

MIXED-SENSE: A Mixed Reality Sensor Emulation Framework for Test and Evaluation of UAVs Against False Data Injection Attacks

Kartik A. Pant, Li-Yu Lin, Jaehyeok Kim, Worawis Sribunma, James M. Goppert, Inseok Hwang

Abstract—We present a high-fidelity Mixed Reality sensor emulation framework for testing and evaluating the resilience of Unmanned Aerial Vehicles (UAVs) against false data injection (FDI) attacks. The proposed approach can be utilized to assess the impact of FDI attacks, benchmark attack detector performance, and validate the effectiveness of mitigation/reconfiguration strategies in single-UAV and UAV swarm operations. Our Mixed Reality framework leverages high-fidelity simulations of Gazebo and a Motion Capture system to emulate proprioceptive (e.g., GNSS) and exteroceptive (e.g., camera) sensor measurements in real-time. We propose an empirical approach to faithfully recreate signal characteristics such as latency and noise in these measurements. Finally, we illustrate the efficacy of our proposed framework through a Mixed Reality experiment consisting of an emulated GNSS attack on an actual UAV, which (i) demonstrates the impact of false data injection attacks on GNSS measurements and (ii) validates a mitigation strategy utilizing a distributed camera network developed in our previous work. Our open-source implementation is available at https://github.com/CogniPilot/mixed_sense

I. INTRODUCTION

In recent years, UAVs (autonomous and semi-autonomous) have proliferated in various applications such as search and rescue, disaster management, agriculture, etc. While these aerial vehicles perform their mission, they are often exposed to security threats imposed by adversaries to disrupt their operations. For instance, the recent events in the Russia-Ukraine war have demonstrated the use of cyber-warfare systems to neutralize drones actively by jamming and spoofing GNSS measurements [1]. These threats often result from vulnerabilities exhibited by sensors and communication channels of UAVs [2]. An intelligent adversary can manipulate the UAV's behavior to its advantage by injecting false data in either the sensor feed, e.g., GNSS spoofing [3], [4], or can disrupt the communication links, e.g., Command-and-Control (C2) and Vehicle-to-Vehicle (V2V) link hijacking [5], to take control over the system.

Test and evaluation of UAVs before deployment is essential to assess the impact of such adversarial false data injection attacks. This allows us to quickly identify and eliminate the effects of the system's vulnerabilities. The current methods for test and evaluation rely either on software-in-the-loop (SiTL) tests (e.g., PX4 SiTL [6]) or controlled

This research is funded by the Secure Systems Research Center (SSRC) at the Technology Innovation Institute (TII), UAE. The authors are grateful to Dr. Shreekant (Ticky) Thakkar and his team members at the SSRC for their valuable comments and support.

The authors are with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47906. Email: {kpant, lin1191, kim2153, wsribunm, jgoppert, ihwang}@purdue.edu

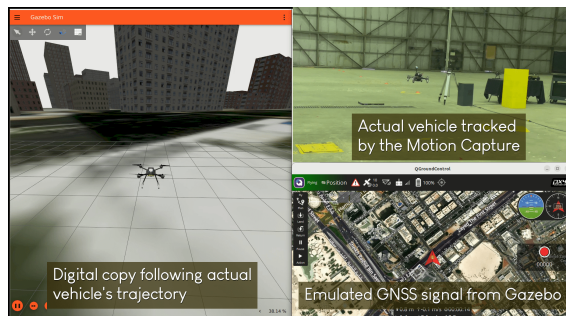


Fig. 1: An instance of Mixed-Reality-in-the-loop (MRiTL) GNSS sensor emulation

hardware tests (e.g., controlled over-the-air GNSS spoofing tests [7]). One of the major drawbacks of such approaches is their lower level of fidelity, which limits their use for cybersecurity applications. To this end, we present a Mixed Reality (MR) sensor emulation framework to analyze and mitigate the impact of false data injection attacks on UAVs. Mixed Reality has emerged as a new paradigm for vehicle-in-the-loop simulation, an extension of the traditional hardware-in-the-loop (HiTL) simulation. It represents the spectrum encompassing both Augmented Reality (AR) and Virtual Reality (VR), synergistically combining elements of the physical and digital world.

The rise of MR technology has been fueled by recent advancements in the hardware, e.g., advanced graphic processors with GPU acceleration, and software frameworks, e.g., gaming platforms such as Unreal Engine and Unity, that render high-fidelity photorealistic simulations in real-time. Several datasets, Synthia [8], virtual KITTI [9], and Blackbird [10], comprising photorealistic renderings of urban environments, are being actively used in robotics research. Another major driving force in the maturation of Mixed Reality simulations is the advancements in Motion Capture (Mocap) technologies, i.e., infrared cameras, laser tracking, etc. Motion Capture facilitates the incorporation of natural motion and behavior of actual vehicles into the simulation, along with its interaction with virtual components.

MR can play a vital role in UAV cybersecurity analysis. For example, conducting a GNSS spoofing experiment in a densely populated urban environment, e.g., Downtown Chicago, can be practically infeasible as it would disrupt the existing operations pertaining to commercial aviation, military, autonomous cars, and air taxis, which rely heavily on GNSS measurements. However, with improved sensor modeling and photorealistic urban environment rendering,

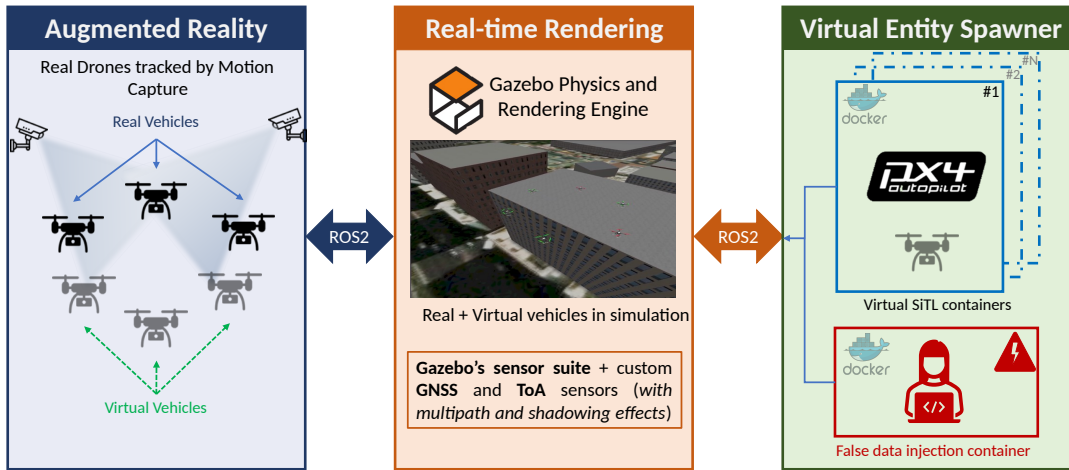


Fig. 2: Overview of our proposed framework. Gazebo’s rendering engine spawns real vehicles tracked in the motion capture and virtual vehicles running as PX4 SiTL instances. Gazebo’s physics engine performs the real-time processing of scene rendering, sensor emulation, and collision detection.

one can utilize Mixed-Reality-in-the-loop (MRiTL) experiments to characterize the impact of such attacks without actually flying the vehicle in the urban canyon. Furthermore, a Mixed Reality sensor emulation framework can be leveraged to conduct sophisticated false data injection (FDI) attacks that simultaneously impact multiple sensors and communication channels. Thus, our proposed framework provides the potential to combine the effectiveness, safety, and adaptability of simulation with real-world physics for UAV cybersecurity applications. In this paper, our main contributions are as follows.

- 1) We propose a Mixed Reality sensor emulation framework to test and evaluate UAVs against FDI attacks.
- 2) We analyze the effects of latency in sensor emulation on MRiTL experiments and provide an empirical approach to tune the emulated sensors’ latency.
- 3) We experimentally demonstrate the effectiveness of our proposed framework on an actual UAV, evaluating the impact of GNSS replay attacks and validating a mitigation strategy that utilizes external position measurements from a distributed camera network.

The rest of the paper is organized as follows. Section II reviews related work on Mixed Reality (MR) testbeds for robotics, especially focusing on UAV cybersecurity. Section III details the system architecture of the proposed framework, focusing on integrating emulated sensors with actual test vehicles. Section IV examines the impact of sensor emulation latency and provides an empirical method for tuning latency in Mixed Reality experiments. Section V presents the experimental validation of the framework, including an emulated GNSS replay attack and the assessment of mitigation strategies using external measurements. Section VI concludes the paper and outlines future research directions.

II. RELATED WORKS

The idea of Mixed Reality goes back to 1994 [11], where it was first described as a continuum between reality and

virtuality, i.e., the merging of physical and virtual objects and environments. Since then, this idea has been widely adopted in various fields of research such as manufacturing [12], robotics [13], and automotive [14]. This section will mainly focus on the recent advances in Mixed Reality for robotics and UAV applications. We also present a brief survey of various testbed developments for UAV cybersecurity.

A. Mixed Reality in Robotics

The recent advancements in 3D graphic rendering have enabled advanced graphic features such as material shading, real-time reflections, volumetric lighting, etc. These developments allow modern robotics simulators to achieve photorealistic rendering of scenes within simulation settings. Gazebo [15], known for its tight integration with the Robot Operating System (ROS), has recently been upgraded to support high-fidelity scene rendering and physics. The Unreal engine, developed initially for game development, has gained popularity in robotics because of its high-fidelity scene rendering capabilities. Based on these advancements in high-fidelity simulation frameworks, several research efforts have been made to incorporate AR and VR technologies for robotics applications [16]. There have been efforts to combine AR/VR to develop Mixed Reality frameworks for robotics [13], [17]. The closest work to our approach is FlightGoggles [18], which utilizes Unreal engine and photogrammetry techniques to generate high-fidelity photorealistic simulations for UAVs. However, their work is limited to sensor emulation of exteroceptive sensors, e.g., cameras, IR beacons, etc. They do not provide sensor emulation of proprioceptive sensors, e.g., GNSS, etc, which are more susceptible to FDI attacks.

B. Testbed for UAV Cybersecurity

Several testbeds have been developed for UAV applications. In [19], the authors have developed an open-source cloud-based testbed specifically for UAVs. It is based on Gazebo and PX4 Autopilot; however, the main focus was

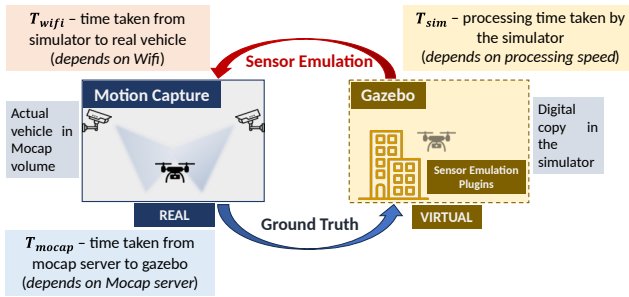


Fig. 3: End-to-end time delay from sensing the motion of the real vehicle to publishing the emulated sensor measurements.

establishing a cloud-based platform for UAV research. The authors in [20] proposed a simulation platform focusing on the cybersecurity analysis of UAVs. A SiTL and HiTL testbed for UAV cybersecurity was proposed in [21]; however, it lacks the fidelity to account for realistic scene rendering and sensor emulation. The authors in [22] presented an experimental approach for fail-secure and fail-safe testing of UAVs and Urban Air Mobility (UAM) systems. However, it is targeted explicitly for false data injection attacks on GNSS measurements and Automatic Dependent Surveillance–Broadcast (ADS-B). Most existing approaches either rely purely on simulations (lacking the required level-of-fidelity) or hardware tests targeted for specific system vulnerabilities, making them ineffective for an end-to-end test and evaluation of UAVs against FDI attacks. To this end, our proposed framework enables a comprehensive cybersecurity analysis of UAVs, combining the real-world physics of UAVs with high-fidelity sensor emulation.

III. MIXED REALITY SENSOR EMULATION

Our sensor emulation framework follows a modular architecture, as shown in Fig. 2. The three major components of the framework are (i) Augmented Reality, (ii) Real-time Rendering, and (iii) Virtual Entity Spawner. Our architecture allows the user to modify the simulation scenario by combining real and virtual components as per the test specification. The core component of our framework is the Gazebo’s physics engine, i.e., Bullet, and its rendering engine, i.e., Ogre2, which performs real-time processing of scene rendering, sensor emulation, and collision checking. Contrary to previous approaches [18], which only provide exteroceptive sensor models (such as cameras, IR beacons, and time-of-flight range sensors), in this work, we extend the sensor emulation capabilities by also including GNSS [23] and Time of Arrival (ToA) sensors (incorporating multipath and line-of-sight effects). We use Gazebo’s improved rendering capabilities to generate photorealistic renderings of the simulation environment by importing high-resolution meshes from Google Earth 3D tiles [24]. Similar to [18], our framework allows users to import the meshes generated using photogrammetry into the simulation environment. All the information related to real and virtual vehicles is combined by Gazebo and made available as ROS topics. The Augmented Reality module consists of Motion Capture

cameras tracking the motion of actual vehicles. The position and orientation data obtained via the Motion Capture system is transported to Gazebo, where digital avatars of the actual vehicles are created. These avatars are dragged inside the simulation environment following the same trajectory as the actual vehicle. The emulated sensor measurements in Gazebo are broadcasted to the actual test vehicles over the ROS2 network. The Virtual Entity Spawner module generates virtual vehicles in the simulation. Each virtual vehicle is created as a docker container instance of PX4 Autopilot firmware, replicating real-world communication and memory constraints. This allows each virtual vehicle to have independent resource allocation, memory management, and networking. The virtual vehicle uses Gazebo’s physics engine for its simulated dynamics, collision checking, and inertial measurements.

The main advantage of this approach stems from actual test vehicles experiencing environmental effects similar to those of an actual flight, e.g., multipath effects near high-rise buildings, etc., all within a controlled environment. It also allows the actual vehicle to interact with virtual components in a synchronized manner. As a result, the proposed method enables the discovery of vulnerabilities that can result from (i) environmental changes, e.g., weather, lighting, etc., adjusted via high fidelity simulations, (ii) unmodelled dynamics, e.g., aerodynamics, battery electrochemistry, motor vibrations, integrated through vehicle’s natural dynamics, and (iii) interaction of actual and virtual vehicles, e.g., communication delays in C2 and V2V link, etc. The only downside to our approach is the limited experimental motion capture space available for UAV’s motion.

IV. LATENCY IN SENSOR EMULATION

One of the significant challenges in MRiTL experiments is the latency in sensor emulation [25]. The latency is attributed to the rendering, data processing, and network overhead caused by the distributed components involved in the sensor emulation pipeline, i.e., the Motion Capture system, the ROS2 network, and the wireless connection to the actual vehicle. It is crucial to faithfully recreate signal characteristics such as latency and noise in the emulated sensor measurements, as the effectiveness of the FDI attack often relies upon such signal parameters. To achieve an accurate representation, the end-to-end latency in sensor emulation must follow a latency distribution similar to that of an actual sensor. In this work, we aim to identify the nominal latency caused by each component involved in the sensor emulation. This provides an empirical approach to tuning the end-to-end time delay¹. Figure 3 represents the critical components responsible for latency in sensor emulation using our proposed framework. The Motion Capture system tracks the actual vehicle and processes the information to generate its 6 DoF pose. Let T_{mocap} be the time a Motion Capture system takes to process the raw images to obtain a 6 DoF

¹A more detailed theoretical analysis of the latency distribution of each sensor and compensation techniques to minimize its impact on sensor emulation will be considered in our future work.

pose and transmit it over the network. Let T_{sim} denote the time the Gazebo simulator takes to render the sensor in simulation and process its data. Finally, let T_{wifi} denote the combined time taken by the Gazebo transport and the ROS2 network to actual vehicles flying in the Motion Capture volume. Thus, the total end-to-end time delay T_{total} in sensor emulation for a particular sensor can be represented as:

$$T_{\text{total}} = T_{\text{mocap}} + T_{\text{sim}} + T_{\text{wifi}} \quad (1)$$

Most commercially available autopilots handle sensor delays as part of their state estimation logic. For example, PX4 Autopilot uses an extended Kalman filter (EKF) for state estimation. The compensation for sensor delay is incorporated in the EKF’s update step. However, the nominal sensor delay must be specified as part of the UAV’s configuration settings. This allows the EKF to compensate for the effects of time delay while fusing measurements from various onboard sensors. To this end, we propose to add a constant delay T_c on top of the end-to-end time delay T_{total} to match the latency distribution of the actual sensor². We find this empirical tuning approach effective through our experimental evaluation.

V. EXPERIMENTAL DEMONSTRATION

This section describes the experimental demonstration of the proposed framework on an actual test vehicle in the MRiTL experiment. We emulate a GNSS meaconing attack to showcase the effectiveness of the proposed framework. We use a PX4 Vision drone with the PX4 Autopilot firmware as our test vehicle for experiments. The drone has a Pixhawk 6C flight controller, a UP Core companion computer, and a depth camera. We conduct our experiments at the Purdue UAS Research and Test Facility (PURT), the world’s largest indoor motion capture facility. It comprises 60 Oqus 7+ Qualisys motion capture cameras distributed across 20,000 square feet and with a 30-foot ceiling, accessible for the UAV for experimental purposes. The PX4 Vision drone is interfaced with emulated GNSS sensor measurements from Gazebo. We use a custom GNSS sensor plugin [23], which utilizes ray tracing to approximate multipath effects due to reflections from buildings and other structures rendered in the simulation. The GNSS sensor measurements are emulated using a Dell Precision-3581 workstation laptop with a 13th-Gen Intel Core i7 processor and an Nvidia RTX A1000 GPU. Figure 4 depicts the empirical value of latency in various components recorded from a 2-minute GNSS sensor emulation experiment. The nominal latency value for the Qualisys Motion Capture system with 60 Oqus 7+ cameras is 6.02 ± 0.88 ms, the processing time taken by Gazebo has a nominal value of 16.01 ± 4.45 ms, and the ping latency (one-side) of the WiFi network has a nominal value of 4.62 ± 0.98 ms. The latency depends on the position of the WiFi router and the actual vehicle. In our setup, the average distance between the UAV and router is 10 ft. We add a delay of 73

²Note that such an approach is only feasible if the end-to-end latency for an emulated sensor is less than the latency of the actual sensor.

ms to the nominal end-to-end latency of 27.42 ± 4.11 ms to match the latency value of a typical uBlox receiver, i.e., 100 ms, set in the PX4Autopilot’s configuration file [6].

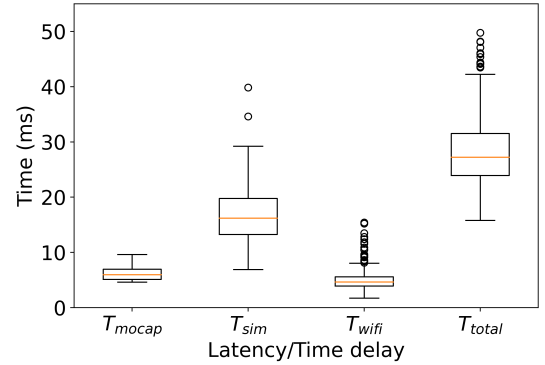
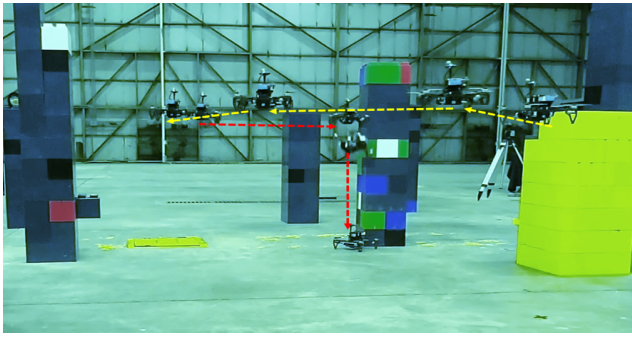


Fig. 4: Empirical latency in each component and total end-to-end latency in GNSS sensor emulation

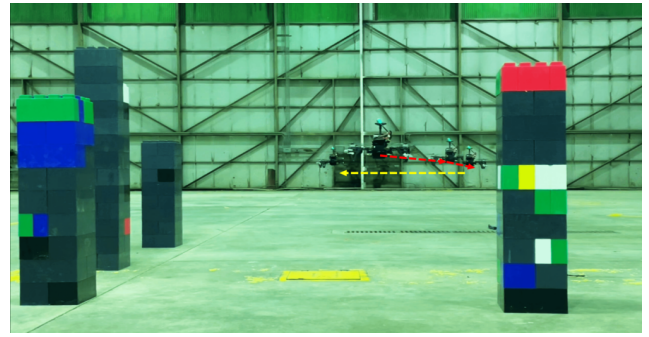
Under nominal operation, the test vehicle performs an autonomous waypoint navigation mission using PX4’s off-board control mode. We then launch an emulated GNSS replay or meaconing attack on the vehicle, injecting false data into the GNSS measurements. The details of GNSS replay/meaconing attacks are presented in Sec. V-A. The vehicle drifts off-course from its desired trajectory without triggering the onboard fault monitoring system, as shown in Fig. 5a. The PX4 Autopilot uses a chi-squared residual test for onboard fault monitoring [6]. The residual signal represents the difference between the measurements and the predicted dynamics of the system. Under nominal conditions, the residual signal follows a normal distribution, and its energy (i.e., residual squared) follows a chi-squared distribution. The chi-squared residual test compares the residual energy against a threshold to flag a faulty measurement signal. The attack is designed to deviate the UAV from the reference trajectory while maintaining the residuals below the detection threshold.

Once the onboard sensing is compromised, a possible method for attack detection is to fuse external measurement redundancies (if available) along with the compromised onboard measurements. However, such an approach is only feasible if secure external measurements are available. Even when such external measurements are available, combining multi-rate and asynchronous measurements becomes a challenging problem. To address these issues, we leverage the idea of the lifting technique for residual generation and attack detection³. We implement a linear observer-based fault detector for attack detection that fuses the external measurements with the onboard measurements to detect stealthy GNSS attacks. Section V-C highlights the details of the attack detector.

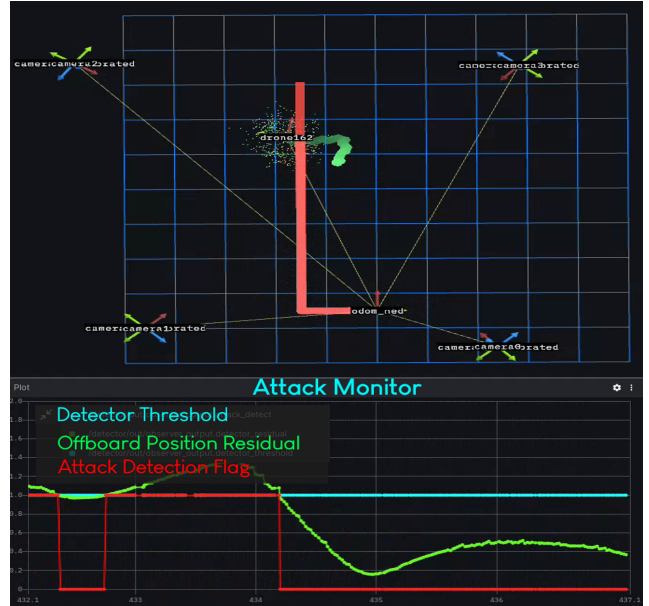
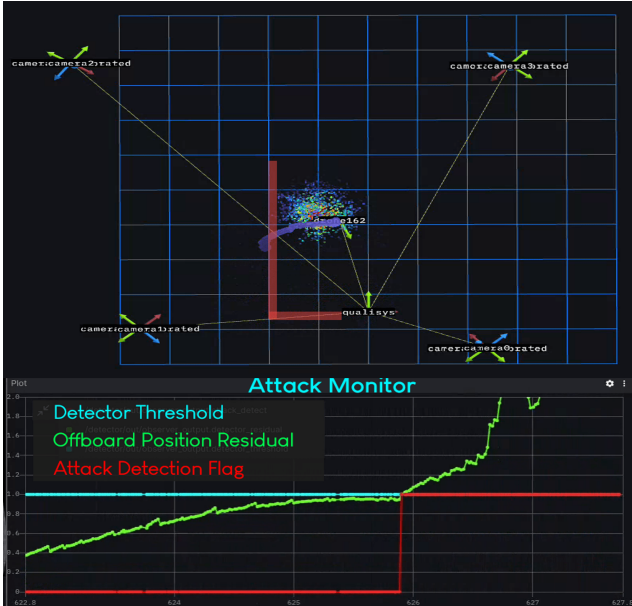
³As the main focus of this work is developing a mixed-reality framework for UAV cybersecurity, we direct interested readers to [26] for more details on lifting techniques for attack detection and mitigation



(a) Emulated GNSS meaconing attack causing the actual vehicle to crash without triggering a failsafe.



(b) Vehicle fuses external position along with GNSS measurements to correct its position once the attack is detected.



(c) (Top Down) Particle filter-based UAV detection and tracking used as external measurements for attack detection. (Run-time Monitor) A linear observer-based detector that combines onboard and external measurements to detect and mitigate attacks.

Fig. 5: Application of the proposed framework for test and validation of UAVs: Assessing the impacts of GNSS Meaconing attacks on the actual vehicle, and validating the use of external measurements for attack detection and mitigation.

For demonstration, we use a camera network as secure external position measurements for detecting stealthy GNSS meaconing attacks. The details of the camera network are explained in Sec. V-B. In real-world applications, such external measurements can be leveraged from existing infrastructure in urban environments; for example, camera surveillance systems for traffic monitoring or radar systems for air traffic control can be utilized to detect and track drones. Finally, we demonstrate measurement reconfiguration where the vehicle fuses the external position measurements with the compromised GNSS measurement shown in Fig. 5b to correct its path. This is done by publishing the external position measurements to the visual odometry topic in PX4 once the attack detector detects the attack.

A. GNSS Meaconing Attacks

We demonstrate the false data injection attacks on GNSS measurements by designing GNSS replay/meaconing attacks. Such attacks have been demonstrated in [27] [28] to be suc-

cessful even against encrypted GNSS communication, i.e., Galileo's Open Service Navigation Message Authentication (OS-NMA). In our demonstration, we assume that an attacker launches a rogue spoofer drone equipped with a GNSS receiver and a radio transmitter to replay its received GNSS messages to the victim drone. The spoofer drone flies close to the victim drone, satisfying the transmitter power constraints and directionality conditions [3]. The victim drone accepts the false data and starts following the trajectory guided by the spoofer drone. This attack is emulated using Gazebo by adding a spoofer drone inside the simulation environment. It follows a trajectory designed such that the onboard fault detection algorithm does not flag the rebroadcasted signals, i.e., residuals below a predefined threshold value.

B. Distributed Camera Network

We utilized our previous work on a distributed camera network for UAV detection and tracking [29]. It combines CNN-based object detection and particle filter tracking to de-

tect and track multiple targets. The camera network provides external position measurements, which are subsequently used by the attack detector module to detect the presence of the emulated GNSS meaconing attack. It is implemented by creating an ad-hoc mesh network of four distributed camera nodes, each consisting of an UP Squared Edge Series with an Intel Atom x7-E3950 single-board computer and an Arducam 1080p USB serial camera.

C. Offboard Attack Detector

We implement the linear observer-based fault detector presented in [26] to fuse the onboard and external position measurements. As mentioned earlier, the external measurements are obtained from the distributed camera network in our demonstration. The attack detector utilizes a lifting approach, i.e., augmentation of the states and measurements for residual generation. It has been shown to be effective for attack detection in the presence of multi-rate and asynchronous measurements.

VI. CONCLUSION

In this paper, we proposed a high-fidelity Mixed Reality sensor emulation framework for testing and evaluating UAVs against false data injection attacks. Our architecture allows an MRiTL experiment combining the physics of the actual vehicle with virtually emulated sensor measurements. We demonstrated our framework on an actual vehicle by generating an emulated GNSS meaconing attack. We also validated the use of external measurements to detect and mitigate the impact of such attacks using our framework. In the future, we will demonstrate the effectiveness of the proposed framework for testing and validating UAV swarms while adding real and virtual vehicles in tandem. This will allow us to model the large-scale interaction of swarms and assess the performance of fault detection, mitigation, and isolation algorithms through MRiTL experiments.

REFERENCES

- [1] D. Hambling, "Ukraine will spoof gps across the country to stop russian drones," *NewScientist*, 2024.
- [2] C. L. Krishna and R. R. Murphy, "A review on cybersecurity vulnerabilities for unmanned aerial vehicles," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pp. 194–199, IEEE, 2017.
- [3] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *Journal of field robotics*, vol. 31, no. 4, pp. 617–636, 2014.
- [4] M. L. Psiaki and T. E. Humphreys, "Gnss spoofing and detection," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016.
- [5] Y.-M. Kwon, J. Yu, B.-M. Cho, Y. Eun, and K.-J. Park, "Empirical analysis of mavlink protocol vulnerability for attacking unmanned aerial vehicles," *IEEE Access*, vol. 6, pp. 43203–43212, 2018.
- [6] PX4Autopilot, "Using the ecl ekf," 2022. https://docs.px4.io/main/en/advanced_config/tuning_the_ecl_ekf.html [Accessed: 02/26/2024].
- [7] H. Sathaye, M. Strohmeier, V. Lenders, and A. Ranganathan, "An experimental study of {GPS} spoofing and takeover attacks on {UAVs}," in *31st USENIX Security Symposium (USENIX Security 22)*, pp. 3503–3520, 2022.
- [8] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3234–3243, 2016.
- [9] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4340–4349, 2016.
- [10] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, and S. Karaman, "The blackbird uav dataset," *The International Journal of Robotics Research*, vol. 39, no. 10-11, pp. 1346–1364, 2020.
- [11] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE TRANSACTIONS on Information and Systems*, vol. 77, no. 12, pp. 1321–1329, 1994.
- [12] A. Malik, H. Lhachemi, and R. Shorten, "I-nteract: A cyber-physical system for real-time interaction with physical and virtual objects using mixed reality technologies for additive manufacturing," *IEEE Access*, vol. 8, pp. 98761–98774, 2020.
- [13] W. Hoenig, C. Milanese, L. Scaria, T. Phan, M. Bolas, and N. Ayanian, "Mixed reality for robotics," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5382–5387, IEEE, 2015.
- [14] M. F. Drechsler, J. Peintner, F. Reway, G. Seifert, A. Riener, and W. Huber, "Mire, a mixed reality environment for testing of automated driving functions," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 3443–3456, 2022.
- [15] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3, pp. 2149–2154, IEEE, 2004.
- [16] Z. Makhataeva and H. A. Varol, "Augmented reality for robotics: A review," *Robotics*, vol. 9, no. 2, p. 21, 2020.
- [17] I. Y.-H. Chen, B. MacDonald, and B. Wunsche, "Mixed reality simulation for mobile robots," in *2009 IEEE International Conference on Robotics and Automation*, pp. 232–237, IEEE, 2009.
- [18] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "Flightgoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6941–6948, IEEE, 2019.
- [19] M. Schmittle, A. Lukina, L. Vacek, J. Das, C. P. Buskirk, S. Rees, J. Sztipanovits, R. Grosu, and V. Kumar, "Openuav: A uav testbed for the cps and robotics community," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCP)*, pp. 130–139, IEEE, 2018.
- [20] A. Y. Javaid, W. Sun, and M. Alam, "Uavsim: A simulation testbed for unmanned aerial vehicle network cyber security analysis," in *2013 IEEE globecom workshops (gc wkshps)*, pp. 1432–1436, IEEE, 2013.
- [21] J. Goppert, A. Shull, N. Sathymoorthy, W. Liu, I. Hwang, and H. Aldridge, "Software/hardware-in-the-loop analysis of cyberattacks on unmanned aerial systems," *Journal of Aerospace Information Systems*, vol. 11, no. 5, pp. 337–343, 2014.
- [22] S. Siewert, K. Sampigethaya, J. Buchholz, and S. Rizor, "Fail-safe, fail-secure experiments for small uas and uam traffic in urban airspace," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, pp. 1–7, IEEE, 2019.
- [23] K. A. Pant, Z. Yang, J. M. Goppert, and I. Hwang, "An open-source gazebo plugin for gnss multipath signal emulation in virtual urban canyons," in *AIAA SCITECH 2023 Forum*, p. 2586, 2023.
- [24] GoogleEarthTiles3D, "Google earth map tiles," 2022. <https://developers.google.com/maps/documentation/tile> [Accessed: 02/26/2024].
- [25] M. Schnierle and S. Röck, "Latency and sampling compensation in mixed-reality-in-the-loop simulations of production systems," *Production Engineering*, vol. 17, no. 3-4, pp. 341–353, 2023.
- [26] D. Sun, M. Cho, and I. Hwang, "On attack detection and identification for the cyber-physical system using lifted system model," *arXiv preprint arXiv:2212.04396*, 2022.
- [27] M. Motalleighbghomi, H. Sathaye, M. Singh, and A. Ranganathan, "Location-independent gnss relay attacks: A lazy attacker's guide to bypassing navigation message authentication," *ACM WiSec 2023*, 2023.
- [28] M. Lenhart, M. Spanghero, and P. Papadimitratos, "Relay/replay attacks on gnss signals," in *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 380–382, 2021.
- [29] Z. Yang, J. M. Goppert, and I. Hwang, "Target tracking system for urban counter-uas using a camera network," in *AIAA SCITECH 2022 Forum*, p. 1474, 2022.