Proposal of a Visual Positioning Architecture for Master-Slave Autonomous UAV Applications

Lucas Coradin Rech[1(✉)], Luciano Bonzatto Junior[1], Guido S. Berger[3,4], José Lima[3,4], Alvaro Rogério Cantieri[2], and Marco Aurélio Wehrmeister[1,5]

[1] Computer Systems Engineering Laboratory (LESC), Federal University of Technology-Campus Curitiba, Curitiba, Paraná 80230-901, Brazil
{lucasrech,lucjun,}@alunos.utfpr.edu.br, wehrmeister@utfpr.edu.br
[2] Applied Robotics and Computation Laboratory - LaRCA- Federal Institute of Paraná, Paraná, Pinhais, Brazil
alvaro.cantieri@ifpr.edu.br
[3] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
{guido.berger,jllima}@ipb.pt
[4] Laboratório para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
[5] University of Münster, Münster, Germany
wehrmeister@uni-muenster.de

Abstract. Autonomous UAVs offer advantages in industrial, agriculture, environment inspection, and logistics applications. Sometimes the use of cooperative UAVs is important to solve specific demands or achieve productivity gain in these applications. An important technical challenge is the precise positioning between two or more UAVs in a cooperative task flight. Some techniques provide solutions, like the GNSS positioning, visual and LIDAR slam, and computer vision intelligent algorithms, but all these techniques present limitations that must be solved to work properly in specific environments. The proposal of new cooperative position methods is important to face these challenges. The present work proposes an evaluation of a visual relative positioning architecture between two small UAV multi-rotor aircraft working in a master-slave operation, based on an Augmented Reality tag tool. The simulation results obtained absolute error measurements lower than 0.2 cm mean and 0.01 standard deviation for X, Y and Z directions. Yaw measurements presented an absolute error lower than 0.5 °C with a 0.02–5 °C standard deviation. The real-world experiments executing autonomous flight with the slave UAV commanded by the master UAV achieved success in 8 of 10 experiment rounds, proving that the proposed architecture is a good approach to building cooperative master-slave UAV applications.

Keywords: Autonomous cooperative UAV · Visual UAV positioning · AR Tag UAV positioning · Master-slave UAV architecture

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
D. Tardioli et al. (Eds.): ROBOT 2022, LNNS 590, pp. 365–375, 2023.
https://doi.org/10.1007/978-3-031-21062-4_30

1 Introduction

Small size autonomous Unmanned Aerial Vehicles (UAVs) had an expressive increase in number and types of practical applications nowadays due to the decrease of this aircraft costs and the offer of new small size hardware and sensors proper to embed in these vehicles. The autonomous control of small UAVs presents technical challenges, like secure flight and collision avoidance, path planning, precise 3D positioning and robust control.

Collaborative arrangements between multiple UAVs are an excellent alternative for some applications. The division of tasks between several aerial vehicles reduces the task duration. Consequently, it increases the efficiency of the process, allowing to solve technical issues such as the reduced flight time of multi-rotor vehicles, for example. The literature presents works in agriculture [1], inspection of energy power plants [2], 3D mapping of an external area [3], among other.

Precise positioning is vital to assure the security of the operation in autonomous multi-UAV tasks, to ensure the correct path following. Nowadays, the most common techniques to achieve precise positioning in UAV flights is Differential Global Navigation Satellite System (GNSS). This technique is an optimum approach to solve the positioning problem, but GNSS systems demand a proper satellite signal receiving to work properly, which is sometimes hard to achieve[4].

Visual positioning techniques provide position data to UAV flights and operations. The standard approaches use monocular, binocular and omni-directional RGB cameras and stereoscopic cameras [5]. The application computer vision techniques to calculate the UAV coordinates is demanding, especially in not-structured environments. This approach presents challenges, like the sensibility of environment light variations, visual occlusion of the target, image capture noise and distortion due to the camera vibration and twist, among others.

A possible approach to decrease the computational demanding is to create auxiliary image marks to provide specific visual data for the correct 3D position estimation. Augmented reality tag positioning, proposed originally for Augmented Reality applications, has been used for this kind of application. April Tags [6] and AR Track Alvar [7] are example of these tools. The present work proposes a relative positioning architecture between two UAVs working in a master-slave cooperative arrangement. An Augmented Reality Tag tool, among a regular RGB camera embedded in the master UAV, is used to provide the relative position between the aircraft. This tag provides the quick and effective identification of each vehicle and calculates its position in the experimental environment.

The experiments are conducted in an indoor laboratory using two small aircraft. This investigation aims to provide a first proof-of-concept of the proposed architecture, allowing the proposal of future practical cooperative applications based on this master-slave architecture. The Sect. 2 reviews the use of visual tags in cooperative autonomous vehicle applications. Section 3 describes the proposed architecture and components. Section 4 presents the experimental set and obtained results. Finally, the Sects. 5 and 6 sections discusses the work results and describe conclusions and future works.

2 Related Works

The computational vision process used to calculate the relative position between two or more cooperative autonomous vehicles is a complex task which commonly demands a lot of hardware capacity. The application of printed tags to provide high visual quality information to the vision algorithms is an excellent choice to decrease the computational demand. The literature presents some investigations that use this kind of tag to provide position data to control a UAV flight.

The paper [8] proposes a UAV vision-based target positioning solution. The work proposes a position control algorithm to follow a ground target represented by a tag. The work presents a 92% reliability in real-world experiments, showing the viability of using this kind of proposal for UAV outdoor positioning.

An indoor AprilTag markers UAV localization is proposed in [9]. A set of markers are fixed in the experimental area walls providing visual clues captured by the UAV frontal camera. An off-board algorithm extracts the relative position between the aircraft and each visible tag and sends it to the aircraft control algorithm. Besides the small size of the flight area, this confirms the reliability of using the tags to provide 3D position data to a multi-rotor UAV flight.

Another similar approach is presented in the work [10]. The proposition uses an AR.Drone 2.0 autonomous controlled by a base station that processes the images of an AprilTag captured by the UAV camera. The experiments present stable aircraft control using the position data processed from the tags.

The work [11] presents an AR-Drone autonomous search algorithm. In this work, the UAV captures the images of a printed tag held in front of it by a human. These images are off-board processed to provide the relative position between the UAV and the tag, allowing the aircraft to following it using a PID flight control.

Using visual tags to provide relative position between autonomous cooperative vehicles is a possible application of this technique. Works present in the literature propose using these tools to allow autonomous landing and cooperative navigation between UGVs and UAVs in most cases.

The work [12] proposes an outdoor UAV navigation system where the aircraft follows a moving automobile and executes an autonomous landing. A group of visual tags are placed, allowing the correct positioning of the landing even when only using the visual position information provided by the tag.

A navigation and route generation system for land vehicles based on the vision obtained by a drone in cooperative work is presented in [13]. The UAV serves as a leader in the displacement. It sends position information related to the ground robot through a TAG fixed to the top of this vehicle. The tag images are captured by a down pointing camera installed in the UAV. This arrangement allows the ground vehicle to follow an obstacle-free path previously determined by the ground vehicle through the vision system.

This short review provides a basis for the present work proposition: the use of a computer vision algorithm based on AR-Tag to allow cooperative navigation between two UAVs. To the best of the authors' knowledge, no similar proposal is present in the literature. The following sections describe the architecture proposal, and experimental results are presented.

3 System Architecture and Components

The positioning architecture is based on visual feedback by capturing images of the slave UAV obtained by the master UAV camera. The experiments use two small drones, a DJI Tello model as a slave and a Bebop Drone model as a master. Drones connect to a base station composed of an Intel Core i7 processor with 16 GB RAM and an Intel® HD Graphics 520 (Skylake GT2) board computer running Ubuntu 16.4 LTS and ROS Kinect. The base station runs the ROS packages to provide communication and control interfaces for both aircraft.

All the architecture components use the Robot Operating System (ROS) to exchange information. [14]. The Bebop drone is connected through the Bebop Autonomy package [15], which allows the base station to read the UAV camera and publish *cmd_vel* commands to control the UAV flight. Bebop drone is an excellent choice to implement this architecture because it is a small size drone with stabilized flight performance and an embedded Full-HD resolution camera. The camera gimbal stabilization minimizes the image shifts during the data acquisition. DJI Tello drone offers embedded flight stabilization control. The small size of this aircraft allows the flight in reduced spaces. A *tello_driver* ROS package implements the Tello drone communication with the ground station. The Ar-Track Alvar Augmented Reality SDK [7] is used to implement the visual position reference system. The Ar Track Alvar package developed by Scott Niekum [15] provides ROS compatibility. The Images of the AR-TAG are captured by the Bebop drone camera and processed by this package that publishes position and orientation data on *ar_pose_marker* a Ros Node.

Both aircraft work in a hot-spot Wi-Fi connection architecture. The connection between the base station and the Bebop Drone works at a frequency of 5.8 GHz using a USB Wi-Fi adapter, while the connection with the Tello Drone works at 2.4 GHz using the PC onboard Wi-Fi. The base station concentrates all the ROS messages and sends commands to the agents to execute their activities.

The Bebop camera is set to point to the ground all the time to keep the perfect visual capture of the tag. Bebop can point it's camera inside a limit of a 75-degree angle so that it will provide a capture in its frontal area. The slave UAV must keep its flight position inside this area during all operation to allow the perfect control. Figure 1 a) shows the architecture coordinate reference system. A C++ code implements a PID control of the Tello drone flight's vertical, horizontal and yaw positions, publishing in two ROS nodes called */position* and */orientation*. The experiments using the Bebop drone as master are performed using manual piloting implemented by C++ code running in the base station. The code captures the commands from the computer keyboard and sends *cmd_vel* messages to the aircraft through the Bebop Autonomy ROS driver. Manual control is chosen in this case to assure the security flight of the aircraft in the reduced space present in the laboratory. Figure 1 b) shows the logical diagram of the proposed architecture and its components.

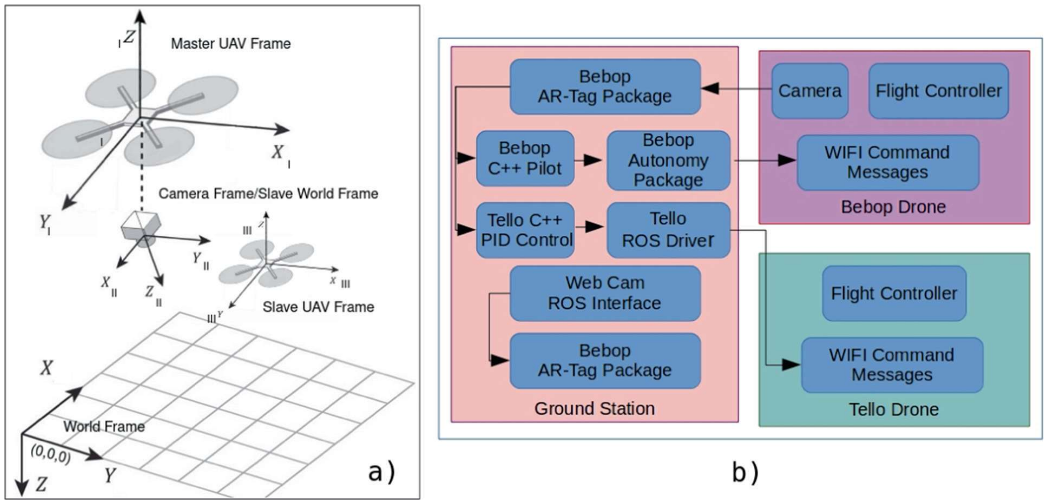

Fig. 1. a) Coordinate system representation; b) Architecture block diagram

4 Experiments Description and Results

The experimental planning aims to provide the first proof-of-concept for the master-slave following the leader's proposed architecture in a laboratory environment. This step is essential to evaluate the reliability of the technique and allow complex cooperative applications using the UAVs in a real-world environment. Three-step experimental planning is described in this section. First, a simulation of the master-slave operation using the visual control based on the AR-Tags is presented.

After that, two different real-world experimental arrangements are tested. First, a fixed position Web Cam captures the Tello Drone tag images during a flight and commands the aircraft using the tag position data and a C++ PID control code. After that, a cooperative master-slave flight where the Tello Drone autonomous follows the Bebop Drone is executed.

4.1 Simulation Experiments

The simulated environment uses the Copellia Sim [16] simulation software to evaluate the proposed UAV autonomous cooperative performance. Two multirotor models running a stabilization PID control LUA script are used in the experiments, allowing to the UAV's keep hovering at a static position and displace from one point to the other in the space without a crash. This stabilization code works as a flight controller hardware of a real-world UAV.

An image sensor is fixed at the bottom of the master UAV and captures the AR Tag images sent to the package, which processes the tag position calculation. Looking to avoid the camera shift during the UAV displacements, a virtual gimbal is programmed to control the camera orientation and keep it constantly pointing to the ground. The AR-Tag Alvar package publishes 6 DOF position and orientation information in */visualization_marker* ROS topic. A ROS master provides the node communication between the simulation software, the AR-Tag package and the external C++ position control code.

A programmed path point are sent to the slave UAV, that displace using the master UAV to provide the 3D position data. The absolute positioning error of the tag measurements and the UAV 3D position are evaluated. Ten experimental rounds are executed for three different paths, a square, a triangle and a cross. The absolute position error for each point is recorded for statistical calculations. Figures 2 a) and b) shows the measurements of the tracking error calculated from the AR-Tag data and the UAV world-frame position for a flight. Figure 3 shows the path followed by each UAV in space and the plot of the absolute error and AR-Tag error for a flight. Table 1 presents the statistic calculations of the absolute error for all the experiment rounds.

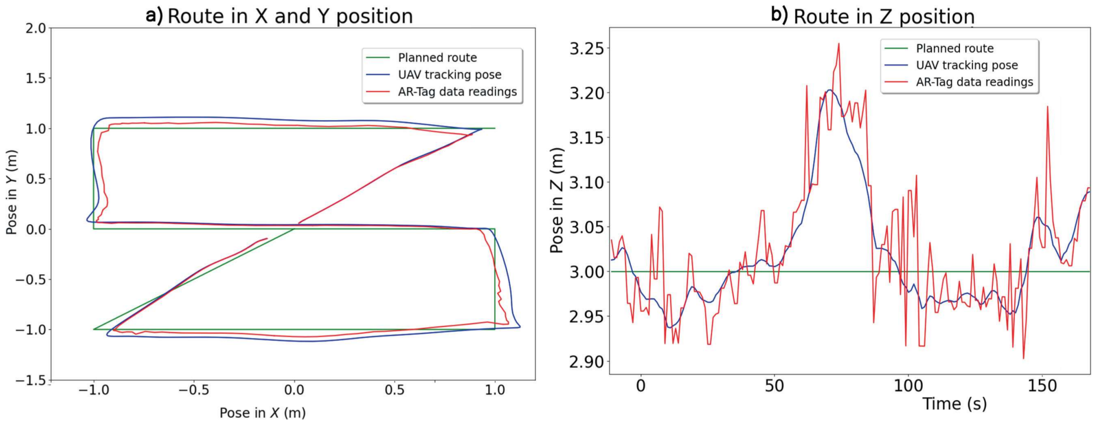

Fig. 2. a) Tracking of UAV position in X and Y axis of a flight master-slave simulation; b) Tracking of UAV height of a flight master-slave simulation

4.2 Real World Experiments

The second step is to evaluate the capacity of the real-world architecture to provide autonomous flight capability in the master-slave arrangement. The laboratory where the experiments are conducted is not equipped with a precise positioning reference system to provide ground truth for the position error measurements. An indirect error measurement is used to evaluate the experimental data based on the AR-Tag position measurements, using the (0,0,0) point of the camera frame as coordinate origins. The relative error between the AR-Tag

Table 1. Mean and Stardard Deviation for X;Y;Z ;Yaw tag measurements. These points refer to the camera frame reference system.

Set Point cm [X,Y,Z,Yaw]	X_Mean	Y_Mean	Z_Mean	Yaw Mean	X_Std	Y_Std	Z_Std	Yaw Std
0;0;230;0	0.45	0.32	0,29	2.45	0.043	0.051	0.058	1.25
60;35;230;0	0.65	0.39	0.28	3.12	0.033	0.040	0.055	1.44
60;-35;230;0	0.60	0.41	0.28	3.10	0.051	0.049	0.067	0.99
35;60;230;0	0.68	0.38	0.33	2.88	0.048	0.054	0.052	2.11
-35;60;230;0	0.71	0.35	0.38	2.75	0.049	0.066	0.050	1.31

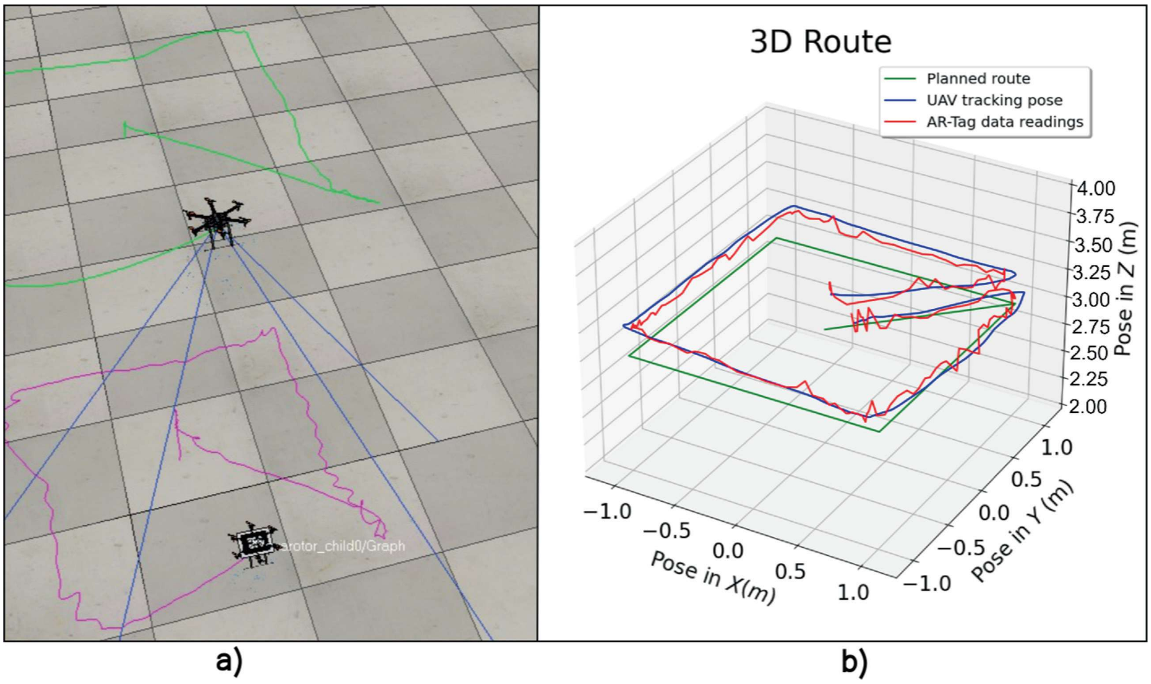

Fig. 3. 3D UAVs route and plot of tracking of UAV position in a follow-the leader flight. a) Image of the simulated environment. b) Plot of the track error.

(X,Y,Z) readings and the center of the camera frame is presented in the experiment graphs. A video of the real-world experiments is available at https://youtu.be/jDQYng34abY.

Autonomous Slave UAV Flight with an Fix Camera Reference. The first experiment evaluates if the tag-based positioning system can provide reliable data to the UAV displacement control algorithm in a controlled environment. A Webcam HD Logitech c920–1080p resolution fixed in the laboratory ceiling provides the AR-Tag capture and sends the data to the base station computer. The camera is set at an approximately 3.2 m height and provides a visual area of 2,5 $\times$ 2,0 m. These readings provide the UAV (X,Y,X,Yaw) data.

The Tello UAV with a 6.0 $\times$ 6.0 size tag fixed on its roof is commanded by the PID control code running in the base station, using the tello_driver package to command the aircraft to follow a square route in the camera vision area. The flight is executed at a 1.0 m height in 10 rounds, and the AR-Tag position data is collected and stored to allow the position graph plot. Figure 4 shows the plot of planned route *versus* the AR-Tag position collected form three flight rounds.

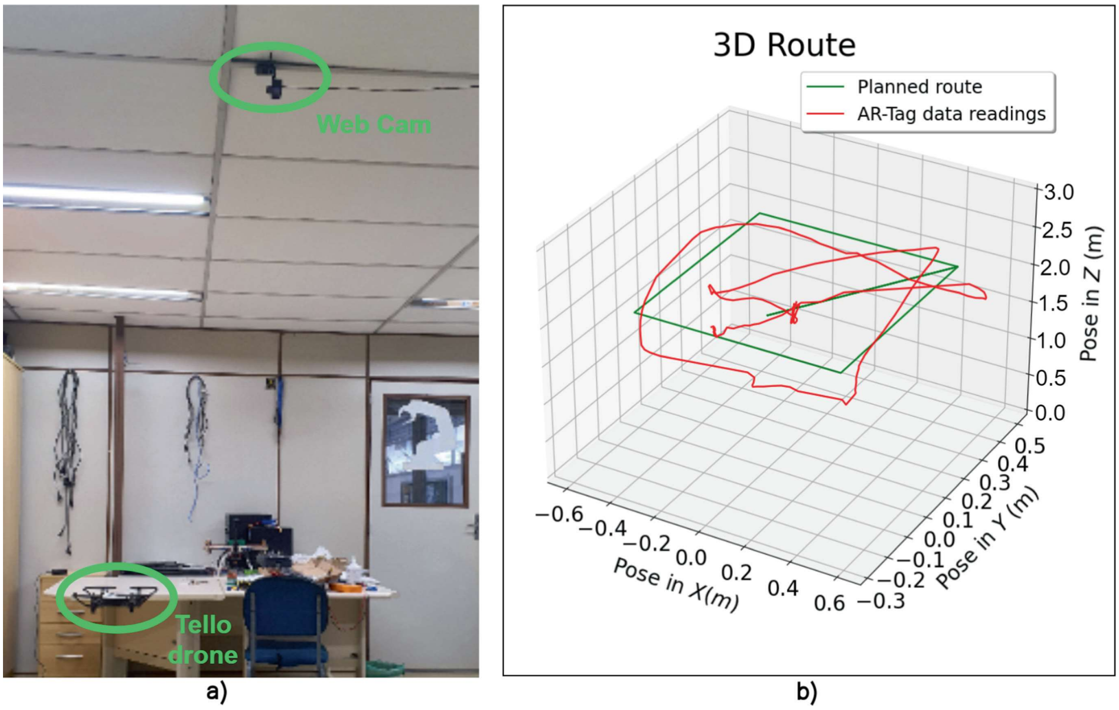

Fig. 4. The Tello UAV autonomous flight plot with the fixed WebCam generating position data. a) Image of the experiment arrangement. b) Plot of the track error.

This experiment indicates that the control algorithm based in the AR-Tag position measurements works properly, so the next step is to evaluate the master/slave UAV arrangement.

Autonomous Master-Slave UAV Follow-the-Leader Flight. The follow-the-leader experiment is based on the Bebop Drone working as the master and capturing images from the tag fixed in the Tello Drone, the slave. The control algorithm is programmed to lead the slave UAV to reach the (0.00, 0.15, –1.00) meter from the center of the image frame captured by the Bebop Drone. Bebop drone is manually commanded to displace in the flight area, and the Tello drone follows it continuously. Experiments is executed in an approximately 2.0-minute flight, at a maxim horizontal velocity of approximately 0.1 m/s. Figure 5 presents a plot of the UAV (X,Y,X) measurements for a flight round.

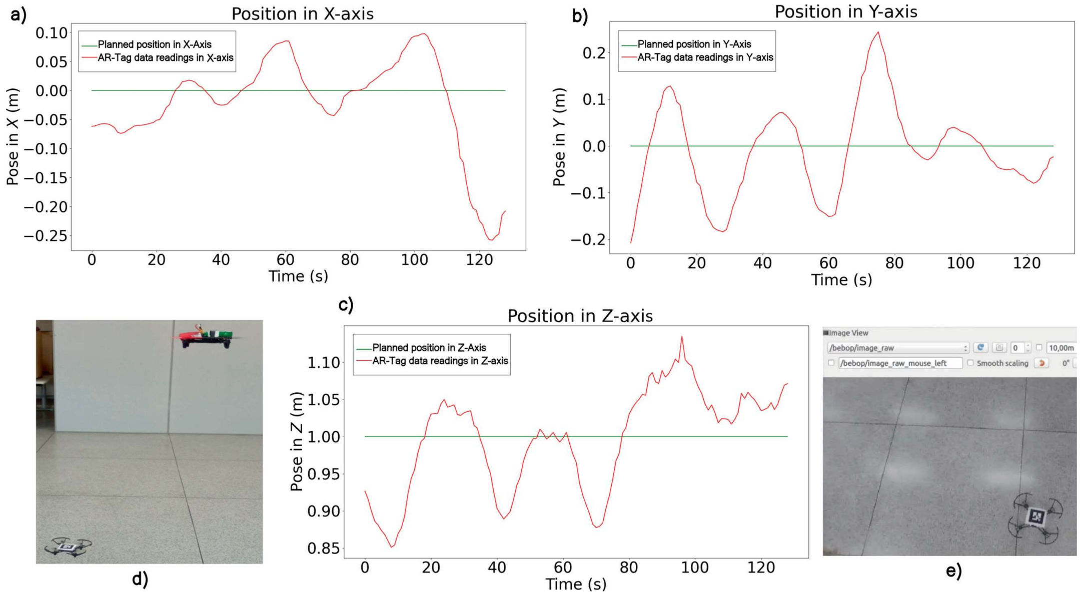

Fig. 5. Plots of the slave UAV position in a following autonomous operation and images of the experiment arrangement. a), b) and c) Plot of the actual *versus* planned pose readings. d) and e) Images of the Bebop drone and Tello drone cooperative flight.

5 Discussion

The experiments indicate that the proposed architecture is a reliable way to create a visual master-slave collaborative UAV architecture. The results presented small absolute error measurements in simulation. In real-world experiments, environment and hardware conditions impact the control stability, mainly in the master-slave flights, due to the Bebop drone image capture limitations.

The camera quality captures are a limitation for this schema. In the first case, the Web Cam offers constraints like slow brightness correction time, low data transfer to the PC due to the 2.0 USB connection and no auto-focus mechanism. These limitations impact the frame captures and the AR-Tag processing, so it is possible to evaluate the use of professional cameras to increase the data position quality and the robustness of the positioning system in indoor environments. Besides the Bebop drone camera acquiring Full HD images, the Bebop Autonomy driver only transmits 640×480 resolution frames to the base station, which decreases the AR-Tag position error calculation. Besides, the executed follow flights presented a proper operation when the tag capture provides good position data to the PID control algorithm.

The camera shift caused by the master UAV frame inclination during the route changes drives the tag position calculation error. Still, the Bebop gimbal was able to compensate for this, preventing the aircraft loss of control. Also, the small tag size demands keeping the slave UAV in a close flight with the master one. Because of that, the camera capture area is reduced, so the master UAV sometimes loses visibility of the slave, and the position control is lost. This difficulty could be minimized if a slave UAV capable of supporting a bigger tag size is used.

Some image frames sent by Bebop drone to the base station fail, which is possible to see in the experiment video. These fail to affect the position data generation of the AR-Tag package and the good work of the PID algorithm. Besides that, as shown in the video, this problem was handled by the control algorithm.

6 Conclusions and Future Works

The experiments confirm the reliability of the master-slave position control proposed architecture for real-world applications under controlled laboratory conditions. The AR-Tag tool's measurements provide adequate quality data for the UAV autonomous flight algorithm in the proposed requirements.

The use of a better resolution and image quality camera, increased capability tag capture arrangements, and multi-sensor fusion data is an interesting way to achieve better results for the proposed architecture. Practical problems must be solved to allow the application of this method in real robot operations, mainly in unstructured outdoor environments. Still, it is possible to consider that a first proof-of-concept of the proposed collaborative architecture is presented in this work.

Future works are planned to investigate new visual positioning data systems and arrangements that could provide better data to the applications, like active light tags, for example. Also, real-world practical problems that could be solved using the cooperative master-slave proposed architecture will be investigated.

Acknowledgment. The authors are grateful to the FCT - Foundation for Science and Technology, Portugal for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI - Research Centre in Digitalization and Intelligent Robotics (UIDB/05757/2020 and UIDP/05757/2020) and SusTEC (LA/P/0007/2021).

References

1. Lee, H.-S., Shin, B.-S., Thomasson, J.A., Wang, T., Zhang, Z., Han, X.: Development of multiple uav collaborative driving systems for improving field phenotyping. Sensors **22**(4), 1423 (2022). https://doi.org/10.3390/s22041423, https://www.mdpi.com/1424-8220/22/4/1423, ISSN 1424–8220
2. Lu, J., Fu, J., Zhang, J., Zhang, K.: Multi-machines and Multi-tasks Scheduling for UAV Power Inspection in Smart Grid. In: Liu, Q., Liu, X., Chen, B., Zhang, Y., Peng, J. (eds.) Proceedings of the 11th International Conference on Computer Engineering and Networks. LNEE, vol. 808, pp. 1579–1587. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-6554-7_174
3. Kanellakis, C., Mansouri, S.S., Fresk, E., Kominiak, D., Nikolakopoulos, G.: Cooperative uavs as a tool for aerial inspection of large scale aging infrastructure. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5040–5040 (2018). https://doi.org/10.1109/IROS.2018.8593996

4. Zimmermann, F., Eling, C., Klingbeil, L., Kuhlmann, H.: Precise positioning of uavs - dealing with challenging rtk-gps measurement conditions during automated uav flights. ISPRS Annal. Photog. Remote Sens. Spatial Inf. Sci. **4**(2W3), 95–102 (2017). https://doi.org/10.5194/isprs-annals-IV-2-W3-95-2017, ISSN 21949050
5. Al-Kaff, A., Martín, D., García, F., de la Escalera, A., Armingol, J.M.: Survey of computer vision algorithms and applications for unmanned aerial vehicles. Expert Syst. Appli. **92**, 447–463 (2018). https://doi.org/10.1016/j.eswa.2017.09.033, ISSN 09574174
6. APRIL Robotics Laboratory. AprilTags Visual Fiducial System (2018). https://april.eecs.umich.edu/software/apriltag
7. VTT Technical Research Centre of Finland. Augmented Reality/3D Tracking (2017). http://virtual.vtt.fi/virtual/proj2/multimedia/
8. Hinas, A., Roberts, J.M., Gonzalez, F.: Vision-based target finding and inspection of a ground target using a multirotor UAV system. Sensors (Switzerland) **17**(12), 2929 (2017). https://doi.org/10.3390/s17122929, ISSN 14248220
9. Kayhani, N., Heins, A., Zhao, W., Nahangi, M., McCabe, B., Schoellig, A.: Improved tag-based indoor localization of uavs using extended kalman filter. In Al-Hussein, M. (ed.) Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC), Banff, Canada, pp. 624–631. International Association for Automation and Robotics in Construction (IAARC), May 2019. https://doi.org/10.22260/ISARC2019/0083, ISBN 978-952-69524-0-6
10. Wang, S., Hu, T.: Ros-gazebo supported platform for tag-in-loop indoor localization of quadrocopter. In: Chen, W., Hosoda, K., Menegatti, E., Shimizu, M., Wang, H. (eds.) IAS 2016. AISC, vol. 531, pp. 185–197. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-48036-7_14
11. Boudjit, K., Larbes, C.: Detection and implementation autonomous target tracking with a quadrotor ar.drone. In: 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 02, pp. 223–230 (2015)
12. Muskardin, T.: A novel landing system to increase payload capacity and operational availability of high altitude long endurance uav. In: 2016 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 495–504 (2016). https://doi.org/10.1109/ICUAS.2016.7502668
13. Hood, S., Benson, K., Hamod, P., Madison, D., O'Kane, J.M., Rekleitis, I.: Bird's eye view: Cooperative exploration by UGV and UAV. In: 2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017, pp. 247–255 (2017). https://doi.org/10.1109/ICUAS.2017.7991513
14. ROS.org. ROS (2019). http://www.ros.org
15. Monajjemi, M.: Bebop Autonomy - ROS Driver for Parrot Bebop Drone (2018). https://bebop-autonomy.readthedocs.io/en/latest/
16. Copellia Robotics. CoppeliaSim (2019). https://www.coppeliarobotics.com/