# Simulation-in-the-Loop: Accelerating the Software Development Cycle for Multi-Agent Autonomous Drones


Andrezza de Melo Bonfim
FPFtech
Manaus, Brasil
andrezza.bonfim@fpf.br

Diógeles da Silva Tamaturgo
FPFtech
Manaus, Brasil
diogeles.tamaturgo@fpf.br

Gilberto Eduardo Santos
FPFtech
Manaus, Brasil
gilberto.santos@fpf.br

Agemilson Pimentel da Silva
TPV
Manaus, Brasil
agemilson.pimentel@tpv-tech.com



## Abstract

The deployment of autonomous Unmanned Aerial Vehicle (UAV) fleets in smart warehouses introduces significant challenges for software validation, particularly regarding multi-agent coordination and collision avoidance in GPS-denied environments. Testing these behaviors in physical industrial settings is often cost-prohibitive and risky for high-value assets. This paper presents an experience report on the development of a custom Simulation-in-the-Loop (SITL) environment built with the Godot Engine and a proprietary UDP-based communication protocol. A key feature of the platform is protocol transparency, allowing the real flight control software to interact with the virtual environment using the exact same datagrams as the physical hardware. The platform enables the concurrent execution of up to three independent agents, each maintaining isolated control pipelines and sensor feeds. Our results demonstrate that this high-fidelity virtual environment achieves high kinematic accuracy (RMSE of 0.042 m/s) and maintains sub-millisecond network latency, allowing for the early detection of race conditions and flight logic inconsistencies. By replicating DJI-specific flight dynamics and shared-space interactions, the solution accelerates the development cycle, reduces testing costs, and ensures a safer transition from code to physical warehouse operations.



## Keywords

Simulation-in-the-Loop (SITL), Software Verification and Validation (VV), Multi-UAV Systems, Industrial Warehousing, Godot Engine


## 1 Introduction

The advancement of Industry 4.0 has consolidated the use of Unmanned Aerial Vehicle (UAV) fleets for real-time inventory automation [1, 2]. However, transitioning from a single prototype to an operational multi-agent fleet is not merely an incremental step; it is an engineering bottleneck. This transition introduces a critical layer of software development complexity, necessitating the validation of collective behaviors and communication network resilience in GPS-denied environments [4].

In the context of warehousing, direct field testing with high-value assets, such as Enterprise-class drones, rapidly becomes unfeasible. The operational risks and logistical costs associated with isolating production areas drastically slow down development cycles [3]. Testing in a real-world environment is hazardous; a single physical collision could paralyze a project for weeks.

This paper reports on the development of a custom 3D simulator capable of simultaneously instantiating three drones within a shared environment. We adopted the Software-in-the-Loop (SITL) practice as a core Software Engineering strategy, enabling production code to be executed against a virtual environment that faithfully emulates physics and hardware interfaces [5, 6]. We selected the Godot Engine and a UDP infrastructure specifically to ensure the complete isolation of each agent. The core of our solution is protocol transparency: the control software is unaware that it is operating within a simulator, as it utilizes the exact same datagrams and communication logic as the physical hardware. This eliminates the need to maintain separate codebases for testing and production. By translating concurrency-focused user story requirements into a robust platform, we created an environment where interaction failures are resolved prior to testing with physical drones.

### 1.1 Industrial Context

This report stems from a strategic partnership between the Paulo Feitoza Foundation (FPFtech) and TPV (Envision) aimed at automating intralogistics processes within the Manaus Industrial Pole (PIM). From a business perspective, the need for a simulation environment was driven by strict operational constraints set by the client's logistics management. The target deployment area is a finished goods warehouse, characterized by intense forklift and personnel traffic during regular business hours. The logistics manager raised valid safety concerns regarding the risk of physical collisions, which could damage high-value enterprise assets (such as the DJI Mavic 3E) or disrupt daily supply chain operations.

To accommodate these stakeholder requirements, the FPFtech development team could not rely solely on the physical warehouse for testing. The team engineered a high-fidelity Digital Twin using the Godot Engine, shifting the 'fail-fast' point from the physical floor to the developer's workstation. This SITL platform decoupled software testing from the physical warehouse's availability, directly addressing the client's risk mitigation policies while preventing development bottlenecks.

The main contributions of this work are:

(i) the architectural design of a multi-agent SITL environment featuring protocol transparency and the Jolt physics backend to run independent control pipelines concurrently;

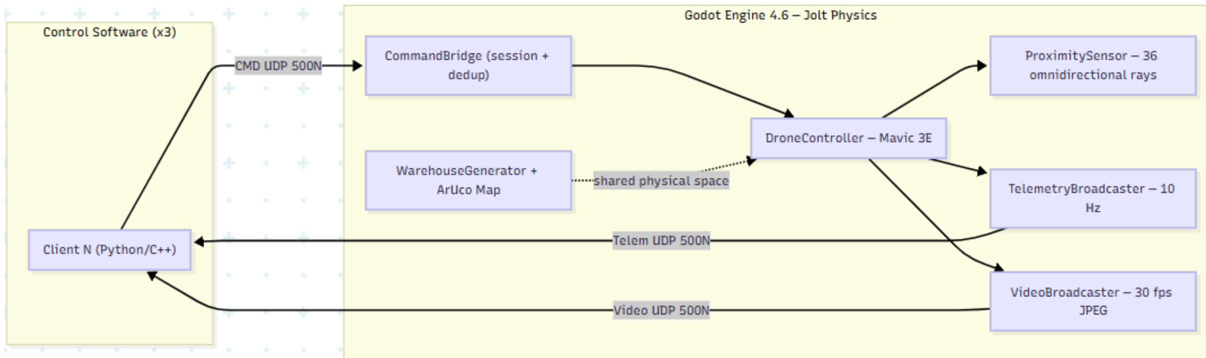


**Figure 1: Architectural design of the multi-agent SITL testing environment, demonstrating protocol transparency via isolated UDP channels (CMD, Telemetry, and Video) connected to the Godot/Jolt engine components.**

(ii) the implementation of a deterministic communication infrastructure using a port-stride strategy to synchronize telemetry, 60 FPS video feeds, and sensor data across the fleet;
(iii) a practical evaluation of how a high-fidelity simulation environment accelerates the V&V cycle and mitigates risks associated with hazardous field testing.

## 2 System Architecture and Validation Methodology

To validate the embedded software pipeline of the DJI Mavic 3E without exposing physical hardware to operational risks, we developed a multi-agent Software-in-the-Loop (SITL) simulator using the Godot Engine 4.6 with the Jolt physics backend. The core principle is *protocol transparency*: the control software emits the exact same UDP datagrams it would send to the physical drone, allowing the same codebase to be used in both simulated and production modes. An architectural overview of this infrastructure is illustrated in Figure 1.

### 2.1 UAV Digital Twin: Modeling the DJI Mavic 3E

The drone is represented as a `CharacterBody3D` with a cylindrical collision shape (radius: 1.01 m; height: 0.50 m), matching the physical envelope of the Mavic 3E including rotating propellers. Flight dynamics follow a first-order kinematic system with acceleration and drag (exponential drag coefficient of 0.92 per physics cycle). A 4S LiPo battery profile maps voltage (13.2 V–16.8 V) to charge percentage, resulting in an estimated 8-minute indoor flight endurance.

Perception is emulated through a synthetic omnidirectional proximity sensor. Every 0.5 s, the system casts 36 rays in 10° increments covering 360° horizontally, each tripled at elevation angles of -20°, 0°, and +20°. To simulate hardware imperfections, a synthetic noise (±30 mm) is injected into measurements. This data is formatted according to the DJI Obstacle Avoidance SDK, ensuring direct compatibility with the navigation algorithms.

### 2.2 Environment Digital Twin: Warehouse and Localization

The warehouse is procedurally generated using modular corridor scenes instantiated in a grid. This allows the replication of physical floor plans via inspector configuration without code changes. Shelves are assigned logical identifiers (e.g., `C02_R`) compatible with Warehouse Management Systems (WMS).

For visual localization, ArUco markers are placed on shelves and exported via a JSON map. The coordinate conversion uses a scale factor of 4.725 units/meter, derived from the physical 8-meter height of the shelves. The virtual camera is synchronized in position and Field of View (70.6°) with the Mavic 3E's physical camera, streaming 640x360px JPEG frames at 30,fps, as illustrated in Figure 2.

### 2.3 Multi-Agent Infrastructure and Communication

The simulator supports up to three concurrent agents using a port-stride strategy to ensure network isolation. Each agent is allocated three UDP channels: a Command Channel (Port 5000+10i) for JSON commands and session management; a Telemetry Channel (Port 5001+10i) streaming 10 Hz packets with 3D pose, velocity, and sensor data; and a Video Channel (Port 5002+10i) transmitting the synchronized feed. Inter-visibility is handled natively by the physics engine, enabling the validation of complex scenarios where one drone dynamically obstructs another's sensors or visual markers

### 2.4 Validation Scenarios

The reliability of the simulator as a testbed was evaluated through three scenarios:

(1) **Kinematic Fidelity:** Comparison between simulated response and an analytical model for velocity commands yielded a Root Mean Square Error (RMSE) of 0.042 m/s, confirming high motion accuracy.
(2) **Network Scalability:** Latency tests with three simultaneous agents showed an overhead of less than 0.8 ms in loopback, with zero packet loss, proving the scalability of the UDP architecture.

(3) **Collision Arbitration:** In 20 controlled intersection runs, the proximity sensors reported obstacles 2.3 s before projected collision. The physics engine registered 100% of collisions when evasion was disabled, positioning the simulator as a reliable "safety arbiter" for software builds.

## 3 Results and Discussion of Software Impact

The evaluation of the multi-agent SITL environment focused on three dimensions: kinematic fidelity, network scalability under load, and the effectiveness of the simulator as a safety arbiter for software validation.

### 3.1 Kinematic Fidelity and Determinism

To ensure that the software validated in simulation behaves consistently when deployed to the physical DJI Mavic 3E, we measured the kinematic error. A control client emitted velocity commands (`set_velocity`) for 5 seconds. The comparison between the simulated response and the analytical model yielded a **Root Mean Square Error (RMSE) of 0.042 m/s**, with a maximum deviation of 0.11 m/s during peak acceleration. This error stems from the 10 Hz telemetry quantization and is negligible for high-level mission planning.

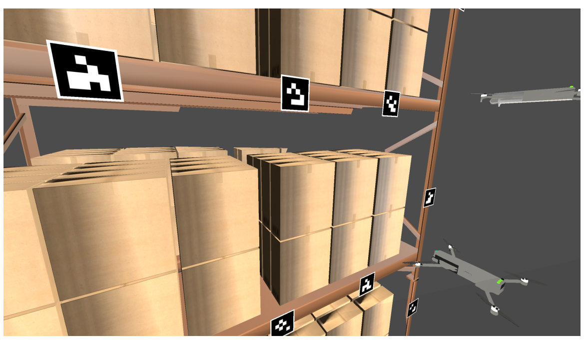

**Figure 2: Drone fleet simulation in a warehouse environment from the perspective of an onboard camera of one of the agents.**

### 3.2 Network Scalability and Multi-Agent Load

A critical requirement was the system's ability to handle multiple independent control pipelines without performance degradation. We measured telemetry latency in a loopback configuration with increasing agent counts:

- **1 Agent:** 0.31 ms latency;
- **2 Agents:** 0.47 ms latency;
- **3 Agents:** 0.79 ms latency.

The zero packet loss recorded across all configurations and the sub-millisecond overhead confirm the efficiency of the proprietary UDP architecture. For the development team, this means that network-related race conditions or synchronization issues can be isolated from flight logic bugs.

### 3.3 Collision Detection as a Safety Arbiter

The simulator's most significant impact on the V&V process is its role as a safety arbiter. In 20 controlled intersection scenarios, the proximity sensors consistently reported obstacles **2.3 s before the projected collision**.

In test runs where evasion logic was deliberately disabled, the physics engine registered collisions in 100% of the cases. From a software quality perspective, this allows the team to adopt a *"fail-fast"* approach: any software version that triggers a virtual collision is immediately flagged as failed by the test script during simulation review. This avoids exposing high-value DJI hardware to risks, reducing the cost of failure to zero during early development stages.

### 3.4 Discussion of Industrial Impact

The transition to this SITL environment significantly optimized the feedback loop for the QA and development teams. To properly contextualize the productivity gains, it is important to define the baseline measurement methodology. The project operates with a seven-person technical team (five developers and two Quality Assurance analysts) running on a three-week deployment cycle (two weeks of active development followed by one week of testing).

Prior to the SITL integration, the baseline validation process required the QA team to conduct all behavioral testing directly on the physical warehouse floor. Any logic failure discovered during these field sessions necessitated halting the operation, returning to the lab for debugging, and scheduling a new physical test. With the introduction of the digital twin, physical warehouse visits were not eliminated, but their scope fundamentally changed. The QA team now pre-validates all multi-agent scenarios in the simulator, comparing expected virtual behaviors with subsequent physical execution.

By ensuring that only mature, pre-validated software builds reach the physical drone, the trial-and-error phase in the real warehouse was drastically minimized. In practical terms, this operational shift resulted in an estimated 85% reduction in the time spent debugging and validating logic failures in the physical facility. By mitigating the need for repetitive physical arena setups and prolonged safety oversight during early testing stages, the engineering team saved approximately 16 man-hours per sprint, substantially increasing overall development velocity.

Additionally, the ability to inject synthetic noise into sensors allows for the validation of software resilience against "flaky" inputs in a way that would be difficult to reproduce deterministically in a physical setting. This combination of reduced verification effort and higher test coverage directly enhances the overall reliability of the final deployment. For the business stakeholders, particularly warehouse managers, this platform serves as a visual validation tool. They can review and approve new multi-agent behaviors in the simulator, ensuring that digital transformation initiatives proceed without introducing operational risks to the physical facility.

### 3.5 Limitations

While the multi-agent SITL environment effectively accelerates software verification, certain limitations must be acknowledged. First, the physics backend relies on a simplified first-order kinematic model rather than a full aerodynamic fluid dynamics simulation. Consequently, the environment does not emulate complex physical phenomena, such as the ground effect or the wake turbulence generated by propellers when multiple drones operate concurrently in tight, enclosed warehouse aisles. While this abstraction is perfectly sufficient for validating high-level flight logic, concurrency, and protocol transparency, low-level flight controller stability under turbulent conditions must still be verified in physical tests.

Second, the synthetic proximity sensor and camera feeds operate under deterministic conditions with injected Gaussian noise. This approach may not fully capture chaotic real-world environmental factors, such as sudden ambient lighting shifts, dust accumulation on lenses, or physical reflections from metallic warehouse structures, which could influence the behavior of the real DJI Mavic 3E perception stack.

## 4 Lessons Learned

The development of the multi-agent SITL environment provided valuable insights into the verification and validation of autonomous systems in industrial settings:

- **The Power of Protocol Transparency:** By ensuring that the simulator speaks the exact same "language" (UDP/JSON) as the physical drone, we eliminated the need for a separate simulation codebase. This reduced maintenance overhead and ensured that bugs found in simulation were 100% reproducible in the real world.
- **Network Simulation is as Critical as Physics:** We learned that in multi-agent systems, software failures often stem from network congestion or latency rather than physical collisions. Implementing a *port-stride* strategy was essential to isolate telemetry streams and maintain real-time performance.
- **Synthetic Noise as a Testing Asset:** One major advantage of the simulator was the ability to inject controlled sensor noise. This allowed us to test the resilience of the navigation software against "flaky" inputs (e.g., erratic infrared readings) in a deterministic way that is nearly impossible to replicate in the physical warehouse.
- **Decoupling Hardware Availability from QA:** The simulator allowed the QA team to begin stress-testing the fleet coordination logic months before the full physical fleet was available or before the warehouse area could be cleared for tests, significantly accelerating the Time-to-Market.
- **Generalizability of the Architecture:** While the specific communication protocol implemented in this study was explicitly tailored to the DJI Mavic 3E and the TPV warehouse, the underlying architectural pattern is highly generalizable. Utilizing a game engine like Godot as a physics backend, coupled with an isolated port-stride UDP strategy for telemetry and control, is a platform-agnostic approach. Other industrial teams can readily adapt this methodology to accelerate V&V pipelines for different robotic platforms utilizing protocols such as MAVLink or ROS.

## 5 Conclusion and Future Work

This experience report detailed the development of a high-fidelity, multi-agent Simulation-in-the-Loop (SITL) environment designed for the validation of autonomous drone fleets in smart warehouses. By leveraging the Godot Engine and a proprietary UDP communication infrastructure, we successfully created a Digital Twin of the DJI Mavic 3E and the TPV industrial warehouse.

Our results demonstrate that the simulator achieves high kinematic fidelity (RMSE of 0.042 m/s) and maintains sub-millisecond network latency for multiple concurrent agents. More importantly, the system acts as a reliable safety arbiter, allowing for the detection of critical logic failures and collisions in a risk-free virtual space. This approach has transformed our V&V process, moving from high-risk, logistically complex field tests to an agile, fail-fast assisted testing environment.

Future developments will focus on:

1. **Hardware-in-the-Loop (HIL) Integration:** Incorporating the physical flight controllers into the loop to account for real-world processing constraints and CPU thermal throttling.
2. **Dynamic Warehouse Generation:** Integrating the WarehouseGenerator directly with the Warehouse Management System (WMS) to replicate real-time inventory layouts and obstacle positions automatically.
3. **AI-Driven Swarm Optimization:** Utilizing the simulation environment to train and validate Reinforcement Learning (RL) models for advanced swarm coordination and energy-efficient path planning.

## Artifact Availability

Repository cannot be shared due to industrial restrictions and confidentiality constraints.

## Acknowledgements

The authors would like to express their profound gratitude to the Paulo Feitoza Foundation (FPFtech) and TPV (Envision) for their strategic partnership and institutional support, which were fundamental for the execution and validation of this project within the Manaus Industrial Pole (PIM).

## References

[1] Samer Al-hamadani and Izzat Al-Darraji. 2025. WAREHOUSE IN INDUSTRY 4.0 BASED DRONE, COMPUTER VISION, AND ARTIFICIAL INTELLIGENCE TECHNOLOGIES: A. *Proceedings on Engineering* 7, 1 (2025), 517–526.

[2] Tiago M Fernández-Caramés, Oscar Blanco-Novoa, Iván Froiz-Míguez, and Paula Fraga-Lamas. 2019. Towards an autonomous industry 4.0 warehouse: A UAV and blockchain-based system for inventory and traceability applications in big data-driven supply chain management. *Sensors* 19, 10 (2019), 2394.

[3] Ismail Laabousse, Mohamed Haidoury, and Younes El Bouzekri El Idrissi. 2025. Advanced Software-in-the-Loop Simulation Environment for UAV Control Algorithms. In *2025 5th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*. IEEE, 1–5.

[4] Christian Llanes, Zahi Kakish, Kyle Williams, and Samuel Coogan. 2024. Crazysim: A software-in-the-loop simulator for the crazyflie nano quadrotor. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 12248–12254.

[5] Khoa Dang Nguyen and Trong-Thang Nguyen. 2019. Vision-based software-in-the-loop-simulation for Unmanned Aerial Vehicles using gazebo and PX4 open source. In *2019 international conference on system science and engineering (ICSSE)*. IEEE, 429–432.

[6] Zijun Zheng, Hao Chen, Yongguang Lu, Kun Xiao, and Xiangke Wang. 2023. Software-in-the-Loop Simulation of Cooperative Fixed-wing Unmanned Aerial Vehicles. In *2023 2nd Conference on Fully Actuated System Theory and Applications (CFASTA)*. IEEE, 302–307.