A Modular High-Fidelity Simulation Framework for Human–UAV Interaction Research in Built Environments

Guohao Wang[1,*], Abdul-Majeed Mahamadu[1], Vijay M. Pawar[1], Honghu Chu[1,3], Diran Yu[1]

[1]Bartlett School of Sustainable Construction, University College London, United Kingdom
[2]College of Civil Engineering and Architecture, Zhejiang University of Water Resources and Electric Power, China

*ucbvanb@ucl.ac.uk

Abstract -

The growing use of unmanned aerial vehicles (UAVs) in built environments has created a need for simulation tools that support realistic and systematic human–UAV interaction (HUI) research. However, many existing HUI studies still rely on simulation setups that simplify flight control, onboard computation, sensing, or environmental conditions, limiting the ecological validity and transferability of their findings. In addition, existing UAV simulators are rarely designed with human interaction as a central concern. This paper presents a modular high-fidelity simulation framework for HUI research in built environments. The framework integrates closed-loop flight control, onboard computation, multimodal interaction, and communication-aware system integration within a unified simulation architecture. A tunnel inspection case study is used to demonstrate the feasibility of implementing the framework in a representative built-environment scenario involving human-in-the-loop UAV operation under GPS-denied conditions, real-time LiDAR-based odometry, and realistic system constraints. The proposed framework provides a reusable and reconfigurable experimental basis for advancing HUI research in complex built environments while reducing reliance on physical experiments.

Keywords -

Human-UAV interaction; Human robot collaboration; Robot simulation; Multi-modal interface

1 Introduction

Unmanned aerial vehicles (UAVs) are increasingly deployed in a wide range of built-environment applications, such as the infrastructure inspection [1]. Their ability to access hazardous, confined, or elevated areas makes them suitable for tasks that are difficult, inefficient, or unsafe for human workers. Existing research on UAVs in these domains has predominantly focused on improving robotic capabilities, such as perception and localization [2], path planning[3], and flight control to enable more reliable and autonomous operation in complex built environments. Despite substantial progress, the structural diversity, clutter, and dynamic changes characteristic of real-world built environments still pose significant challenges to fully autonomous UAV operation [4]. In addition, the integration of UAVs into these everyday environments, particularly with respect to how human occupants and operators perceive, supervise, collaborate with, and otherwise interact with UAVs, remains relatively underexplored.

Effective deployment of UAVs in built environments requires understanding not only UAV–environment interaction but also human–UAV interaction. Human-UAV interaction have been researched from two complementary perspectives. The first concerns the interaction between UAV operators and the UAV, where research has focused on investigating the effect of different teleoperation and supervision interfaces on on workload, situation awareness, and task performance [5, 6, 7]. The second concerns the interaction between UAVs and human beings who coexist with them, such as workers and residents. These studies have emphasized perceived safety and comfort [8], appropriate distance and motion behavior in proximity to humans [9], social acceptability, and issues such as trust and perceived intrusion into personal or work spaces.

Simulation is a key enabler for the development and evaluation of human-UAV interaction systems in built environments, as it allows potentially risky scenarios to be explored safely and repeatedly at low cost. Conventional UAV simulators, however, have largely been designed around other objectives, such as flight-dynamics testing [10] or large-scale training of learning-based controllers [11, 12, 13], and typically devote limited attention to modeling human interaction in built environments. In most existing HUI studies, the UAV control loop is detached from real flight-control software, perception algorithms rely on idealized or highly simplified sensor models, or interaction takes place in gamified, static virtual environments that do not reflect the spatial complexity and variability of real built spaces. These simplifications can undermine the ecological validity of HUI experiments and limit the extent to which findings about interfaces, autonomy levels, or collaborative behaviors transfer to real-world deployments.

To address these limitations, this work proposes a modular high-fidelity simulation framework for human–UAV interaction research in built environments. The framework integrates realistic built-environment modeling with closed-loop flight control, onboard real-time computation, and communication-aware system integration by organising these core components in a modular manner, enabling systematic investigation of how humans perceive, supervise, and intervene in UAV operation under realistic environmental and system constraints. A system demonstration in a representative tunnel-inspection scenario is further presented to validate the feasibility of implementing and integrating the core components of the proposed framework in a realistic human–UAV interaction setting.

2 Related works

2.1 Human-UAV interaction research in built environments

Human-UAV interaction research has gained an increasing focus in recent years with the rising of maturity of drone technology and emergent need of construction automation. Existing HUI studies in construction can be broadly grouped into two research branches. The first branch investigates the impact of drone deployment on spatially coexisting construction workers, with a primary focus on safety and human factors. Representative studies employ immersive virtual reality (VR) environments to examine workers' safety awareness [14], physical risks arising from worker–drone proximity [9], and mental workload under different interaction levels [8] or communication modalities [7, 6]. Collectively, these studies advance the understanding of how UAVs can safely coexist with construction workers, offering empirical evidence to support risk-aware deployment, safety-oriented training, and informed design of human–drone coexistence in shared workspaces. The second branch focuses on interactions between drone operators and UAV systems, particularly in inspection and monitoring tasks. Prior research has examined operator cognitive ergonomics with intelligent assistance [5], and VR-based training systems to support skill acquisition and performance assessment [15].

Despite their contributions, existing human–UAV interaction studies commonly rely on low-fidelity or simplified simulation settings, where UAV flight control, sensor modeling, and communication are abstracted from real system implementations. As a result, such setups may fail to reflect the coupled effects of vehicle dynamics, perception uncertainty, and control latency, thereby limiting the ecological validity and transferability of interaction findings to real construction environments.

2.2 Existing UAV simulation frameworks

Simulation plays a central role in UAV research by enabling safe, repeatable, and cost-effective evaluation of control, perception, and interaction methods prior to real-world deployment. Existing UAV simulators can be broadly categorized according to their primary design objectives. Physics-based simulators and software-in-the-loop (SITL) environments, such as Gazebo [10], and PX4-based SITL frameworks, emphasize accurate multirotor dynamics and closed-loop flight control, making them suitable for validating low-level control strategies under realistic disturbances. However, their environment representations are typically geometrically simple and offer limited support for visually rich scenes or human interaction.

In contrast, visually realistic simulators built on game engines, including Unity or Unreal-based platforms and AirSim [16], provide high-quality rendering, material appearance, and lighting conditions that benefit perception-oriented studies. These platforms, however, often rely on simplified UAV dynamics and do not consistently support tight integration with real flight-control software or high-frequency, noise-aware virtual sensing. A further class of learning-oriented simulators, such as Flightmare [11], Gym PyBullet Drones [12], and PyFlyt [13], prioritizes training efficiency by abstracting environment geometry, sensing pipelines, and human input channels, which limits their suitability for interaction studies that depend on spatial and perceptual fidelity.

Overall, existing UAV simulators have been developed primarily to support specific control, perception, or learning objectives, rather than the systematic study of human–UAV interaction in realistic built environments. Moreover, most simulation platforms are not coupled with Building Information Modeling (BIM) or digital-twin representations, constraining their ability to reflect the spatial complexity and as-built conditions of construction and infrastructure assets. As a result, despite their complementary strengths, current tools rarely provide an integrated, high-fidelity simulation framework that simultaneously supports realistic built-environment modeling, autopilot-based closed-loop control, virtual sensing, and multimodal human–UAV interaction.

3 Modular high-fidelity human-UAV interaction simulation framework

This section presents a modular high-fidelity simulation framework designed to support systematic research on human–UAV interaction in built environments. As illustrated in Fig. 1, the framework comprises five core components: a multimodal user interface, a simulator, a companion computing unit (CCU), a UAV flight controller, and a communication layer. These components are organised in a modular manner and defined by functional roles rather than fixed implementations, allowing them to be replaced or reconfigured for different platforms, interaction modalities, and experimental setups. High-fidelity simulation of human–UAV interaction is achieved by incorporating

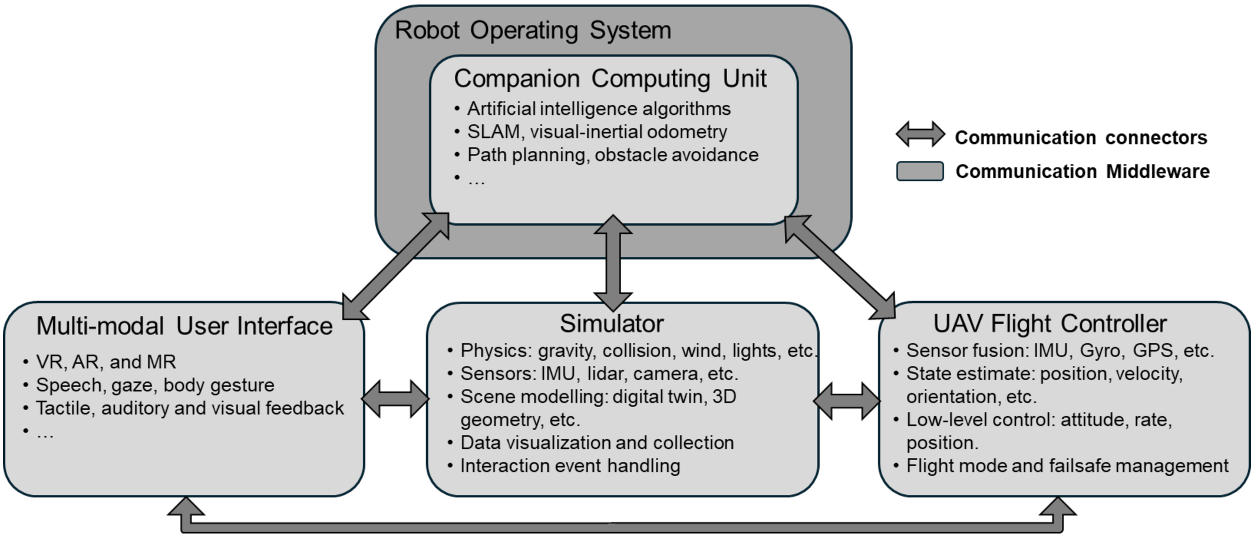

Figure 1. Overview of the proposed modular high-fidelity human-UAV interaction simulation framework

realistic system-level constraints, including flight dynamics, onboard computation, and communication-related integration. By preserving these constraints while decoupling human interaction, autonomous computation, and low-level flight control, the framework enables flexible integration and evaluation within a unified simulation environment. The following subsections describe each component in detail.

3.1 Simulator

3.1.1 Realistic simulation of built environments

Simulation of the built environment is a foundational element of human–UAV interaction research, as it constrains where the UAV can operate, what sensors can perceive, and how humans interpret distance, risk, and task progress, thereby shaping navigation strategies, inspection viewpoints, and perceived task difficulty. As built environments are often dynamic and cluttered, the way how environmental complexity is represented in simulation has a direct impact on the external validity and generalizability of experimental findings.

Built-environment simulation can be characterized along three complementary dimensions: geometric, visual, and physical representation. Geometric representation concerns the accurate modelling of spatial layout and structural elements that constrain vehicle motion and sensing; visual representation relates to surface appearance and illumination conditions that affect both human perception and vision-based algorithms; and physical representation captures the dynamic interaction between the UAV and its surroundings through gravity, rigid-body dynamics, collisions, and simplified environmental disturbances. Realistic simulation can therefore be developed by combining geometry-preserving 3D models (e.g. from BIM, CAD, or digital twins) with modern rendering and physics engines, so that both virtual sensors and human operators experience conditions representative of real built environments.

3.1.2 UAV and sensor simulation

The modeling of the UAV platform and its onboard sensors directly shapes both UAV behavior and human perception of the UAV and its surrounding environment. Flight behavior is governed by UAVs' physical structure and sensing configuration, which together influence localization capability, stability, collision avoidance, and fine motion control. Sensor data further drives low-level flight control as well as higher-level autonomy modules, and simultaneously forms the primary source of information through which operators assess safety, task progress, and system state via visual feedback and telemetry. As a result, UAV and its sensor modeling affects not only internal system operation but also the realism and validity of the resulting human–UAV interaction experience.

In typical human–UAV interaction scenarios, the aerial platform is most commonly a multirotor UAV equipped with a mission-dependent combination of sensors, such as IMUs, cameras, range sensors and lidars to support operation in both open and GPS-denied built environments [17]. A simulation framework should therefore be capable of representing common multirotor configurations and sensor suites, and of reproducing their characteristic sensing and motion constraints in a controllable manner. In the proposed framework, the UAV is modeled as a rigid body with appropriate geometric, mass, and collision properties, while detailed flight dynamics are governed by a software-in-the-loop flight controller coupled with a physics engine rather than by ad-hoc kinematic models. Onboard sensors are implemented as modular components with configurable fields of view, update rates, and noise characteristics, publishing measurements through standard ROS message interfaces. By decoupling UAV geometry and physics from sensor abstraction and from the control and autonomy stack, the framework supports flexible configuration of platforms and payloads while preserving realistic sensing and motion behavior suitable for systematic human–UAV interaction studies.

3.2 UAV flight controller

The flight controller, or autopilot, is the core onboard control system responsible for transforming sensor measurements and high-level commands from multi-modal user interfaces into actuator outputs that stabilize and navigate the UAV. Modern flight-control software (e.g., PX4, ArduPilot) integrates sensor-fusion algorithms to estimate vehicle states, such as attitude, angular rates, altitude, and velocity from multiple onboard sensors. Based on these estimates, the flight controller executes low-level control laws and manages higher-level flight modes, while also enforcing safety mechanisms such as failsafe behaviors and emergency procedures.

Accurately simulating UAV flight-control mechanisms, rather than relying on idealized or gamified motion models, is essential for capturing real system dynamics and constraints and for deriving meaningful insights into interaction quality. Identical commands may result in different closed-loop behaviors depending on the operational context (e.g., GPS-available versus GPS-denied environments) and the level of command abstraction. Continuous low-level inputs, such as joystick control, fundamentally differ from discrete high-level commands issued via language or AR-anchored waypoints. Consequently, perceived responsiveness and stability are strongly influenced by how controllers interpret, filter, and execute user inputs. Flight modes, control laws, and safety mechanisms also play a critical role in shaping operator controllability, workload, and trust, particularly in cluttered built environments.

In the proposed framework, the flight controller is integrated in a software-in-the-loop configuration and maintains bidirectional links with the simulator, the companion computing unit and the multi-modal user interface. It receives synthetic sensor inputs from the simulator and reference inputs from the companion computing unit or the multimodal user interface for state estimation and control, and returns actuator commands and estimated states to the rest of the system. Off-the-shelf SITL autopilots such as PX4 or ArduPilot can be integrated to provide realistic flight-control behavior in simulation.

3.3 Companion computing unit

The companion computing unit (CCU) is an additional onboard computer that complements the flight controller by hosting computationally demanding, higher-level functions, including vision- or LiDAR-based localization, mapping, path planning, semantic perception, and human–UAV interaction logic. In practice, the CCU is typically realized using embedded computers (e.g., Raspberry Pi) mounted on the UAV and connected to the flight controller. In human-UAV interaction studies, this layer mediates between human intent and the flight controller by interpreting user commands, combining them with task and environment information, and generating motion or behavior references that are consistent with the current autonomy mode and mission objectives.

In the proposed framework, the CCU is implemented as a ROS-based computation layer that can run either on a dedicated companion computer or on the simulation host. This allows researchers to emulate different onboard computing platforms by constraining host CPU, memory, or GPU resources, and to study how algorithm performance and interaction behavior vary under embedded-like computational budgets. The CCU interfaces with both the simulator and the flight controller by subscribing to virtual sensor streams and simulated UAV states, executing localization and planning modules, and transmitting motion references via the communication middleware introduced in Section 3.5. In parallel, it exposes ROS topics, services, or actions to multimodal interaction interfaces for command input and feedback. Through this integration, the CCU serves as a modular and replaceable core of the autonomy and interaction stack, enabling new algorithms or behaviors to be introduced without modifying the simulator or flight controller, and thereby supporting systematic human–UAV interaction studies on a stable underlying platform.

3.4 Multi-modal user interface

In human-UAV interaction research, the interface is the primary medium through which humans perceive the UAV and its environment and convey their intent back to the system. Unlike fully autonomous UAV applications, human-UAV interaction explicitly requires a human in the loop, and the design of the interaction interface often becomes the main independent variable under investigation, shaping task performance, workload, and trust [6]. A human-robot interaction interface can be decomposed into a control interface and a feedback interface [1]. The control interface captures human intent and translates it into high-level commands. The flight controller then takes these commands as references and performs the corresponding low-level closed-loop control for UAV motion execution. Common

UAV control interfaces include joysticks, touchscreens, VR controllers [15], body gestures [7] and higher-level inputs such as language [7] or gaze[18]. The feedback interface conveys information about the UAV state, environment, and task progress back to the operator. Visual feedback is typically provided through onboard camera views, 3D scene visualization, maps, and overlaid annotations, while auditory and haptic cues are increasingly used to signal proximity to obstacles, changes in autonomy level, or critical events. Recent work has begun to investigate multimodal interfaces that augment visual feedback with additional sensory channels, such as audio or haptic cues, to enhance operators' situational awareness and support performance under high-workload or cluttered environmental conditions [19].

There are mainly two ways to integrate the above interfaces into the simulation framework. One approach is to connect interaction devices through their software development kits or native APIs to the simulator, where user inputs are captured and abstracted into high-level interaction events. These events are then forwarded to the robotics stack by mapping them to ROS messages, services, or actions, decoupling device-specific details from control and autonomy logic. Alternatively, devices can be interfaced directly with ROS, where dedicated interface nodes read raw input from hardware drivers or middleware libraries and publish normalized command messages that can be consumed by simulation and flight-control modules. By adopting these patterns, a high-fidelity simulation framework remains open to a wide range of control and feedback devices while providing a structured multimodal interface layer suitable for human-UAV interaction studies.

3.5 Communication layer: middleware and connectors

To enable the coordinated operation of heterogeneous components such as simulator, flight controller, autonomy, and human interaction, these modules must communicate within a unified and consistent communication framework. Accordingly, this study incorporates a communication layer into the proposed framework to interconnect and coordinate all system components. This layer comprises a shared middleware backbone and a set of communication connectors. In the proposed framework, this backbone is organized around the Robot Operating System (ROS)[1], which offers a common message abstraction and naming scheme for exchanging state, sensor, and command information among modules. To integrate heterogeneous components that use their own native communication formats and protocols, the framework employs a set of communication connectors that translate between component-specific interfaces and ROS messages. For example, autopilots based on uORB can be connected via uXRCE-DDS or MAVROS bridges, game engine-based simulators such as Unity interface through ROS bridges, and external devices such as AR headsets can be integrated using dedicated converters (e.g., a HoloLens-ROS bridge [2]). These connectors ensure that each subsystem can communicate using its native conventions while still participating in a unified ROS-based data flow.

The design of the communication layer has direct implications for both system integration and interaction quality. From a systems perspective, reliable message routing and consistent time stamping are prerequisites for coordinating sensing, control, and interaction modules within the simulation framework. From a human-UAV interaction perspective, transport mechanisms (e.g., TCP or UDP-based links and their associated middleware configurations) influence communication latency and bandwidth, which in turn affect perceived responsiveness and stability when operators issue commands or interpret feedback. By structuring communication around ROS and dedicated message converters, the framework provides a modular and extensible data-flow architecture that tightly integrates simulation, control, autonomy, and interaction, while enabling systematic analysis and tuning of latency and reliability across experimental scenarios.

4 System demonstration

To demonstrate the applicability of the proposed framework for human–UAV interaction studies, this section presents a representative UAV-based tunnel visual inspection system, while explicitly incorporating real-time flight control, mapping and localisation, and communication-related integration that are often simplified or omitted in conventional simulation studies. At the same time, the system is designed to be extensible, allowing advanced AI-based modules and alternative interaction modalities to be integrated within the same architecture to support more intuitive human–UAV interaction and to extend the framework to other built-environment scenarios.

4.1 Unity-based simulator

Unity 2022.3.6f1 is adopted as the core simulation engine in the proposed system due to its open ecosystem, cross-platform compatibility, active developer community, and flexible authoring tools for interactive 3D content. These characteristics make it well suited for constructing visually and physically plausible built-environment scenarios while supporting tight integration with external robotics software stacks. The developed simulation environment is shown in Fig. 2.

[1] https://www.ros.org/

[2] https://github.com/soliagabriel/holoROS

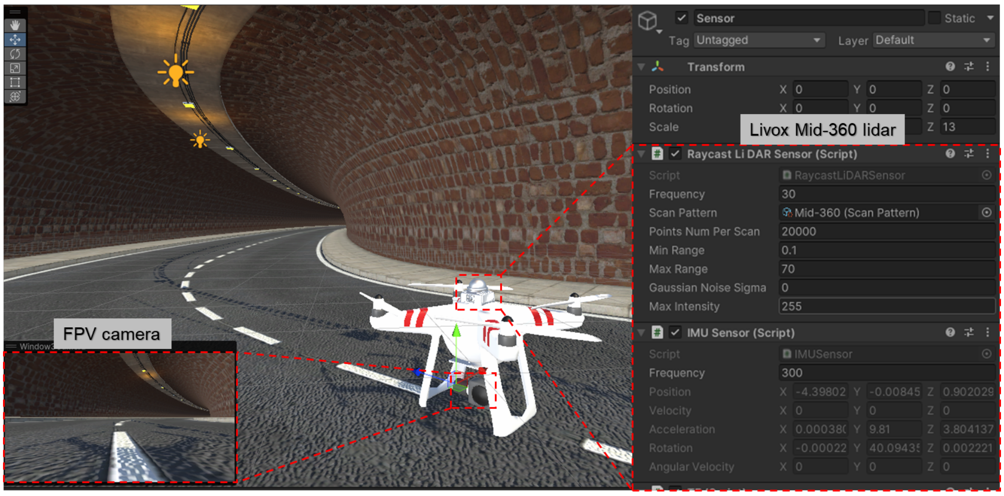

Figure 2. Unity simulation environment for UAV-based tunnel inspection

On top of Unity, the AirSim plugin [3] is used to model the UAV platform (DJI Phantom 4) and to interface with an external flight controller. AirSim provides two control modes, *SimpleFlight* and *External Flight Controller*; in this work, the latter is selected and connected to a PX4 software-in-the-loop instance to supply realistic low-level flight dynamics and control behavior. The tunnel environment is generated using the tunnel maker [4] package, which enables parametric adjustment of tunnel geometry for different inspection scenarios. Blender [5] is employed to edit UV mappings and surface materials, while Unity's Universal Render Pipeline is used for real-time rendering. Mesh-based collision geometry is used to represent physical tunnel constraints and support UAV collision interactions. In addition, LiDAR and IMU sensors are modeled via the UnitySensors [6] package to provide virtual range and inertial measurements for running a LiDAR–inertial odometry algorithm in GPS-denied tunnel conditions. Communication between Unity and ROS-based applications is realized through the ROS–TCP Connector [7] and UnitySensorROS [8] packages, which stream simulated sensor data and UAV states to the companion computing layer and receive control commands in return.

4.2 PX4-based flight control

Flight control in the demonstrated system is provided by a PX4 [9] software-in-the-loop (SITL) autopilot interfaced with AirSim via MAVROS [10]. To assess closed-loop position-control performance, a hover experiment was conducted in which a fixed position setpoint was maintained over a 20 s interval in the simulated tunnel environment. As illustrated in Fig. 3, the UAV remains tightly clustered around the commanded position with centimeter-level deviations. Quantitatively, the position

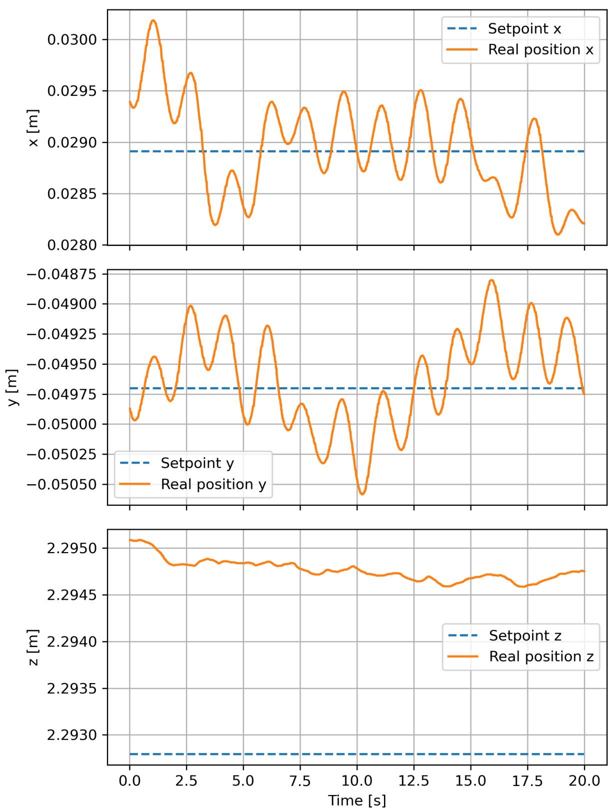

Figure 3. Result of PX4 hovering position control

[3] https://microsoft.github.io/AirSim/
[4] Unity tunnels maker asset
[5] https://www.blender.org/
[6] https://github.com/Field-Robotics-Japan/UnitySensors
[7] https://github.com/Unity-Technologies/ROS-TCP-Connector
[8] https://github.com/Field-Robotics-Japan/UnitySensorsROSArchive
[9] https://docs.px4.io/main/en/
[10] https://github.com/mavlink/mavros

tracking errors are small, with RMSE values of 0.0006 m, 0.0004 m, and 0.0078 m in x, y, and z, respectively, and a three-dimensional position RMSE of 0.0079 m. The maximum absolute errors are 0.0026 m in x, 0.0010 m in y, and 0.0534 m in z, corresponding to a maximum 3D deviation of 0.0534 m, indicating that the PX4 SITL integration provides sufficiently stable and accurate closed-loop behavior for subsequent localization and human-UAV interaction experiments.

4.3 Real-time computing for localization and mapping in GPS-denied tunnel environment

Real-time localization in the tunnel environment is provided by a Fast-LIO2 [2] LiDAR–inertial odometry module running on the ROS-based companion computing layer, which subscribes to simulated LiDAR and IMU streams from Unity and publishes pose estimates and a dense point-cloud map used by the flight controller and visualization tools. Figure 4 shows a representative example in which the reconstructed map in RViz closely matches the corresponding Unity tunnel geometry, indicating consistent localization and mapping performance under GPS-denied conditions. On an Intel® Core™ i9-14900HX CPU with 32 cores, the Fast-LIO2 process used on average 235.1% CPU (minimum 222.0%, peak 248.0%), corresponding to approximately 2.35 fully utilized cores, i.e., about 7.4% of the total processing capacity, and an average memory footprint of roughly 546 MB (peak ~699 MB). The pose topic was published at an average rate of 30 Hz (mean inter-message interval ≈ 33 ms, with observed intervals between 14 ms and 100 ms), matching the configured LiDAR frame rate and confirming that localization operates in real time within the proposed framework.

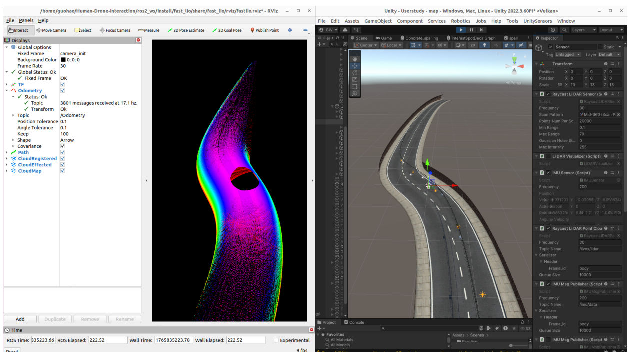

Figure 4. Mapping and localization

4.4 Joystick-based UAV control

To emulate the most common interaction modality used in real-world UAV operation, the system implements a joystick-based control interface. An Xbox controller is adopted as the input device. Flight-related joystick inputs are mapped and transmitted directly through QGroundControl to the PX4 flight controller, using the standard Mode 2 configuration, in which the left stick controls collective thrust and yaw rate, while the right stick controls pitch and roll for translational motion. In addition to basic flight control, task-related interaction inputs are handled separately through Unity's Input System. Specifically, the two trigger axes are mapped to control the pitch angle of the simulated gimbal-mounted camera, and the `A` button is assigned to trigger image capture. These inputs are processed within Unity and forwarded to the ROS-based system for execution during flight. A supplementary video demonstrates the joystick-based UAV operation within the proposed framework. Please refer to this *video link*.

5 Conclusion

This paper presented a modular high-fidelity simulation framework for human–UAV interaction research in built environments, addressing the common disconnect between interaction-focused studies and realistic UAV system constraints. By integrating built-environment simulation with closed-loop flight control, onboard computation, and modular communication, the framework enables interaction research to be conducted under system conditions that more closely reflect real-world UAV deployment. The tunnel inspection case study demonstrated the framework's ability to support human-in-the-loop operation, real-time localization, and interaction in GPS-denied environments, illustrating its suitability as a baseline experimental platform. Overall, the proposed framework shifts human–UAV interaction research from simplified, abstracted simulations toward more system-grounded and reproducible experimentation in complex built environments.

6 Acknowledgements

This work was financially supported by UCL Bartlett School of Sustainable Construction PhD studentship and the Zhejiang Provincial Natural Science Foundation (Grant No. LQN26E080033), and the Hangzhou Municipal Natural Science Foundation (Grant No. 2025SZRJJ1193).

References

[1] Sruthy Agnisarman, Snowil Lopes, Kapil Chalil Madathil, Kalyan Piratla, and Anand Gramopadhye. A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection. *Automation in Construction*, 97:52–76, 2019.

[2] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odom-

etry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022.

[3] Zhenyu Liang, Liu Yang, Zhaolun Liang, Jeff Chak Fu Chan, Zhaojie Zhang, Mingzhu Wang, and Jack C. P. Cheng. Optimized UAV view planning for high-quality 3D reconstruction of buildings using a modified sparrow search algorithm. *Advanced Engineering Informatics*, 65:103344, May 2025.

[4] Cheng Zhang, Yang Zou, Feng Wang, Enrique del Rey Castillo, Johannes Dimyadi, and Long Chen. Towards fully automated unmanned aerial vehicle-enabled bridge inspection: Where are we at? *Construction and Building Materials*, 347:128543, 2022.

[5] Fatemeh Dalilian and David Nembhard. Cognitive and behavioral markers for human detection error in AI-assisted bridge inspection. *Applied Ergonomics*, 121:104346, November 2024.

[6] Boyi Hu, Shuyan Xia, Zixian Zhu, Jiun-Yao Cheng, Yue Luo, Idris Jeelani, and Masoud Gheisari. Exploring the effect of human-drone communication modality on safety and balance control in virtual construction environments. *Ergonomics*, 68(5):723–736, May 2025.

[7] Zixian Zhu, Jiun-Yao Cheng, Idris Jeelani, and Masoud Gheisari. Using gesture and speech communication modalities for safe human-drone interaction in construction. *Advanced Engineering Informatics*, 62:102827, October 2024.

[8] Woei-Chyi Chang and Sogand Hasanzadeh. Mental workload in worker-drone communication in future construction: Considering coexistence, cooperation, and collaboration interaction levels. *Advanced Engineering Informatics*, 65:103110, May 2025.

[9] Zixian Zhu, Idris Jeelani, and Masoud Gheisari. Physical risk assessment of drone integration in construction using 4D simulation. *Automation in Construction*, 156:105099, December 2023.

[10] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2149–2154, Sendai, Japan, September 2004.

[11] Yunlong Song, Selim Naji, Elia Kaufmann, Antonio Loquercio, and Davide Scaramuzza. Flightmare: A flexible quadrotor simulator. In *Proceedings of the 2020 Conference on Robot Learning*, pages 1147–1157, 2021.

[12] Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P. Schoellig. Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[13] Jun Jet Tai, Jim Wong, Mauro Innocente, Nadjim Horri, James Brusey, and Swee King Phang. Pyflyt–uav simulation environments for reinforcement learning research. *arXiv preprint arXiv:2304.01305*, 2023.

[14] Mariusz Szóstak, Abdul-Majeed Mahamadu, Abhinesh Prabhakaran, David Caparros Pérez, and Kofi Agyekum. Development and testing of immersive virtual reality environment for safe unmanned aerial vehicle usage in construction scenarios. *Safety Science*, 176:106547, August 2024.

[15] Yu Li, Muhammad Monjurul Karim, and Ruwen Qin. A Virtual-Reality-Based Training and Assessment System for Bridge Inspectors With an Assistant Drone. *IEEE Transactions on Human-Machine Systems*, 52(4):591–601, August 2022. Conference Name: IEEE Transactions on Human-Machine Systems.

[16] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[17] Feng Wang, Yang Zou, Cheng Zhang, Joao Buzzatto, Minas Liarokapis, Enrique del Rey Castillo, and James B. P. Lim. UAV navigation in large-scale GPS-denied bridge environments using fiducial marker-corrected stereo visual-inertial localisation. *Automation in Construction*, 156:105139, December 2023.

[18] Sunwoong Choi, Zaid Abbas Al-Sabbag, Sriram Narasimhan, and Chul Min Yeum. Gaze-based Human-Robot Interaction System for Infrastructure Inspections. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9571–9577, May 2024. arXiv:2403.08061 [cs].

[19] Pengxiang Xia, Hengxu You, and Jing Du. Visual-haptic feedback for ROV subsea navigation control. *Automation in Construction*, 154:104987, October 2023.