# Synchronization of DJI Tello Group Flights Indoors

Olha Pohudina[1,2,*,†], Antonietta Sivo[1,†], Nicola Cordeschi[1,†], Luigi Alfredo Grieco[1,†] and Michele Lagioia[1,†]

[1]Department of Electrical and Information Engineering, Politecnico di Bari, Bari, Italy
[2]Department of Information Technology of Design, National Aerospace University «KhAI», Kharkiv, Ukraine

**Abstract**
The synchronization of indoor collaboration for a group of Unmanned Aerial Vehicle (UAV)s presents challenges due to instability of the Global Positioning System (GPS) signal, cluttered environments, and dynamic obstacles. This paper investigates the accuracy of the navigation of two UAVs during a synchronized flight using DJI Tello UAVs. Additionally, it studies the tracking problem by introducing delays into the system to provide a more accurate representation of a real-world control scenario. In addition to the theoretical analysis, this paper presents a set of experiments in which different synchronous flight control options are tested and compared under different conditions. A Proportional Integral Derivative Controller (PID) is considered, which is adapted to control the maintenance of a predetermined distance between two UAVs. Several simulations are conducted to evaluate the performance of the above approaches. The obtained results demonstrate that the proposed methods for evaluating positioning accuracy during the execution of synchronized actions by a UAV group are applicable to future indoor localization analysis scenarios.

**Keywords**
small unmanned aerial vehicles, indoor navigation, synchronized flight, inertial measurement unit

## 1. Introduction

The coordinated operation of robots is being utilized in automation projects in high-tech industries to leverage their collective capabilities. Coordinating the control of multiple devices presents significant technical challenges, particularly when using low-cost UAV models. For instance, the DJI Tello has Visual Positioning System (VPS) and comes with a Python library that enables synchronized flight of multiple UAVs. The objective of this work is to improve the quality of DJI Tello synchronization by selecting a control architecture and configuration that ensures UAV coordination, reduces latency in sending commands, improves flight stability, and prevents possible collisions. In general, synchronization is fundamental to improve the positioning accuracy of UAVs flights, especially in indoor localization scenarios, such as those related to logistic automation [1]. For example, inspections and verification of the status of goods in autonomous warehouses is fundamental [2]. We consider the flight of two DJI Tello using three different synchronization modes, controlled by one or two ground control stations. The adopted approach combines automatic control methods, which are implemented using the DJI Tello library, and computer vision modules, which are proposed using the OpenCV library. An external Vicon system is used to monitor the position of the UAV. Synchronization refers to the alignment in timing of commands executed by each UAV in the swarm. In this case, stabilization is necessary to ensure that UAV accurately follows the desired trajectory, avoiding oscillations or unstable behavior that could jeopardize the fulfillment of the flight objective. The stability of UAV relies on the quality of its sensor suite, which for the DJI Tello includes an Inertial Measuring Unit (IMU) and two cameras. Additionally,

*IPIN-WCAL 2025: Workshop for Computing & Advanced Localization at the Fifteenth International Conference on Indoor Positioning and Indoor Navigation, September 15–18, 2025, Tampere, Finland*
*Corresponding author.
†These authors contributed equally.
o.pohudina@phd.poliba.it (O. Pohudina); a.sivo1@phd.poliba.it (A. Sivo); nicola.cordeschi@poliba.it (N. Cordeschi); alfredo.grieco@poliba.it (L. A. Grieco); m.lagioia2@studenti.poliba.it (M. Lagioia)
https://github.com/OlhaPohudina/SynchronizationDJITello/ (O. Pohudina)
0000-0001-5689-2552 (O. Pohudina); 0009-0003-0602-4185 (A. Sivo); 0000-0003-2021-835X (N. Cordeschi); 0000-0002-3443-6924 (L. A. Grieco); 0009-0000-7911-2248 (M. Lagioia)

the effectiveness of the control algorithms plays a crucial role in processing the data collected by these sensors and adjusting the motor power in real time to correct the UAV's position and orientation. As well, the wireless communication between the control station and UAVs introduces delays in data transmission, which can change the relevance of information related to position, speed, and direction of UAV. If this information is delayed due to interference or channel variability, the response of local UAV controllers may be inaccurate, jeopardizing its stability [3]. To ensure the coordination of UAVs group, it is necessary to design the communication system and the control system together. Recent studies [4] [5] have shown that there is a maximum transmission delay threshold beyond which the stability of the system can no longer be guaranteed. This value depends on the controller gain and the characteristics of the radio channel. It has also been shown that as the distance between UAVs increases, the probability of exceeding this threshold and thus violating stability increases significantly. To improve the quality of DJI Tello synchronization and ensure the coordination of UAVs during flight, it is important to ensure that the UAV is in the planned position at the appropriate time. In the current case, it is important to ensure a constant distance during the movement of two UAVs. To select a strategy for controlling synchronous actions, the following architectures are proposed for selection:

- independent control - each UAV is controlled by a separate ground station that is directly connected to it. The operator's responsibility is to send the same commands to all UAVs simultaneously. However, a situation of non-simultaneous commands execution can be possible, which depends on the operators' actions;
- centralized control - a single ground station sends the same synchronized commands to both UAVs using a common Wireless-Fidelity (Wi-Fi) network;
- leader-slave control: the slave UAV autonomously follows the leader UAV.

## 2. Architecture of synchronous flight organization

The following architectures are proposed to develop the three variants of synchronized flight, previously described. In the configurations, increasing levels of complexity and cooperation, both in network management and in coordination between UAVs, are considered. The leader UAV is visually identified by a QR code attached to its back.

In the first architecture, shown in Figure 1, each UAV operates in Access Point (AP) mode, creating its own independent Wi-Fi network. Each UAV is connected to a control station, like a computer. These two separate control stations run the same Python script in parallel to send flight commands. In this configuration, both UAVs receive the same sequence of instructions and follow the same flight plan. The synchronization of movements relies heavily on the simultaneous execution of scripts at the two control stations, without any explicit coordination mechanism between the two systems, relying solely on manual synchronization.

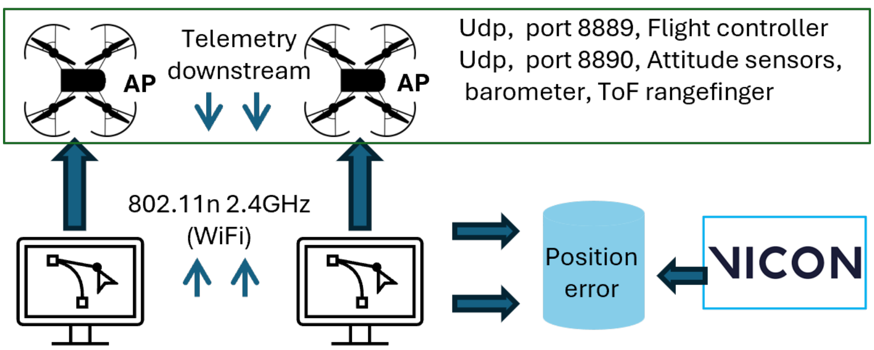


**Figure 1:** Independent control of two DJI Tello.

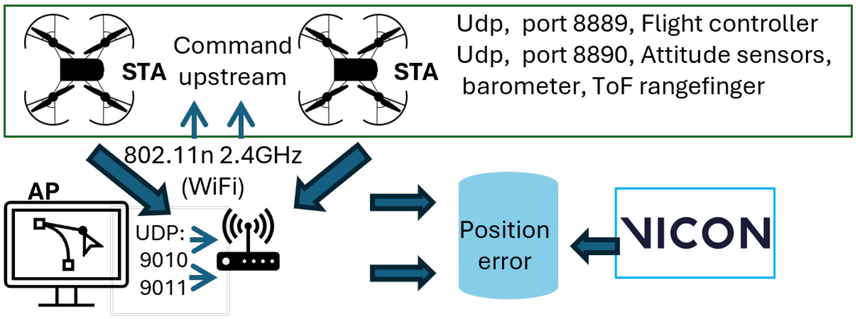


**Figure 2:** Centralized control of two DJI Tello.

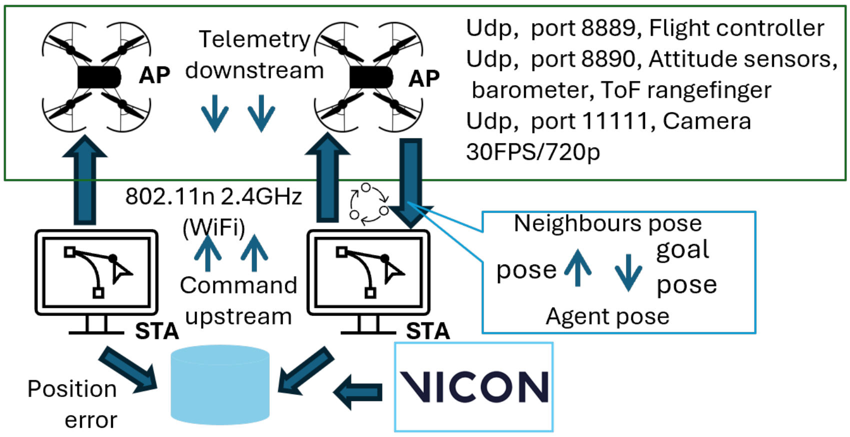


**Figure 3:** Leader-follower cooperative control.

Communication between computers and UAVs is based on the protocol User Datagram Protocol (UDP):

- commands are transmitted from the computer to the UAV via UDP port 8889;
- telemetry data transmitted by the UAV is received downstream by the host computer via UDP port 8890.

The independent architecture can be used in a scenario with a limited number of UAVs and a sufficient number of operators. In case of good operator training, it is possible to achieve high maneuvering accuracy and system responsiveness in the presence of interference and obstacles. The second architecture, shown in Figure 2, uses a centralized approach. In this scenario, two UAVs operate in Station (STA) mode and are connected to a Wi-Fi network, which is created by an external router configured as an access point [6]. In this local network, the router acts as a central hub for traffic management and provides Dynamic Host Configuration Protocol (DHCP) service, dynamically assigning an IP address to each connected device, including the host computer and the UAVs.

The ground control station is connected to the same network and sends UAV flight commands via UDP packets using port 8889. The results of the commands are sent by the UAV control system to ports 9010 and 9011 (one for each UAV). The ground station uses one Python script to transmit identical commands simultaneously. The centralized structure increases the scalability of the architecture. However, using a router as an access point can introduce additional delays and affect the frequency of receiving response data from UAV. In addition, in the presence of environmental disturbances (e.g., obstacles), UAVs may struggle to maintain synchronous flight.

The third architecture, shown in Figure 3, uses an approach based on the leader-slave model. In this scheme, the slave control program UAV recognizes the position of the QR code attached to the back of the leader UAV.

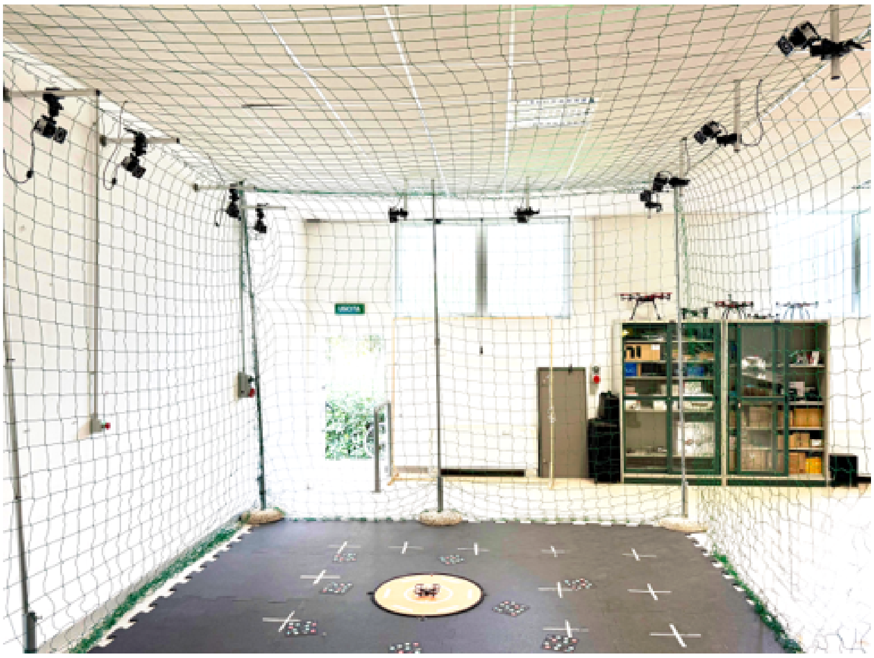

**Figure 4:** Experiment work area.

The leader operates under a script from an independent ground station, which sends flight commands via Wi-Fi, similar to previous flight organization methods. The slave UAV is connected to the second ground station, which receives a video stream on port 11111, recognizes the leader's position using a QR code, computes corrective actions, and transmits control commands to maintain a fixed distance. This architecture is more adaptive and can prevent the risk of collisions.
All three considered architectures include the Vicon system, which provides UAV positioning tracking. The experiments are carried out inside a secured area in our laboratory [7], along the perimeter of which Vicon system cameras are installed, as shown in Figure 4.

## 3. Methods for implementing autonomous flight

After evaluating various solutions for communication and control management in systems with multiple UAVs, the software implementation of synchronous modes is presented. For all methods, a control interface is implemented which assumes the following control options: take-off, landing, circle, square flights, and emergency stop. The battery charge level is proposed to be displayed, which affects the operation of the control system: a charge level below 20% disallows take-off. The thread module is used to perform flight procedures in separate threads, preventing the graphical interface from blocking during the execution of instructions. The djitellopy library, and in particular the Tello class, provides a Python interface for direct interaction with the DJI Tello UAV over the Wi-Fi network, allowing you to send commands and receive sensor data. For the flight of the slave UAV, another interface is implemented, which allows you to adjust and, if necessary, change the control coefficients. The model for detecting the QR code of the slave UAV is divided into three main layers:

- Layer 1: Pose Detection and Error Estimation. This layer is responsible for processing visual data to determine the position of the QR code and calculate errors relative to the desired pose. QR Controller receives a video stream from the front camera (CF - camera frame) and extracts the pose (position and orientation) of the QR code: $\{x, y, z, \psi\}$. Error Estimation computes errors $\{e_x, e_y, e_z, e_\psi\}$ in relation to the target pose $\{x_d, y_d, z_d, \psi_d\}$. The values of the desired pose are the following: $x_d = 0.7, \quad y_d = 0, \quad z_d = 0, \quad \psi_d = 0$. In this way, the slave UAV will be at a distance of 70 cm from the master UAV, at which the control errors are calculated as the difference between the desired pose and the detected one:

$$e_x = x_d - x, \quad e_y = y_d - y, \quad e_z = z_d - z, \quad e_\psi = \psi_d - \psi.$$

- Layer 2: PID controller. Here, the errors obtained for each axis of motion are processed. Each controller calculates a specific control action that will be sent to the actuator. PID calculates the control signal $u(t)$ for each axis ($x, y, z, \psi$) as the sum of three PID controller contributions: $u(t) = P + I + D$. Through extensive experimentation, it was possible to progressively calibrate the PID controller gains. The tuning was done dynamically by allowing real-time intervention in the parameters, allowing immediate observation of how changes affect the system's response. The control action $u(t)$ for the different axes ($x, y, z, \psi$) is composed by the proportional, integrator and derivative components. For each recommended PID the gain values are also provided, determined experimentally to ensure stable and efficient operation of the control system.
  - Proportional term (P):

    $$P_x = K_{Px} \cdot e_x, \quad P_y = K_{Py} \cdot e_y, \quad P_z = K_{Pz} \cdot e_z, \quad P_\psi = K_{P\psi} \cdot e_\psi.$$

    Recommended $K_P$ values: $K_{Px} = 0.75, \quad K_{Py} = 0.75, \quad K_{Pz} = 0.65, \quad K_{P\psi} = 0.6.$

  - Integral term (I):

    $$I_x(t) = K_{Ix} \int_0^t e_x(\tau)\, d\tau, \quad I_y(t) = K_{Iy} \int_0^t ey(\tau)\, d\tau,$$
    $$I_z(t) = K_{Iz} \int_0^t e_z(\tau)\, d\tau, \quad I_\psi(t) = K_{I\psi} \int_0^t e_\psi(\tau)\, d\tau.$$

    Recommended $K_I$ values: $K_{Ix} = 0.0012, \quad K_{Iy} = 0.0012, \quad K_{Iz} = 0.002, \quad K_{I\psi} = 0.01.$

  - Derivative term (D):

    $$D_x = K_{Dx} \cdot \frac{de_x(t)}{dt}, \quad D_y = K_{Dy} \cdot \frac{de_y(t)}{dt}, \quad D_z = K_{Dz} \cdot \frac{de_z(t)}{dt}, \quad D_\psi = K_{D\psi} \cdot \frac{de_\psi(t)}{dt}.$$

    Recommended $K_D$ values: $K_{Dx} = 0.4, \quad K_{Dy} = 0.4, \quad K_{Dz} = 0.4, \quad K_{D\psi} = 0.4.$

  The recommended values ensure the stability of the slave UAV movements and the speed when correcting errors in the distance from the master UAV, both for translational movements ($X, Y, Z$) and for rotation around the vertical axis (yaw angle $\psi$).
- Layer 3: Drive and Engine Control. The control commands, produced by the PID controller, are transformed into low-level signals necessary to actuate UAV's motors. This layer is not accessible for modification in DJI Tello UAV, it is responsible for executing the motion along the trajectory.

The accuracy of executing the motion along the trajectory will be assessed by comparing the actual trajectory with the planned one in the next section.

## 4. Experiments and results

Analyzing the trajectories of the two DJI Tello UAVs reveals how accurately they follow the desired path, highlighting any deviations and execution errors.

The planned flight path, indicated by the blue and red dotted lines in Figure 5, represents the expected results of the execution of the sent commands, while the actual flight path (blue and red) is displayed using Vicon motion capture data. The shape and continuity of the trajectories also allow us to assess the stability of the flight. In Figure 5, each column corresponds to the different experimental architecture (1- independent control, 2- centralized, and 3- cooperative). The figures at the top show circular trajectories,

**Table 1**
Average values of the position error (in cm) for Tello1 and Tello2.

| # Experiment | Routine | Tello Error 1 (cm) | Tello Error 2 (cm) |
|---|---|---|---|
| 1 | Circle | 18.0 | 38.9 |
| 2 | Circle | **16.4** | **35.4** |
| 3 | Circle | 15.3 | 28.0 |
| 1 | Square | 14.5 | 18.7 |
| 2 | Square | **10.1** | **16.1** |
| 3 | Square | 8.1 | 30.1 |

while those at the bottom show square trajectories. The position error estimated in these experiments represents the distance, frame-by-frame, between the ideal planned trajectory and the one actually flown by the UAVs. Thus, it represents a spatio-temporal error, as it relies on both the spatial distance between the two corresponding points and their accurate temporal relationship. During flight, it is evident that the error increases due to the accumulation of inaccuracies in the IMU sensors. These errors are partially compensated by the internal controller, whose calculations do not depend on the flight starting point, but on the relative position at the start of the execution of the last DJI Tello library command.

The average values of the position error, calculated in the different experiments and deduced from the previous graphs, are summarized in Table 1. It should be noted that, in Experiment 3 (leader-follower), the position error still indicates the deviation from an ideal trajectory, but it is not directly comparable with the other cases. In this context, the value of the error also includes the dynamics of adaptation of the follower to the leader's movements, influenced by the controller's response and unexpected variations in the leader's trajectory.

## 5. Conclusions

The results indicate that a good level of multi-UAV coordination can be achieved with low-cost platforms, if they are supported by a specific architecture and appropriate compensation mechanisms. Although hardware limitations cannot be eliminated, the combined use of computer vision and distributed control

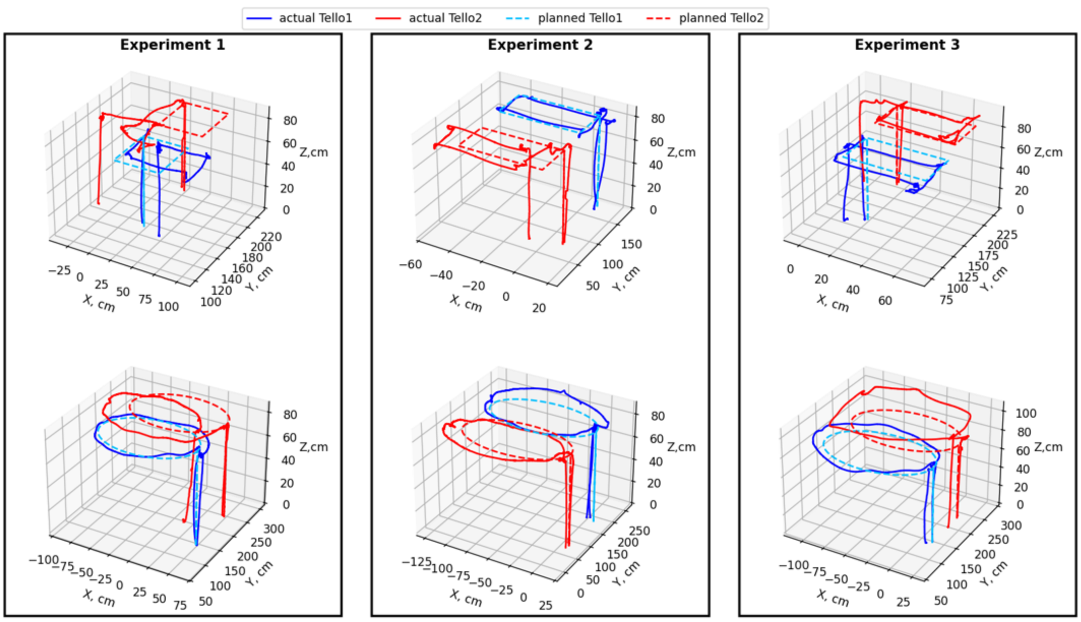


**Figure 5:** 3D Trajectories of Two Tello UAVs.

techniques can overcome many operational limitations. A hybrid solution between centralized and master-slave architectures will be proposed for further research, in which the master UAV and a group of autonomous slaves use computer vision to maintain coordination. This approach will also combine the advantages of scalability, direct control, and collision avoidance, making it suitable for more complex operational scenarios. An external independent positioning system that can compensate for IMU errors should also be added. The obtained results demonstrate that the proposed methods for evaluating positioning accuracy during the execution of synchronized actions by a UAV group are applicable to future indoor localization analysis scenarios. In particular, they can be used effectively in systems that integrate visual data, IMU sensors, and external positioning measurement technologies [8]. This will allow us to estimate both the average positioning error and the latency when synchronous commands are executed.

## Acknowledgments

Project co-funded by the European Union – Next Generation Eu - under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 3.3 - Decree No. 117 (2nd March 2023) of Italian Ministry of University and Research - Concession Decree No. 2332 (22nd December 2023) of the Italian Ministry of University and Research, Project code D93D23000380003, within the Italian National Program PhD Programme in Autonomous Systems (DAuSy).

## Declaration on Generative AI

During the preparation of this work, the author(s) used X-GPT-4 and Gramby in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

## References

[1] K. Kim, S. Kim, J. Kim, H. Jung, Drone-assisted multimodal logistics: Trends and research issues, Drones 8 (2024). URL: https://www.mdpi.com/2504-446X/8/9/468. doi:10.3390/drones8090468.

[2] H. Liu, Q. Chen, N. Pan, Y. Sun, Y. An, D. Pan, Uav stocktaking task-planning for industrial warehouses based on the improved hybrid differential evolution algorithm, IEEE Transactions on Industrial Informatics 18 (2022) 582–591. doi:10.1109/TII.2021.3054172.

[3] T. Zeng, M. Mozaffari, O. Semiari, W. Saad, M. Bennis, M. Debbah, Wireless communications and control for swarms of cellular-connected uavs, in: 2018 52nd Asilomar Conference on Signals, Systems, and Computers, 2018, pp. 719–723. doi:10.1109/ACSSC.2018.8645472.

[4] Z. Ding, X. Wang, C. Cai, L. Jia, Z. Xu, Multi-uav intelligent decision-making method with layer delay dual-center mappo for air combat, Applied Intelligence 55 (2025) 811.

[5] T. Zeng, O. Semiari, W. Saad, M. Bennis, Joint communication and control for wireless autonomous vehicular platoon systems, IEEE Transactions on Communications 67 (2019) 7907–7922. doi:10.1109/TCOMM.2019.2931583.

[6] O. Pohudina, M. Kovalevskyi, M. Pyvovar, Group flight automation using tello EDU unmanned aerial vehicle, in: 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), volume 2, 2022, pp. 151–154. URL: https://ieeexplore.ieee.org/document/9648704. doi:10.1109/CSIT52700.2021.9648704, ISSN: 2766-3639.

[7] Telematics lab – politecnico di bari, https://telematics.poliba.it/, 2025. Accessed: 2025-08-13.

[8] M. Martens, M. U. De Haag, Uwb-based localization of suas swarms as part of an indoor u-space evaluation range, in: 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), 2024, pp. 1–7. doi:10.1109/DASC62030.2024.10749585.