3$ corresponds to a normalised version of q^* , and $\lambda \in \mathbb{R}$ is a positive constant to be suitably tuned.

We propose to exploit visual error δ_1 as the sole feedback for the observer presented in the sequel.

III. CONTROL SYSTEM

This section presents a nonlinear controller endowing a quadrotor with the capability of autonomously reaching and stabilising with respect to a visual target. Only onboard vision along with IMU rotational data are employed by the control system. The linear (translational) velocity of the vehicle is estimated online with a nonlinear observer. The stability of both the observer and the closed-loop system has been proved with Lyapunov theory in [15]. In the sequel, we first briefly introduce the proposed control system. Then, we show how it will be used for landing.

Let v be the linear velocity of the vehicle with respect to a fixed frame and expressed in the camera frame. We propose in [15] the following nonlinear observer to estimate v :

$$\begin{cases} \dot{\hat{v}} &= \hat{z} - k_{2o} \mathbf{A}^{-1} \tilde{\delta}_{1o} \\ \dot{\hat{z}} &= -\omega \times \hat{v} + \frac{1}{m} \mathbf{f} - k_{2o} \mathbf{G} \tilde{\delta}_{1o} - \xi \\ \tilde{\delta}_{1o} &= \delta_1 - \hat{\delta}_1 \\ \dot{\hat{\delta}}_1 &= -\omega \times \delta_1 - \mathbf{A} \mathbf{Q} \hat{v} + k_{1o} \tilde{\delta}_{1o} \\ \mathbf{G} &= k_{1o} \mathbf{A}^{-1} - \frac{d}{dt} \mathbf{A}^{-1} \\ \xi &= \mathbf{Q} \mathbf{A}^\top \tilde{\delta}_{1o}, \end{cases} \quad (4)$$

where $\hat{v}$ is the estimate of v and where the involved matrices write:

$$\begin{cases} \mathbf{Q} = \sum_{i=1}^n \frac{1}{d_i} \bar{\mathbf{Q}}_i \in \mathbb{R}^{3 \times 3} & \text{and } \bar{\mathbf{Q}}_i = \mathbf{I}_3 - \mathbf{s}_i \mathbf{s}_i^\top \\ \mathbf{A} = [\mathbf{q}_0^*]_\times + \lambda \mathbf{q}_0^* \mathbf{q}_0^{*\top} \in \mathbb{R}^{3 \times 3} \\ \dot{\mathbf{A}} = \mathbf{A}[\boldsymbol{\omega}]_\times - [\boldsymbol{\omega}]_\times \mathbf{A} \end{cases} \quad (5)$$

with k_{1o} and $k_{2o} \in \mathbb{R}$ positive tuning gains. As for $\boldsymbol{\omega}$, it represents the angular velocity of the vehicle expressed in the camera frame, $[\cdot]_\times$ denotes the skew-symmetric matrix, while d_i is the image depth of point P_i . Vector $\mathbf{f} \in \mathbb{R}^3$ corresponds to the total linear forces acting on the vehicle and can be approximated as follows:

$$\mathbf{f} = -u \mathbf{e}_3 + m g \mathbf{R}^\top \mathbf{e}_3, \quad (6)$$

where u is the value of the applied thrust, $\mathbf{e}_3 = (0 \ 0 \ 1)^\top$, m is the total mass of the UAV, and g is the gravity acceleration.

The observer defined above is exponentially stable. Indeed, considering the following Lyapunov storage function $V_o = \frac{1}{2} |\tilde{\boldsymbol{\delta}}_{1o}|^2 + \frac{1}{2} |\tilde{\mathbf{v}}_o|^2$, where $\tilde{\mathbf{v}}_o = \mathbf{v} - \hat{\mathbf{v}}$ is the estimation error on the velocity, it has been shown in [15] that time variation $\dot{V}_o$ of V_o satisfies $\dot{V}_o \leq -\alpha V_o$ (with $\alpha > 0$), thus the exponential stability. Although the image depth is involved as a parameter above, the control system has shown to be robust thereof. Only a rough approximation is enough, such that the obtained experimental results show that the system is robust to errors up for instance to 200%.

Next, we use the obtained estimate $\hat{v}$ to achieve visual control without velocity measurements. We propose the following integral-backstepping image-based control law [15]:

$$\begin{cases} \frac{1}{m} \mathbf{f} := \boldsymbol{\xi} + c_1 \mathbf{A}^\top \left(c_1 c_2 \boldsymbol{\delta}_2 + \alpha_2 \boldsymbol{\sigma}_2 \right) \\ \begin{cases} \boldsymbol{\delta}_2 := \boldsymbol{\delta}_1 - \frac{1}{c_1} \mathbf{A}^{-\top} \hat{\mathbf{v}} \in \mathbb{R}^3 \\ \boldsymbol{\sigma}_2 = \int \boldsymbol{\delta}_2 dt, \end{cases} \end{cases} \quad (7)$$

where $\boldsymbol{\delta}_2$ is a second backstepping error and $\boldsymbol{\sigma}_2$ is its integral. The integral has been injected to account for the eventuality of the presence of modeling errors in the system. Indeed, we noticed a considerable error between the commanded (input) and achieved thrust in the quadrotor used in the reported experiments. Without $\boldsymbol{\sigma}_2$ the system falters and destabilises even when using the precise velocity measurements from a motion capture system.

The proposed control law relates the desired roll/pitch attitude and thrust the UAV needs to achieve. Indeed, from a desired value (f_x^*, f_z^*, f_z^*) of $\mathbf{f}$, the desired roll ϕ (around X-axis), pitch θ (around Y-axis), and thrust u of the vehicle

can be extracted as follows [15]:

$$\begin{cases} \theta = \arcsin\left(-\frac{1}{m g} f_x^*\right) \\ \phi = \arcsin\left(\frac{1}{m g c_\theta} f_y^*\right) \\ u = m g c_\theta c_\phi - f_z^*, \end{cases} \quad (8)$$

where c_x denotes $\cos(x)$.

Control law (7) ensures that $\boldsymbol{\delta}_1$ converges to zero without velocity measurements and is globally asymptotically stable, provided that [15]:

$$\begin{cases} k_{1o} > 0 \\ c_1 > \frac{1}{2} \max\left(\frac{1}{^H \lambda_{\min}} \|\mathbf{A}\|_F, \frac{1}{^Q \lambda_{\min}} \|\mathbf{A}^{-\top}\|_F\right) \|\mathbf{Q}\|_F \\ k_{2o} > \frac{\|\mathbf{A}\|_F \|\mathbf{Q}\|_F}{^Q \lambda_{\min} - \frac{1}{2 c_1} \|\mathbf{A}^{-\top}\|_F \|\mathbf{Q}\|_F} \\ c_2 > ^H \lambda_{\max} + \frac{2}{c_1} \left(\|\mathbf{A}\|_F + \frac{k_{2o}}{c_1} \|\mathbf{A}^{-\top}\|_F \right) \|\mathbf{Q}\|_F, \end{cases} \quad (9)$$

where $^H \lambda_{\min}$ and $^H \lambda_{\max}$ are respectively bounds of the minimum and maximum eigenvalues of $\mathbf{H} = \mathbf{A} \mathbf{Q} \mathbf{A}^\top$. Notice that $\mathbf{H}$ inherits positive definiteness from $\mathbf{Q}$. As for $\|\cdot\|_F$, it denotes the Frobenius norm. It can be demonstrated that the time variation of the following Lyapunov storage function is definite negative [15]:

$$V = \frac{1}{2} |\boldsymbol{\delta}_1|^2 + \frac{1}{2} |\boldsymbol{\delta}_2|^2 + \frac{1}{2} \alpha_2 |\boldsymbol{\sigma}_2|^2 + \frac{1}{2} |\tilde{\boldsymbol{\delta}}_{1o}|^2 + \frac{1}{2} |\tilde{\mathbf{v}}_o|^2. \quad (10)$$

IV. AUTONOMOUS LANDING

We show in the present section how to exploit the above-presented control law (7) for autonomous landing. Let δ_z be the third component of $\boldsymbol{\delta}_1$, introduced by (3). This component is in fact correlated to the distance from the observed object to the camera frame center. The higher the distance the larger δ_z and the inverse is true. Since in our configuration the camera is oriented downward, δ_z is proportional to the relative altitude of the vehicle from the target. We exploit this fact to achieve autonomous landing.

Assume the vehicle-attached down-looking camera observes a target characterised by a set of visual points (landmarks for instance), similarly as considered in Section II. The robotic task is that the vehicle autonomously lands on this target without any external aid. In addition, no prior knowledge on the target geometry or on the landmarks is used. Our method uses only the image of the target along with the information from the IMU's gyro. We thus propose to first align the vehicle with the target center of gravity. This can be achieved by setting the first two components of $\mathbf{b}$ introduced just after (2) to null value. Once the vehicle is hovering on the target, the last component is

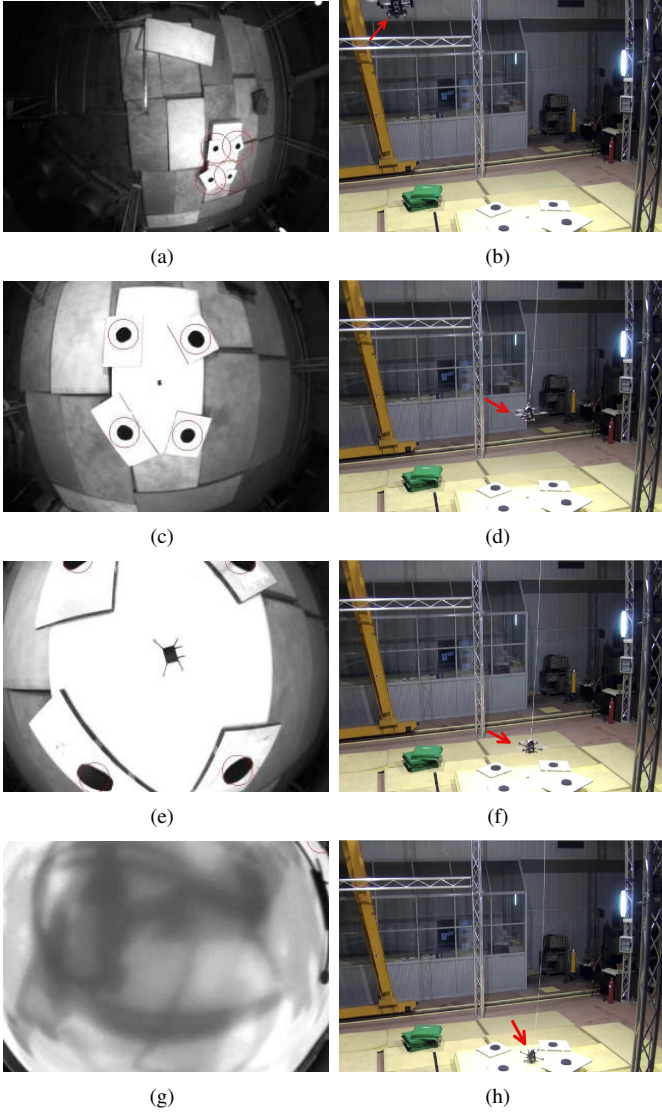


Fig. 3. Experimental result with a quadrotor autonomously reaching, stabilising, and finally landing on a visual target without linear velocity measurements. The target, that can be seen in images (a) and (b), is characterised by the four black dots (landmarks) surrounded with red circles. The circles are generated and displayed by the image processing. No prior knowledge on the target geometry nor on the landmarks is used by the control scheme. (a) Initial image of the target captured by the onboard camera just before visual servoing is launched. (b) Image taken at the same instant but from an offboard camera (used only for recording). The vehicle is then at a relative position of (2.1, 1.8, 3.1) m from the target. For the sake of clarity the UAV is indicated with the red arrow. (c)-(d) Images taken after the UAV achieved the first phase of the landing: hovering centered on the target. (e)-(f) The UAV is finally at less than 30 cm above the target after starting with an initial error of 3 m. Just after that, visual servoing is automatically turned off and the thrust is decreased until the vehicle touches the ground.

then continuously decreased until the vehicle is one step from landing.

Let $\mathbf{b}_c = (0 \ 0 \ b_{cz})^\top \in \mathbb{R}^3$ be the first desired direction of the visual features so that the vehicle would hover and get centered on the target. The corresponding visual error is denoted δ_c , as given by (2). Once the norm $|\delta_c|$ of δ_c is under a certain threshold, which would mean that the UAV is

alighted enough with respect to the target, a new direction is introduced for landing. Let it be $\mathbf{b}_l = (0 \ 0 \ b_{lz})^\top \in \mathbb{R}^3$. In contrast to $\mathbf{b}_c$, $\mathbf{b}_l$ is varying. Its third component b_{lz} decreases from an initial value equal to b_{cz} , so as to yield autonomous landing on the target center of gravity. This phase corresponds to a roughly-vertical descent of the vehicle down to the target. To summarise

$$\mathbf{b} = \begin{cases} (0 \ 0 \ b_{cz})^\top & \text{if } |\delta_1| > \epsilon_\delta \\ (0 \ 0 \ b_{lz,t})^\top & \text{otherwise,} \end{cases} \quad (11)$$

such that $b_{lz,t} = b_{lz,t-1} - |\Delta_b|$, with t being the time variable. Thus, in the landing phase the third component is decremented by $|\Delta_b|$ for each time step until the vehicle is close enough to the target. Threshold ϵ_δ is set as a switch between the two described phases: get the vehicle hovering and centered on the target, and roughly-vertical final landing motion.

V. EXPERIMENTAL RESULTS

The proposed control scheme is tested for autonomous landing of a real flying quadrotor. The latter consists in an ASCTEC Pelican of roughly 1.5 Kg total mass and is endowed with an MIO-2261N-S8A1E motherboard featuring a 1.86 GHz with 4 GB dual core processor. As for the sensor suite, it consists of a low-cost IMU steaming at 100 Hz, and a UI-1221LE uEYE camera equipped with fisheye lenses and providing images with 640×480 pixel resolution at the cadence of 70 Hz. The proposed method is implemented in C++ programming language on the motherboard onboard the UAV, while the communication between the onboard modules is managed with ROS middleware software¹. Ground truth is provided by an Optitrack² motion capture system, which consists of twenty S250e infrared cameras covering the vehicle operational space.

Before visual servoing is launched, the vehicle is initially teleoperated to a place far from the target, so as to verify the capability of the system in real conditions. This step is realised by using the measurements of the vehicle 3D positions provided by Optitrack in a Cartesian controller. Figures 3(a) and 3(b) show respectively an image frame from the onboard camera, and a snapshot image from an offboard camera, when the vehicle reached that initial location. Next, visual servoing is launched and the UAV is in a fully-autonomous mode, where only the onboard images along with IMU's gyro data feed the control system. Control laws (4), (7), and (11) solely control the quadrotor, by generating desired roll/pitch attitude angles and thrust to the off-the-shelf autopilot. The observer and control gains are manually and empirically set as follows: $c_1 = 0.1$, $c_2 = 9$, $\lambda = 2.5$, $k_{1o} = 10$, $\mathbf{K}_{2o} = 6e^{-2} \mathbf{Q}^{*-1}$, and $\alpha_2 = 0.4$, where $\mathbf{Q}^*$ corresponds to the value of $\mathbf{Q}$ at the desired configuration (i.e. location centered on the target just before the descent manoeuvre). In general conditions, the value of $\mathbf{Q}^*$ can be a priori known.

¹<http://www.ros.org>

²<http://www.naturalpoint.com/optitrack>

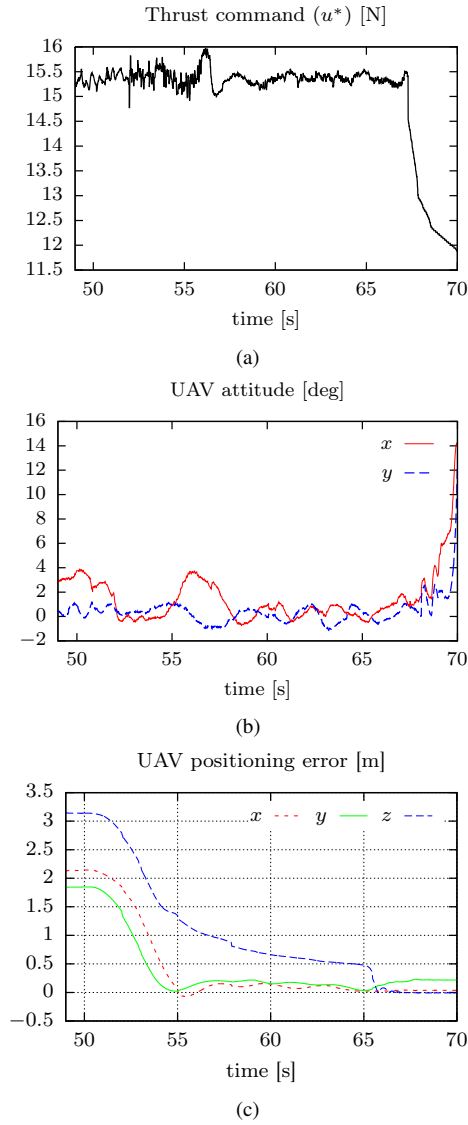


Fig. 4. Plots corresponding to Fig. 3. Figure 4(c) shows the relative position of the onboard camera with respect to the target; more precisely, with respect to a marker roughly centered on the latter. (b) 'x' and 'y' stand for roll and pitch angles, respectively.

Corresponding experimental results are shown in Figs. 3, 4, and 5. The eight figures of Fig. 3 illustrate the different phases for landing. The initial location from which the UAV starts autonomous landing is shown with both the corresponding image frame from the onboard camera [Fig. 3(a)] and a snapshot from an offboard recording camera [Fig. 3(b)]. Figures 3(c) and 3(d) show images taken when the UAV reached the hover configuration roughly-centered on the target, just before the vertical descent is undertaken. Then, the UAV starts roughly-vertical descent motion to end up centered only 50 cm above the target, as displayed by Figs. 3(e) and 3(f). Finally, visual servoing is automatically switched off and the vehicle loses elevation until it touches the ground as can be seen in Figs. 3(g) and 3(h). The evolution of the relative 3D position of the vehicle with respect to the target's during the whole autonomous

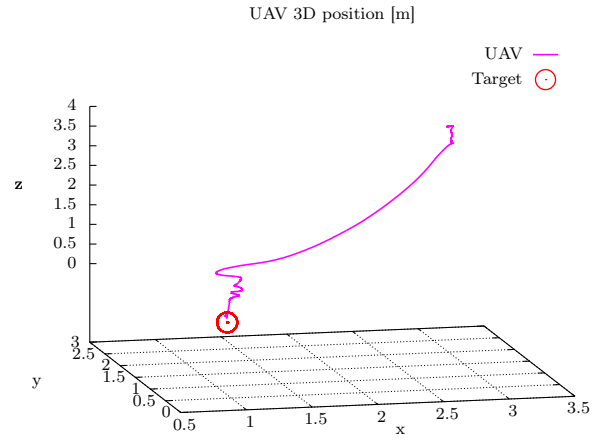


Fig. 5. 3D path of the UAV (of the onboard camera, more precisely) to autonomously land on the target, whose center is also indicated.

manoeuvre is reported in Fig. 4(c). As planned, the vehicle first gets centered with respect to the target, such that we can see indeed that the values of the first two coordinates (X and Y axes, that represent the horizontal plane) of its position converge to those of the target. This phase starts at time 51 s and ends at time 55 s, wherein the value of the desired direction is given by the first relationship of (11). Note that a marker is affixed roughly-centered on the landing pad, as can be seen in Fig. 3(e). The marker position is sensed with Optitrack motion capture system and represents the reference to induce the positioning error plotted in Fig. 4(c). Afterwards, the UAV enters the second phase as it starts the descent manoeuvre to approach and land on the target, such that the control system automatically switches to the second relationship of (11) to update the value of desired direction *b*. We can see indeed that while the UAV keeps centered on the landing pad, the relative altitude (Z axis) decreases to finally hover at only less than 50 cm above the target. This phase corresponds to time [55 - 65] s. Finally, visual servoing is switched off and the thrust is decreased until the vehicle touches the ground. The motion of the vehicle in the 3D space is depicted in Fig. 5. The value of the thrust command is plotted in Fig. 4(a) and the roll, and pitch angles of the vehicle attitude are plotted in Fig. 4(b). Note that the 3D motion of the vehicle is smooth, though the occurrence of the above phase transitions. These results thus provide an experimental validation and proof-of-concept of the proposed approach based on onboard vision for autonomous landing. Though we omitted explicitly measuring the success rate of the proposed approach in the real hardware experiments, we can nevertheless report that we did not record any failure of the control system after all the different involved gains have been tuned.

As already stated, the final phase consisting in the vertical descent [just after the vehicle had reached the position captured in Fig. 3(f)] is realised in an open-loop fashion, such that the thrust is gradually decreased until the vehicle touches the ground. This phase corresponds to a vehicle motion of about 50 cm, from the location captured in Fig. 3(f) to that

captured in Fig. 3(h). As soon as the UAV is close enough to the target, the latter no longer fits the camera field of view, which subsequently leads to occlusion. That is the reason why the visual servoing is switched off. Nevertheless, as alternatives to the open-loop final manoeuvre adopted in this work, we propose the following solutions:

- i) A fisheye camera with large field of view or a (ultra) wide angle camera would address the problem.
- ii) If the above proposition cannot be met, then we would propose to close the remaining dozens of centimeters gap in an open-loop control fashion. Indeed, since the distance to be achieved is short, and in addition is known to be vertical, one could impose a vertical descending and slow motion to the vehicle. The control objective would be to regulate the vehicle linear velocity. The measures of the latter to be used for feedback could be obtained by integrating the onboard accelerometer data, which would be reliable enough for so very short displacement.
- iii) Finally, sonar or laser sensors could be employed to guide the vehicle through the remaining short vertical distance.

VI. CONCLUSIONS

In this work we have proposed and successfully validated through real-hardware experiments a control system for precise autonomous landing of a quadrotor UAV on a visual target in a GPS-denied environment. Only onboard sensing has been employed: a single down-looking camera and IMUs gyro. It is worth remarking that we did not use the accelerometer. The control system only needs that the landing pad be characterised by a set of visual features, which is quite feasible in real world scenarios. It does not require that the environment be highly textured. The control system consists of a nonlinear observer estimating the linear velocity of the vehicle and a backstepping nonlinear control law to visually and automatically position the UAV with respect to the pad, both of which use the same spherical visual feature extracted from the pad image. It generates desired attitude and thrust to the UAV autopilot.

The future works would be on autonomously landing on a moving pad, and testing in the presence of external wind perturbations.

REFERENCES

- [1] L. Marconi and A. Isidori, "Robust output regulation for autonomous vertical landing," in *IEEE Int. Conf. on Decision and Control*, pp. 3590 – 3595, 2000.
- [2] S. Saripalli, J. Montgomery, and G. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 371 – 380, 2003.
- [3] O. Bourquard and F. Chaumette, "Visual servoing of an airplane for auto-landing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1314 – 1319, October 2007.
- [4] R. Brockers, S. Susca, D. Zhu, and L. Matthies, "Fully self-contained vision-aided navigation and landing of a micro air vehicle independent from external sensor inputs," in *Proc. SPIE 8387, Unmanned Systems Technology XIV*, 83870Q, May 2012.

- [5] F. Christian, F. Matthias, F. Flavio, W. Manuel, and S. Davide, "Continuous on-board monocular-visionbased elevation mapping applied to autonomous landing of micro aerial vehicles," in *IEEE Int. Conf. on Robotics and Automation*, May 2015.
- [6] D. Sergiu, G. Florian, and A. Elli, "Autonomous approach and landing for a low-cost quadrotor using monocular cameras," in *Computer Vision - ECCV 2014 Workshops - Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I*, pp. 209–222, 2014.
- [7] A. Johnson, J. Montgomery, and L. Matthies, "Vision guided landing of an autonomous helicopter in hazardous terrain," in *IEEE Int. Conf. on Robotics and Automation*, pp. 3966 – 3971, May 2005.
- [8] V. Desaraju, N. Michael, M. Humenberger, roland Brockers, S. Weiss, and L. Matthies, "Vision-based landing site evaluation and trajectory generation toward rooftop landing," in *Proceedings of Robotics: Science and Systems*, (Berkeley, USA), July 2014.
- [9] S. Lange, N. Sunderhauf, and P. Protzel, "A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments," in *IEEE Int. Conference on Advanced Robotics*, pp. 1–6, 2009.
- [10] Y. Shuo, Y. Jiahang, Y. Lu, and L. Zexiang, "Precise quadrotor autonomous landing with SRUKF vision perception," in *IEEE Int. Conf. on Robotics and Automation*, pp. 2196– 2201, May 2015.
- [11] B. Herisse, F.-X. Rusotto, T. Hamel, and R. Mahony, "Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008.
- [12] A. Gautam, P. Sujit, and S. Saripalli, "A survey of autonomous landing techniques for UAVs," in *Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, pp. 1210 – 1218, May 2014.
- [13] T. Hamel and R. Mahony, "Visual servoing of an under-actuated rigid body system: An image based approach," *IEEE Trans. on Robotics and Automation*, vol. 18, pp. 187–198, 2002.
- [14] N. Guenard, T. Hamel, and R. E. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Trans. on Robotics*, vol. 24, pp. 331–340, 2008.
- [15] R. Mebarki, V. Lippiello, and B. Siciliano, "Nonlinear visual control of unmanned aerial vehicles in GPS-denied environments," *IEEE Trans. on Robotics*, vol. 31, no. 4, pp. 1004–1017, 2015.