Robotica (2018) volume 36, pp. 1493–1509. © Cambridge University Press 2018
doi:10.1017/S0263574718000516

Sliding mode collision-free navigation for quadrotors using monocular vision

Diego Mercado†*, Pedro Castillo‡ and Rogelio Lozano‡§

†*Department of Mechanical and Aerospace Engineering, Rutgers, The State University of New Jersey, 98 Brett Road, Piscataway, NJ 08854-8058, USA*
‡*Sorbonne Universitès, Université de Technologie de Compiègne, CNRS, Heudiasyc UMR 7253, CS 60 319, 60 203 Compiègne cedex, France. E-mail: pedro.castillo@hds.utc.fr*
§*LAFMIA UMI 3175 CINVESTAV-CNRS, Avenida Instituto Politècnico Nacional 2508, San Pedro Zacatenco, 07360 Mexico City, CDMX, Mexico. E-mail: Rogelio.Lozano@hds.utc.fr*

(Accepted May 26, 2018. First published online: June 20, 2018)

SUMMARY
Safe and accurate navigation for autonomous trajectory tracking of quadrotors using monocular vision is addressed in this paper. A second order Sliding Mode (2-SM) control algorithm is used to track desired trajectories, providing robustness against model uncertainties and external perturbations. The time-scale separation of the translational and rotational dynamics allows to design position controllers by giving a desired reference in roll and pitch angles, which is suitable for practical validation in quadrotors equipped with an internal attitude controller. A Lyapunov based analysis proved the closed-loop stability of the system despite the presence of unknown external perturbations. Monocular vision fused with inertial measurements are used to estimate the vehicle's pose with respect to unstructured scenes. In addition, the distance to potential collisions is detected and computed using the sparse depth map coming also from the vision algorithm. The proposed strategy is successfully tested in real-time experiments, using a low-cost commercial quadrotor.

KEYWORDS: UAVs; Quadrotors; Autonomous navigation; Sliding mode; Collision avoidance; Computer vision.

1. Introduction

Applications involving flying robots, also known as Unmanned Aerial Vehicles (UAVs), have astonishingly spread out all over the world, in a wide range of tasks where they are used mainly for remote sensing. In particular, four rotor rotorcrafts have attracted special attention from both research groups and industry, thanks to their simplified mechanics and control, and their great maneuverability which allows for vertical take-off and landing, hover and aggressive flight in small spaces. Such applications include, among others, remote monitoring, surveillance, patrolling, search and rescue, transportation and film recording.

In order to successfully accomplish their mission in an autonomous or semi-autonomous operation, besides the control algorithm, all UAVs require a good estimation of their state vector, with high fidelity and fast rate. Moreover, aerial robots normally present extra challenges steaming from their limited size and payload, constraining their sensing capabilities. Even more, small UAVs are intended to be inexpensive, and the use of high-precision sensing equipment is precluded. Primarily, the position, orientation, translational speed and angular rate are needed; hence, they are equipped with a set of sensors normally including an Inertial Measurement Unit (IMU), an optic flow sensor, ultrasound range finders and cameras. A Global Positioning System (GPS) is often used for localization, however, in many scenarios GPS measurements are unavailable, or corrupted due to jamming or environmental constrains.

* Corresponding author. E-mail: die.ravell88@gmail.com

In fact, localization in GPS-denied environment continues to be an open problem of great interest in the field. Several recent works about this subject can be found in the literature. For example, in ref. [1] the autonomous control problem of a quadrotor in GPS-denied environment is addressed using a miniature laser range finder as the main onboard sensor.

Monocular vision appears as a good option for UAV's perception of their environment, since cameras offer huge amount of information in a compact, lightweight and inexpensive device, at the cost of computational effort. For instance, in ref. [2] a vision-based navigation strategy for UAVs using a single embedded camera observing natural landmarks is presented. Also, in ref. [3] the authors proposed a vision-based algorithm to track and land on a known moving target. A vision-based anticipatory controller for the autonomous indoor navigation of an UAV is addressed in ref. [4], where a dual feedforward/feedback architecture was used as the UAV's controller and the K-NN classifier using the gray level image histogram as discriminant variables was applied for landmark recognition. Additionally, in ref. [5] optic flow information from a camera is fused with inertial measurements using an Extended Kalman Filter (EKF) to estimate metric speed, distance to the scene and also acceleration biases.

More in particular, in the last years, some important works on pose estimation using monocular vision have appeared in the literature. They make use of a powerful technique conceived for Augmented Reality (AR) applications, where a hand-held camera is tracked through a small unstructured workspace. This technique, which is called Parallel Tracking and Mapping (PTAM), is based on the generation of a sparse map composed of thousands of points from the special features on the scene, which can be tracked at frame-rate with high accuracy.[6] In addition, PTAM has been successfully adapted for UAV's localization, by extending the generated map and solving the absolute scale estimation problem using extra information from the quadrotor's embedded sensors. In refs. [7–10], the authors introduce implementations of an onboard vision-based UAV controller for navigation on unknown environments without any external assistance. The drawback of these works is that they consider expensive aerial prototypes to get a fully embedded system.

Other teams, contrary to the above cited works, have obtained similar results using only a low-cost commercial UAV coupled with a ground-station to compute externally the algorithms.[11,12] In these research works the system is composed of the PTAM algorithm, an EKF and a linear control. The contribution was to find a solution to the estimation problem for the absolute scale of the generated sparse map, by fusing the visual information with inertial and/or altitude measurements. Furthermore, they provide their complete implementation as open-source code.

Once the vehicle's pose problem is solved, other key components for the success of autonomous navigation of aerial vehicles are the obstacle detection and avoidance. Several teams have explored this matter using different approaches and in several cases adding more sensors, for example, in ref. [13] the authors proposed a complete system with a multimodal sensor setup for omnidirectional obstacle perception consisting of a 3D laser scanner, two stereo camera pairs and ultrasonic distance sensors. Detected obstacles were aggregated in egocentric local multiresolution grid maps. Planning of collision-free trajectories and reactive obstacle avoidance were tested in real-time experiments. However, this approach requires a huge payload for the multiple sensors involved.

Monocular cameras arise again as an excellent alternative to detect and avoid obstacles for small UAVs. First works were based on optical flow algorithms[14] or perspective cues,[15] however, these methods cannot handle frontal obstacles well. In ref. [16], the problem of detecting frontal obstacles for an UAV is examined through a method to detect relative size changes in image patches, with highly confident results in experiments. Similarly, in ref. [17] the authors presented a strategy for a quadrotor with a monocular camera to locally generate collision-free waypoints. In that paper, PTAM is used for navigation, then a dense depth map is computed from a small set of images, finally a 2D scan is rendered and suitable waypoints are obtained, but extensive extra computations are required.

In this work, the vehicle's localization and obstacle detection and avoidance problems are solved using visual information coming only from a camera. The sparse depth map obtained from the special features on the image and provided by the PTAM algorithm is modified and used to locate the aerial vehicle in unstructured scenes. Moreover, this algorithm is improved and extended to detect possible collisions in the horizontal plane. In addition, no extra sensors are needed in the prototype to perform this task because we consider that, for a wide range of applications, no extra computation is needed to generate a dense map for collision avoidance, despite the fact that the sparse map is noisy and weak for low-texture regions. The advantage of this algorithm is that it could be implemented in commercial aerial vehicles having a camera. In our case, a quadcopter Parrot AR.Drone is used.

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

Linear or nonlinear control algorithms can be used for autonomous navigation (the choice depends on the application, background of control knowledge, desired characteristics of the algorithm and implementation complexity in the embedded system, in case of real validation). Real-time applications involve a set of practical problems including modeling uncertainties, measurements noise and external disturbances mainly due to natural conditions, such as wind. It is then of great interest for the controller to acknowledge for these constrains and be robust against them. Robust control for UAVs have been subject of several studies, to cite some examples, in ref. [18] a robust global trajectory tracking is derived for underactuated UAVs, in a hierarchical scheme where the inner-loop is the attitude which is represented using quaternions in a hybrid system's framework, while the global position trajectory tracking is accomplished robustly in the outer-loop. The proposed scheme was tested experimentally in yaw with the help of a motion capture system. Moreover, in ref. [19] a robust trajectory tracking control for quadrotors is derived through a hierarchical scheme, where the position control is achieved by means of a feedback linearization plus a robust compensation technique, taking into account parametric perturbations, nonlinear and coupled dynamics, external disturbances and time delays. Experimental validation is provided using a precise GPS and an optic flow sensor. High order sliding mode techniques are well-known and popular, for their inherent robustness properties with decreasing of the undesired chattering effect typical in sliding mode strategies. Some applications of the sliding mode theory to UAVs can be found in the literature, for example, in ref. [20] the author presents a regulation controller for the Planar Take-Off and Landing (PVTOL) vehicle using feedback linearization in combination with a sliding mode, while in ref. [21] the author introduces an attitude controller for mini UAVs using a sliding mode controller and a fuzzy inference mechanism. In ref. [22] a sliding mode fault-tolerant control for an octorotor is proposed, taking advantage on the redundant rotors in this kind of configuration. Also, in ref. [23] a 2-SM is used to achieve attitude control, while position control is addressed using the sliding mode theory in ref. [24], both for quadrotors.

For autonomous navigation, a second order sliding mode controller was designed in this work, to follow desired trajectories in a robust way with respect to uncertainties and perturbations, making the closed-loop system suitable for outdoor applications. Such controller is inspired in our previous work,[25] and modified to improve the closed-loop system behavior and facilitate the parameter tuning. Furthermore, real-time experiments were carried-out to validate the control strategy.

This controller is capable of following all kinds of mostly translational maneuvers for trajectories in the $3D$ space. However, since only a frontal camera is used for perception, collision avoidance is only guaranteed for frontal obstacles or previously detected obstacles for lateral motions.

Our contribution can be summarized in the following points:

- A 3D trajectory tracking control algorithm based on high order sliding mode, robust against perturbations and uncertainties, is proposed and successfully validated in real-time experiments for quadrotors.
- Several tools were assembled, adapted and improved to offer solutions for aerial autonomous tracking. This includes extending the use of the PTAM algorithm to detect and avoid collisions, without extra expensive calculations or sensors.
- An effective strategy for obstacle avoidance in the horizontal plane was developed taking into account the visual information and without requiring expensive cost computation. No extra sensors are required for this task.

The rest of this paper is organized as follows: the monocular vision algorithm for pose estimation and the experimental platform are introduced in Section 2, the proposed control strategy for robust trajectory tracking with collision avoidance is described in Section 3. Main results from real-time experiments are presented in Section 4. Last but not the least, Section 5 states the conclusions and perspectives for future work.

2. Experimental Setup

The experimental setup is simple, which is an advantage of the proposed work, and composed by a quadcopter Parrot A.R. Drone and vision algorithms for localization and perception.

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

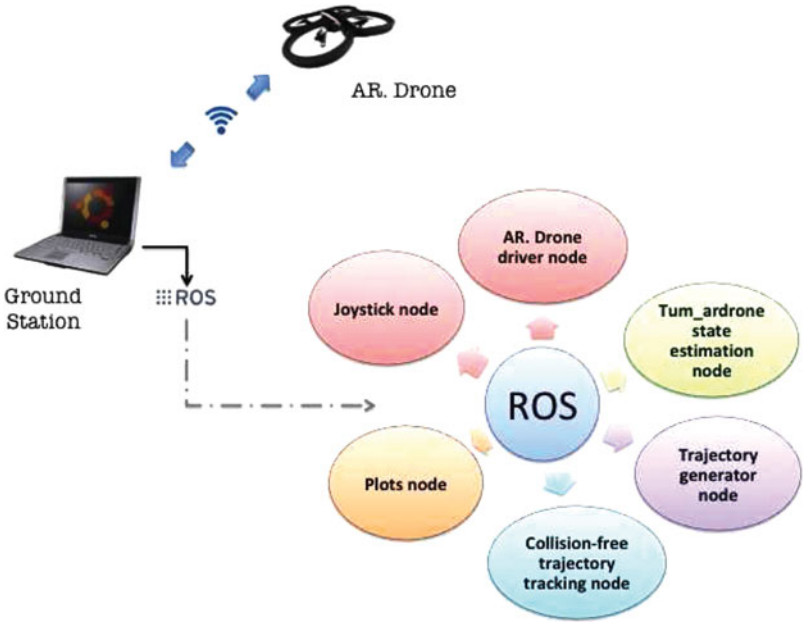

Fig. 1. Overview of the experimental setup and the ROS nodes. Significant changes were made in the Trajectory generator node, Collision-free node and State estimation node. This last node was customized to recover the pointcloud.

2.1. Prototype

The AR.Drone is a well-known and inexpensive commercial quadrotor which can be used safely close to people and is robust to crashes. It measures 53×52 cm^2 and weights 0.42 Kg. It is equipped with three-axis gyroscopes and accelerometers, an ultrasound altimeter, an air pressure sensor and a magnetic compass. The prototype has an inner algorithm for fusing data from the inertial sensors and estimating its attitude. Also, it provides video streams from two cameras, the first one is looking downwards with a resolution of 320×240 pixels at a rate of 60 fps, and is used to estimate the horizontal velocities with an optic flow algorithm. The second camera is looking forward, with a resolution of 640×360 at 30 fps and is in general employed for transmitting video when the vehicle is flying. No algorithm for this video stream is reported for this prototype. All sensor measurements can be sent to a ground station at a frequency of 200 Hz. In addition, it is equipped with an on-board autopilot to control roll, pitch, altitude velocity and yaw rotational speed (ϕ, θ, $\dot{z}$ and $\dot{\psi}$), according to external references. These references are considered as control inputs and computed and sent at a frequency of 100 Hz. A WIFI communication is used between the drone and the ground station.

It is in general hard to modify the software or the hardware; hence, no modifications in the prototype were realized. The idea is to focus the research efforts in the development of new algorithms and solutions using this kind of test-beds. For our application, we propose to use the open-source ROS (Robot Operating System) in the ground station. The video stream from the frontal camera is used with the monocular vision algorithm and fused, all in real-time on ROS, with the measurements coming from the accelerometers, gyrometers, altimeter and the downward camera to estimate the vehicle position and perceive the environment (obstacle avoidance).

Figure 1 depicts an overview of the full experimental setup and the nodes used in ROS. From Fig. 1, the *AR. Drone driver node* connects the ground station to the aerial vehicle. The *Joystick node* is used to control the quadrotor in manual mode. *Tum_ardrone state estimation node* computes the PTAM algorithm for localization. The *trajectory generator node* was included for adding new trajectories. The *Collision-free trajectory tracking node* includes the control algorithm. The *parameter adjust node* gives the user the possibility of tuning the algorithms gains online. And finally the *plots node* helps for visualizing graphically the quadrotor performance.

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

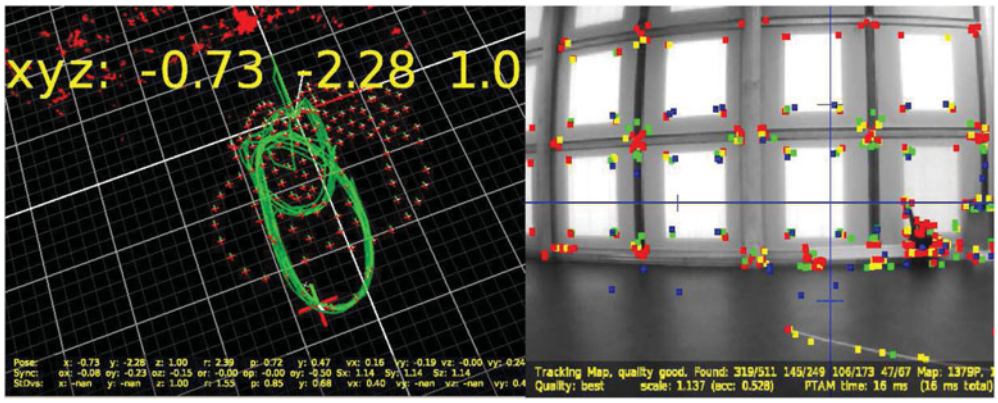

Fig. 2. UAV localization w.r.t. the sparse depth map (left). Characteristic features on the image (right).

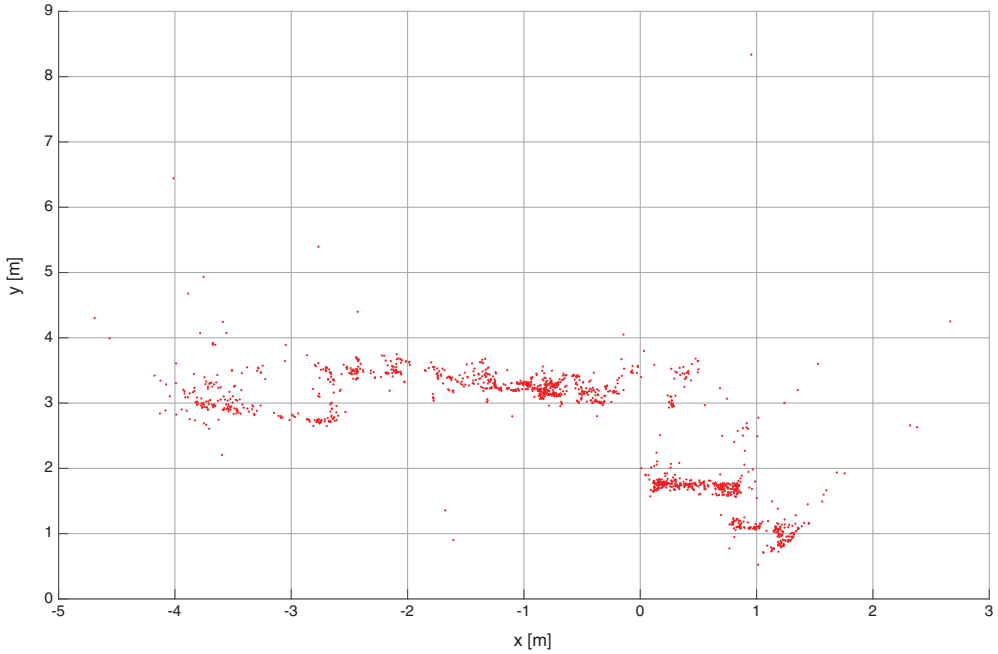

Fig. 3. Horizontal projection of the sparse depth map obtained by PTAM.

2.2. Monocular vision localization

The aerial vehicle position is obtained using computer vision and inertial data fused with an EKF algorithm. The vision algorithm, based on PTAM, estimates camera pose in an unstructured scene.[6,9,11] The algorithm executes in parallel the vision information for the tracking and mapping. Even if the PTAM is a good solution for pose estimation, it was conceived for mostly static and small scenes, and an absolute scale for the map is not provided. This could be considered as a drawback for MAV's (Micro Aerial Vehicles) applications. Nevertheless, in refs. [11] and [12], the authors proposed a nice solution fusing data measurements coming from an IMU, a camera, ultrasounds sensors and using a scale estimator and an EKF. One advantage of this solution is that the vision approach can be found as open-source for ROS.

For the present work, the estimation module in ROS proposed by ref. [11] was modified and enhanced for obstacle detection and avoidance. Also, the control algorithm code was completely replaced for a new one in order to easily implement and validate different control strategies and help to tune the required gains (parameter adjust node). Furthermore, a trajectory generator was also included for tracking autonomously different kind of time-varying trajectories, rather than just way-points. Finally, the localization algorithm was modified to recover the pointcloud of the depth map generated by the PTAM algorithm and sent it to another node to estimate the distance to potential

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

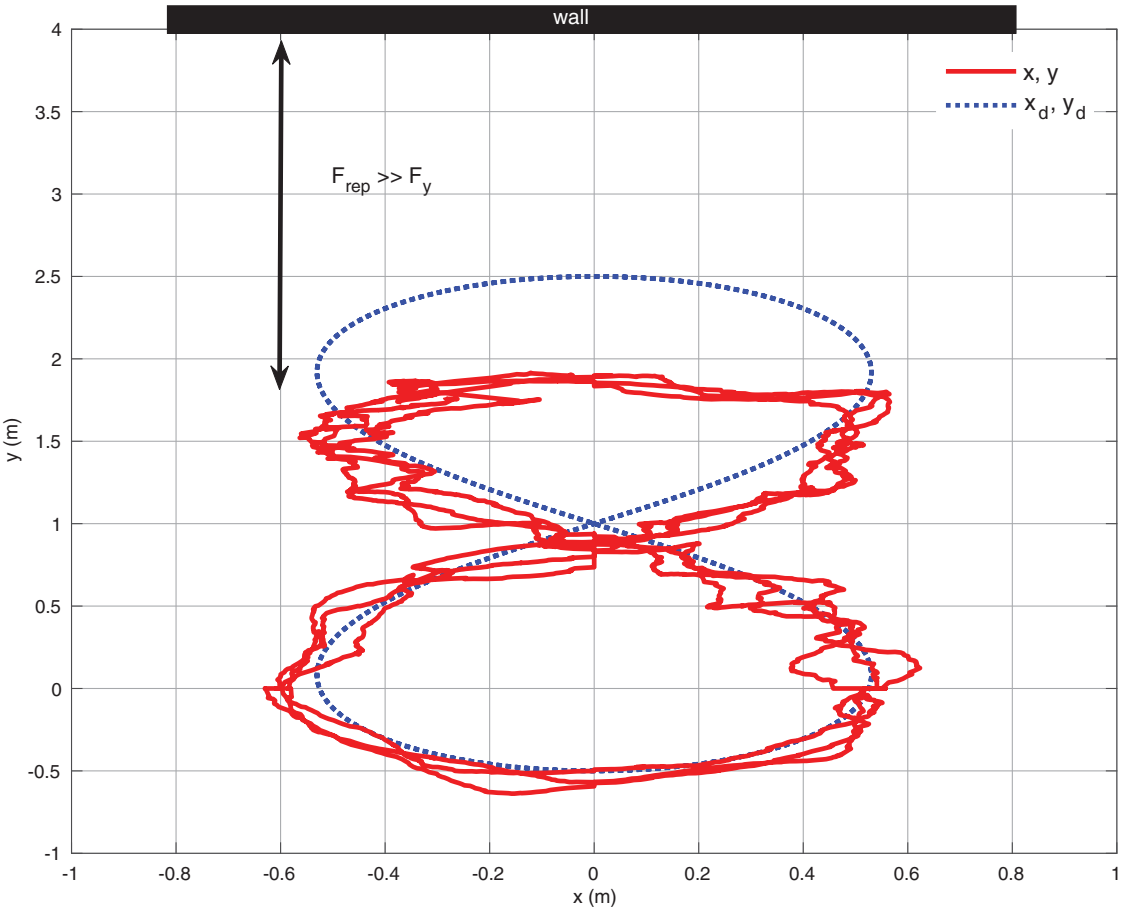

Fig. 4. Collision-free trajectory tracking. The aerial vehicle is capable of following the trajectory except when the obstacle is detected and the repulsive forces is induced. The desired path is a lemniscate and the wall is represented by the black rectangle.

frontal collisions, see Fig. 2, as explained in Section 3.2. In this way, the operator can select online between the different programmed trajectories, control laws and operation modes, as well as modify in real time any parameter for tuning.

3. Collision-Free Autonomous Tracking

Let us consider a simplified version of the well-known dynamic model of a quadrotor:[26]

$$m\ddot{\xi} = TRe_3 - mge_3 + w \tag{1}$$

$$\ddot{\Phi} \approx \Gamma \tag{2}$$

with mass m and the gravity constant g. Also, $\xi = [x\ y\ z]^T$ defines the position of the quadrotor with respect to an inertial frame, $\Phi = [\phi\ \theta\ \psi]^T$ stands for the Euler angles roll, pitch and yaw. $T \in \Re^+$ defines the total thrust produced by the motors and $\Gamma \in \Re^3$ denotes the control torque produced by the differential of rotors velocities. $R \in SO(3)$ represents the rotation matrix from the body fixed frame to the inertial one, and $e_3 = [0\ 0\ 1]^T$. Finally, $w \in \Re^3$ is an external and unknown disturbance vector.

3.1. 2-Sliding mode trajectory tracking control

Time-scale separation allows us to hierarchically design separate controllers for the rotational and translational dynamics.[27] This is possible considering that the close loop rotational subsystem is much faster than the translational one. In addition, taking into account the prototype characteristics,

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

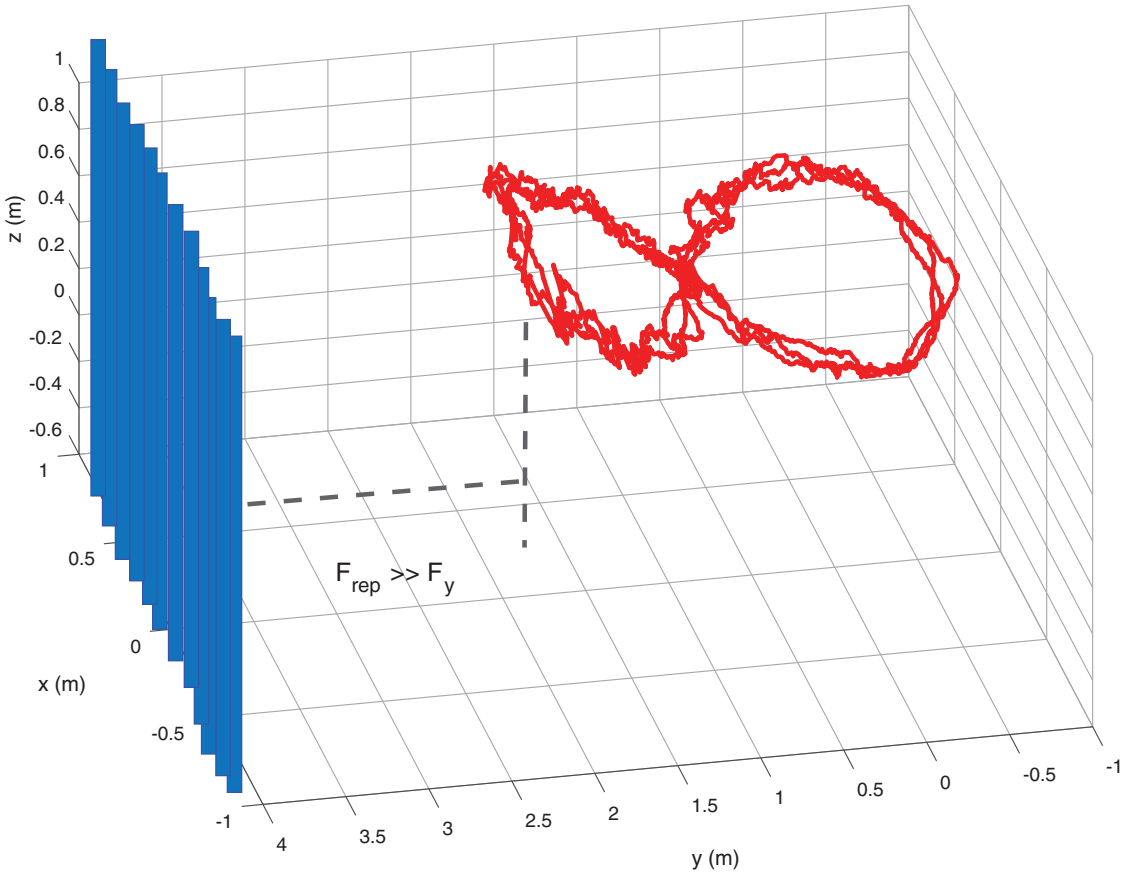

Fig. 5. 3D Collision-free lemniscate trajectory tracking close to a wall. The solid line is the vehicle path. Note that it is flattened in the y axis at 2m from the wall. It is a consequence of the potential field for avoiding the obstacle.

the desired references for the attitude are used as control inputs for the trajectory tracking. Note that this is suitable for most commercially available UAVs with internal autopilot.

Then, the control goal is to design a trajectory tracking control for performing outdoor flight autonomous navigation. It is required to give robustness properties to the algorithm for dealing with external perturbation and unknown uncertainties produced by the changing weather conditions, mainly the wind. A second order sliding mode was selected for this purpose.

Define the desired position ξ_d and the position error as $\bar{\xi} = \xi - \xi_d$, then substituting into (1) leads to

$$m\ddot{\bar{\xi}} = (TRe_3)_d - mge_3 - m\ddot{\xi}_d + w. \tag{3}$$

Let us consider the so called switching function

$$\sigma = \dot{\bar{\xi}} + k_1 \int \bar{\xi}\, \mathrm{d}t, \tag{4}$$

where $k_1 \in \Re^{3x3}$ is a diagonal positive definite gain matrix. Then, computing the second time derivative of the previous equation, it yields

$$\begin{aligned} \ddot{\sigma} &= \dddot{\bar{\xi}} + k_1 \ddot{\bar{\xi}} \\ &= \frac{1}{m}(TRe_3)_d - g\dot{e}_3 - \dddot{\xi}_d + k_1 \ddot{\bar{\xi}} + \frac{\dot{w}}{m}. \end{aligned} \tag{5}$$

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

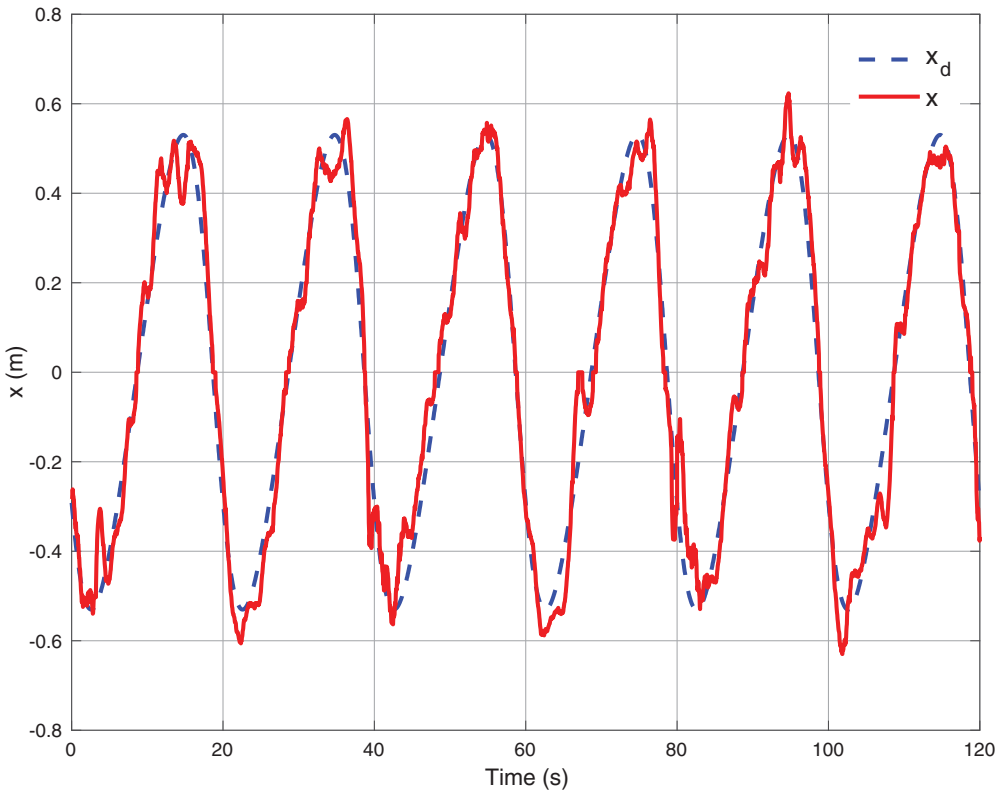

Fig. 6. x-state response (solid line) when the quadrotor follows the lemniscate trajectory (dashed line). Observe the good performance of the controller when tracking the desired path in the x-axis. As can be seen in Figs. 4 and 5 the wall is located in the y axis only.

Consider now $u = (TRe_3)_d$ to be the control input. Then, using the twisting algorithm theory in ref. [28], the following discontinuous controller is proposed

$$u = m(ge_3 + \ddot{\xi}_d - k_1\dot{\tilde{\xi}} - r_1 Sgn(\sigma) - r_2 Sgn(\dot{\sigma}) - k_2\dot{\sigma}), \tag{6}$$

where $r_1,\ r_2,\ k_2 \in \Re^+$ are constant control parameters. Introducing the above into (5), the first three terms will annihilate the continuous dynamics of the closed-loop system and (5) becomes

$$\ddot{\sigma} = -r_1 Sgn(\sigma) - r_2 Sgn(\dot{\sigma}) - k_2\dot{\sigma} + \frac{w}{m}. \tag{7}$$

Observe that the first two terms introduce a discontinuous control while the final term $(-k_2\dot{\sigma})$ induces a proportional feedback of the position error. It is aimed to improve the controller behavior and facilitate the parameter tuning.

The vectorial sign function is defined for a vector $\Upsilon = [\upsilon_1\ \upsilon_2\ \upsilon_3]^T$ as

$$Sgn(\Upsilon) = \begin{bmatrix} sgn(\upsilon_1) \\ sgn(\upsilon_2) \\ sgn(\upsilon_3) \end{bmatrix}; \quad sgn(\upsilon_i) = \begin{cases} 1 & \upsilon_i > 0 \\ 0 & \upsilon_i \in [-1, 1]. \\ -1 & \upsilon_i < 0 \end{cases} \tag{8}$$

Then, using the Lyapunov stability theory for non-smooth Lipschitz continuous Lyapunov functions,[29] the stability of the closed-loop system can be analyzed. Let us consider the following Lipschitz continuous Lyapunov function

$$V = r_1||\sigma||_1 + \frac{1}{2}\dot{\sigma}^T\dot{\sigma} \tag{9}$$

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

Table I. Parameters.

k_1	k_2	$\varepsilon[m]$	k_{rep}	$d_s[m]$
2	25	0.5	4	5

differentiating (9) with respect to time, everywhere but on $\sigma = 0$ where V is not differentiable, leads to

$$\dot{V} = r_1 Sgn(\sigma)^T \dot{\sigma} + \ddot{\sigma}^T \dot{\sigma} \tag{10}$$

and after some manipulations

$$\dot{V} \leq ||\dot{\sigma}||_1(-r_2 + \frac{1}{m}||w||) - k_2\dot{\sigma}^2 \tag{11}$$

if the external disturbance is bounded $||w|| \leq ma$, for some $a > 0$, and choosing $r_2 > a$ we have $\dot{V} \leq 0$.

Applying an extended version of the LaSalle invariant principle, and following a similar procedure as the one presented in ref. [25], we can show that choosing $r_1 > r_2 + a$, the largest invariant set where $\dot{V} = 0$ contains only the origin $\sigma = \dot{\sigma} = 0$, and all the trajectories of the system converge to zero.

Observe that

$$R_d e_3 = \begin{bmatrix} R_{dx} \\ R_{dy} \\ R_{dz} \end{bmatrix} = \frac{(TRe_3)_d}{T_d} \tag{12}$$

with $T_d = \|(TRe_3)_d\|$.

For an easy implementation, propose ψ_d constant. Using the short notation $s\alpha = \sin(\alpha)$, $c\alpha = \cos(\alpha)$, it is possible to find ϕ_d and θ_d explicitly as

$$\phi_d = \arcsin\left(R_{dx}s\psi_d - R_{dy}c\psi_d\right) \tag{13}$$

$$\theta_d = \arcsin\left(\frac{R_{dx}c\psi_d + R_{dy}s\psi_d}{c\phi_d}\right), \tag{14}$$

where ϕ_d and θ_d are the desired roll and pitch angles provided to the quadcopter vehicle.

3.2. Collision avoidance

One of the main challenges in aerial autonomous navigation, is the perception of unknown environments with the extra difficulty of the limited payload. In this section, we extend the results from monocular vision for detecting possible frontal collisions using only the sensors embedded in the prototype. Particularly, the visual information is used to estimate the distance to obstacles from the horizontal projection of the sparse depth map computed by the PTAM algorithm, as the one showed in Fig. 3.

It consists of a set P of n points $p_i(x_i, y_i, z_i),\ i = 1, \ldots, n$ obtained from the characteristic features of the image stream provided by the frontal camera of the quadrotor. PTAM uses these points as a map to estimate the pose of the UAV. The horizontal projection of this point cloud is used for computing the distance to frontal obstacles, we will consider for the first tests that the obstacles will have the same height (this stands for walls, columns, etc.). However, this horizontal projection results in a very noisy depth map and should be given special care for obstacles presenting low-texture surfaces for the vision algorithm.

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

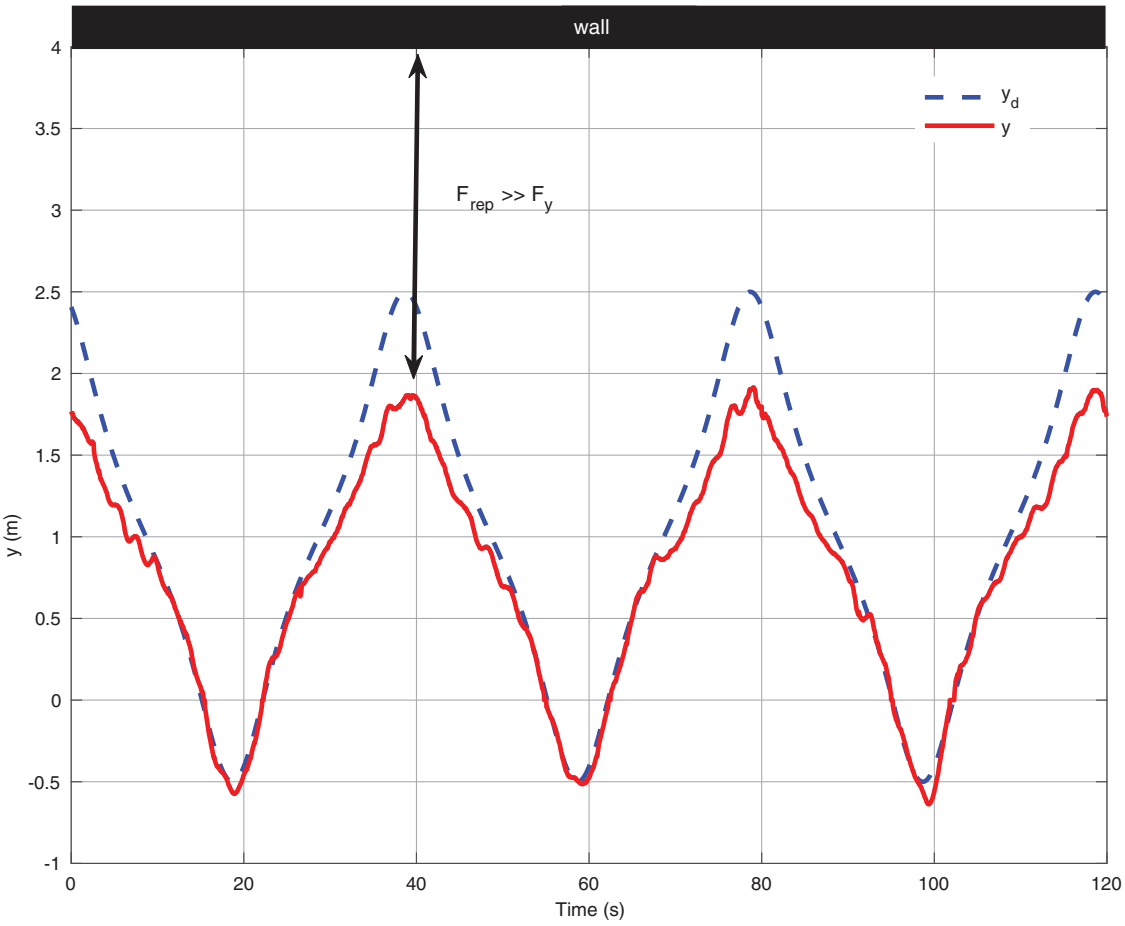

Fig. 7. y-state response (solid line) when the aerial vehicle tracks the lemniscate trajectory (dashed line). Note from Figs. 4 and 5 that the repulsive force is larger than the controller computed for this axis $F_{\text{rep}} \gg F_y$, and then, the controller gives priority to avoid collisions and as a consequence it did not reach the desired trajectory.

Therefore, we define the estimated distance to frontal obstacles d_y as

$$\begin{aligned} d_y &= y - \tfrac{1}{\eta_y} \sum_{i \in \Omega_y} y_i \\ \Omega_y &= \{p_i(x_i, y_i, z_i) \in P |\, x_i \in [x - \varepsilon, x + \varepsilon]\} \end{aligned} \tag{15}$$

i.e., the average depth, w.r.t. the position of the quadrotor along y, of the η points inside certain lateral range ε from the lateral position of the quadrotor x. Analogously, we can obtain the estimated distance to lateral obstacles d_x.

In order to avoid collisions, we apply a potential field, such that if distance d_i $(i : x,\ y)$ falls bellow certain safe distance d_s, then, a repulsive force F_{rep_i} will be exerted as follows

$$F_{\text{rep}_i} = \left\{ \begin{array}{c|l} 0 & d_i > d_s \\ -k_{\text{rep}_i}(\frac{1}{d_i} - \frac{1}{d_s})(\frac{1}{d_i^2}) & d_i \leq d_s \end{array} \right. . \tag{16}$$

Finally, retaking the 2-SM trajectory tracking control (13) and (14), the full collision-free trajectory tracking strategy becomes

$$\phi_d = \arcsin\left(R_{dx} s\psi_d - R_{dy} c\psi_d\right) + F_{\text{rep}_y} \tag{17}$$

$$\theta_d = \arcsin\left(\frac{R_{dx} c\psi_d + R_{dy} s\psi_d}{c\phi_d}\right) + F_{\text{rep}_x}. \tag{18}$$

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

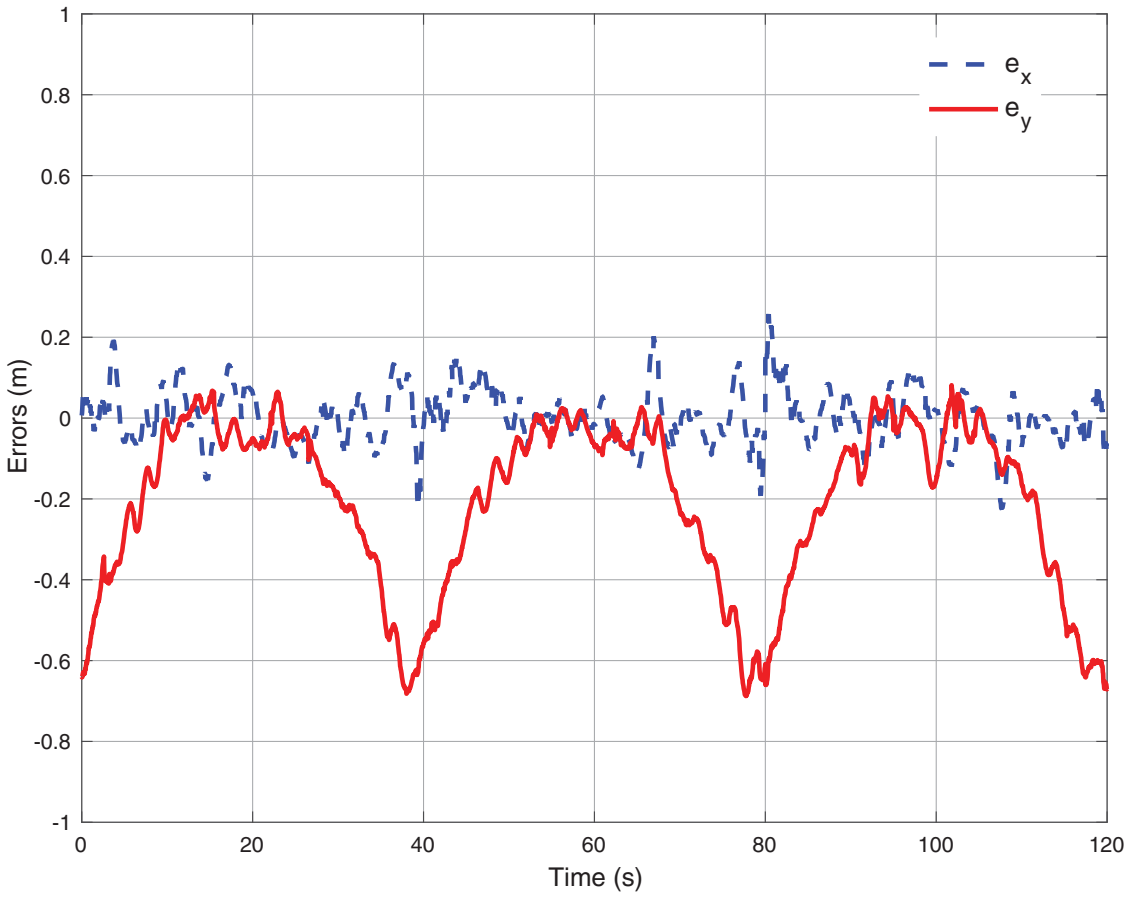

Fig. 8. Error states when tracking the trajectory. From Fig. 4 notice that the desired path is followed almost perfectly and then the error in the x-axis is small (dashed line in this figure). Nevertheless, from Fig. 5 note that the desired path is not well tracked because the potential field appears producing an error in the y-axis (solid line in this figure).

It must be noted that lateral obstacle detection becomes much more challenging, since only a frontal camera is used for this purpose.

4. Real-Time Experiments

Extensive experiments were executed to validate the proposed control-vision collision-free scheme. Two practical validations were developed; first, for autonomous trajectory tracking close to a wall and, second, when the quadcopter is surrounding a column-kind obstacle. The parameters used for the first case were tuned by trial and error and are presented in Table I. For surrounding a column, the gains were relaxed due to the difficulty of the problem while dealing with a lateral obstacle and using only a frontal camera for detection.

4.1. Trajectory tracking

In this test, the mission was to follow autonomously a lemniscate trajectory with a length of 3 m, see Fig. 4. The challenge in this mission is to introduce a wall close to the trajectory, such that the vehicle, using the monocular camera detects the wall and activates the repulsive algorithm for collision avoidance.

Figures 4 and 5 (3D view) illustrate the experiment where we can appreciate how the vehicle modifies its trajectory when it is close to the wall. Notice here that the repulsive force F_{rep} becomes bigger than the force generated by the tracking control F_y while approaching the wall, producing the deflection with respect to the desired trajectory. This is expected since the safety of the system is more important than accomplishing the tracking mission.

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

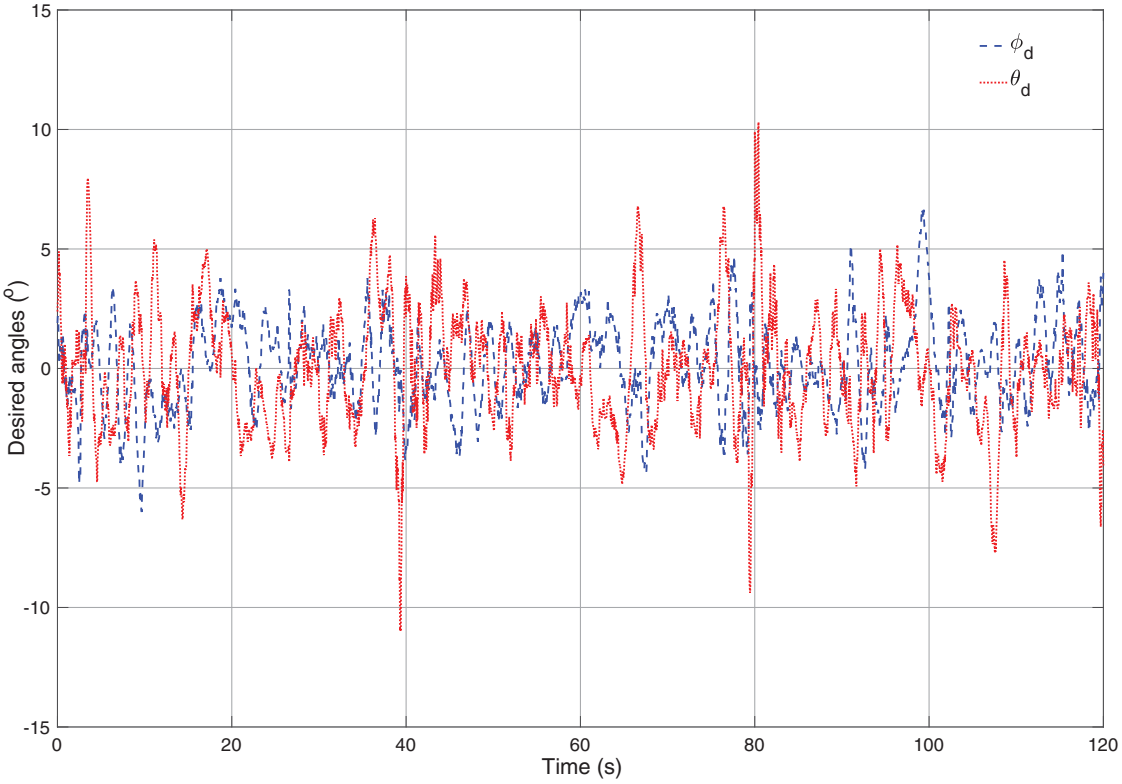

Fig. 9. Desired angles computed for free collision and trajectory tracking. These angles act as virtual control inputs for the trajectory tracking controller.

Figures 6 and 7 introduces the x and y responses of the quadrotor while following the lemniscate trajectory. In this experiment the wall is in front of the y axis. Observe in these figures that the quadrotor carries out very well the path tracking in the x coordinate, minor errors can be appreciated due to the aggressive nature of the lemniscate path on this axis, see Fig. 6. Note that when the vehicle approaches the wall (see Fig. 7) it cannot follow the desired trajectory on the y axis, this is due to the repulsive force in the control scheme which becomes bigger than the trajectory tracking law.

In Fig. 8 the tracking errors $\bar{\xi}_x$, $\bar{\xi}_y$ are displayed. These errors always remain bounded in small values except for the time when the UAV get close to the wall. The control inputs for the internal autopilot are depicted in Fig. 9. It is important to point out that small gains have been chosen in the discontinuous terms for the real-time implementation to avoid aggressive commands. This is due to the discontinuous nature of the controller, to solve this issue, it is considered to explore in future developments a super-twisting algorithm, to attenuate the undesired chattering effect.

From this experiment we can conclude that the trajectory tracking control performs well even when an obstacle is present. Meanwhile, we corroborate that obstacle avoidance is possible for obstacles like flat walls, using only a frontal camera for detection and without any extra expensive computation. In order to test the collision avoidance strategy in a more challenging scenario, Section 4.2 address the position control problem when an obstacle is present in the path. In this case, the aerial vehicle needs to surround the obstacle to arrive to the desired position.

4.2. Surrounding an obstacle

In this case, the objective consists in going from point (0,0) to (0,9) without clashing against the obstacle localized halfway, as depicted in Fig. 10. The obstacle is placed in an unknown position for the quadcopter and has an irregular shape, as shown in Fig. 11. Therefore, the practical objective will be that the vehicle detects it and surrounds it in order to accomplish its mission without collision.

The experimental results are presented in Figs. 10–14. In Fig. 10 we can appreciate the performed trajectory by the UAV in the horizontal plane. We can see how the quadrotor was able to detect the obstacle and change its course to continue its mission without collision. Hence, a frontal repulsive

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

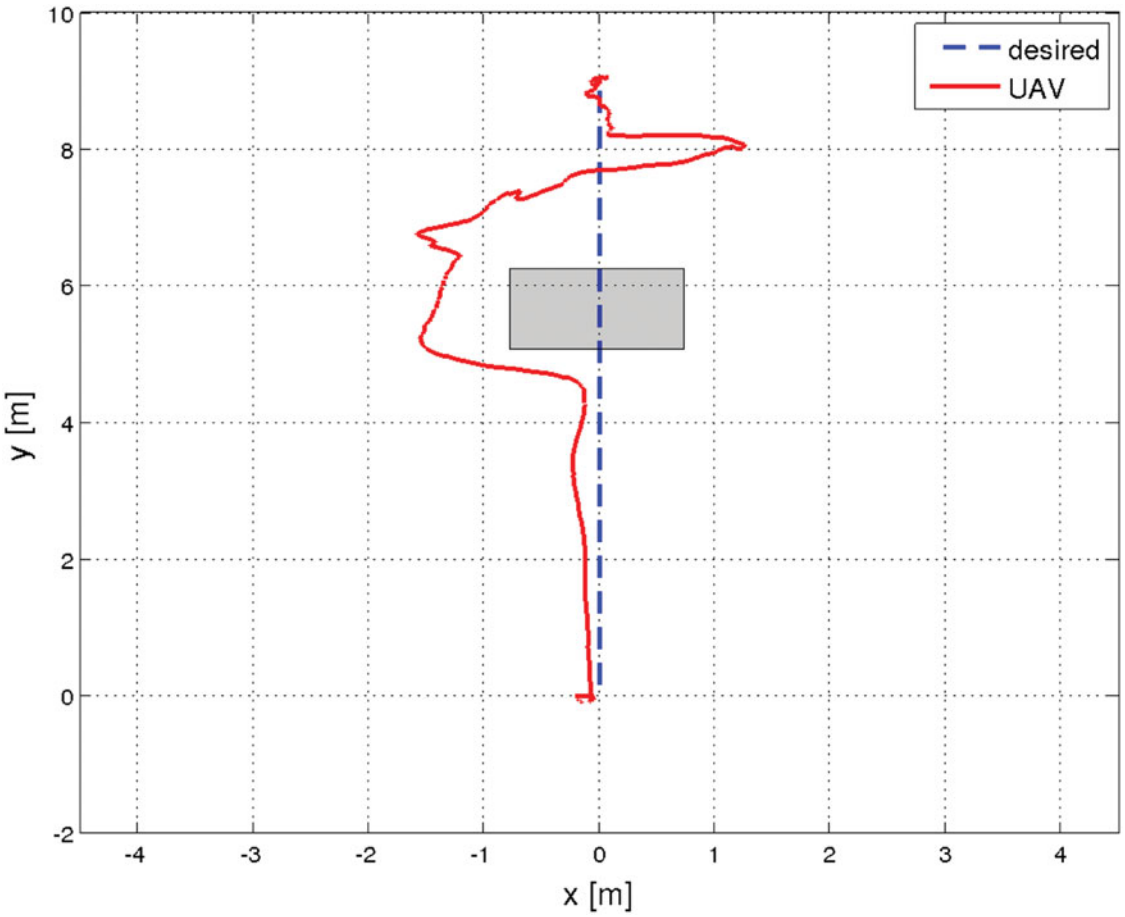

Fig. 10. Quadrotor response (solid line) when following a desired trajectory (dashed line) with an obstacle.

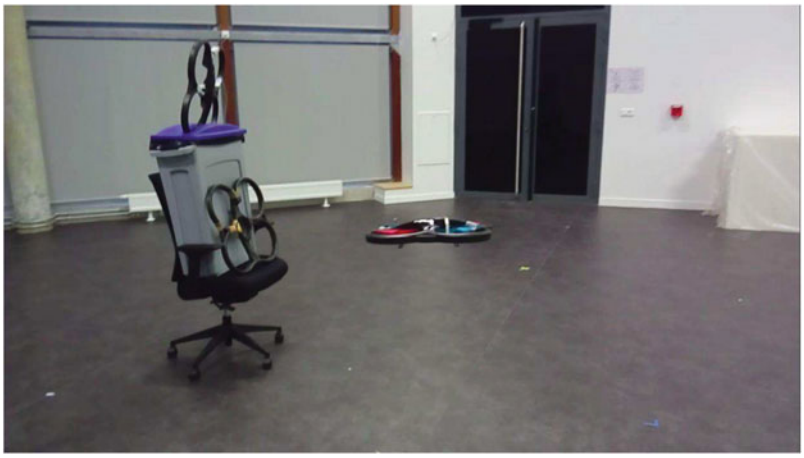

Fig. 11. Obstacle avoidance experiment. Note that the obstacle has an irregular form and the vision algorithm is capable of detecting it.

force is exerted to prevent the UAV from colliding, and once the UAV is close to the obstacle, a lateral repulsive force allows to evade the obstacle and continue to the goal.

As previously stated, the control gains were relaxed in this experiment due to its difficulty. This can be appreciated in Fig. 10 where the aerial vehicle follows the desired path but some errors can be appreciated in the x axis.

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

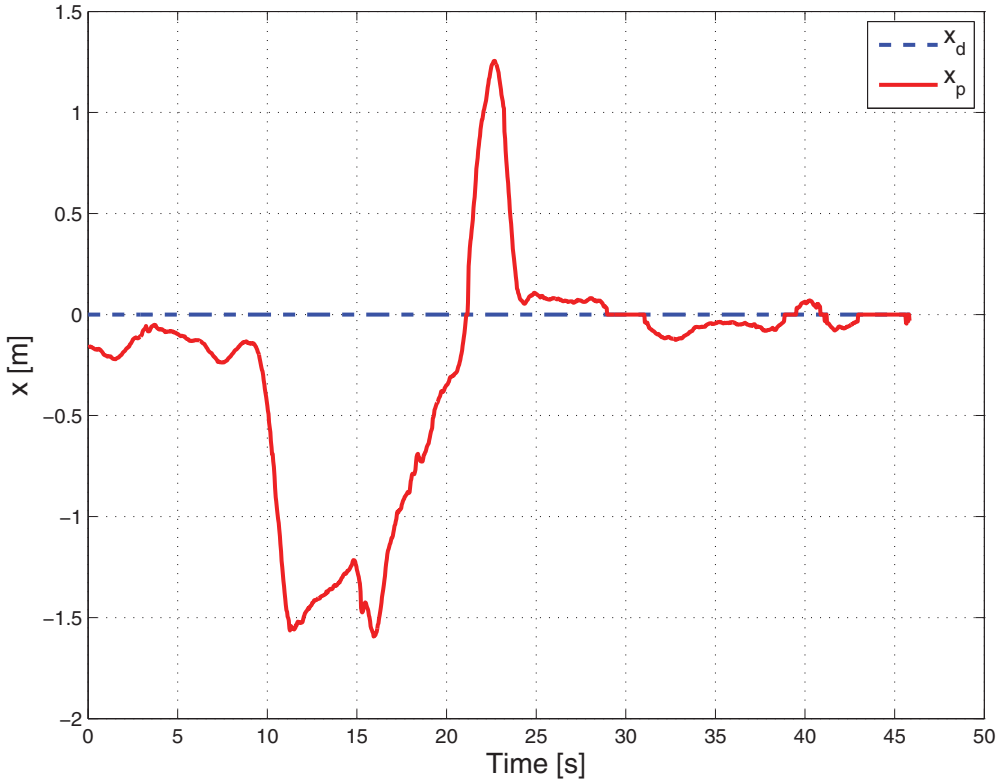

Fig. 12. *x*-axis response (solid line) when the vehicle follows a line trajectory (dashed line) and avoiding an obstacle in the trajectory.

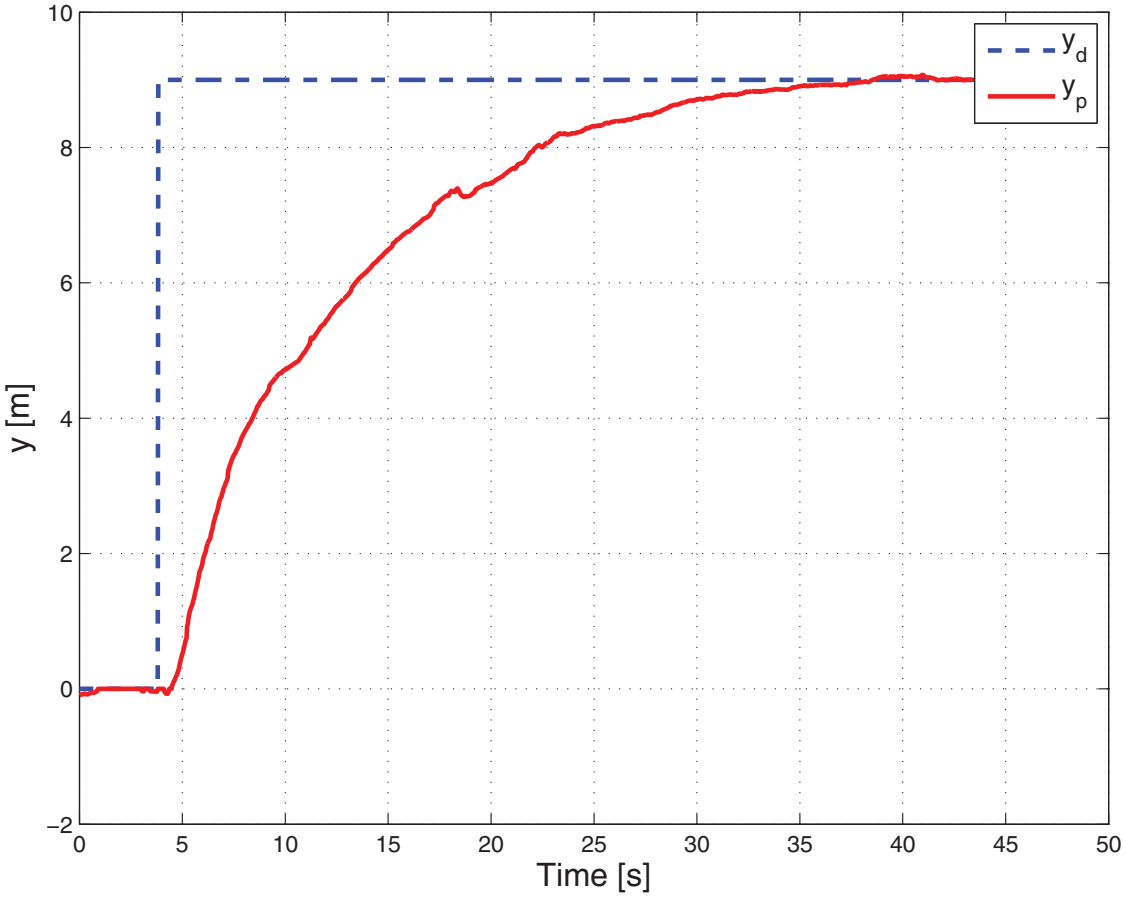

Fig. 13. *y*-state performance when the aerial vehicle surrounds an obstacle in the desired trajectory.

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

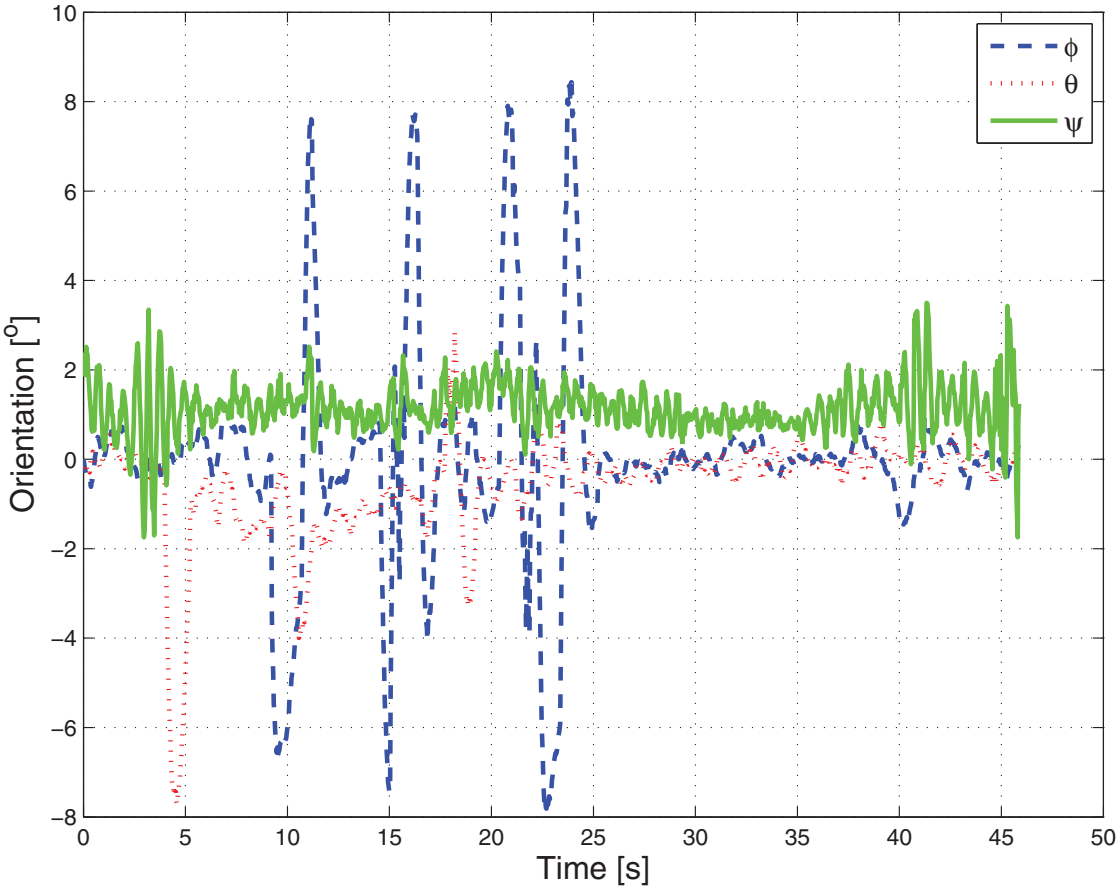

Fig. 14. Attitude quadrotor response for the experiment. Observe that the yaw angle remains constant and the pitch angle has a significant value at the beginning because there is no obstacle in the trajectory (the vehicle is moving in this axis) and later moves slowly when the obstacle is present. It is different for the roll angle that at the beginning remains almost constant and changes to produce a lateral displacement in the drone for avoiding the obstacle.

In Figs. 12 and 13 we can appreciate the x and y responses. From the x position in Fig. 12, we can observe that there is a big overshoot at time 22 s, once the quadrotor passes the obstacle and tries to retake its way to the goal. This is attributed to the relaxed gain adjustment in the x coordinate. Meanwhile, in Fig. 13 we can appreciate how the helicopter converges to the reference almost exponentially, despite the presence of the obstacle.

Finally, Fig. 14 shows the quadrotor orientation while performing the task. It is interesting to note the attitude maneuvers performed by the UAV in order to quickly go to the target at time 4 s, and in order to evade the obstacle at $t = 9$ s. We can also observe some peaks at $t = 15$ s and $t = 21$ s probably due to noise in the point cloud.

It is important to point out that the obtained results are quite satisfactory, taking into account that only a monocular camera is used to locate the quadrotor and to detect collisions, instead of using an expensive motion capture system and/or extra range sensors as others teams. Henceforth, the obtained results can be easily reproduced on outdoor flight tests.

Some experiments of this work were recorded in a video[(1)].

5. Conclusion and Future work

Collision-free navigation for a quadrotor using only a monocular camera was presented and experimentally validated in this paper. Despite the simplicity of the proposed collision avoidance technique, it satisfactorily accomplished its goal.

[(1)]`https://youtu.be/A-6zgwdjn3k`

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press

A second order sliding mode control proved to be adequate for accurate trajectory tracking with the presented setup, accomplishing well the tracking objective while adding robustness against external perturbations. However, special attention must be payed when implementing this kind of controllers because of their discontinuous nature, in this case, the discontinuous terms were kept bounded by choosing small gains. If required, a second order sliding mode control using the super-twisting algorithm could be used to attenuate the chattering effect.

Monocular-vision based navigation showed to be a powerful tool for MAVs in GPS denied environments, as well as an exciting research area to be explored.

Future work includes extending the obtained results for outdoor tests, and to improve the obstacle detection algorithm for including the lateral direction. This could be performed using more sophisticated techniques for point-cloud filtering or adding other sensors.

Acknowledgment

This work has been sponsored by the French National Networks of Robotics Platforms - ROBOTEX (ANR-10-EQPX-44).

References

1. Y. Song, B. Xian, Y. Zhang, X. Jiang and X. Zhang, "Towards autonomous control of quadrotor unmanned aerial vehicles in a GPS-denied urban area via laser ranger finder," *Optik-Int. J. Light Electron Opt.* **126**(23), 3877–3882 (2015), ISSN 0030-4026.
2. J. Courbon, Y. Mezouar, N. Guenard and P. Martinet, "Vision-based navigation of unmanned aerial vehicles," *Control Eng. Pract.* **18**(7), 789–799 (2010), ISSN 0967-0661.
3. Y. Bi and H. Duan, "Implementation of autonomous visual tracking and landing for a low-cost quadrotor," *Optik-Int. J. Light Electron Opt.* **124**(18), 3296–3300 (2013), ISSN 0030-4026.
4. D. Maravall, J. de Lope and J. Fuentes, "Vision-based anticipatory controller for the autonomous navigation of an UAV using artificial neural networks," *Neurocomputing* **151**(1), 101–107 (2015), ISSN 0925-2312.
5. P. Li, M. Garratt, A. Lambert and S. Lin, "Metric sensing and control of a quadrotor using a homography-based visual inertial fusion method," *Robot. Auton. Syst.* **76**, 1–14 (2016), ISSN 0921-8890.
6. G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japan (2007).
7. M. Achtelik, M. Achtelik, S. Weiss and R. Siegwart, "Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In- and Outdoor Environments," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China (2011).
8. S. Weiss, M. Achtelik, S. Lynen, M. Chli and R. Siegwart, "Real-time Onboard Visual-Inertial State Estimation and Self-Calibration of MAVs in Unknown Environments," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Saint Paul, MN, USA (2012).
9. S. Weiss, D. Scaramuzza and R. Siegwart, "Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments," *J. Field Robot.* **28**(6), 854–874 (2011).
10. S. Weiss, S. M. Achtelik, S. Lynen, M. Achtelik, L. Kneip, M. Chli and Siegwart, R. "Monocular vision for long-term micro aerial vehicle state estimation: A compendium," *J. Field Robot.* **30**: 803–831. doi:10.1002/rob.21466, (2013).
11. J. Engel, J. Sturm and D. Cremers, "Camera-Based Navigation of a Low-Cost Quadrocopter," *Proceedings of the IEEE International Conference on Intelligent Robot Systems (IROS)*, Vilamoura, Portugal (2012).
12. J. Engel, J. Sturm and D. Cremers, "Scale-aware navigation of a low-cost quadrocopter with a monocular camera," *Robot. Auton. Syst.* **62**(11), 1646–1656 (2014).
13. M. Nieuwenhuisen, D. Droeschel, M. Beul and S. Behnke, "Obstacle Detection and Navigation Planning for Autonomous Micro Aerial Vehicles," *Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, USA (2014).
14. A. Beyeler, J. Zufferey and D. Floreano, "3D Vision-Based Navigation for Indoor Microflyers," *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, Roma, Italy (2007).
15. C. Bills, J. Chen and A. Saxena, "Autonomous MAV Flight in Indoor Environments Using Single Image Perspective Cues," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China (2011).
16. T. Mori and S. Scherer, "First Results in Detecting and Avoiding Frontal Obstacles From a Monocular Camera for Micro Aerial Vehicles," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany (2013).
17. H. Alvarez, L. M. Paz, J. Sturm and D. Cremers, "Collision Avoidance for Quadrotors With a Monocular Camera," *Proceedings of The 12th International Symposium on Experimental Robotics (ISER)*, Marrakech and Essaouira, Morocco (2014).
18. R. Naldi, M. Furci, R. G. Sanfelice and L. Marconi, "Robust global trajectory tracking for underactuated VTOL aerial vehicles using inner-outer loop control paradigms," *IEEE Trans. Autom. Control* **62**(1), 97–112 (2017).

19. H. Liu, W. Zhao and Z. Zuo, "Robust control for quadrotors with multiple time-varying uncertainties and delays," *IEEE Trans. Ind. Electron.* **64**(2), 1303–1312 (2017).
20. C. Aguilar-Ibaez, "Stabilization of the PVTOL aircraft based on a sliding mode and a saturation function," *Int. J. Robust Nonlinear Control* **27**, 843–859 (2017).
21. F. Yeh, "Attitude controller design of mini-unmanned aerial vehicles using fuzzy sliding-mode control degraded by white noise interference," *IET Control Theory Appl.* **6**(9), 1205–1212 (2012).
22. H. Alwi and C. Edwards, "Sliding mode fault-tolerant control of an octorotor using linear parameter varying-based schemes," *IET Control Theory Appl.* **9**(4), 618–636 (2015).
23. L. Derafa, L. Fridman, A. Benallegue and Ouldali, "Super Twisting Control Algorithm for the Four Rotors Helicopter Attitude Tracking Problem," *Proceedings of the 11th International Workshop on Variable Structure Systems*, Mexico City, Mexico (2010).
24. R. Xu and Ü. Özgüner, "Sliding Mode Control of a Quadrotor Helicopter," *Proceedings of the 45th IEEE Conference on Decision & Control*, San Diego, CA, USA (2006).
25. D. Mercado, P. Castillo, R. Castro and R. Lozano, "2-Sliding Mode Trajectory Tracking Control and EKF Estimation for Quadrotors," *Proceedings of the 19th IFAC World Conference*, Cape Town, South Africa (2014).
26. P. Castillo, R. Lozano and A. Dzul, *Modelling and Control of Mini-Flying Machines* (Springer-Verlag, Londres, 2005).
27. S. Bertrand, N. Gunard, T. Hamel, H. Piet-Lahanier and L. Eck, "A hierachical controller for miniature VTOL UAVs: Desing and stability analysis using singular perturbation theory," *Control Eng. Practice* **19**, 1099–1108 (2011).
28. Y. Shtessel, C. Edwards, L. Fridman and A. Levant, *Sliding Mode Control and Observation* (Birkhäuser Basel, Springer New York, 2010).
29. D. Shevitz and B. Paden. "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Automat. Control* **39**, 1910–1914 (1994).

https://doi.org/10.1017/S0263574718000516 Published online by Cambridge University Press