


Autonomous flying with quadcopter using fuzzy control and ArUco markers

Jan Bacik¹  · Frantisek Durovsky² · Pavol Fedor¹ · Daniela Perdukova¹

Received: 14 June 2016 / Accepted: 22 January 2017 / Published online: 13 February 2017
© Springer-Verlag Berlin Heidelberg 2017

Abstract In this paper, we present an approach which enables a low-cost quadcopter to fly various trajectories autonomously. Artificial landmarks are used for pose estimation, and a fuzzy controller is utilized to generate steering commands. The presented system can navigate a low-cost quadcopter along a predefined path without the need for any additional external sensors. In addition to a full description of our system, we also introduce our software package for Robot Operating System, which allows the robotics community to experiment with proposed mapping algorithm.

Keywords Quadcopter · ArUco · Autonomous flying · ROS · Fuzzy control

1 Introduction

Research in the field of autonomous micro-aerial vehicles (MAVs) is currently divided into several different approaches: One part of the community focuses on improving the accuracy of quadcopter control and has led to notable advances such as aggressive flight maneuvers [1, 2], ping pong [3] and even collaborative construction tasks [4]. While these developments are certainly impressive, such systems nonetheless require external motion capture systems to perform these demanding tasks properly. Another part of

the community focuses on outdoor flights where GPS-based pose estimation is possible. Although complete solutions in this field are already available [5], researchers continue to work on improvements in stabilization, [6], photogrammetry, obstacle detection and avoidance.

However, by far the largest part of the community is interested in the most popular topic in the field of MAV's—autonomous flying in indoor GPS-restricted environments.

A wide variety of different approaches to this task have already been investigated. These range from methods which enable the quadrotor to learn a map offline from previously recorded manual flight [7], to approaches utilizing wireless network sensors [8] and hybrid ultrasound systems [9]. However, the most popular and challenging approach relies entirely on MAV onboard sensors with the use of Simultaneous Localization and Mapping (SLAM) algorithms.

Recent research has improved the accuracy of SLAM methods by combining data from laser scanners, stereo-cameras and inertial measurement units (IMU) [10], but such a complex sensing system requires high initial investment and significant computer power. In order to overcome this problem and to enable even low-cost platforms to fly autonomously, SLAM techniques using monocular sensor systems have been explored. Parallel Tracking And Mapping (PTAM) algorithm has been used successfully [11], but this method requires the MAV to remain in more or less constant visual contact with the initial starting scene, a restriction which ultimately limits the actual operating area of the MAV. Stabilizing controllers based on optical flow have also been investigated [12], but these systems tend to drift over time and are therefore not considered suitable for reliable long-term navigation.

A much easier and more intuitive method of low-cost MAV indoor localization and navigation may lie in the tech-

✉ Jan Bacik
jan.bacik.2@tuke.sk

¹ Faculty of Electrical Engineering and Informatics, Department of Electrical Engineering and Mechatronics, Technical University of Kosice, Letna 9, 042 00 Kosice, Slovakia

² Faculty of Mechanical Engineering, Department of Automation, Mechatronics and Robotics, Technical University of Kosice, Park Komenskeho 8, 042 00 Kosice, Slovakia

nique of distributing augmented reality (AR) markers in the environment, and a number of studies have already investigated this possibility. Among the earliest research in this field was that published by Eberli et al. [13] in which the camera pose was estimated from the elliptic appearance of a circle in perspective projection. By estimating 5DoF and assuming that yaw is controlled independently, this approach enables the quadrotor to take off, hover over the marker and land autonomously.

An alternative monocular pose estimation technique was proposed by Lamberti et al. [14]. A marker-less visual odometry algorithm exploiting SIFT descriptors was combined with a marker-based tracking method relying on AR markers. Markers were distributed at precise locations in the environment and served as reference points for the quadrotor. When flying from one marker to another, the position of the quadrotor was estimated entirely from the visual odometry algorithm and corrected once the next marker had been detected. One major drawback of this method, however, was the fact that the positions of the reference markers had to be hardcoded to the MAV prior to flight.

AR markers were also used in a 2014 study published by Hartmann et al. [15] which focused on improving the precision of vertical landings of quadrotors. The landing platform was tagged by several markers mounted on exact positions which thereby enabled the MAV to localize the landing area from different angles and to perform precision landings.

The research presented in our study was inspired by recent work from Munoz-Salinas et al. [16] from the University of Cordoba. In their latest paper, a novel mapping and localization approach using planar markers was presented and compared with state-of-the-art Structure from Motion and Simultaneous Localization and Mapping Techniques. The “Marker Mapper” method which they introduced is capable of obtaining better maps and localization results under a wider range of viewpoints than up-to-date SLAM techniques such as LSD-SLAM or ORB-SLAM2.

Inspired by these results, we decided to experiment with the Marker Mapper approach and to use it for real-time MAV localization and mapping. Since the original Marker Mapper method is intended to be used offline to process previously recorded video footage, our implementation simplified the existing pipeline to allow near real-time performance (Fig. 1). In order to prove its effectiveness for MAVs, we developed our work further and achieved fully autonomous flight with an off-the-shelf quadrotor relying solely on our mapping method and fuzzy control. The overall accuracy of the proposed system was tested under different lighting conditions and measured by a laser tracker station during flight. This paper is also accompanied by a video which demonstrates the robustness of our approach and the ability to follow a pre-

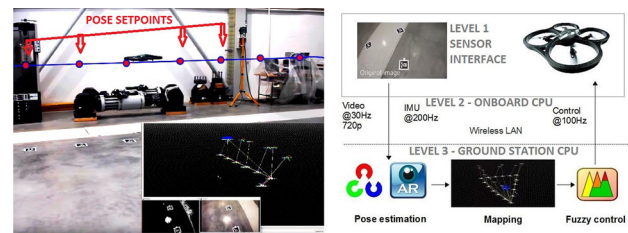


Fig. 1 Our approach enables low-cost quadcopter to follow trajectories defined by artificial landmarks



Fig. 2 Intel miniITX-based ground station for off-board processing and control

defined trajectory accurately. It is available online at: http://youtu.be/3mCu_7Ek2fs.

2 Hardware platform

The Parrot AR Drone, a commercially available quadcopter, was used as the platform for our experiments. The low unit cost of the Parrot in comparison with other MAVs is not the only advantage which it offers; the AR Drone is also robust and can safely operate in close proximity to people. Its main drawback, however, is the lack of flexibility; neither the MAV’s onboard software nor the hardware itself can be easily modified.

The AR Drone is equipped with two cameras, an ultrasound, pressure altimeter, a 3-axis gyroscope, an accelerometer and a magnetometer. The frontal 720p HD camera has a 92° diagonal field of view at a streaming rate of 30 fps. The bottom camera covers a 64° field at 60 fps with a resolution of 320 × 240 pixels. Communication with the Parrot is effected through wireless LAN from the ground station (Fig. 2).

The onboard software running a 1 GHz 32 bit ARM Cortex A8 processor uses sensor data to control the roll Φ , pitch Θ heading Ψ and vertical velocity \dot{z} according to external set points received from control system every 10 ms. The AR Drone weighs 420 g and measures 53 × 52 cm.

3 ArUco pose estimation and mapping

ArUco library was originally developed by Rafael Muñoz and Sergio Garrido [17] to generate a set of AR fiducials and to simplify the estimation of 6DoF marker poses. It is based on OpenCV and enables the detection of various tag dictionaries. Its detection pipeline is much more effective and faster than the pipeline of the similar AprilTag library.

In general, ArUco markers are synthetic square markers comprising a wide black border with an inner binary matrix. The black border facilitates the fast detection of the image and the binary codification allows marker identification, the application of error detection and correction techniques.

The marker detection process is performed in two main steps—the detection of marker candidates and codification analysis.

In the detection process, adaptive thresholding and contour extraction in combination with additional extra filtering is used in order to detect square shapes which are candidates to be markers.

The goal of the second step is to analyze inner codification. Perspective transformation is used initially in order to obtain the marker in canonical form; Otsu thresholding method is then applied to separate white and black sections, and bit by bit analysis is finally performed in order to determine whether the marker belongs to a specific tag dictionary. The position of the four marker corners in the image and the real size of the marker are sufficient to allow the system to estimate its full pose in absolute scale.

In this standard use, the camera is identified with world origin, while the marker position is calculated in respect to the camera frame. If C denotes camera frame and M-marker frame, then the pose of the marker with respect to the camera is expressed as:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_C = T_{MC} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_M \tag{1}$$

where T_{MC} is a transformation matrix calculated as:

$$T_{MC} = \begin{bmatrix} (e_0^2 + e_1^2 - e_2^2 - e_3^2) & 2(e_1e_2 + e_0e_3) & 2(e_1e_3 - e_0e_2) & t_x \\ 2(e_1e_2 - e_0e_3) & (e_0^2 - e_1^2 + e_2^2 - e_3^2) & 2(e_2e_3 + e_0e_1) & t_y \\ 2(e_1e_3 + e_0e_2) & 2(e_2e_3 - e_0e_1) & (e_0^2 - e_1^2 - e_2^2 + e_3^2) & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

with e_0, e_1, e_2, e_3 expressing quaternion components, and t_x, t_y, t_z translational components of marker positions.

This approach allows us to estimate marker pose whether the camera is identified as world origin, but for mapping purposes an inverted approach in which the camera position is

estimated with respect to the static markers is more convenient.

The same principle is also used in the AprilSLAM technique originally developed by Chao Qu and Gareth Cross from Kumar Robotics. But while AprilSLAM recalculates the position of each visible marker continuously, our approach is slightly different. In our implementation, the world’s origin is defined by the first detected marker while each consecutive marker is fixed to these world’s coordinates firmly immediately after the detection. While AprilSLAM is only aware of the locations of currently visible markers (and even these are continuously being re-calculated), our algorithm stores the positions of all of the previously detected markers. This pipeline may not be precise at the sub-millimeter level, but in combination with the more effective detection pipeline of ArUco library in general, results in the robust and smooth localization system viable even for real indoor MAV flights. For a fuller understanding of the proposed algorithm, see the original video available at the *ArUco_mapping* ROS Web site: <https://youtu.be/MIOy9qt-K4Y>.

So, in case of inverted principle, the transformation between camera frame and marker is:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_M = T_{MC}^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_C \tag{3}$$

Creating a chain of marker pose estimations, or calculating the position of the second marker with respect to the first, and the third marker to the second, etc., is the key to successful mapping (Fig. 3). In general, the n th marker position can be computed as:

$$T_{MnWo} = T_{M1M0}T_{M2M1}T_{M3M2} \dots T_{MnMn-1} \tag{4}$$

The only prerequisite of serialized mapping is the simultaneous occurrence of the $n - 1$ th marker within a camera image at the moment when the n th marker is detected for

the first time. By calculating the pose of the newly detected marker in respect to the previous one, it is possible to keep track of the whole trajectory and to locate the absolute camera position in respect to the world origin during the whole

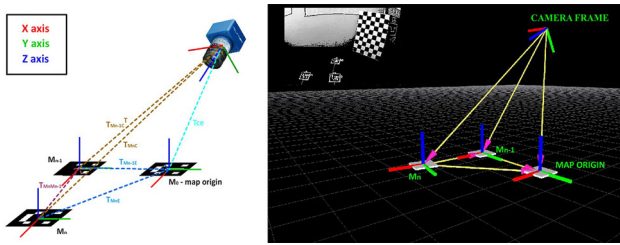


Fig. 3 Localization and mapping approach—serialization as a main principle

flight. In a real-world scenario, this limits the mutual distances of following landmarks when creating a trajectory, since at least two landmarks must be visible at the same time. If two or more markers are visible, camera pose is estimated with respect to all of them, but in order to maximize the accuracy of the system, the value used as the actual camera pose is calculated from the closest marker.

4 Fuzzy control

Robust and reliable localization is not the only requirement for autonomous flying—a precise and accurate control mechanism is also vital. PID controllers are commonly used for MAV, but recent years have seen an increased interest in the field of fuzzy control [18]. Fuzzy logic is a proven control method for nonlinear systems in situations in which exact parameter identification is difficult or even impossible. Its inference mechanism also allows simpler transition of human experience into the control algorithm and this is of considerable benefit in the development of autonomous MAV flight. Several interesting studies have been proposed in recent years describing dynamical models, simulations, identification and the development of fuzzy controllers for quadrotors [19–22,24].

Our approach is very similar to the work recently published by Indrawati et al. [25]. This study also deals with the fuzzy control of a Parrot AR Drone, and the authors define a set of linguistic rules and membership functions according to the experience of a human pilot in an approach very similar to that outlined in our study. However, the main difference when compared to our approach is the fact that our controller is capable of controlling not only movement in X-, Y-, Z-axis, but also the heading of the MAV, thereby allowing circular motions in real flight as shown in Fig. 4.

A simplified mathematical model of the quadcopter is shown in Fig. 5. The sum of forces F_i and momentum M_i serves as the input for the Newton–Euler motion equation, a formula which is typically used to derive mathematical models of systems with 6DoF.

Newton–Euler motion equations describe the movement of the center of gravity of the body caused by the actuation of

Circular motion

- simultaneous move in 2 linear and 1 one angular direction

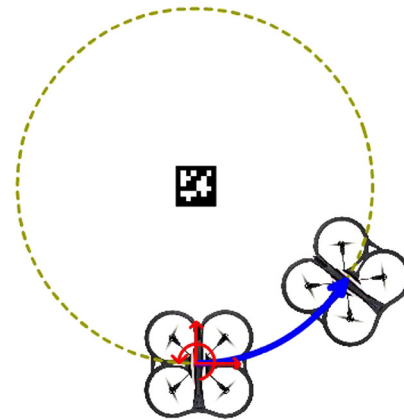


Fig. 4 Additional fuzzy control of heading enables circular motions

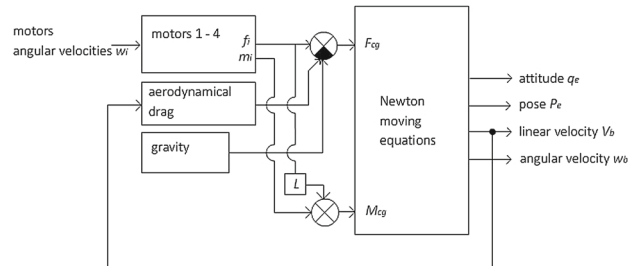


Fig. 5 Simplified mathematical model of quadcopter

external forces and momentum. The first Eq. (5) expresses the change in quadcopter rotation ω'_b due to the effect of momentum on individual axes. Changes of rotation are dependent on the magnitude of momentum and mass distribution within the drone, which is represented by the momentum of inertia. Angular rates $\omega_x, \omega_y, \omega_z$ together with input forces are used to calculate changes in linear velocities V'_b (Eq. 6). Changes of angular rates are transformed to the change of attitude quaternion q'_E (Eq. 7), and the last equation transforms the linear velocities V_x, V_y, V_z expressed in the body frame to changes in pose P'_E in Earth frame (Eq. 8).

$$\omega'_b = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}^{-1} \begin{bmatrix} \text{Ltg}(\pi/2)k(\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \\ \text{Ltg}(\pi/2)k(-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) \\ \sum_{i=1}^4 b\omega_i^2 \end{bmatrix} - \begin{bmatrix} I_z\omega_y\omega_z - I_y\omega_x\omega_z \\ I_x\omega_x\omega_z - I_z\omega_x\omega_z \\ I_y\omega_x\omega_y - I_x\omega_x\omega_y \end{bmatrix} \tag{5}$$

$$V'_b = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 k\omega_i^2 \end{bmatrix} - \frac{1}{m} \begin{bmatrix} c_x V_x \\ c_y V_y \\ c_z V_z \end{bmatrix} - m \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} - \begin{bmatrix} V_z m\omega_y - V_y m\omega_z \\ V_x m\omega_z - V_z m\omega_x \\ V_y m\omega_x - V_x m\omega_y \end{bmatrix} \tag{6}$$

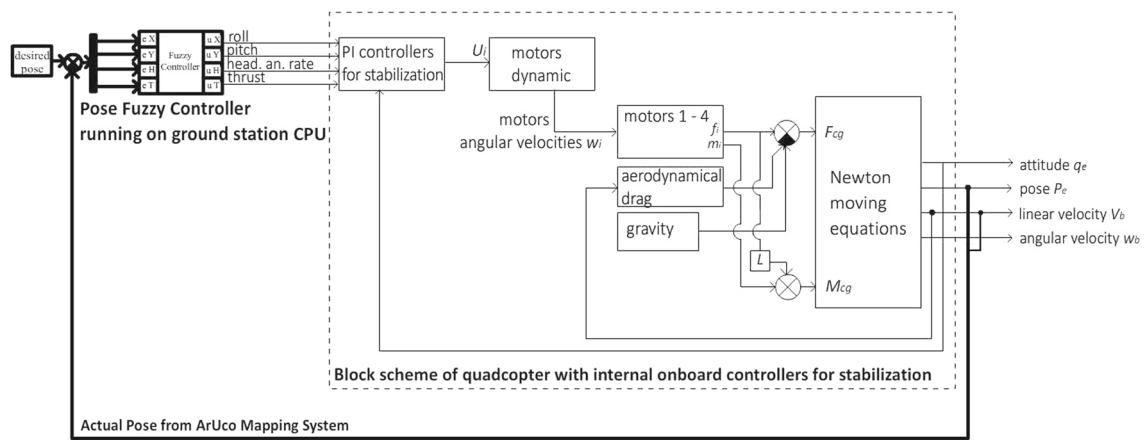


Fig. 6 Control diagram—the quadcopter is a nonlinear system with four inputs and position feedback

$$q'_E = \begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{33} & q_{32} & q_{33} \end{bmatrix} \quad (7)$$

$$P'_E = \begin{bmatrix} 2q_0^2 - 1 + 2q_1^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 2q_0^2 - 1 + 2q_2^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 2q_0^2 - 1 + 2q_3^2 \end{bmatrix}^{-1} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \quad (8)$$

In contrast to systems which use direct thrust control for each motor, the Parrot AR Drone uses built-in controllers to stabilize each axis. The system therefore expects steering commands in the form of total thrust, roll, pitch and heading set points.

The built-in stabilization system of the Parrot permitted the use of a fuzzy controller to control the quadcopter position and heading directly through four system outputs (See Fig. 6). For the purposes of this study, a discrete fuzzy controller with a standard PI structure—input variables e , and Δe , with output Δu for all four control variables—was chosen.

In the case of our controller, regulation deviation e describes the difference between the desired and current speeds. Changes in regulation deviation Δe provide information about the actual dynamics of the system for each axis—if the system accelerates or decelerates. The control effort Δu is the output value—speed set point in particular axis.

In theory, a Mamdani fuzzy controller is comprised of four main blocks [23]—fuzzifier, knowledge base, inference mechanism, defuzzifier. If a normalized signal is required, two additional blocks normalization and denormalization are added to the controller. The most important element is the base of rules which implements the relationship between the input and the output of the system. The rule-base houses a collection of IF–THEN rules summarizing the knowledge base that underpins the decisions made by the fuzzy controller.

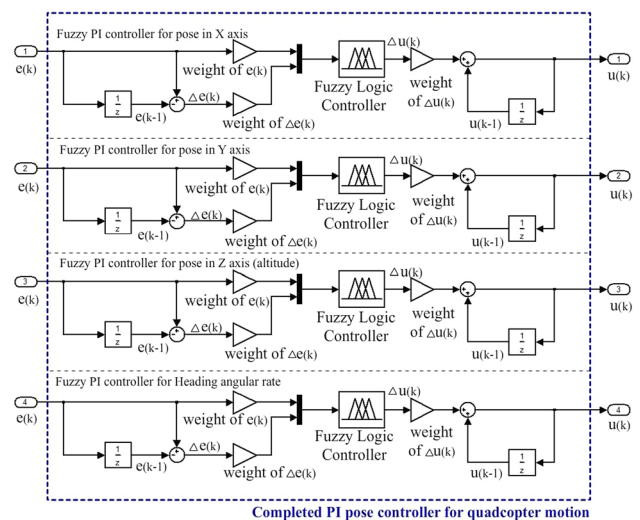


Fig. 7 Fuzzy control diagram with Mamdani’s interference method and standard PI structure

In our study, a set of linguistic rules illustrated in Fig. 7 was obtained from previously conducted experiments and the experience of human pilots.

The following value range is used in the tables:

- N2—Large Negative
- N1—Small Negative
- Z—Zero
- P1—Small Positive
- P2—Large Positive

Table 1 shows the set of rules for the X, Y axes plus heading control, while the second table provide rules for altitude control—Z-axis.

The main problem in quadcopter control is the fact that the propellers are unable to generate negative thrust, and this places significant limits on altitude control. In a real flight,

Table 1 Set of linguistic rules for X, Y and heading controller with standard symmetric distribution

		e -regulation deviation				
		N2	N1	Z	P1	P2
Δe change of reg. deviation	N2	N2	N2	N2	Z	Z
	N1	N2	N2	N1	Z	P1
	Z	N2	N2	Z	P1	P2
	P1	N1	Z	P1	P1	P2
	P2	Z	Z	P2	P2	P2

Table 2 Set of linguistic rules for Z controller with non-symmetric distribution

		e -regulation deviation				
		N2	N1	Z	P1	P2
Δe change of reg. deviation	N2	N2	N2	N2	N2	N1
	N1	N2	N2	N2	N1	Z
	Z	N2	N1	Z	P1	P1
	P1	N1	Z	Z	P1	P1
	P2	Z	Z	P1	P2	P2

it means that in order to stop the quadcopter at a specific height during takeoff, it is necessary to reduce actual thrust just before the desired altitude is reached. This limitation affects the set of rules for the altitude— Z -axis controller as shown in Table 2.

In order to convert the linguistic rules to real output values, standard triangular membership functions were used and these are listed in Fig. 8. Membership functions are defined for each input ($e, \Delta e$) and output (Δu) values by five clusters.

The variables of the inference system were represented as membership functions with each input normalized to the range of $\langle -1, 1 \rangle$. Defuzzification was based on the singleton weight centers method due to the short processing time required.

As is shown in the experiments section, our fuzzy controller was capable of controlling the quadcopter in fully autonomous mode.

5 Experiments and results

In order to analyze properties of the proposed system, three different experiments were conducted.

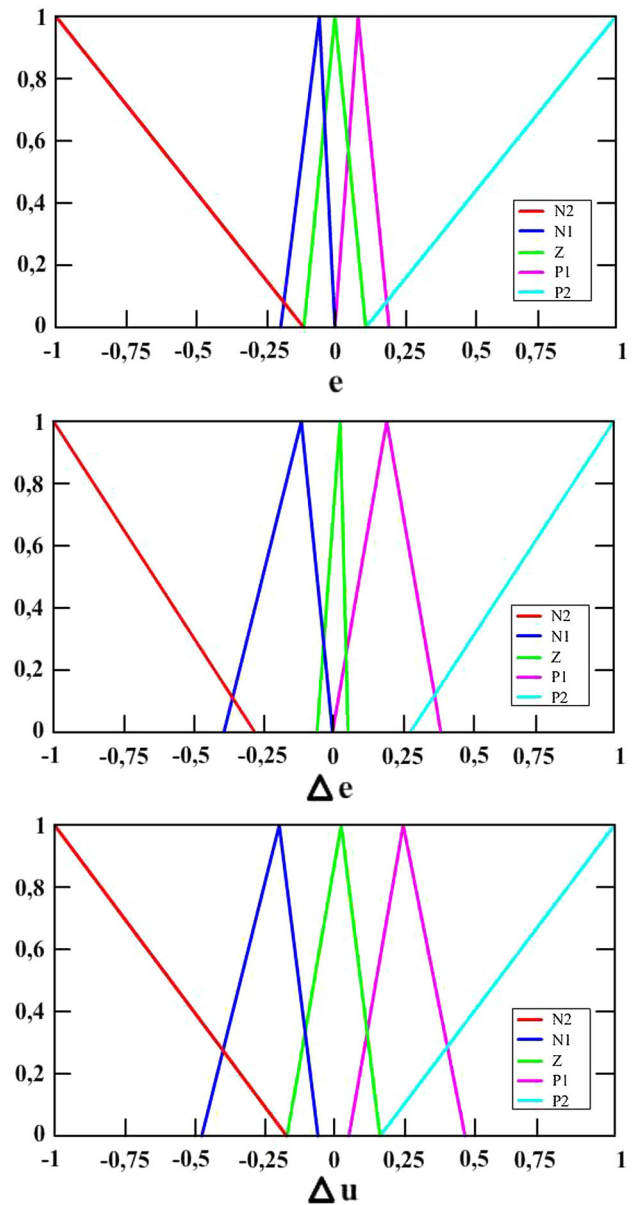


Fig. 8 Triangular membership functions of linguistic variables— $e, \Delta e, \Delta u$ are defined by five clusters

The goal of the first measurement was to estimate the overall accuracy of the mapping algorithm. In order to guarantee the precision of our measurements, we printed our testing track in real-world scale on a 4 m long roll of paper. The seven markers printed on the sheet, each of which are 135 mm wide, were placed 585mm apart. Three different cameras (industrial BlueFox MLC200, Parrot AR Drone front HD camera and wide-lens Genius F100, each properly calibrated) were used to collect data. During this measurement, the cameras were manually guided along the trajectory while estimating the position of each new marker with respect to the previous one. This difference between real and estimated positions

Table 3 Measurement of pose estimation accuracy—real vs. estimated pose

	Marker Axis	Marker							SD (cm)
		1	2	3	4	5	6	7	
Real position	<i>x</i>	0	0.5850	1.1700	1.7550	2.3400	2.9250	3.5100	
	<i>y</i>	0	0	0	0	0	0	0	
Bluefox MLC200W	<i>x</i>	0	0.5884	1.1837	1.7698	2.3624	2.9539	3.5389	1.80
	<i>y</i>	0	−0.0078	−0.0157	−0.0181	−0.0123	−0.0189	−0.0199	1.54
A.R.Drone front camera	<i>x</i>	0	0.6254	1.2230	1.8112	2.4054	3.0008	3.5881	6.14
	<i>y</i>	0	0.0107	0.0083	0.0096	0.0114	0.0172	0.0227	1.33
Genius F100	<i>x</i>	0	0.6175	1.2331	1.8397	2.4482	3.0490	3.6647	9.45
	<i>y</i>	0	−0.0049	−0.0165	−0.0186	−0.0277	−0.0242	−0.0494	2.35

was essential for evaluating the accuracy of the system. The results obtained with all three devices are shown in Table 3.

The most accurate device according to the results listed in Table 2 was the industrial Bluefox MLC200W camera which demonstrated an error of only 1.80 cm in the *x*-axis. This low level of absolute error is a result of the small distortion of the high-quality 2.4 mm lens in combination with proper calibration using a large scale calibration grid. The lower accuracy of the results obtained from the AR Parrot HD camera (6.14 cm) and the wide-lens Genius F100 Parrot (9.45 cm) shown in Fig. 9 can likely be attributed to the low-cost nature of the optics used in these cameras, however this combination would still be acceptable for various robotics navigation applications.

The goal of the second experiment was to measure the accuracy of the overall localization. For the purposes of this experiment, 32 markers were distributed over the floor in random positions. A Parrot AR drone was equipped with an additional 360° reflector and its front camera was tilted down by 45°. The measurement was performed in an underground parking lot in order to simulate real-world conditions. During this experiment, the quadcopter was manually controlled to fly over the area of markers while two different sources were used to estimate its position.

The first source was the camera stream in conjunction with the proposed algorithm. The algorithm was run on the ground station and generated a map of markers according to the incoming images while simultaneously localizing the Parrot in space and capturing the trajectory of the Parrot’s actual flight. When several markers were visible within a frame, the current location was calculated with respect to the position of the marker closest to it. The second source used to determine the ground truth position of the quadcopter was a Leica laser station (Fig. 10). The laser station provided ground truth data since it was able to pinpoint the location of the 360° reflector attached to Parrot chassis approximately 10 times per second with an error margin of less than 1 mm.

In order to compare the precision of the proposed method, the trajectories acquired during real flight were aligned using

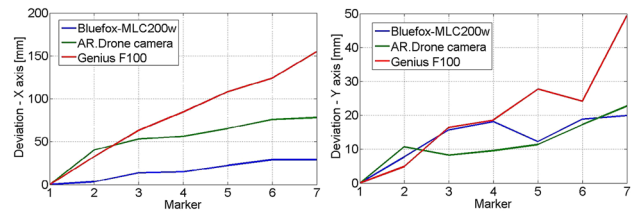


Fig. 9 Absolute error of marker pose estimation for three different cameras



Fig. 10 Measurement of localization accuracy

standard ROS tools. As is shown in the axis decomposition in Fig. 11, even though several measurement disturbances were detected, the precision of our localization system was still convincingly accurate—the average overall error within this experiment was less than 7 cm. Additional filtering of estimated values might improve the overall precision even further by removing the noise and measurement disturbances causing plot spikes. We assume that their occurrence and size is closely related to the camera chip size and lens distortion. When using a standard 4:3 camera format, the measurement error of the marker detected at the very right edge of the frame was greater than the error of the marker detected at the very top of the frame. The reason for this is the level of distortion

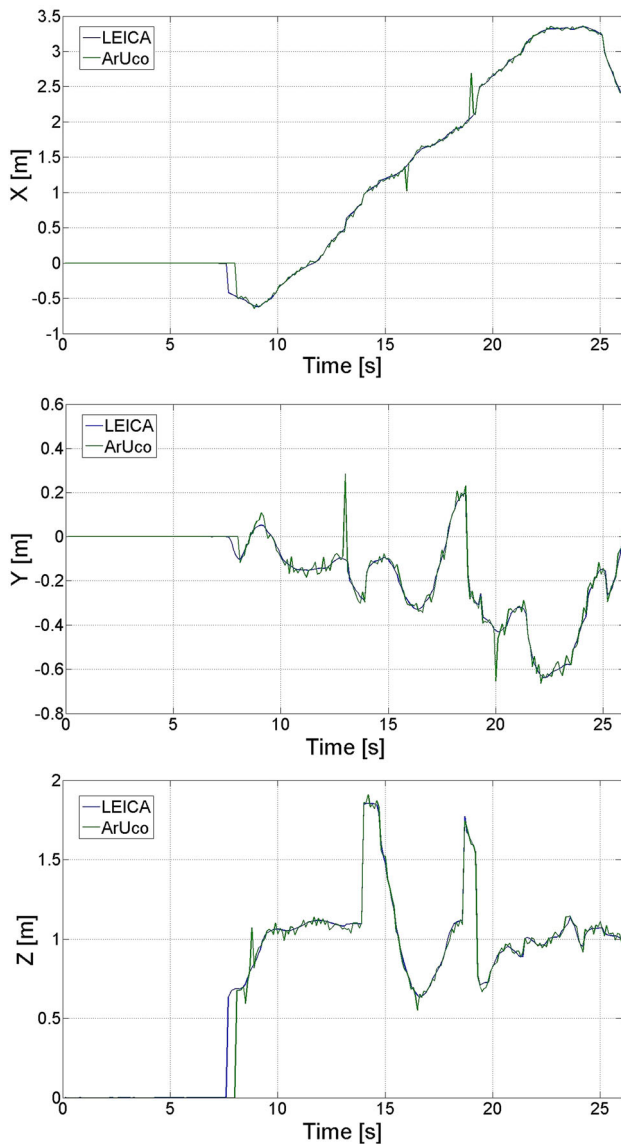


Fig. 11 Laser Station data compared with ArUco estimation, an error was within 5–7 cm range

which is closely coupled with the absolute distance from the optical center axis.

Figure 12 combines data from previous plots and provides a comparison of estimated and ground truth trajectory in 3D space.

Our final experiment (Fig. 13) combined both ArUco-based localization and fuzzy control into the operation of a fully autonomous flying quadrocopter. In order to test this system, two different trajectories, rectangular and triangular, were created in an empty warehouse. Markers were arranged in ascending order with the exactly determined orientation the x -axis of every n th marker pointed to the center of the $n + 1$ th marker in order to provide direction for the off-board processing system.

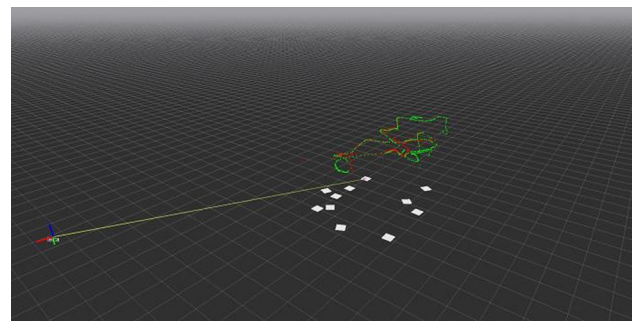


Fig. 12 Aligned data from laser tracker station (red color) and ArUco-based localization (green color) (color figure online)

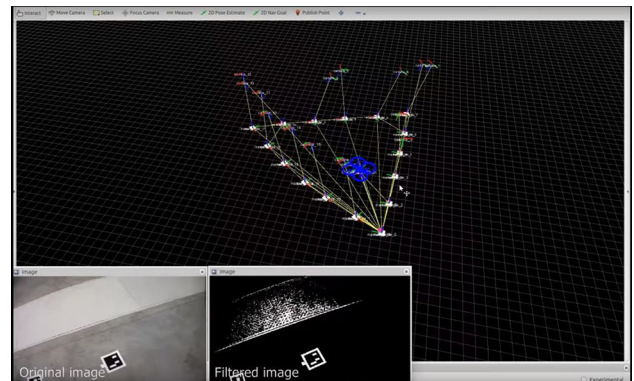


Fig. 13 Final experiment—Parrot autonomously following triangle trajectory defined by ArUco markers

As in the previous experiment, the front Parrot HD camera was tilted down by 45° and the video stream was transferred to the ground station, where marker positions as well as Parrot's actual position were determined in near real time. In addition to the pure localization performed in the previous experiment, the system was in fully autonomous operation in this situation. The fuzzy controller calculated steering commands in accordance with the feedback from the ArUco-based localization. The Parrot was guided by controlling program to approach currently visible marker with the highest ID. However, due to a combination of mutual marker distances, height of flight and camera field of view, the next marker with higher ID was detected before the closer one was actually approached—a new goal was set before the previous was reached, resulting in the continuous forward movement seen in the final video.

6 Conclusion

In this study, a marker-based system for the autonomous flying of quadrocopter is presented. It was inspired by recent progress in the field of marker-based localization and mapping and enables simultaneous localization and navigation

of the MAV in unstructured, GPS-denied environments using only a single onboard camera and augmented reality markers. Our implementation which also available as a standard ROS package called *ArUco_mapping* is capable of creating both 2D and 3D maps of detected markers in a real-time, while simultaneously estimating the position of the MAV with respect to a coordinate system defined by the first detected marker. In order to demonstrate the practicality and robustness of marker-based navigation even for low-cost hardware platforms, we developed our system further using fuzzy control and achieved a fully autonomous flight. The average localization error in optimal conditions was less than 2 cm with the manually guided camera and less than 7 cm for the onboard camera during a real flight.

By adding fuzzy logic for MAV control and utilizing a mapping approach, we achieved a fully autonomous flight in which the quadcopter was able to navigate visually and to follow a predefined trajectory while estimating its own absolute position in a world reference frame.

As a suggestion for further study, the precision of the mapping approach could be improved even further by integrating closing-loop detection algorithms which have been recently developed for keypoint-based SLAM methods. The stability of the quadrotor in flight could also be refined by examining the flight data using Lyapunov function. In order to reach faster system response and more accurate control in flight, a nonlinear estimation based on the mathematical model described in the fuzzy control section could also be added to the existing system.

Acknowledgements The authors wish to thank the project VEGA 459 1/0464/15 for its support. This work was supported by the Slovak Research and Development Agency under the contract No. APVV-15-0750.

References

- Bry A et al (2015) Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments. The International Journal of Robotics Research. 34(7):969–1002. doi:[10.1177/0278364914558129](https://doi.org/10.1177/0278364914558129)
- Hehn M, Raffaello D'A (2012) Real-time trajectory generation for interception maneuvers with quadcopters. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 7. - 12.10.2012; Vilamoura. IEEE; 2012. p. 4979–4984. DOI:[10.1109/IROS.2012.6386093](https://doi.org/10.1109/IROS.2012.6386093)
- M. Q. Lindsey M Q, Mellinger D, Kumar V (2012) Construction of cubic structures with quadrotor teams. In: Proceedings of Robotics: Science and Systems (RSS); Los Angeles, USA. MIT Press; 2012. p. 177–184
- Grzonka S, Grisetti G, Burgard W (2009) Towards a navigation system for autonomous indoor flying. In: IEEE International Conference on Robotics and Automation (ICRA); 12. - 17.5.2009; Kobe. IEEE; 2009. p. 2878–2883. DOI:[10.1109/ROBOT.2009.5152446](https://doi.org/10.1109/ROBOT.2009.5152446)
- Ascending technologies [Internet]. 2016 . Available from: <http://www.asctec.de/>
- Krishnakumar R, Rasheed AM, Kumar KS, >> """, October 21, (2015) Enhanced hover control of quad tilt frame UAV under windy conditions. International Journal of Advanced Robotic Systems. 2015:1–14. doi:[10.5772/61231](https://doi.org/10.5772/61231)
- Krajník T, Vonásek V, Fišer D, Faigl J (2011) AR-Drone as a Platform for Robotic Research and Education. Communications in Computer and Information Science. 161:172–186
- Rullan-Lara JL, Sanahuja G, Lozano R, Salazar S, Garcia-Hernandez R, Ruz-Hernandez JA (2013) Indoor Localization of a Quadrotor Based on WSN: A Real-Time Application. International Journal of Advanced Robotic Systems. 10:48. doi:[10.5772/53748](https://doi.org/10.5772/53748)
- Kim SJ, Kim BK (2012) Dynamic Ultrasonic Hybrid Localization System for Indoor Mobile Robots. IEEE Transactions on Industrial Electronics. 60(20):4562–4573. doi:[10.1109/TIE.2012.2216235](https://doi.org/10.1109/TIE.2012.2216235)
- Droeschel D, Schreiber M, Behnke S (2013) “.” UAV-g 8/2013. (2013). Omnidirectional perception for Lightweight UAVs using a continuously rotating 3D laser scanner. In: UAV-g 8/2013; 4. – 6.9. 2013; Rostock, Germany. The international archives of the photogrammetry, remote sensing and spatial information sciences, XL-1 W 2; 2013. p. 107–112. DOI:[10.5194/isprsarchives-XL-1-W2-107-2013](https://doi.org/10.5194/isprsarchives-XL-1-W2-107-2013)
- Engel J, Sturm J, Cremers D (2012) Accurate figure flying with a quadcopter using onboard visual and inertial sensing. In: Workshop on Visual Control of Mobile Robots /ViCoMoR at the IEEE/RJS International Conference of Intelligent Robots and Systems (IROS); 7.-12.10.2012; Vilamoura, Portugal. IEEE; 2012
- Zingg S, Scaramuzza D, Weiss S, Siegwart R (2010) MAV navigation through indoor corridors using optical flow. In: Robotics and Automation; 3-7 May 2010; Anchorage, AK. 2010. p. 3361 - 3368. DOI:[10.1109/ROBOT.2010.5509777](https://doi.org/10.1109/ROBOT.2010.5509777)
- Eberli D et al (2011) Vision based position control for MAVs using one single circular landmark. Journal of Intelligent & Robotic Systems. 61(1):495–512. doi:[10.1007/s10846-010-9494-8](https://doi.org/10.1007/s10846-010-9494-8)
- Lamberti F et al (2013) Mixed marker-based/marker-less visual odometry system for mobile robots. International Journal of Advanced Robotic Systems. 10(1):1–11. doi:[10.5772/56577](https://doi.org/10.5772/56577)
- Hartmann P, Ben C, Moormann D (2014) Navigation for Vertical Precision Landing Based on Optical Tracking of a Spatial Retroreflective Marker Array. In: 29th Congress of the International Council of the Aeronautical Sciences; 7-12 September; St. Petersburg, Russia. 2014. p. pp. 1-8
- Muñoz-Salinas, Rafael, et al (2016) “Mapping and Localization from Planar Markers.” arXiv preprint [arXiv:1606.00151](https://arxiv.org/abs/1606.00151)
- Garrido-Jurado S et al (2014) Automatic generation and detection of highly reliable fiducial markers under occlusion. Journal Pattern Recognition. 47(6):2280–2292. doi:[10.1016/j.patcog.2014.01.005](https://doi.org/10.1016/j.patcog.2014.01.005)
- Kendoul F (2012) Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. Journal of Field Robotics. 29(2):315–378. doi:[10.1002/rob.20414](https://doi.org/10.1002/rob.20414)
- Derafa L, Madani T, Benallegue A (2006) Dynamic Modelling and Experimental Identification of Four Rotors Helicopter Parameters. In: Proceedings of the IEEE International Conference on Industrial Technology; 15.-17.12.2006; Mumbai, India. IEEE; 2006. p. 1834–1839. DOI:[10.1109/ICIT.2006.372515](https://doi.org/10.1109/ICIT.2006.372515)
- OstafewCh J, Schoellig AP, Barfoot TD (2014) Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments. In: IEEE International Conference on Robotics and Automation (ICRA); 31.5. - 7.6.2014; Hong Kong. IEEE; 2014. p. 4029–4036. DOI:[10.1109/ICRA.2014.6907444](https://doi.org/10.1109/ICRA.2014.6907444)
- Gautam D, Ha Ch (2013) Control of a quadrotor using a smart self-tuning fuzzy PID controller. International Journal of Advanced Robotic Systems. 10:1–9. doi:[10.5772/5691](https://doi.org/10.5772/5691)

22. Poundsa P, Mahonyb R, Corkec P (2010) Modelling and control of a large quadrotor robot. *Control Engineering Practice*. 18(7):691–699. doi:[10.1016/j.conengprac.2010.02.008](https://doi.org/10.1016/j.conengprac.2010.02.008)
23. Mamdani EH, Assilian S (1975) An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*. 7(1):1–13
24. Szlachetko B, Lower M, Nguyen NT, Hoang K, Jędrzejowicz P (2012) On Quadrotor Navigation Using Fuzzy Logic Regulators, Computational Collective Intelligence. Technologies and Applications: 4th International Conference, ICCCI 2012, Ho Chi Minh City, Vietnam, November 28–30, 2012. Proceedings, Part I
25. Indrawati V, Prayitno A, Kusuma T A (2015) Waypoint Navigation of AR.Drone Quadrotor Using Fuzzy Logic Controller, TELKOMNIKA, Vol.13, No.3, September 2015, pp. 930 ~ 939