CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Journal Paper

A Drone Secure Handover Architecture validated in a Software in the Loop Environment

Enio Filho*
Filipe Gomes
Stéphane Monteiro
Sérgio Penna*
Anis Koubâa*
Eduardo Tovar*

*CISTER Research Centre

CISTER-TR-230604

2023/06/29

A Drone Secure Handover Architecture validated in a Software in the Loop Environment

Enio Filho*, Filipe Gomes, Stéphane Monteiro, Sérgio Penna*, Anis Koubâa*, Eduardo Tovar*

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: enpvf@isep.ipp.pt, 1181059@isep.ipp.pt, 1180697@isep.ipp.pt, sdp@isep.ipp.pt, aska@isep.ipp.pt, emt@isep.ipp.pt

https://www.cister-labs.pt

Abstract

The flight and control capabilities of uncrewed aerial vehicles (UAVs) have increased significantly with recent research for civilian and commercial applications. As a result, these devices are becoming capable of flying ever greater distances, accomplishing flights beyond line of sight (BVLOS). However, given the need for safety guarantees, these flights are increasingly subject to regulations. Handover operations between controllers and the security of the exchanged data are a challenge for implementing these devices in various applications. This paper presents a secure handover architecture between control stations, using a Software in the Loop (SIL) model to validate the adopted strategies and mitigate the time between simulation and real systems implementations. This architecture is developed in two separate modules that perform the security and handover processes. Finally, we validate the proposed architecture with several drone flights on a virtual testbed.

A Drone Secure Handover Architecture validated in a Software in the Loop Environment

A Drone Secure Handover Architecture validated in a Software in the Loop Environment

Enio Vasconcelos Filho[1], Filipe Gomes[1,2], Stéphane Monteiro[1,2], Ricardo Severino[1,2], Sergio Penna[1], Anis Koubaa[1,3], Eduardo Tovar[1]

[1]CISTER – Research Centre in Real-time Embedded Computing Systems, Instituto Superior de Engenharia do Porto, Rua Alfredo Allen 535, 4200-135 Porto, Portugal;
[2] PORTIC, Polytechnic Institute of Porto, Portugal;
[3]Prince Sultan University, Saudi Arabia;

E-mail: enpvf@isep.ipp.pt, 1181059@isep.ipp.pt, 1180697@isep.ipp.pt, sev@portic.ipp.pt, sdp@isep.ipp.pt, akoubaa@psu.edu.sa, emt@isep.ipp.pt

Abstract. The flight and control capabilities of uncrewed aerial vehicles (UAVs) have increased significantly with recent research for civilian and commercial applications. As a result, these devices are becoming capable of flying ever greater distances, accomplishing flights beyond line of sight (BVLOS). However, given the need for safety guarantees, these flights are increasingly subject to regulations. Handover operations between controllers and the security of the exchanged data are a challenge for implementing these devices in various applications. This paper presents a secure handover architecture between control stations, using a Software in the Loop (SIL) model to validate the adopted strategies and mitigate the time between simulation and real systems implementations. This architecture is developed in two separate modules that perform the security and handover processes. Finally, we validate the proposed architecture with several drone flights on a virtual testbed.

1. Introduction

Uncrewed Aerial Vehicles (UAVs), also known as drones, are an increasingly popular area of interest for researchers due to their growing utility and vast array of applications. The miniaturization of computing systems, innovations in communication technologies, and advancements in UAV flight controllers have enabled the development of complex UAV systems that cater to multiple commercial, industrial, and government applications [9]. As such, drones are expected to outstrip any other user base in the aviation market [4].

The non-existence of a single model for drones in delivery systems presents a vast field for research. Among these models, the flight control systems have been widely studied, while the architectures that promote flight capability beyond the line of sight (BVLOS) still have few studies. These flights require safety guarantees, which include the existence of a flight authority, usually playing a role of Ground Station (GS). The GS is responsible for instructing the Drone about the actions to be taken during the delivery mission and is responsible for acting in case of any failure. It establishes constant communication with the drone, ensuring control and precision over its actions.

The most used way to guarantee this communication is by applying a handover system [16], that can be defined as vertical or horizontal handover model. Horizontal handover is characterized by the exchange of radio or communication links within the same network, while vertical handover uses frequency exchange [7]. The second handover model is based on the exchange of Drone control authority when needed. Thus, it assumes the existence of two or more GS, distant from each other and which can take control over the Drone and finalize the delivery process, taking responsibility for its actions. The use of different GS presents an additional

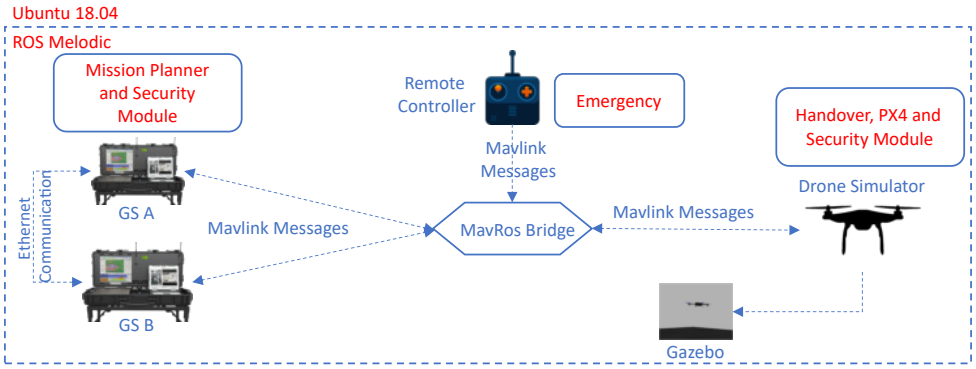

Figure 1. Simulation Architecture

challenge in the implementation of BVLOS flights since, besides the Drone's control needs, it is necessary to have a structure that guarantees the security of communications, avoiding the presence of a GS that is not part of the process and that may jeopardize the mission completion.

A critical aspect for ensuring the reliability of the operations performed by drones comes from their validation platforms. Therefore, several works have already been done in this area, such as the one presented in [10], that started to create a realistic environment for evaluation of drones controllers. This work introduced the MAV3DSim, with a fixed-wing or a quadrotor. However, this simulator does not address the drone communication protocol, the Mavlink [8]. A similar platform was proposed by [5], with RotorS presenting a realistic framework but introducing the Mavlink protocol. This simulator also integrates with the Robot Operating system (ROS), increasing its applicability due to integration flexibility. However, the most used UAV simulator platform is PX4 [13]. This simulator includes several drone models and a flexible platform based on ROS and Mavlink. Several works have been done on this platform due to its accuracy in physical representation and usability [11, 6].

In this context, the contributions of this paper can be described as follows:

- Development of a secure communication architecture between the Drone and two GS, with the capacity to perform handover between them and safety guarantees in case of communication failures.
- Validation of the handover architecture using an open-source simulator, PX4, with the implementation of a SIL model. This model aims to minimize the differences between the system implemented in the simulation and the one to be implemented in the actual drone.
- Demonstrate a series of Security measures to increase drone's operation trustability and reliability.

The rest of this work is structured as follows: In Section 2, we present the general simulation architecture, while in Section 3, we present the Software in The Loop design, with its modules and their functions. The validation process of the architecture and the Handover is presented in Section 4. Finally, the conclusions and future works are presented in Section 5.

2. Simulation Architecture

The simulator chosen to implement this architecture was PX4 because of its flexibility in terms of programming and its integration with ROS and Gazebo and open-source code. Thus, besides choosing a drone model mimicking the characteristics of an accurate model, the modules built in ROS can be reused in a real scenario. The flight controller used is a simulation of the PixHawk [13] controller, providing a standard for delivering hardware support and software stack, allowing hardware and software ecosystems to build and maintained in a scalable way

[12]. In this simulation, the flight controller can be operated by a remote control, manually operated in case of failures, or by a GS. In both cases, the communication is performed using the Mavlink protocol, commonly used in drone control [3], to standardize this communication between the equipment modules. As the PX4 is integrated with the ROS, we also use it to encapsulate these messages using MavRos, which translates the Mavlink data to the model to be transmitted by any of the devices that need to send data. When the messages are delivered, the opposite path is taken, with the messages being received in MavRos, converted to MavLink, and finally provided to the flight controller. The proposed simulation architecture is presented in Fig. 1. Each GS has a mission planner application and a data security application. The drone simulator contains, besides the PX4, responsible for flight control, a module responsible for handover, and a data security module. Finally, the drone's movement is displayed in the Gazebo, a 3D visualization tool that accurately represents what physically happens with the simulation elements. The simulation was performed in an Ubuntu 18.04 environment with ROS Melodic.

In this work, it is considered that the GSs authorized to communicate with the Drone during its flight and perform the handover process communicate over a secure Ethernet network. This network allows GSs to be known and exchange access keys among themselves. These keys will be used in the reliability of the drone communication since they will serve to encrypt and decrypt the messages exchanged between GSs and Drone before, during, and after the flights. All the communication between the GSs is realized through ROS topics, and these topics are closed after the password sharing, avoiding any intrusion tentative. Each GS has a unique system ID recognized by other authorized GS and listed by the Drone controller.

The Handover and security applications have been developed in separate modules to work independently. Thus, interfaces were designed so that their implementation was independent of the flight controller and the GS chosen. This model also allowed the configuration of these modules in a separate device from the others, configuring a Software in The Loop (SIL) architecture.

3. Software in the Loop - SIL

The use of simulators to validate architectures and models of autonomous systems is a strategy that has been known for a long time. It is allied to the use of more and more natural systems that can mimic the physics of the environment very accurately. However, the differences between the devices implemented in simulation and those used in actual experiments still delay the processes and validate the models adopted [15]. In this work, we use a Software in the Loop (SIL) implementation to avoid common testing problems by keeping in simulation the devices and algorithms that do not require validation. At the same time, we use the same hardware that will be used in the real device to validate the security handover module. This application combines the fidelity of the hardware being used with the flexibility and low simulation cost. Thus, code reuse can be maximized and validation time minimized [2].

On the Drone side, the integrated SIL platform acts as a layer between the PX4 controller and the interface for receiving the messages sent by the GSs. For the initial implementation, Jetson Xavier NX [14] was chosen, which will be used for the security analysis of the messages and for performing command handover between the GSs. On the GSs side, the SIL was implemented as a software block, receiving the data from the Mission Planner in the GS and transmitting it to the transmission layer. The Drone's onboard SIL development contains two independent applications, one responsible for the security and message identification, while the other handles the Handover process between the GSs. Thus, the first is called the security module (SM), while the second is the handover application (HA). This independent modeling allows the SM to be used in other applications, such as camera control and management of different drones, acting as an interface to provide security to the final application. This model also allows the same SM

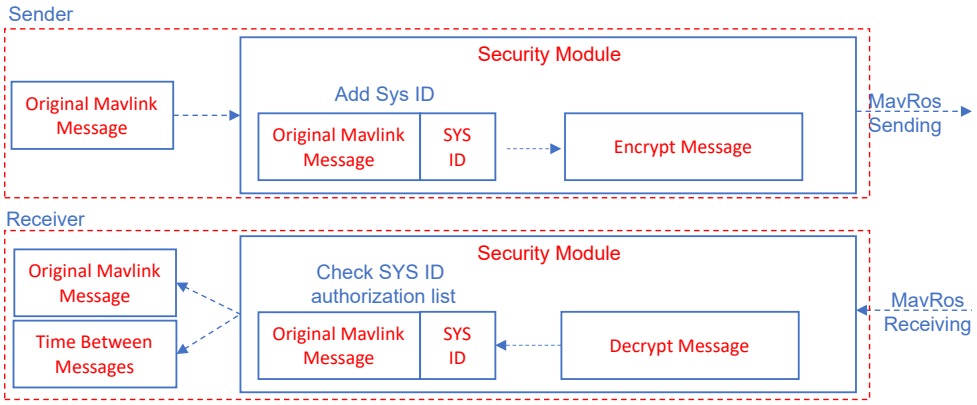

Figure 2. Security Module

application to be used in GSs, validating the messages sent by the drone.

3.1. Security Module - SM

The SM goal is to reduce the effects of malicious attacks against the system by providing reliability over untrusted communication models. Thus, the communication between Drone and GS has increased security level through its use at both ends of the communication. Its operation starts after the application defines the message sent to the recipient. At this moment, the SM is called and receives the message to be shipped, already in Mavlink format. A SYS ID is added to this package, referring to the sender. Finally, the message is encrypted using a secret key previously exchanged between the two devices. The encrypt process uses the Fernet model with symmetric encryption, which is ideal for UAV communications [1]. The encrypted message is then sent to the recipient using MavRos. Upon receiving the message, the recipient sends it to its SM, which begins the validation process, initially using the key exchanged to decrypt the received message. If the message is correctly interpreted, the SM also validates if the sender has the authorization to send the message. Only at this point, the message is passed to the HA. All this process is illustrated in Fig. 2.

To increase the quality of communication and provide safety features, the SM also provides Time Between Messages (TBM) checking. This information is delivered to the application together with the validated message. If the delay increases unexpectedly, the application can choose how to manage the process.

3.2. Handover Application - HA

The HA was developed as an introductory module to the drone flight controller. Its function is to establish the initial connection between GS and Drone, allowing the encryption key to exchange between them, verify the continuity of communication, and finally perform the exchange of flight command authority, making the handover between the GSs. The password sharing process follows the same process as the one executed between the GSs. The continuity of the communication between the modules is achieved by using a heartbeat (HB) message. After the connection is established, both sides send a signal to indicate that the communication is still open and that the signals sent are being received. In this way, the SM can validate the time between messages, and the HA can take safety actions in case of data or packet loss.

The handover process is illustrated in Fig. 3. Initially, the HA establishes communication between GS A and the Drone. After the password exchanges and the definition of the flight plan, the Drone starts from the *take-off point*. Both send HB messages to validate the communication during the flight in GS A's coverage area. After entering the area also covered by GS B, called the *Handover Region*, the HA receives messages from both GSs. The HA checks to see if GS B

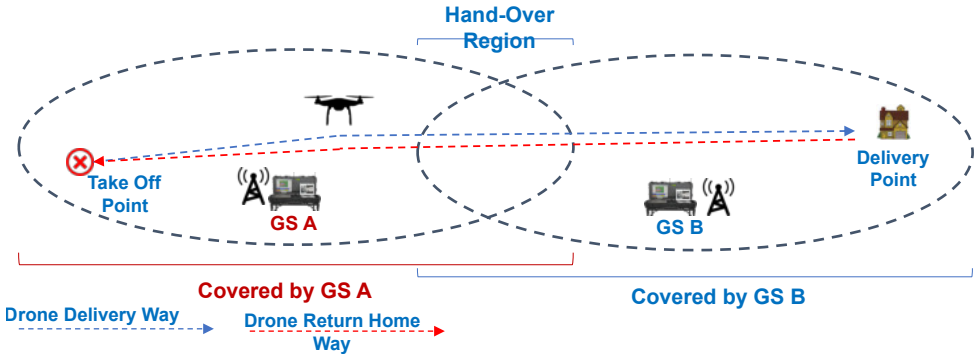

Figure 3. Handover Process

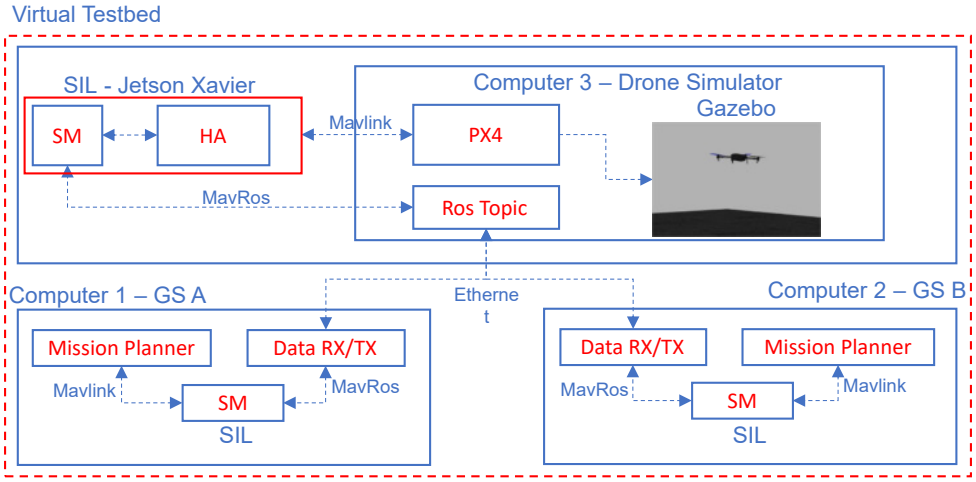

Figure 4. Virtual Testbed Implementation

is authorized to take control of the flight and, after checking, informs GS A that it will make the exchange. After confirmation, it then handles only the messages sent by GS B until it makes the handover. Finally, the delivery process ends at the *Delivery Point*. After the delivery is completed, the Drone takes off again and starts the *Return Home* mode. This process performs the inverse of the algorithm described above, beginning the flight in the GS B coverage area, flying to the *Handover Region*, validating the communication with the GS A, and ending its flight when landing at the take-off point.

4. System Evaluation

The implemented architecture is shown in Fig. 4. It offers the main modules of the virtual testbed, composed of the proposed simulations and implemented models. Then, the validation of the drone flight, including takeoff, handover, package delivery, and return home, was performed in several test missions and demonstrated in the link: https://www.youtube.com/watch?v=JAO1l1yoqYc. In this virtual testbed, the communication between the GSs and between GSs and Drone was performed over an ethernet link encapsulating the ROS topics communication. In this development, Computer 1, which includes the GS A, was defined as the ROS master, while the others were described as clients.

4.1. Handover Execution

The handover process was validated with the connection messages exchanged between the GSs and the drone, as depicted in Fig. 5. In this figure, the Drone is connected to the GS A (with

Figure 5. handover Execution

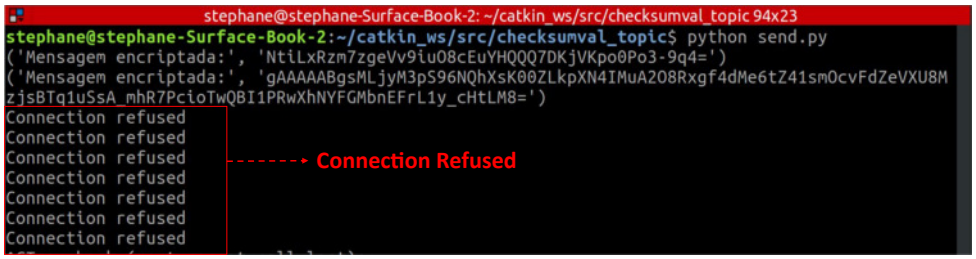

Figure 6. Refused Connection

a Sys ID 4) and then starts to receive messages from another GS, defined with a Sys ID 3). As this is the Sys ID from GS B, and it is allowed, the drone performs the handover as expected.

Thus, we evaluate the possibility of a malicious attack during this process. Initially, we tried to make the connection using a GS that did not have access to the cryptography key used. However, as expected, this connection was refused by the SM since the attempt to decrypt the message failed, indicating a possible attack attempt, as shown in Fig. 6. In a second scenario, assuming the existence of another GS on the network, able to access the encryption data and try to connect to the Drone, it will have its connection denied for not having authorization for this operation. In this case, the link is blocked by HA, which contains the list of GSs authorized to access the Drone's control during its flight.

4.2. Delay detection and Emergency Landing

During the entire flight, the SM monitors the communications, decrypting the messages and checking the delay between them. Thus, it monitors the connection status between Drone and GS and the time between messages. The SM reports only the delay between messages during the usual communication process. However, a loss of connection is indicated when this delay exceeds a previously defined threshold. The delay detection script was tested using a fake delay, simulating situations where messages would take longer than usual to be delivered. This simulation was obtained by decreasing the frequency of messages published in the topic, setting the frequency lower than accepted. This way, the companion board receiving the messages would consider it inadequate to have a safe and stable mission. It is expected that the drone would be able to detect connection delays in a real scenario. This procedure is exemplified in Fig. 7.

4.3. Packet flow measurement and evaluation

UAV systems require lightweight communication with the GSs. So, the number of packets sent through the network should below, and the size of each package should be minimal. Thus, the SM adds data to the usual Mavlink message payload, increasing its size. Under these conditions, we evaluated the packet flow between drone and GS to guarantee the architecture

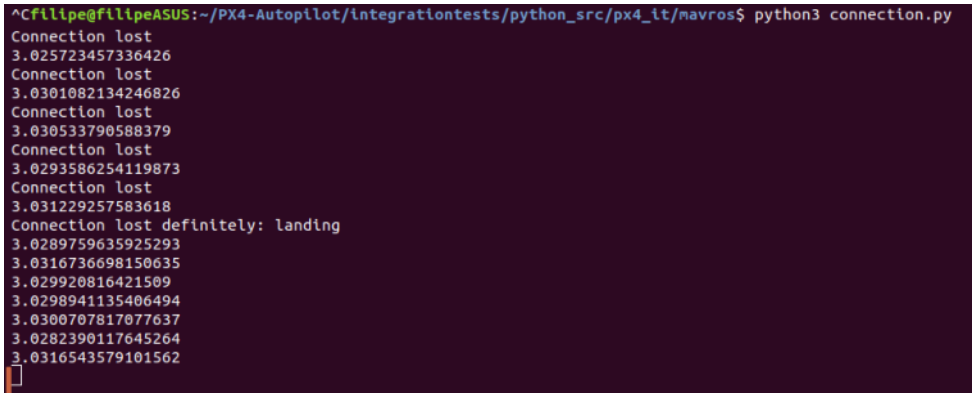

Figure 7. Inter Message Delay

viability after the SM implementation. This evaluation was performed with the WireShark packet analyzer. Two aspects of the packet flow were evaluated: the number of packets flowing through the network and the size of each package. These parameters were tested with and without the safety protocol, comparing a simple mission plan with the mission plan obtained when the developed safety protocol is applied. The results are presented in Table 1, showing that, even though the average packet size did not increase significantly, the number of packets exchanged between GS and Drone increased by 21%, given the presence of HB-type messages. This implies an increase in the average throughput in the network, caused by the increased security of the communications. The results match the expected as the implementation of the SM would increase the packet size. However, despite this slight increase the network traffic, the communication is still reliable. There are no significant changes compared to a standard mission plan with a mission plan with the wireless safety layer.

Table 1. Mission Plan Packet Evaluation

Data	Packets per Minute	Mean Packet Size (bytes)
Standard mission plan	161	433.2
Mission plan with SM	196	435.3

5. Conclusions and Future Works

This paper presents a safe and secure architecture for performing handover in BVLOS flights using Drones. This architecture gave the use of two mutually independent modules, responsible respectively for message security (SM) and the validation of the authority responsible for the flight and the execution of the handover (HA). This system allows for secure communication even in a non-secure environment, providing reliability and resiliency for the drone operation. The validation of these structures relied on implementing a SIL architecture, which proves the independence of the modules used concerning the simulation tool and their willingness to be used in an accurate drone model. The testing process of the implemented architecture showed the applicability of the theoretical concepts presented. Furthermore, it allowed us to observe the system's behavior under conditions of simple security attacks, with the resolution of these attacks by blocking the connection. Besides this, it showed the system's ability to detect message delay conditions and take safety actions based on this detection.

As a suggestion for future work, we understand that the SM can be extended to protect many possible attacks, increasing its applicability. Furthermore, its construction allows it to be developed later as a security layer applied in other contexts, such as autonomous vehicles.

Moreover, we hope to be able to validate the application of the architecture presented here in a real testbed, validating delays and actual attacks.

Acknowledgments
This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UIDP/UIDB/04234/2020); and by FCT and the EU ECSEL JU under the H2020 Framework Programme, within project ECSEL/0010/2019, JU grant nr. 876019 (ADACORSA). The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Netherlands, Austria, France, Sweden, Cyprus, Greece, Lithuania, Portugal, Italy, Finland, Turkey. The ECSEL JU and the European Commission are not responsible for the content on this paper or any use that may be made of the information it contains.

References

[1] S. W. Bray. *Implementing Cryptography Using Python*. John Wiley & Sons, July 2020. Google-Books-ID: bLLxDwAAQBAJ.
[2] S. Demers, P. Gopalakrishnan, and L. Kant. A Generic Solution to Software-in-the-Loop. In *MILCOM 2007 - IEEE Military Communications Conference*, pages 1–6, Oct. 2007. ISSN: 2155-7586.
[3] K. Domin, E. Marin, and I. Symeonidis. Security Analysis of the Drone Communication Protocol: Fuzzing the MAVLink protocol. In *Symposia Informatic Theory Benelux*, Security, Reliability and Trust, pages 198–204, Universite du Luxemburg, 2016. Orbilu.
[4] European GNSS Supervisory Authority. *European Global Navigation Satellite Systems (EGNSS) for drones operations: white paper*. Publications Office, LU, 2019.
[5] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart. RotorS—A Modular Gazebo MAV Simulator Framework. In A. Koubaa, editor, *Robot Operating System (ROS)*, volume 625, pages 595–625. Springer International Publishing, Cham, 2016. Series Title: Studies in Computational Intelligence.
[6] J. T. Jang, A. Santamaria-Navarro, B. T. Lopez, and A.-a. Agha-mohammadi. Analysis of State Estimation Drift on a MAV Using PX4 Autopilot and MEMS IMU During Dead-reckoning. In *2020 IEEE Aerospace Conference*, pages 1–11, Mar. 2020. ISSN: 1095-323X.
[7] R. Khoder, R. Naja, N. Mouawad, and S. Tojme. Vertical Handover Network Selection Architecture for VLC Vehicular Platoon Driving Assistance. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–6, Aug. 2020. ISSN: 2166-9589.
[8] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui. Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey. *IEEE Access*, 7:87658–87680, 2019. Conference Name: IEEE Access.
[9] A. P. Lamping, J. N. Ouwerkerk, and K. Cohen. Multi-UAV Control and Supervision with ROS. In *2018 Aviation Technology, Integration, and Operations Conference*, Atlanta, Georgia, June 2018. American Institute of Aeronautics and Astronautics.
[10] I. Lugo-Cárdenas, G. Flores, and R. Lozano. The MAV3DSim: A Simulation Platform for Research, Education and Validation of UAV Controllers. *IFAC Proceedings Volumes*, 47(3):713–717, 2014.
[11] C. Ma, Y. Zhou, and Z. Li. A New Simulation Environment Based on Airsim, ROS, and PX4 for Quadcopter Aircrafts. In *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pages 486–490, Apr. 2020. ISSN: 2251-2446.
[12] L. Meier. Pixhawk | The hardware standard for open-source autopilots, 2022.
[13] L. Meier, D. Honegger, and M. Pollefeys. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6235–6240, May 2015. ISSN: 1050-4729.
[14] NVIDIA. The World's Smallest AI Supercomputer, 2021.
[15] O. Shechtman, S. Classen, K. Awadzi, and W. Mann. Comparison of Driving Errors Between On-the-Road and Simulated Driving Assessment: A Validation Study. *Traffic Injury Prevention*, 10(4):379–385, July 2009.
[16] M. Tayyab, X. Gelabert, and R. Jäntti. A Survey on Handover Management: From LTE to NR. *IEEE Access*, 7:118907–118930, 2019. Conference Name: IEEE Access.