

Uwb-based target localization and following system for UAV

Giorgio Pellegrini, *giorgio.pellegrini@studenti.unitn.it*, E.N. 239449

Elias Fontanari, *elias.fontanari@studenti.unitn.it*, E.N. 231497

Abstract—Thanks to their increasingly compactness and versatility, Unmanned Aerial Vehicles (UAVs) usage in modern robotic tasks has grown in many different fields. Precision agriculture, environment exploration, search and rescue and fire fighting are only examples of the amount of possible UAVs applications. Among all, the interest in leader-follower tasks has grown, given its potential use (e.g. cinematographic shootings). In this work, a localization and following application is presented. The system is based on a new Ultra-wideband (UWB) kit, composed by a single and a double UWB antennas. Differently from other UWB-based leader-follower solutions, the proposed architecture uses not only ranging measures but also Angle of Arrival (AoA). Inherently it has all the advantages of UWB-based among visual-based system (low battery consumption and low computational effort), plus an already embedded way to localize a target in polar coordinate. The system performances were assessed by extensive simulations and by testing a real setup, using a motion capture system as ground truth. Although the UWB-pair kit presents high errors and non linearity for the AoA measurements, the tests have shown promising behaviour with an overall good following accuracy.

I. INTRODUCTION

UAVs are becoming useful partners in many situations and their low cost, compactness and versatility, has grown the researches interest in a lot of different applications. For instance the effectiveness of UAVs usage was assessed in different application problems, such as crop disease detection [1], complex environment exploration [2], fire monitoring, detection and fighting [3].

In all of the possible different applications, different control strategies are being used and refined in the years. One of the most investigated problem is target localization and following. This can be done using GPS position of both target and follower, but this solution remains too environmental restrictive, since in some situations, as for indoor application where satellites have poor precision or can even completely fail [4], it is not suitable.

Nowadays most of the solutions to overcome this environment-related problems, are vision-based [5]. They can rely on the flexibility in target changing and are already embedded in most of the commercial drones, for video shooting. The problems with these solutions is that they are based on computationally expensive image processing algorithm, and they have high power consumptions. Since computational limitation and battery duration are two well known UAVs problems [6], the needing of more "light" solutions is researched. A category of localization technology that shows high performance with a low power consumption and low computational effort, is the Radio Frequency (RF) based. These solutions make use of metrics such as Time of Arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA), and Received Signal Strength (RSS), to perform localization. Some example of the technologies that can provides these measurements are Wi-Fi, Bluetooth, Ultra-Wideband (UWB), Zigbee, Radio Frequency Identification Devices (RFID), Long-Range Radio (LoRA), and cellular networks [7]. Among all these possible solutions, UWB presents some unique characteristics like low-cost, low latency, low energy consumption, and centimeter-level accuracy. Moreover, as for the other RF-based application, UWB localization does not suffer from low-visibility conditions. For these reasons, many researchers in the field have studied localization, following and many other robotic applications

using UWB-based architectures.

For instance, Zhang and Duan [8] have developed an assisted PDoA positioning method for elderly care. They have developed a method to get rid of the Non-Line-Of-Sight (NLOS) error for elderly localization in elder-care structures. The method start with a first coarse localization with pure PDoA measures, subsequently refined by finding the actual antennas that are in Line-Of-Sight, providing an eventual NLOS compensation, and a final convex regression estimation.

A work more focused on following, is the one of Santoro et al. [9], in which they have proposed a UWB-based solution, that fuse with a Kalman filter, ranging measures coming from three UWB anchors, deployed on the drone in a triangular shape to avoid position ambiguities, with a target which is assumed to move with a linear motion model. More specifically, they have shown that baseline length has an inversely proportional relationship with localization accuracy and with an Hardware-in-The-Loop simulation, and even with preliminary experimental-results on a commercial drone, they have demonstrate the functionality of the proposed following solution. In [10] the authors present a side-by-side following method for a differential drive robot. They use PDoA measures developing a recursive least square parameter adaptation algorithm (RLS-PAA), which performs a control parameter regression while the target person is walking. On experimental test, considering a safe side distance of 0.5 meters, they have achieved a true person-robot distance beyond 0.430 meters for a circle trajectory and 0.482 meters for a square trajectory, respectively, following quite well the shape of the target path.

To our knowledge, not a lot of work that make use of both TDoA and PDoA for localization, are present in the literature, with a complete absence of following application involving these measures. An example could be the optimization-based position algorithm presented by Chen et al. [11]. The proposed method starts with an initial location estimate, obtained by optimizing a TDOA cost function (considering a set of N measures obtained by N anchors deployed in the environment). Next, a PDOA, or a hybrid TDOA-PDOA cost function, is optimized using a particle swarm optimizer to obtain the final location estimate. This procedure has been proven to be effective in improving localization performance relative to pure TDOA methods.

Differently from the reviewed target following work [9, 10], our work has the aim of asserting if TDoA and PDoA measures (from where range and AoA are calculated respectively), can be enough to follow a target with enough accuracy w.r.t. the wanted strategy. Indeed no filtering, except of a PDoA averaging, is primarily performed to calculate the controls. To this end, we used a special kind of Ultra-wideband (UWB) pair of antennas, developed by Qorvo. This kit is composed by a classic single UWB antenna and a double one, with their own Nordic companion board for the firmware. The kit was not developed and commercialized specifically to be compact and to be used on a moving robotic agent. In fact, it has a non negligible weight and encumbrance, that of course impact the overall system performance, both in power consumption and stability. However, since this work has the aim of studying the effectiveness of the

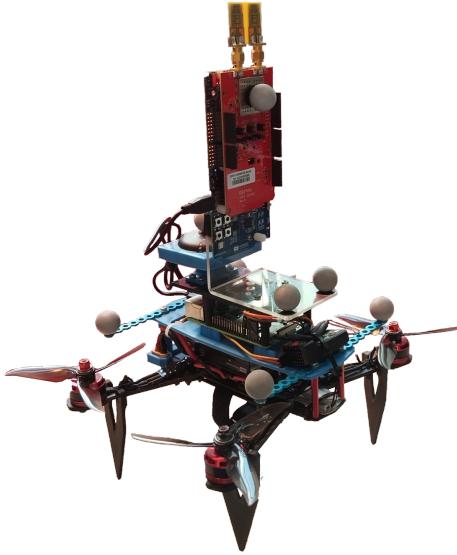


Figure 1. compact 250mm wheelbase Quadcopter used for the test. It is clearly visible the double antenna on top of the drone, as well as the marker used for the Mocap localization.

TDoA/PDoA-based target following, this is not a problem. Thanks to the presence of an array of two antennas, the double UWB, which is deployed on the robotic agent, calculate both range and AoA w.r.t. the single UWB (i.e. the target). These two quantities are the minimum needed to localize the target w.r.t. the agent local reference frame in polar coordinates. In the proposed architecture, as can be seen in Figure 1 the double UWB is mounted on a a compact 250 mm wheelbase UAV, whereas the target UWB could be kept by a human being or mounted on another robotic agent. The main contributions of this work are:

- The development of an algorithm for target localization, given ranging and AoA measures;
- The development of a Gazebo Software-in-The-Loop simulation to evaluate the proposed following strategy;
- The Qorvo DW3 QM33 SDK UWB kit characterization;
- The development of a repeatable experimental setup, involving Motion Capture for ground truth, to test the described real system.

The rest of the paper is organized as follow: in Sec. II we formally introduce the problem, the sensed quantities, and the control law. In Sec. III the quadcopter architecture, together with a description of the robot used as target and of the used motion capture system is exposed. Following, in Sec. IV, after a brief ROS2 introduction, we describe the simulated model of quadcopter and how it interacts with the simulated sensor, in the simulation environment. The implementation details, are instead exposed in Sec. V, whereas in Sec. VI the UWB sensor kit is characterized. In Sec. VII the results of both the simulation and the real test, are presented and compared and in Sec. VIII some conclusions and future improvements are discussed.

II. PROBLEM FORMULATION

As previously introduced, the problem addressed in this work, is the classical *leader-follower* application using relative range and angle. The policy to be maintained by the follower during the leader motion, is to stay at a constant distance (in the x-y plane), with an AoA of zero. The target (leader) is equipped with the single UWB (i.e. the tag), whereas the quadrotor is equipped with the double UWB

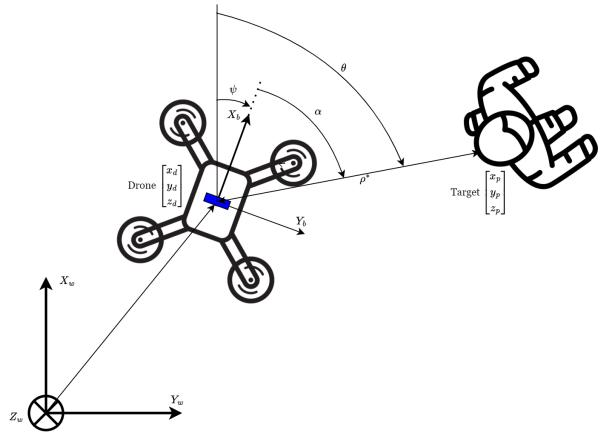


Figure 2. Application model scheme, in which the actual planar range ρ^* , the angles and both the drone local and the global reference frames, are highlighted. In blue is represented the double UWB antenna, deployed perpendicular to the drone's x direction.

(i.e. the follower). Let us define the coordinates of both tag \mathbf{p} and follower \mathbf{d} , in the fixed global reference frame

$$\mathbf{d} = [x_d, y_d, z_d]^T, \quad \mathbf{p} = [x_p, y_p, z_p]^T \quad (1)$$

As said, the quantities delivered by the sensor are the range between the two antennas ρ , and the angle of arrival of the tag w.r.t. the follower α . The range can be expressed as

$$\rho = \|\mathbf{d} - \mathbf{p}\| = \sqrt{(x_d - x_p)^2 + (y_d - y_p)^2 + (z_d - z_p)^2} + \eta = \bar{\rho} + \eta \quad (2)$$

Where η is the measure uncertainty, which is supposed to be normally distributed, zero-mean and white, i.e. $\eta \sim N(0, \sigma_\rho^2)$. The variance of this quantity, is related to the UWB signal bandwidth and of course with the actual distance $\bar{\rho}$ [12]. The range measurement is also affected by a bias introduced by the timestamp delay, which is different for each type of radio and is function of the actual distance $\bar{\rho}$ [13]. Since in the proposed application the measured distance (without considering the initial conditions), should stay in a relatively strict band, a calibration in that zone of interest, should be enough to mitigate the bias. Given that, the bias is supposed to be negligible. In the end, Line-of-sight (LOS) and negligible shadowing conditions are assumed, and so also Non-line-of-sight and shadowing sources of error [14], are neglected.

Clearly the measure obtained is the pure 3D range which is not the wanted quantity, since the described policy is to maintain a certain distance in x-y plane. With the described configuration, it is not possible to directly obtain this value. But if we suppose that the quadrotor can collect measurements about its height h w.r.t. to the ground (e.g. with a distance sensor, or any type of global positioning system) and assuming a fixed known value for the target height, it is possible to calculate the projection of the range in the x-y plane ρ^* . The assumption about the target height, can be simply made by knowing at which height the tag will be kept. Since in our test the tag is placed on a ground robotic agent, we have simply measured the height and saved it as the constant h_t . In the end the projection ρ^* can be calculated as follow:

$$\rho^* = \sqrt{\rho^2 - (z_d - h_t)^2} \quad (3)$$

where the double UWB antenna height $z_d = z + h_d$, i.e. the sum of the quadrotor height estimates z and the relative height of the double antenna w.r.t. the flight controller h_d (which is a construction

constant).

For what concerns the angle of arrival, if we consider the double UWB face placed perpendicular to the x drone axis (as depicted in Figure 2), it is possible to define the angle of arrival α measured by the UWB, as follow:

$$\alpha = \theta - \psi + \xi \quad (4)$$

where ψ is the yaw orientation of the quadrotor, θ is the geometric angle between the follower and the tag in the x-y plane and ξ is the measure uncertainty, which, as for the range, is supposed to be normally distributed, zero-mean and white, that is $\xi \sim \mathcal{N}(0, \sigma_a^2)$. The same considerations and assumptions done for the range uncertainty, are also valid for the angle uncertainty. This angle is obtained from the PDoA, that is the difference in phase of the signal received by the two antennas of the follower, by means of the following relation:

$$\alpha = \arcsin\left(\frac{\Delta\phi \cdot \lambda}{2\pi \cdot d}\right) \quad (5)$$

where $\Delta\phi$ is the PDoA, λ is the signal wavelength and d is the wheelbase between the two, in the double UWB, antennas. This conversion from PDoA to AoA is performed directly inside the Nordic companion board of the antenna, given that all the constant like λ and d are given in construction phase. Moreover, as will be later explained, this conversion is performed by knowing the calibration PDoA offset and the number of elements averaging it. Both those quantities can be modified from serial port, to obtain the better possible value in the calibration phase.

A. Control law

Let us define the wanted policy, as a vector with the wanted range, angle and height $r_w = [\rho_w, \alpha_w, z_w]^T$ and the estimated range, angle and height at time t $r_a(t) = [\rho^*(t), \alpha(t), z(t)]^T$. The control law has the scope of minimizing and maintaining low the difference $r_a - r_w$ in time. To do so three different control strategies are implied:

- a PID controller to generate the x-y velocities setpoints
- a setpoint control for the yaw
- a setpoint control for the height

The PID controller is based on the difference between the actual measured range ρ^* and the wanted value ρ_w , and by defining $d = \rho^* - \rho_w$, a scalar control value V_s can be calculated:

$$V_{sk} = K_p \cdot d_k + K_d \cdot \left(\frac{d_k - d_{k-1}}{\Delta t} \right) + K_i \cdot I_{dk} \Delta t \quad (6)$$

Where K_p , K_d , K_i are the proportional, derivative and integral gain respectively, Δt is the time-step and I_d is the cumulative integral value which can be defined as follow:

$$I_{dk} = I_{dk-1} + d_k$$

The subscript k defines the discrete k instant at which the control is calculated. In order to obtain the actual x-y pair velocities, a last projection step has to be performed. Since the quantity V_{sk} represents the velocity to be applied in the direction of the target, the V_{xk} and V_{yk} control velocity, expressed in the global frame, can be obtained as follow:

$$\begin{bmatrix} V_{xk} \\ V_{yk} \end{bmatrix} = V_{sk} \cdot [\cos(\theta_k) \quad \sin(\theta_k)]^T \quad (7)$$

where θ_k , as already mentioned, is the geometric planar angle between the target and the drone and can be derived as:

$$\theta_k = \alpha_k + \psi_{k_{estim}} \quad (8)$$

where $\psi_{k_{estim}}$ is the quadrotor yaw angle, estimated by the flight controller. Obviously this quantity is affected by both the uncertainty

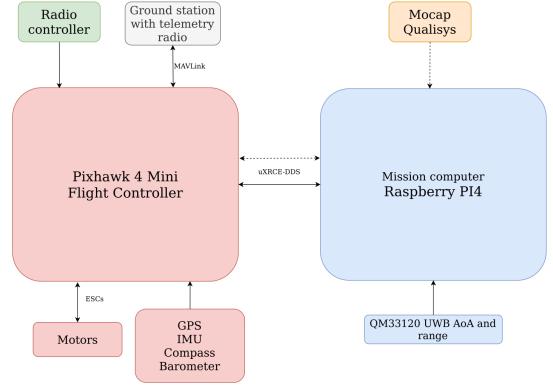


Figure 3. Quadcopter architecture.

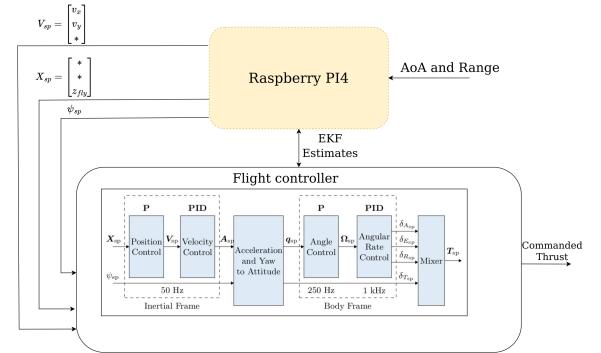


Figure 4. Flight control loop: * stands for not controlled: if in X_{sp} it means that control computed starting from Velocity Control, if in V_{sp} computed by not empty Position Control input.

of the AoA measure α_k and of $\psi_{k_{estim}}$.

The two obtained control velocities, are fed to the flight controller as a velocity setpoint $V_{sp} = [V_{xk}, V_{yk}, *]$, whereas θ is fed as the yaw setpoint, ψ_{sp} , to orient the drone toward the target. Lastly the height is controlled by feeding the flight controller with a position setpoint, $X_{sp} = [*], z_w]$ ¹.

III. LEADER-FOLLOWER SYSTEM SETUP

In the proposed leader-follower application, the target carries a single antenna UWB transceiver while the chaser is a QAV250 quadcopter equipped with a Pixhawk4 Mini flight controller, a Raspberry Pi4 and a double antenna UWB transceiver that communicates through serial port to the mission computer, supplying to it range and angle of arrival of the target w.r.t. the drone.

A. Quadcopter architecture

The architecture and the interactions between the components of the vehicle are schematized in Figure 3.

The flight controller (FC) runs the PX4 Autopilot firmware, responsible for controlling propellers and for state estimates. The Raspberry using the received target location information and the

¹The * simply means that the quantity is not given so not controlled

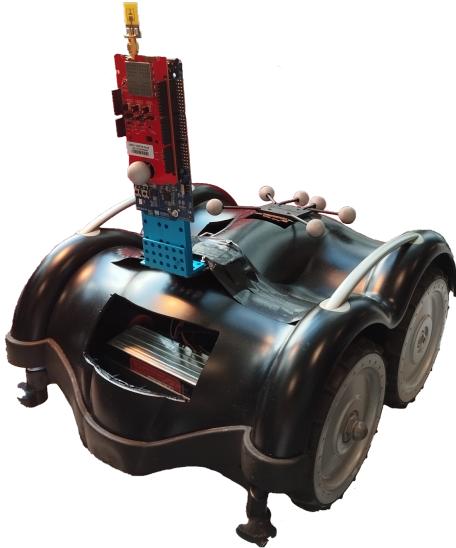


Figure 5. DDR target. It is visible the mounted single antenna, as well as the presence of the Mocap markers, used to get its global position.

drone state estimates, provides the control velocity setpoints, together with the yaw and the height setpoints (as explained in Sec. II.A), to the FC. The flight controller then, using a combination of nested P and PID controllers, converts the inputs in motor thrust commands (as depicted in figure Figure 4).

Since PX4 has the possibility of using a uXRCE-DDS middleware (a client running on the FC and an agent on the Raspberry) to allow uORB messages (that contains information such as command or vehicle estimates and status) to be published and subscribed on a companion computer as they were ROS2 topics, the Raspberry can interact with the flight controller using these topics (better details about ROS2, topics and publishing/subscribing will be later exposed). The advantage of using the middleware ROS2 is a great flexibility, since all the devices connected to the same network can access to all the topics published, even those of the FC interfaced to the Raspberry. In our case this is particularly useful, in order to integrate Motion capture as the global reference for the UAV, as it will be explained later, and of course to send the FC control inputs.

B. Target

To ensure that the UWB tag keeps a constant height during the motion and to avoid possible dangerous situations that involves human beings during testing, as target we used a customized differential drive lawn-mower robot Figure 5, with the UWB tag rigidly mounted on it. During our tests a person was in charge of driving this DDR with a radio controller.

C. Qualisys motion capture system

In order to have a precise pose ground truth, we performed our tests indoor in a dedicated room with a system of seven Qualisys Arqus cameras installed. The system is able to precisely locate the reflective markers that can be mounted on the bodies to track, using the different views of the cameras. To track a body, a fixed configuration of at least 3 markers has to be mounted on it. However, to avoid possible troubles with marker occlusions, is suggested to place more markers. In our case, we tracked the target and the quadcopter to evaluate their motion.

Since in indoor GPS is not available, the quadcopter pose estimate,



Figure 6. Simulation's SITL image.

done by the MoCap, is used also as global pose, both in position and orientation. The way in which it is done will be explained in Sec. V.C. In addition we have disabled the compass and the barometer contribution to the vehicle's EKF, deactivating the related parameters in the flight controller firmware, since MoCap provides alone, a millimeter-level positioning precision. Beyond the uselessness, the reason of this deactivation lies in the indoor unreliability of these type measurements, due to environmental interference or misleading information (i.e. altitude and magnetic orientation have no meaning in the global reference frame of the MoCap lab).

IV. SIMULATION

A. ROS2 Brief introduction

Before diving into the simulation, is important to explain the ROS2 functioning. The Robot Operating System (ROS) is a set of software libraries and tools for building robot applications. A ROS2 application, can be seen as a network, or better a graph, composed by ROS2 elements, processing together at the same time. The principal ROS2 elements are the nodes, which are the actual computational elements. Each node can send and receive data from other nodes via topics, services, actions, or parameters. ROS2 Nodes can be software Robotic components, like the Flight controller, or can be user programmed. The base principle in ROS2 nodes communication is the publisher/subscriber protocol. For what concerns topics (which are the only communication medium used in this work), each node can be a publisher, so it can "change" the topic value, or a subscriber, so it can "read" the topic value. In this way, ROS2 is an effective communication interface between different hardware and software, with the only requirement of being connected to the same network. A representation example of a ROS2 network, is the one of our Simulation depicted and explained in Figure 7.

A user programmed Node is typically defined inside the so called ROS2 packages, in which the Node executable can be interfaced with user defined topics and messages, or with some defined by others. For example the PX4 team have developed a series of packages in which they define all the messages topics, services and actions needed to fully interact with the flight controller (regardless if it is the simulated one or a real one, since they are equal in software). Packages can be programmed both in C++ or Python, and can be grouped together in ROS2 workspace. For better details, we refer to the ROS2 Documentation[15].

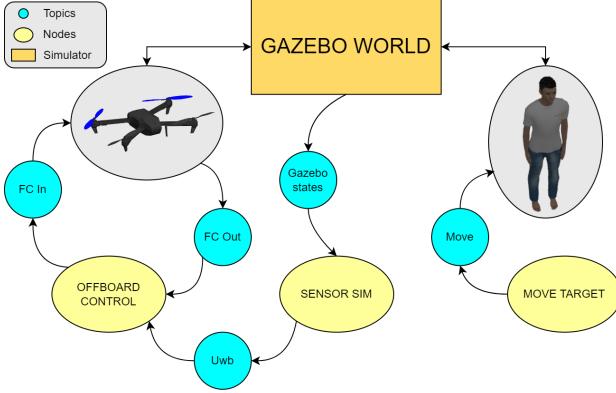


Figure 7. ROS2 simulation Network. The nodes (yellow ellipses) can be subscriber or publisher to the topics (light blue circles). In this way, for example, the offboard node can retrieve the drone's state estimates from the FC, and send to it the controls.

B. Simulation

To assess and validate the developed control law, is necessary to test it on a simulation that well resembles the reality. This is possible conducting software in the loop simulations, with the same flight controller available as hardware and possibly with the same vehicle. Working with the PX4 flight stack, as it is the one that runs on our flight controller, the best simulation environment to use is Gazebo, a graphic simulator with an accurate physics. It is also easily customizable through the use of plugins permitting to include other effects as the wind. Moreover, it interacts well with ROS2, allowing for a lot of flexibility in use.

The vehicle on which we tested our control law in simulation is the Iris quadcopter, the main model for gazebo PX4 simulations (even though it is different from the one we used in real tests). This model guarantees a great level of simulation accuracy: the motors and the generated thrusts are modelled accurately using specific plugins, and even its navigation sensors are corrupted appropriately to be similar to real-case scenarios. Even the flight controller is simulated as the real one, having the same estimation algorithms, controller and characteristics.

To simulate the target we defined a simple object, with the appearance of a human, that can translate in the x and y direction. To make it move, we created a ROS2 node that publishes on the gazebo ROS2 topics the position of the target, making it move. To make it move at a certain speed and with the desired trajectory, nearby points are sent to gazebo at a constant rate.

To simulate the range and AoA sensor, it is necessary to know the global position in the gazebo world of both target and drone, together with the yaw of this last one. Once that these information are known, and in our case it is simply done with another ROS2 node that subscribes to the pose topic of the objects in gazebo, with simple geometric relations it is possible to determine the true angle and range and then corrupt them with noise, to simulate the sensor. In our simulations, we simply added normally distributed, zero mean and white noise to both quantities (referring to the values obtained in the characterization of the true sensor). This simulated sensor node, supplies the simulated sensor readings to the offboard-control node which, by accessing to the drone estimates, compute the velocity setpoints as in (7), together with the yaw setpoint (8) and the position setpoint for the height control, sending them to the FC. It is interesting to note that, since the flight controller is fed via ROS2 even in the real case, the same offboard control node can be used in the real application, of course using the range and

AoA measures given by the real UWB sensor. A Detailed scheme of the ROS2 network is presented in Figure 7. Simulation results are exposed later, in Sec. VII.A.

V. IMPLEMENTATION DETAILS

In the following section, some details about the implementations will be exposed. For details about the source code, we refer to our git-hub repository [16]

A. Offboard control node

The heart of the following algorithm, as already mentioned, resides in the Raspberry Pi 4 and it is the offboard control node. In particular, the tasks that it performs are the following:

- when started, it subscribes to all the required topics (flight controller state estimates, vehicle status and UWB readings) and it creates even all the required publishers that communicate with the FC (vehicle commands and offboard heartbeat signal²). Then it creates a timer with period $dt = 0.1$ seconds that serves as an endless loop (except in case of program termination) in which at every step the setpoints are sent to the FC. In the main loop there are two different modalities, according to the value of a flag of the node: takeoff and following. The flag is initiated in takeoff mode. However, in both the modes the offboard heartbeat signal is sent once per step.
- when the main loop starts, the quadcopter executes a takeoff at the wanted height. This is done by sending to the FC the position setpoint $X_{sp} = [0, 0, z_w]^T$. Since when the flight controller starts up it set the global origin where it is located, it is a simple vertical takeoff. The takeoff is considered done when the drone position estimate is within a tolerance near to the setpoint. At that time the flag switches to following mode.
- once in following mode, at every time step the control law explained in Sec. II.A is employed to compute the setpoints needed to follow the target. To do so, it takes the UWB topic values (i.e. the last measured range and AoA) published by the sensor node, that will be later described, and calculate the controls as explained. Since in ROS2 topic's values remains equal until they are changed by a publisher, the offboard node, if for unforeseen reasons the UWB sensor do not update the measures, keep the old values until they are changed. If we fall in this situation our control strategy is divided into three cases:

- If the taken message remains equal for n_{msg_0} time steps, the offboard node command a hoovering. This is done sending null x and y velocity setpoints, z_w as z position and $\psi_{k_{estim}}$ as yaw;
- If the messages still remain identical for more than n_{msg_1} steps, the node command the FC to land on the spot and disarm;
- Until the message remains equal for less than n_{msg_0} steps, the control is performed normally using the last values.

The described behaviour lasts until the drone is disarmed due to the absence of UWB messages or until a manual exit from the offboard mode commanded by the radio controller.

B. UWB sensor node

The double-antenna UWB module has its own microcontroller on which the proprietary firmware runs. It communicates the range and

²The offboard heartbeat signal is a specific "proof of life" message that has to be sent to the FC, at least with a frequency of 2 Hz, to stay in offboard mode, otherwise PX4 will switch to position mode.



Figure 8. QM33120WDK1 Ultra-Wideband (UWB) Transceiver Development Kit. The Nordic nRF52840DKs (the blue boards behind), are the actual companion boards on which the firmware runs.

angle sensor readings to the Raspberry via serial communication. To exploit the flexibility of ROS2, we decided to create a node on the mission computer that parses the serial communication to extract the values of interest and publish them, with a frequency of 10Hz using a custom ROS2 message that contains the fields range and AoA, on the UWB custom topic.

If no new sensor readings are received, that is when the serial communication with the microcontroller stops or in the eventuality of parsing errors, the node does not publish anything. In this way, the offboard control node, that holds memory of the last received message, realizes if new information are available and if the UWB sensor works properly.

C. Mocap bridge pose node

In the case of indoor flight with a motion capture system, it is necessary to provide Mocap's pose estimate of the drone to the flight controller in order to fly. This is done with a free-to-use Qualisys node [17] executed on a device that shares the network of the PC on which the motion capture system runs. The pose of all the bodies, defined and tracked in the MoCap system, are collected and published.

With another node, called bridge, we simply subscribe and publish to the dedicated input topic of the flight controller. Inside this bridge node, we manipulate the content of the Qualisys' topics to change the global reference frame position, and send this manipulated data to the FC. The global frame is rigidly translated in the position of the drone before takeoff.

VI. UWB TRANSCEIVER DEVELOPMENT KIT

As already underlined, for the target position estimate, we have used the Ultra-Wideband (UWB) Transceiver Development Kit produced by Qorvo Figure 8, for which we refer to the official website page [18] and datasheet [19] for details. For this kit, is also available a firmware that can be requested from Qorvo. Reading the code, we have deduced the following main information:

- AoA and distance, are fed directly to the serial port in an encoded string with other quantity like PDoA, status and block number;
- the firmware can run in different configurations, depending on the antenna type and the selected communication channel³. *Jolie* and *Monalisa* are the two antenna types. The possible channels are instead 9 and 5. It follows that the antenna can be configured in 4 different ways, using the parameter ANTENNA;

³The channel type, allows to change between the two supported IEEE standards: IEEE Std. 802.15.4™-2020[20] and IEEE Std. 802.15.4z™-2020[21]

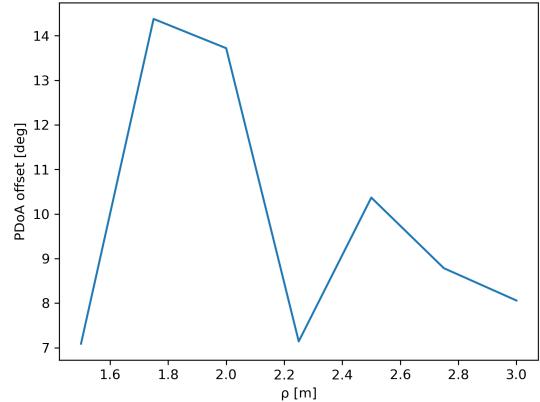


Figure 9. PDOAOFF at various distances.

JOLIE9, JOLIE5, MONALISA9, MONALISA5. Our antenna is a *Jolie* and the default channel is 9, meaning that the ANTENNA parameter is equal to JOLIE9 by default;

- PDoA raw measures, are adjusted by means of a stepped linear regression. The steps of this linear regression are defined by a pair of look up table (different for each channel and antenna type): one for the PDoA intervals, and one for the slopes that the liner regression model has to have inside these intervals. All the values were clearly calculated after laboratory test, aiming to reduce the non linear effects present in the raw PDoA measure;
- the PDoA value truly used to calculate AoA by means of the relation (5), is the average of the last PAVRG values, corrected with PDOAOFF. PAVRG and PDOAOFF are user definable constants by default equal to 10 and 0 respectively. The first defines the values over which perform the moving average: an high value can mitigate the high frequency error, but can worst the sensor performance in detecting PDoA changes; on the other side a lower value can detect even rapid changes in the actual PDoA, but the high frequency error contribution is not lowered. PDOAOFF defines instead the offset to apply at the average result as a user calibration.

A. Characterization

In order to understand the behavior of the kit, we have performed a sensor characterization to asses what follows:

- Performance difference using the two supported IEEE standards (i.e. channel 9 or 5), comparing the results at different angles and distances;
- Performance at different tag heights and angles;
- Performance putting the tag behind the double-UWB anchor;

Since the first tested channel configuration was the default one (i.e. 9), the last two results were extracted only for this configuration. By the way, channel type should not change noticeably the validity of the extracted information (e.g. if the kit performance are worsen by a tag height change, the channel should not be a discriminant).

All the listed assessment, where performed using MoCap as ground truth and even for precise positioning (i.e. to place the tag at precise distances, angles and eventually heights w.r.t the double UWB).

Since the UWB kit declared working region is the half-plane in front of the double UWB, to compare the behaviour under the two different IEEE standards, we have perform tests in this region, at

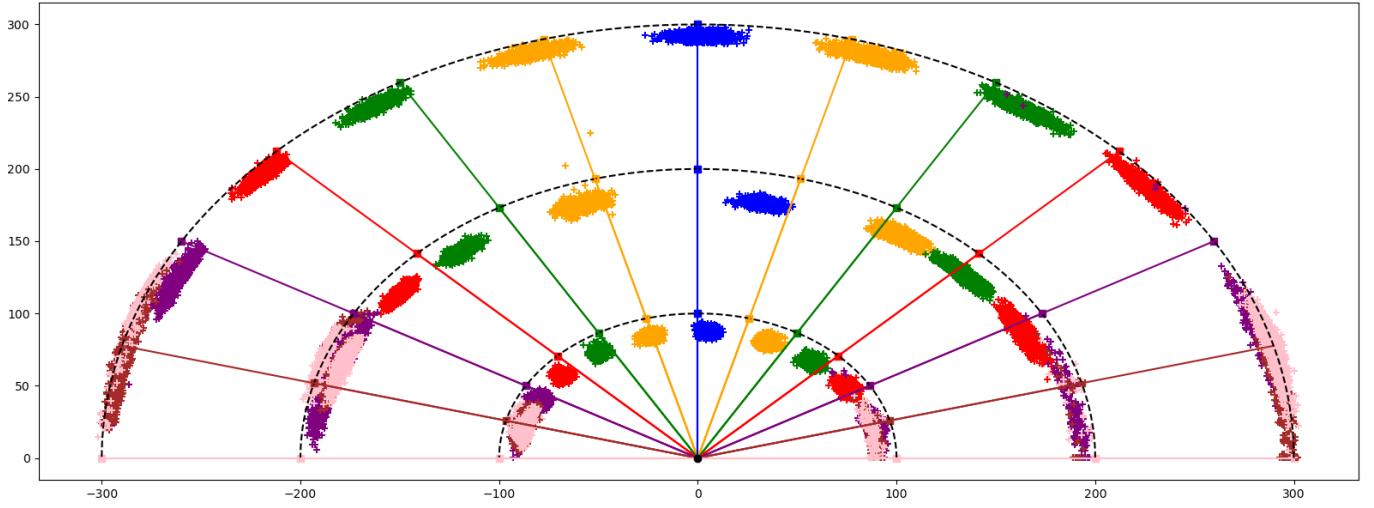


Figure 10. Location test conducted with channel 9 standards, expressed in centimeters.

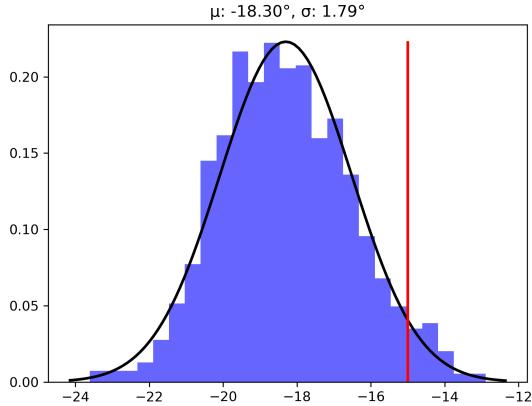


Figure 11. Channel 9 AoA measurements distribution at -15 degrees and 200 centimeters.

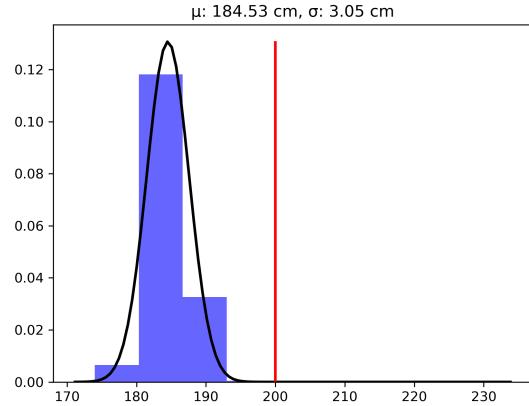


Figure 12. Channel 9 range measurements distribution at -15 degrees and 200 centimeters

different discrete values of range and angles. Given that this half-plane can be defined with a polar coordinate pair, w.r.t. the double antenna, with undefined radius and angles from -90 to $+90$ degrees (being 0 degrees when the tag is in front of the double, with angle positive in clockwise sense), we have performed the tests as follow:

- For channel 9, at three different range $\rho_9 = [100, 200, 300]$ cm and angles from -90 to $+90$ degree with a step of 15;
- For channel 5, at two different range $\rho_5 = [150, 250]$ cm and angles from -90 to $+90$ degree for the first range and from -60 to $+60$ degree for the second, again with a step of 15.

As suggested by the manufacturer, before the data collection we have calibrated the PDoA offset placing the modules at a distance of at least 1.5 meters at 0 degrees, collecting 1000 measures of the PDoA, averaging them to estimate the PDOAOFF value to set. For the channel 9 configuration, we determined it at a distance of 2.5 meters, thinking that the range did not have much influence. However, before the channel 5 data collection, we realized that PDoA offset changes with distance, as can be seen in Figure 9. Hence, for channel 5, the PDoA calibration was performed at a distance of 1.5 meters, which is congruent with the following policy for the indoor test.

a) *Channel Comparison:* The results for channel 9, are depicted in Figure 10. The graph shows different behaviours at different distances. For example at a distance of 3 meters the estimated angle is quite good, whereas at 2 meters it shows different behaviours for the left and the right part of the half-plane. Indeed the left part shows better angle measures than the right one. Despite this, the measured angles are generally bad. This behaviour could be caused by the great variation of PDOAFF at 2 meters w.r.t. 2.5, as shown in Figure 9. At 1 meter the results are slightly better, but still there is asymmetry between positive and negative part. Another characteristic to denote is that, despite the distance, at angles greater than 75 degree (and -75), the measurements are not reliable and tend to be much more scattered. For what concerns the range, a not constant negative bias is always present. The data clouds, shows instead an almost constant spread, indicating that the standard deviation of both range and angles, are constants. A pair of distribution examples, are presented in Figure 11, 12. Both shows a bias w.r.t. the actual angle and range, but their distributions are gaussian-like, with $\sigma_\alpha = 1.79^\circ$ and $\sigma_\rho = 3.05$ cm and mean error of 3.30° for the angle and of 15.47 cm for the range. Comparing these values with the manufacturer's specifications, reported in Table I, is

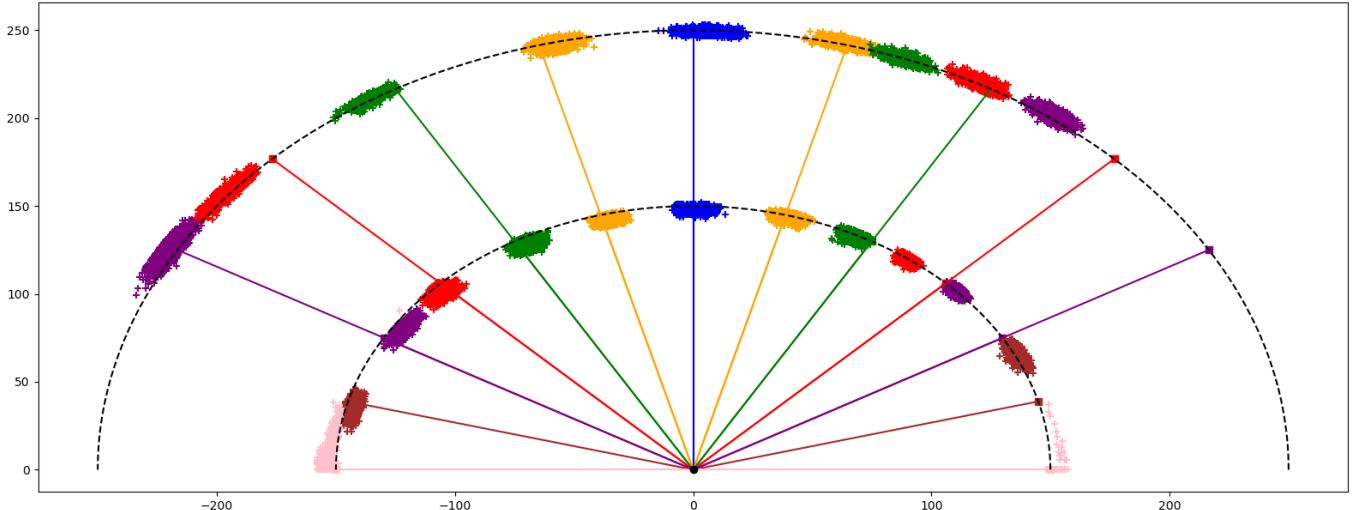


Figure 13. Location test conducted with channel 5 standards, expressed in centimeters.

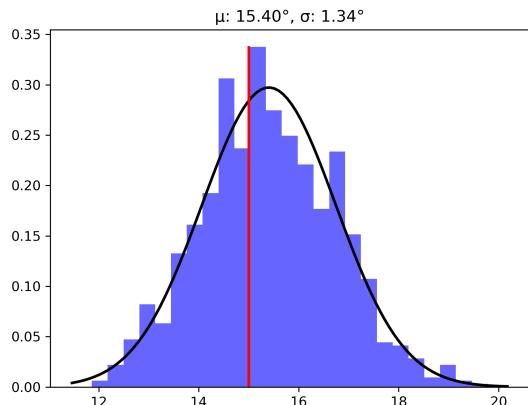


Figure 14. Channel 5 AoA measurements distribution at +15 degrees and 150 centimeters.

Table I
MANUFACTURER DECLARED LOCATION ACCURACY CHARACTERISTICS.
SEE DATASHEET P.17 FOR REFERENCE[19]

	accuracy	standard deviation	Units
Range	± 6	3	cm
AoA CH9	± 6.25	2.5	deg
AoA CH5	± 6.25	2	deg

possible to see that the standard deviation is in accordance with the declared one both for the angle and the range. For what concerns the accuracy, the angle is inside the interval declared while it is not the case for the range. However, as can be seen in Figure 10, if instead of the measurement at 2 meters and -15 degrees one would consider the data at the same distance and angle mirrored, the angle would not be anymore in the accuracy interval.

The test for the channel 5, is depicted in Figure 13. As previously specified, this test was conducted by tuning the PDoA offset at a distance congruent with the one of the following task, that is 1.5 m, in order to reduce the error in the working region. To this end, even the tested ranges were changed to a pair 1.5 and 2.5 m.

Also for this test, the asymmetry between negative and positive

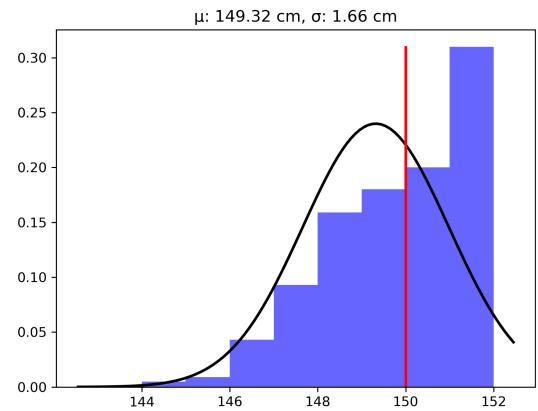


Figure 15. Channel 5 range measurements distribution at +15 degrees and 150 centimeters

region is clear. Despite the slightly lower accuracy in range, the left region, shows better performance in angle measurements. Overall the general accuracy is visibly better for this configuration, and more precisely in the cone from -15 to +15 degrees, the localization is very good. This is an important results since, as will be later shown, during the simulation the angle measured by the simulated UWB remains almost always inside this region. Two representative distribution for angle Figure 14 and range Figure 15 measurements are presented. Their distributions are gaussian-like, with $\sigma_\alpha = 1.34^\circ$ and $\sigma_\rho = 1.66$ cm and mean error of 0.4° for the angle and of 0.68 cm for the range. Comparing this results with the declared specifications for channel 5 Table I, both accuracies and standard deviations are congruent. In particular the accuracy error, both for range and AoA, is very low.

As an overall conclusion, channel 5 outperformed channel 9, making it our choice for the presented application.

b) *Performance for height change:* in our Leader-follower application, the angle estimation robustness against the height difference between leader and follower, is an important requirement. Indeed, the target is supposed to be a ground robot, so the tag is placed at a low height. To asses the kit performance

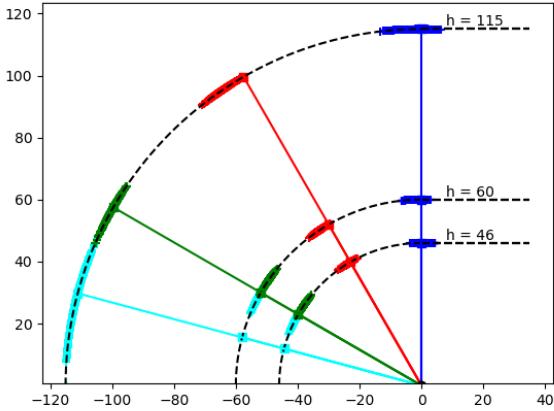


Figure 16. Angle measurements at different heights and at a constant x-y distance of 200 cm.

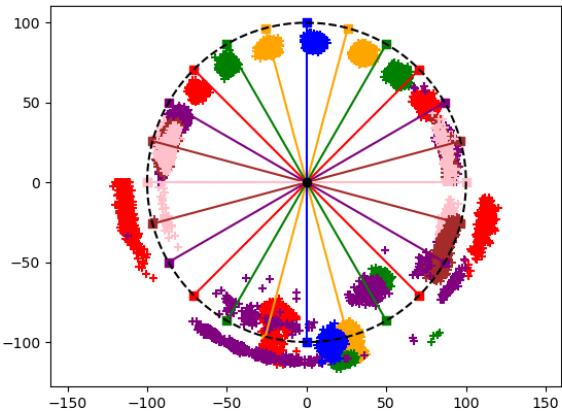


Figure 17. Localization comparison between front and behind taken measures, expressed in centimeters.

at different heights, we have placed the target antenna at 3 different heights $h = [46, 60, 115]$ centimeters and at different angles $\alpha = [0, -30, -60, -75]$ degrees. The results are exposed in Figure 16. It can be seen that the height difference does not remarkably change the measured angle. Only for the measurements at -75 degree the measures are bad but, as previously stated, this is an angle related behaviour, so it does not invalidate the test's outcomes.

c) Behind the double UWB results: in order to fully understand the sensor behaviour in different situations, a test with the target antenna behind the double at a distance of 1 meter was conducted. The results, together with the equivalent measures at one meter taken in front, are presented in Figure 17. As expected, given the manufacturer indication, the measures behind are quite chaotic, and both angles and ranges are mostly wrong. Even the data clouds are spread differently, not resembling the constant behaviour seen for the front taken measures.

d) Conclusions: This evaluation kit has the potential to become a valid tool in Leader-follower applications, but some extra work on the sensor is needed. The main drawback is that the angle offset is

a non linear function of the distance, and for application in which a dynamic range change is involved, this UWB pair is not suitable. To compensate for this behaviour, some work in modeling this offset's non-linearity is suggested. For what concern our purposes, this kit is precise enough, since in the region of interest (i.e. at 1.5 meters ± 15 degrees), it works pretty well. Even in the case of having wider angle and distances, since localization maintain his sign (e.g. if the true angle is 60 degree, it measure 45 not -45), our policy tends to converges towards the 0 degree angle and 1.5 meters distance. Therefore, even in the presence of large accuracy error, the quadcopter is still somehow able to perform decently. Another modification effect on the measures noticed, that is not presented in this characterization, is the change in localization given by the relative orientation between the antennas. Indeed the best measures are obtained if the tag is placed radially w.r.t. the double, whereas important angle errors can be introduced if this condition is not respected (especially at short distances). Probably this error is due to the non isotropic tag antenna. This can overall truly impact the localization performances. In the real test, whose results will be later exposed, this last behaviour is surely responsible for a great amount of the localization error.

VII. RESULTS

A. Simulation

To test the validity of the following algorithm in various motion conditions, we setup a simulation in which the target moves at a velocity of 5 km/h in two ways: in a first phase doing a straight line and then changing direction and doing a sine wave trajectory, testing the quadcopter behaviour during curved motion. The noise that affects the simulated range and AoA sensor, is white gaussian with zero mean and the standard deviations, in the light of what has emerged during the sensor characterization, are set equals to those declared by the manufacturer, as in Table I for channel 5. The set following distance is 1.5 meters and the flight height 2 meters. In Figure 18 is possible to see in yellow, the trajectory of the target that starts at the origin and in blue the quadcopter. Is possible to see that the quadcopter is able to follow accurately the movements done by the target, even in curve, if the curvature radius is not too small and compatible with the demanded following distance.

The AoA sensed by the sensor during the motion is visible in Figure 19. After the first phase of takeoff and stabilization behind the target, it is possible to see that the measured angle is almost always inside a interval of ± 15 degrees, except in the case of the sharp change of direction made when the curved motion starts. During the curves the quadcopter is not able to obtain a sensed angle of 0, however the angle achieved is sufficiently low. A clarification has to be done about the sudden angle sign change at 50 second. This behaviour, is due to the use of the function atan2 in the simulated angle calculations. Indeed, at that time the target-drone y position difference used as numerator in atan2, change sign (is visible in Figure 18). Despite this, the angle absolute value is stabilized around 10 degrees during the curves.

To evaluate the performance in the following distance, Figure 20 can be considered. It can be seen that on average the demanded following distance is achieved. The peak around 20 seconds is due to the target initial movements acceleration, that instantaneously pass from a static state, to move at the set velocity. Of course the drone cannot stabilize at the same rate, so an initial gap is created. Instead the negative-positive peak pair around 40 seconds is caused by the target's change of direction. In that moment the

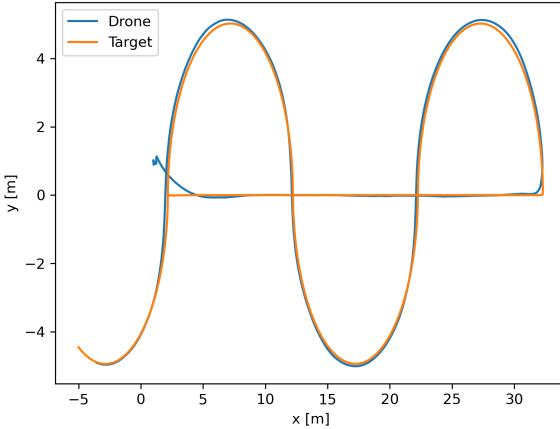


Figure 18. Target and drone trajectory in simulation.

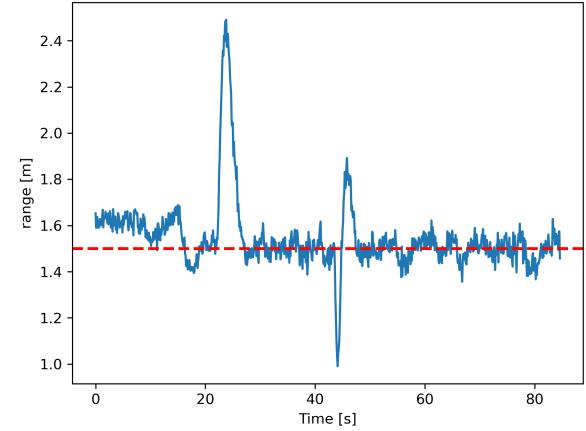


Figure 20. Measured range in simulation. The red dotted line, indicates the wanted range from the following policy.

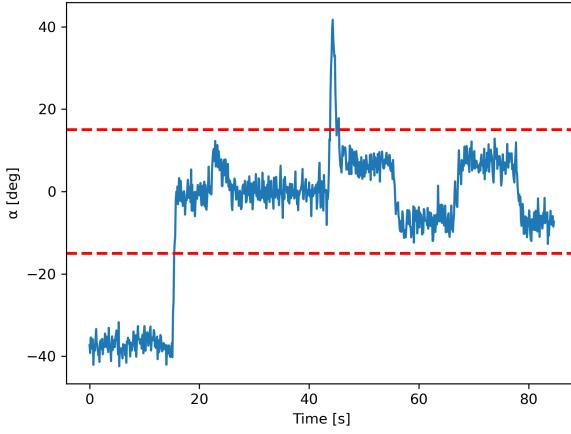


Figure 19. Measured AoA in simulation. The two red dotted lines, indicates the ± 15 deg region in which the kit performs better. The fact that the measured angle remains inside this region, is an important results for the real implementation.

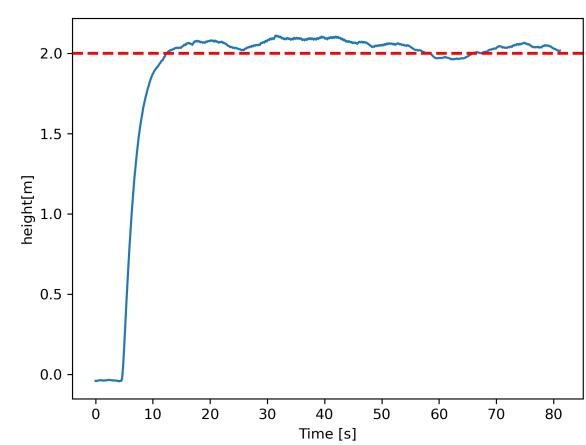


Figure 21. Measured flight height in simulation. The red dotted line, indicates the wanted height from the following policy.

quadcopter stops and waits due to the presence of the integral control effect on the velocity setpoints generation, necessary to have a following distance without bias, counteracting the proportional control behaviour. Without the integral control, the proportional would cause the quadcopter to move backwards at that moment, mitigating that peak pair. However, exception made for this extreme case, our strategy seems to be appropriate so we decided to use the integral, although in some cases it may imply lowerings of the following distance. For safety purposes, in order to mitigate this effects, other strategies should be investigated.

Even the flight height is well respected, as can be seen in Figure 21 and it seems to be independent w.r.t. events such as motion type changes. The estimation accuracy of the target position in the global frame done by the quadcopter using the UWB simulated sensor, is shown in Figure 22, and it can be noted that is always reliable. More in detail, this can be seen in particular in Figure 23. The error is never greater than 0.3 meters for both the axis. It can also be seen that is dependent on the type of motion: in the first phase, where the target move in a

straight way, the error is low and approximately null on average. At the moment of the direction change, when the target starts the curved motion, it expands for both the axis and the mean done on a window relative to that moment is not null. However, the fact that the error is never too large guarantees acceptable following performance.

To test the robustness of the following system, a simulation with noises affected by larger standard deviations, 10 centimeters for the range and 10 degrees for the AoA was performed. From Figure 24, it can be seen that the quadcopter is still able to execute its task, albeit roughly.

Overall, these results suggests that our strategy works, at least in the simulated conditions: some effects indeed are neglected, as the fact that the quadcopter used in the real application, is smaller and different from the simulated Iris drone as well as the inertial modifications due to the mounted UWB sensor, that are not considered.

B. Real test

To test the validity of the UWB kit in a real application, and to further prove the effectiveness of the following algorithm, as previously explained, some real tests were performed. All the

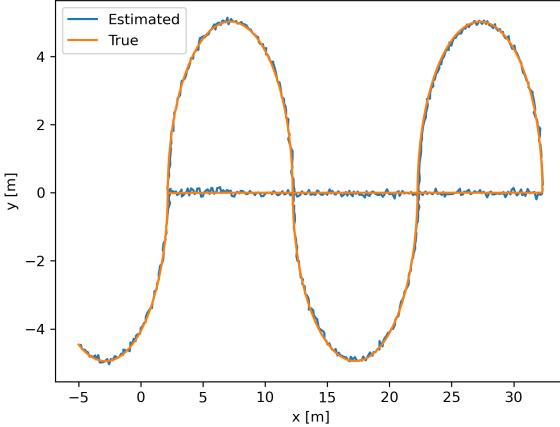


Figure 22. True and estimated pose of the target.

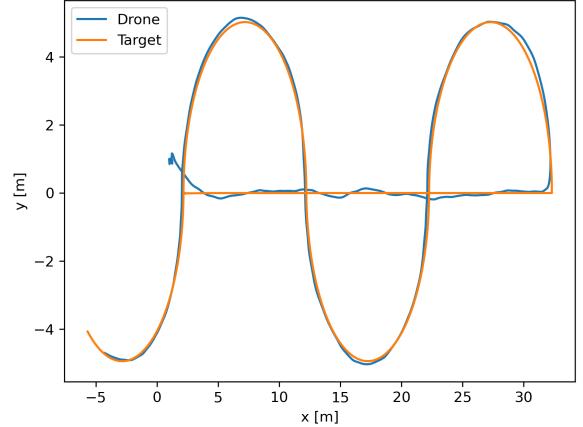


Figure 24. Target and drone trajectory in simulation with high noise.

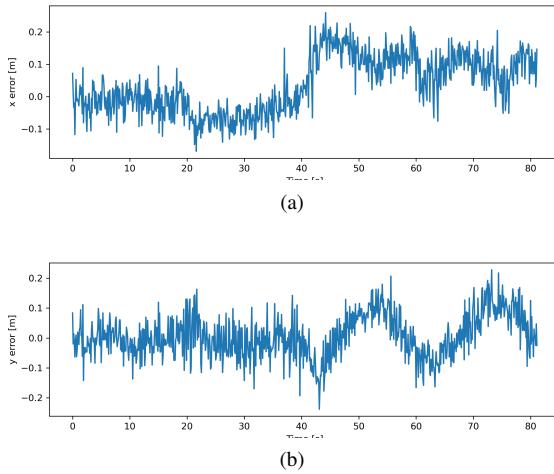


Figure 23. (a) target localization x error [m]. (b) target localization y error [m].

presented results are paired with the relative MoCap ground truth for comparison. The presented results are the best of the conducted tests. In order to obtain a decent motion inside the restricted Mocap laboratory environment, the test were conducted choosing a drone flight height of 0.4 meters (to ensure marker visibility) and a following distance of 1.2 meters. This choice is for sure restricting the UWB kit performance, since better results are obtained at larger distances (as suggested by the manufacturer), but this choice is restricted by the lab dimensions that do not allow to perform good trajectories with higher distances.

In Figure 25 the trajectory of both drone and target, given by MoCap, are presented. It is clear that the drone almost follow the square-like trajectory of the target, suggesting the effectiveness of the following strategy. For what concerns the policy compliance, in Figures 26, 27 AoA and range obtained by the sensor with the relative MoCap ground truth, are depicted. From both can be highlighted the presence of a non-linear bias that majorly depends by the angle. This confirms the behaviour seen in the characterization, in which the angle bias is a function of the actual angle. Nonetheless, the error is maximum in the target turning phases, in which the DDR robot makes almost sharp change of direction. This concretize

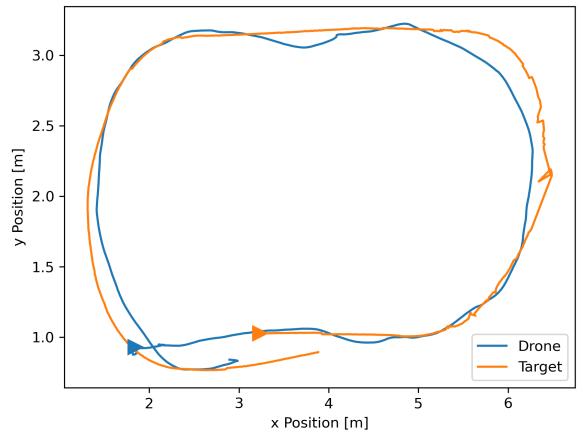


Figure 25. Target and drone trajectory in real test.

the behavior mentioned in the characterization conclusions, for which the measures are worsen by a non radial facing of the tag w.r.t. the double UWB. All these effects sum up, obtaining a non constant bias, especially for the AoA. Nevertheless, if we do not consider the biases, in general the policy is followed. Surely, an outdoor test performed in an open environment, with a more relaxed policy (i.e. a greater following distance), should achieve better results. For what concerns the height, as described, is controlled directly by sending constants z setpoints. The flight height followed by the quadcopter, is presented in Figure 28. The height is maintained with an error similar to the one of the simulation.

The accuracy of the target position drone's estimate, performed using the UWB sensor and its own position estimate, is shown in Figure 29. As can be seen, the estimate follows in general the truth, but is clearly affected by the angle and range biases previously discussed. To better asses the localization performance, the x and y error in time are exposed in Figure 30. The error in both the axes, is bounded in ± 0.4 meters, which is almost twice the one obtained in the simulation. This is clearly due to the described biases.

In conclusion this test brings up the UWB problems discussed in Sec. VI, affecting the final results. Overall the drone behaviour

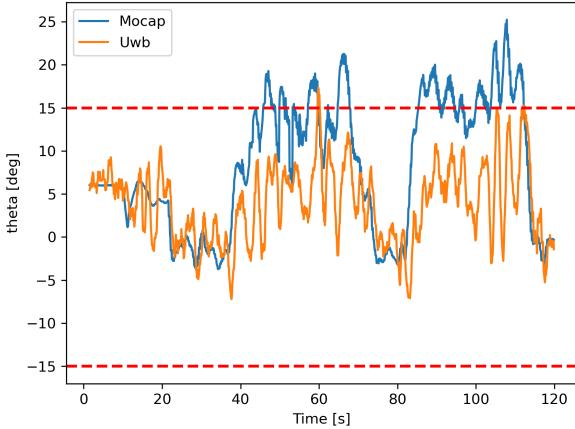


Figure 26. angle of arrival between drone and target. The sensor measure, is compared with the one obtained with MoCap. The red dotted line, indicates the wanted range from the following policy

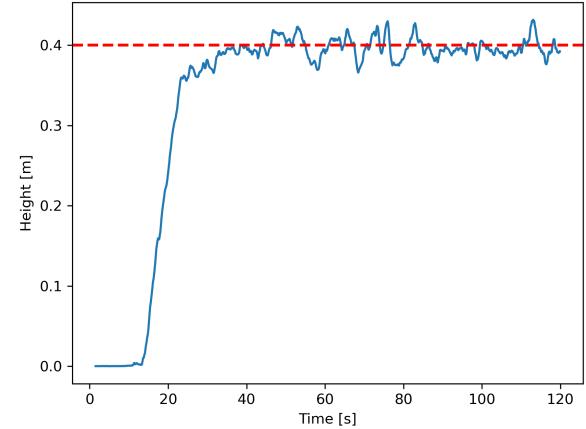


Figure 28. Measured flight height. The red dotted line, indicates the wanted height from the following policy.

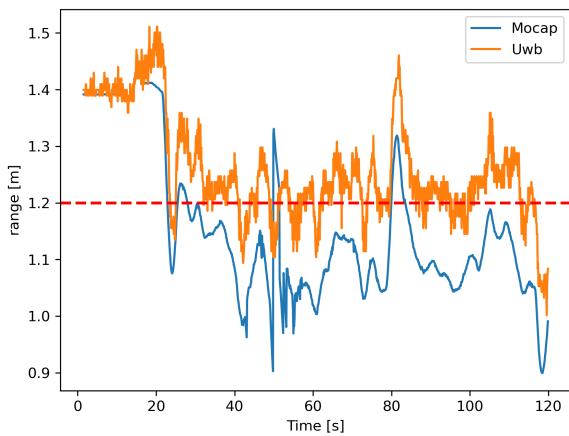


Figure 27. Range between drone and target. The sensor measure, is compared with the one obtained with MoCap.

itself, is quite good, and the target localization remains bounded in an acceptable range. As stated, the performance could be better in larger environments with greater following distance. Further tests in this direction, are advised as future work.

VIII. CONCLUSIONS

Our work, through simulations and real tests, has demonstrated and validated the functioning of the developed solution, and the potential of this new type of UWB in the field. Nevertheless, it is clear that there is still room for improvements. In particular, other refinements on the UWB tuning and on its firmware, could provide better target pose estimation and consequently better performances. Indeed, as suggested in Sec. VI, modelling the non-linearity of the PDoA w.r.t. the distance, for example developing a proper look up table, could improve the performances and the robustness. From the manufacturing point of view, the development of a pure isotropic target antenna, can improve the measurement performance to, solving the antennas mutual orientation related problems pointed in Sec. VI conclusions.

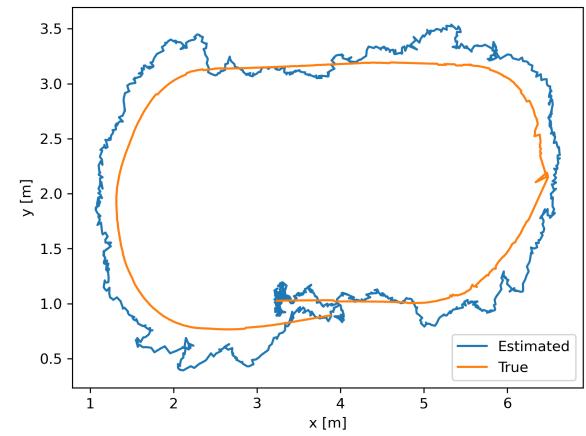


Figure 29. True and estimated pose of the target.

For what concerns our strategy, we directly use the measures without filtering and fusion. Investigation in filtering methods that also consider prior information about the motion of the target, are suggested to improve the strategy robustness. Also our PID controller could be better tuned, performing some optimization on the gains.

REFERENCES

- [1] Tej Bahadur Shahi, Cheng-Yuan Xu, Arjun Neupane, and William Guo. Recent advances in crop disease detection using uav and deep learning techniques. *Remote Sensing*, 15(9), 2023. URL [\[Ref\]](#)
- [2] Vít Krátký, Pavel Petráček, Tomáš Báča, and Martin Saska. An autonomous unmanned aerial vehicle system for fast exploration of large complex indoor environments. *Journal of Field Robotics*, 38(8):1036–1058, 2021. URL [\[Ref\]](#)
- [3] Chi Yuan, Youmin Zhang, and Zhixiang Liu. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Canadian Journal of Forest Research*, 45(7):783–792, 2015. URL [\[Ref\]](#)
- [4] Beiyang Yang and Erfu Yang. A survey on radio frequency based precise localisation technology for uav in gps-denied

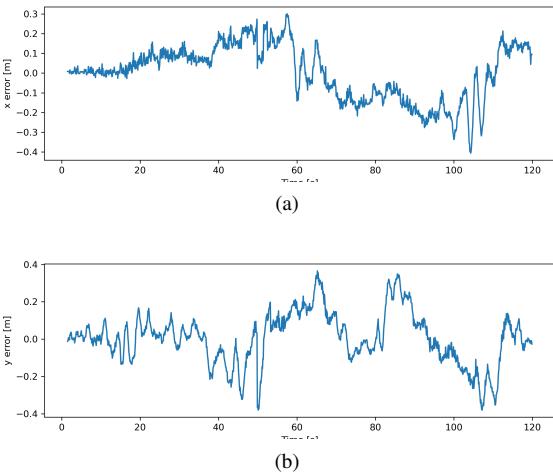


Figure 30. (a) target localization x error [m]. (b) target localization y error [m].

- environment. *Journal of Intelligent and Robotic Systems*, 103, 10 2021. URL [\[Ref\]](#)
- [5] Guoquan Huang. Visual-inertial navigation: A concise review. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9572–9582, 2019. URL [\[Ref\]](#)
 - [6] Syed Agha Hassnain Mohsan, Nawaf Qasem Hamood Othman, Yanlong Li, Mohammed Alsharif, and Muhammad Khan. Unmanned aerial vehicles (uavs): Practical aspects, applications, open challenges, security issues, and future trends. *Intelligent Service Robotics*, 01 2023. URL [\[Ref\]](#)
 - [7] Meisam Kabiri, Claudio Cimarelli, Hriday Bavle, Jose Luis Sanchez-Lopez, and Holger Voos. A review of radio frequency based localisation for aerial and ground robots with 5g future perspectives. *Sensors*, 23(1), 2023. URL [\[Ref\]](#)
 - [8] Yan Zhang and Linfu Duan. A phase-difference-of-arrival assisted ultra-wideband positioning method for elderly care. *Measurement*, 170:108689, 2021. URL [\[Ref\]](#)
 - [9] Luca Santoro, Matteo Nardello, Marco Calliari, William Cechin Guarienti, Giordano Luchi, Daniele Fontanelli, and Davide Brunelli. Catch-me-if-you-can infrastructure-less uwb-based leader-follower system for compact uavs. In *2022 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pages 1–7, 2022. URL [\[Ref\]](#)
 - [10] Guang Xue, Hanchen Yao, Yuanchao Zhang, Jiangyin Huang, Lili Zhu, and Houde Dai. Uwb-based adaptable side-by-side following for human-following robots. In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 333–338, 2022. URL [\[Ref\]](#)
 - [11] Hui Chen, Tarig Ballal, Nasir Saeed, Mohamed-Slim Alouini, and Tareq Y. Al-Naffouri. A joint tdoa-pdoa localization approach using particle swarm optimization. *IEEE Wireless Communications Letters*, 9(8):1240–1244, 2020. URL [\[Ref\]](#)
 - [12] Glenn MacGougan, Kyle O’Keefe, and Richard Klukas. Ultra-wideband ranging precision and accuracy. *Measurement Science and Technology*, 20(9):095105, jul 2009. URL [\[Ref\]](#)
 - [13] Vincenzo Di Pietra, Paolo Dabone, Marco Piras, and Andrea Maria Lingua. Evaluation of positioning and ranging errors for uwb indoor applications. In *International Conference on Indoor Positioning and Indoor Navigation*, 2019. URL [\[Ref\]](#)
 - [14] Qinglin Tian, Kevin I-Kai Wang, and Zoran Salcic. Human body shadowing effect on uwb-based ranging system for pedestrian tracking. *IEEE Transactions on Instrumentation and Measurement*, 68(10):4028–4037, 2019. URL [\[Ref\]](#)
 - [15] Open Robotics. *ROS2 Humble Documentation*, 2024. URL [\[Ref\]](#)
 - [16] Giorgio Pellegrini and Elias Fontanari. *Leader-follower UWB*, 2024. URL [\[Ref\]](#)
 - [17] ZP Yang. *Qualisys_ros node*, 2022. URL [\[Ref\]](#)
 - [18] Qorvo. *QM33120WDK1 product page*, 2022. URL [\[Ref\]](#)
 - [19] Qorvo. *QM33120WDK1 Datasheet*, 2022. URL [\[Ref\]](#)
 - [20] IEEE. Ieee standard for low-rate wireless networks. *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pages 1–800, 2020. URL [\[Ref\]](#)
 - [21] IEEE. Ieee standard for low-rate wireless networks—amendment 1: Enhanced ultra wideband (uwb) physical layers (phys) and associated ranging techniques. *IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020)*, pages 1–174, 2020. URL [\[Ref\]](#)