

AIRHILT: A Human-in-the-Loop Testbed for Multimodal Conflict Detection in Aviation

Omar Garib¹, Jayaprakash D. Kambhampaty¹, Olivia J. Pinon Fischer¹, and Dimitri N. Mavris¹

Abstract—We introduce **AIRHILT** (Aviation Integrated Reasoning, Human-in-the-Loop Testbed), a modular and lightweight simulation environment designed to evaluate multimodal pilot and ATC assistance systems for aviation conflict detection. Built on the Godot engine [1], AIRHILT synchronizes pilot and air traffic controller (ATC) communications, visual scene understanding from camera streams, and ADS-B surveillance data within a unified, scalable platform. The environment supports pilot- and controller-in-the-loop interactions, providing a comprehensive scenario suite covering both terminal area and en route operational conflicts, including communication errors and procedural mistakes. AIRHILT offers standardized JSON-based interfaces enabling researchers to easily integrate, swap, and evaluate various automatic speech recognition (ASR), visual detection, decision-making, and text-to-speech (TTS) models. We demonstrate AIRHILT through a reference pipeline incorporating fine-tuned Whisper ASR, YOLO-based visual detection, ADS-B-based conflict logic, and GPT-OSS-20B structured reasoning, presenting preliminary results from representative conflict scenarios. The AIRHILT environment and scenario suite are openly available, supporting reproducible research on multimodal situational awareness and conflict detection in aviation. The complete repository is available at [github].

I. INTRODUCTION

Aircraft situational awareness and conflict detection currently rely on accurate multi-aircraft surveillance and radio communications, with command and control of airspace operations managed centrally by human air traffic controllers (ATCs). However, human ATCs and pilots are vulnerable to overwork, fatigue, and loss of attention, increasing the risk of operational errors and conflict events [2], [3].

To alleviate workload pressures and reduce operational errors, there is a growing need for assistive aviation systems that incorporate recent advancements in *automatic speech recognition* (ASR) and *vision-based detection*, alongside existing aircraft and radar surveillance data (e.g., ADS-B, radar). Such systems could proactively identify hazards such as traffic conflicts and runway incursions, while enhancing situational awareness and minimizing additional workload for pilots and controllers.

However, the current testing landscape for such aviation assistive systems presents notable challenges. Physically collocated test environments are costly, time-consuming, and require careful scheduling of limited ATC and pilot availability. These sophisticated facilities typically include several pilot and controller workstations along with integrated displays and are commonly utilized for operational



Fig. 1. **AIRHILT at a glance.** Cockpit view from the simulation showing the setting used for pilot-in-the-loop evaluations. AIRHILT synchronizes radio traffic, vision feeds, and ADS-B to test end-to-end assistive warning pipelines.

scenario studies [4], [5]. Yet, their availability is increasingly constrained by growing global aviation traffic and rising research demands associated with emerging aviation concepts such as unmanned aerial systems (UAS) and advanced air mobility (AAM) [6], [7]. Although rigorous testing at physical facilities remains essential for advanced development and certification phases, it is impractical for early-stage concept evaluations. Additionally, the rapidly expanding design space, driven by new machine learning models and varied computational, sensing, and communication architectures, further underscores the necessity of flexible, efficient, simulation-based environments suitable for rapid, systematic evaluations.

Such a simulation-based environment would significantly broaden access to aviation situational awareness research, allowing researchers worldwide to efficiently explore and identify promising candidate systems while conserving limited pilot and ATC resources. To address these needs, we introduce **AIRHILT**, a simulation environment explicitly designed to facilitate research into multimodal AI assistance systems through pilot and controller-in-the-loop experimentation.

Contributions. Our contributions in this effort are as follows:

- 1) **An open simulation environment** that synchronizes pilot-to-ATC communications, ATC control tower camera views, aircraft-mounted camera streams, and ADS-B/radar data, facilitating streamlined experimentation with various intelligent assistive system designs and

*This work was not supported by any organization

¹Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology

- supporting pilot and ATC in-the-loop interactions.
- 2) A **scalable scenario suite** currently consisting of six conflict scenarios, including three terminal operations and three en route encounters, modeling different causes such as communication errors and human errors, with parameterized variations including noise artifacts, visibility conditions, and geometric factors (e.g., positions, speeds).
 - 3) A **reference multimodal pipeline** demonstrating environment capabilities through interchangeable components such as Whisper-based ASR, YOLO-based visual detection, and structured large language model (LLM)-based decision layers, with preliminary performance metrics recorded.
 - 4) **Reproducible artifacts** including the simulation environment, code, scenarios, scripts, and clear documentation to support community use.

The remainder of this paper is structured as follows: Section II provides background on air traffic management operations and the key components relevant to aviation situational awareness. Section III outlines related challenges to building such systems. Section IV details the environment and interfaces. Section V presents the designed scenarios. Section VI describes the reference pipeline and presents preliminary results from two scenarios. Section VII provides some concluding remarks, discusses current limitations and introduces avenues for future work.

II. BACKGROUND

Aviation operations encompass both air traffic management and conflict detection, including the determination, sequencing, and issuance of clearances from departure through en route and approach phases [8], [9], as well as the identification and mitigation of hazards such as wildlife encounters, mechanical issues, and other unexpected conflicts [10]. Effective traffic management maintains prescribed separation between aircraft and operational efficiency under current operational conditions. Controllers integrate surveillance data (e.g., ADS-B and radar) provided via tower infrastructure, direct visual observations from the control tower, and standardized voice communications with pilots to formulate and issue clearances and instructions [11]. In parallel, pilots routinely manage additional hazards, including wildlife (e.g., birds) activity, uncooperative or untracked intruder aircraft, and onboard mechanical anomalies, by synthesizing external visual cues with onboard sensor data and established procedures.

Many unsafe conditions arise from (i) *traffic-management breakdowns* (e.g., violations of separation minima or runway occupancy conflicts) and (ii) *aircraft or airspace hazards* (e.g., bird strikes, uncooperative intruder aircraft, or mechanical anomalies). It is important to note that existing radar-based surveillance systems such as Automatic Dependent Surveillance-Broadcast (ADS-B), and collision-avoidance systems such as the Traffic Collision Avoidance System (TCAS/ACAS), already provide critical support. However, incidents have occurred and continue to occur even with

these systems in place. For example, the Überlingen mid-air collision highlighted critical vulnerabilities when ATC instructions conflict with TCAS resolution advisories (RAs), reinforcing that TCAS RAs must always take precedence over controller clearances [12]. Moreover, it is important to note that TCAS RAs are intentionally inhibited at low altitudes [11], impacting its operation to resolve situations such as runway incursions. These realities motivate the development of an assistive layer that fuses visual streams, voice communications, and surveillance data to identify and flag potential conflicts *before* situations escalate to require intervention from safety systems such as TCAS.

A. Operational Communication Loop

We focus on the standard pilot–controller communication loop used across terminal and en route operations. At a high level:

- 1) Pilot *establishes contact with the appropriate ATC facility* (e.g., tower or approach).
- 2) Controller *issues an instruction or clearance*.
- 3) Pilot *reads back the instruction or clearance*.
- 4) Controller *monitors the readback, correcting if necessary*, after which the pilot executes.

Failures can occur at multiple points (e.g., mishearing, incorrect readback, or delayed compliance), motivating assistive monitoring across modalities.

B. Multimodal Components and Perception Tasks

Recent progress in multimodal AI provides essential building blocks for assistance: (i) **ASR** for transcribing and parsing ATC/pilot communications; (ii) **vision detection models** operating on tower or onboard cameras for surface/aircraft/vehicle detection and tracking; and (iii) **surveillance and trajectory analytics** from ADS-B/radar data. However, performance thresholds and latency budgets required for operational usefulness remain under-specified, and real systems must handle missing or degraded modalities.

1) ASR for ATC/Pilot Communications: Several automatic speech recognition (ASR) models have been proposed to reduce the Word Error Rate (WER) of ATC and pilot radio transmissions, with the aim of integration into ATC and pilot communication workflows. A comprehensive review of recent approaches can be found in the special collection by Helmke et al. [13]. Notably, substantial improvements in WER have recently been achieved by fine-tuning OpenAI's Whisper ASR model [14] on simulated and synthetic ATC speech datasets [15], [16].

Despite recent WER reductions, the safety impact remains uncertain. WER measures transcription accuracy, not how recognition errors influence pilot/controller performance during critical events. As noted by van Doorn et al. [17], concrete performance requirements for safety management are not well specified. Moreover, the events where ASR would help most are rare and highly context dependent (airport geometry, traffic density, fatigue).

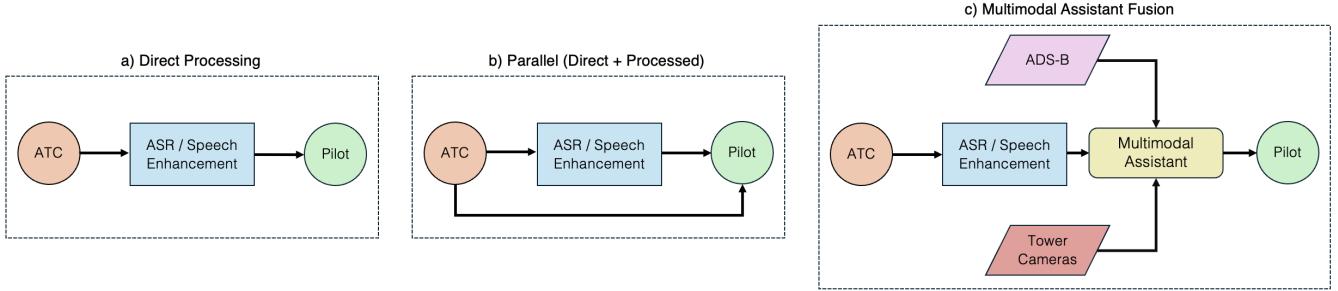


Fig. 2. Canonical topology alternatives for assistive processing in the ATC–pilot loop. (a) *ASR/SE first*: audio is enhanced (SE) and/or transcribed (ASR) first; the resulting output directly informs the advisory presented to the pilot; (b) *Parallel paths*: raw audio reaches the pilot while a copy is processed by ASR/SE in parallel; (c) *Assistant-gated fusion*: ASR/SE, vision, and ADS-B are fused before any advisory is issued. All variants incorporate vision detections and ADS-B tracks within a common decision layer that outputs graded advisories to the pilot and/or controller.

2) *Visual Scene Understanding*: Image processing advances in the ATC context include the identification of aircraft in the air [18], [19], on the ground [20], [21], [22], or in operational contexts [23]. These models are often built on top of the YOLO object detection architecture [24].

C. Architectural Topologies

Assistive systems offer various design choices: placement of ASR/Speech enhancement (SE) relative to pilot audio (ASR/SE-first or parallel), fusion approach for vision and ADS-B (early or late), and recipient of advisories (pilot, controller, or both). Figure 2 illustrates three representative configurations, highlighting the need for flexible simulation environments to systematically evaluate diverse architectures across various operational scenarios.

III. CHALLENGES

Evaluating air traffic situational awareness systems presents four key challenges: (1) moving away from subtask accuracy into operational safety gains, (2) modeling rare failure scenarios with limited real-world data, (3) integrating multimodal context in evaluation frameworks, and (4) enabling simulation environments that support human-in-the-loop input and intelligent system responses.

A. Operational Evaluation

Subtask performance metrics, such as ASR’s WER, are insufficient alone to characterize safety system performance. Scenario-based usage of the systems must be used to evaluate their contribution to preventing unsafe outcomes.

B. Human-in-the-Loop Operation

A related challenge for the design of the simulation environment is to incorporate humans in the scenarios via simulated ATC and pilot workstations. To actually execute operational evaluations as described above, it becomes imperative that human behaviors can be incorporated into the chain of events.

C. Scarcity of Failure Data

Safety-critical events are rare and underrepresented in public datasets, making the evaluation of assistive systems for conflict detection challenging. Therefore, these systems typically require evaluation through carefully constructed simulated scenarios developed with expert input and supplemented by synthetic generation and real ATC data when available. However, effectively decreasing the sim-to-real gap between actual incidents and simulated scenarios remains a significant challenge.

D. Multimodal Context Integration

Current evaluations often rely on narrow, single-channel data inputs. Richer simulation frameworks should be able to synchronize visual, audio, and state data (e.g., from Tar-tanAviation [25]) to reflect the true complexity of operational environments.

IV. SIMULATION ENVIRONMENT AND INTERFACES

To enable efficient and scalable evaluation of different multimodal pilot-assist architectures with representative human-in-the-loop interactions, we introduce *AIRHILT*, a lightweight, modular simulation environment built upon the Godot engine. *AIRHILT* offers a unified simulation platform that synchronizes pilot-to-ATC communications, camera-based visual perception, and ADS-B traffic data, enabling systematic evaluation of assistive systems designed for aviation conflict detection and resolution.

A. Design Goals and Scope

The design of *AIRHILT* directly targets the operational and methodological challenges outlined in Section III, structured around the following primary design goals:

1) *Modularity and Interoperability*: All components communicate via stable REST/JSON interfaces, allowing ASR, vision, decision, and text-to-speech (TTS) modules to be swapped with minimal code changes.

2) *Reproducibility*: Each simulation run leverages deterministic seeding, stable event identifiers, and unified timestamps, enabling consistent and repeatable experimentation.

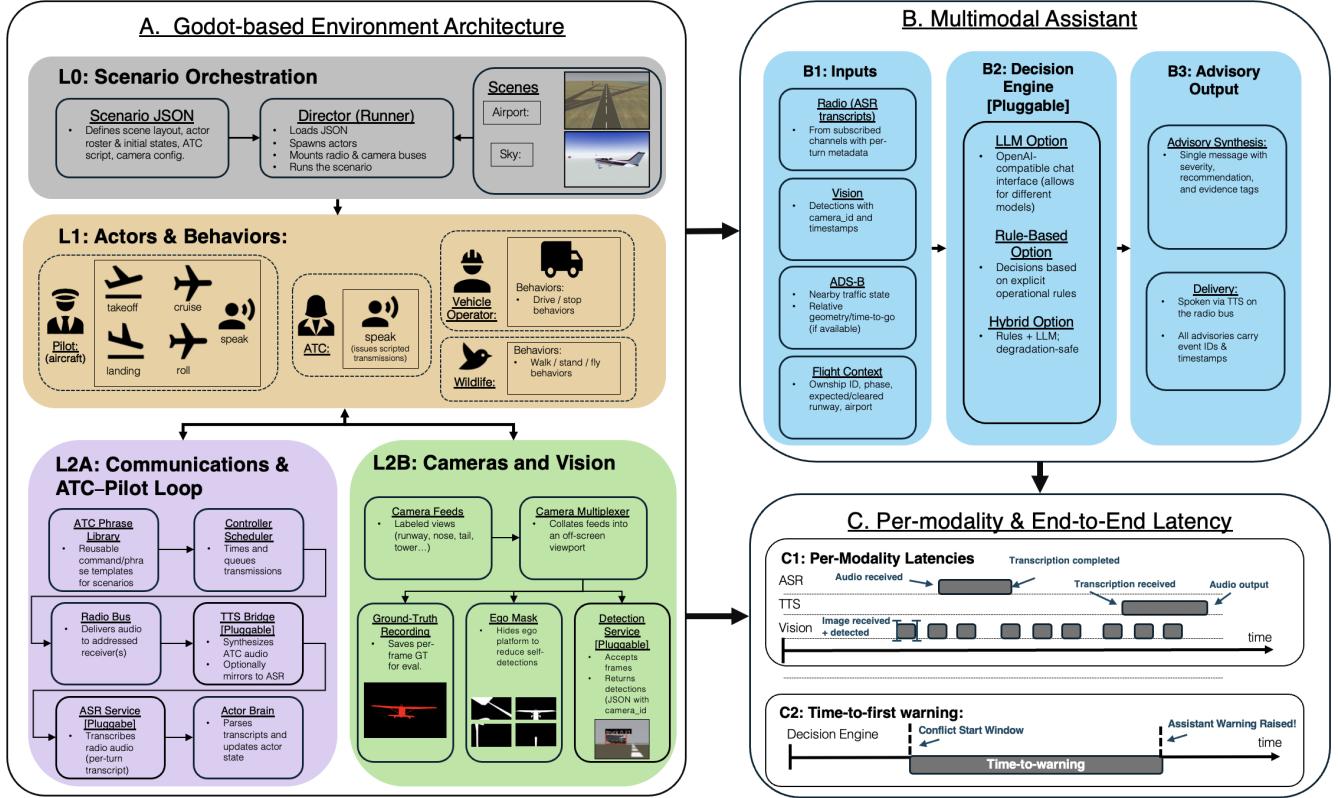


Fig. 3. Modular simulation environment and onboard assistant architecture. A) Godot-based environment with scenario orchestration (L0), actors (L1), and I/O subsystems (L2A/L2B). B) Multimodal assistant with pluggable decision engine and advisory output. C) Built-in logging for per-modality and end-to-end latencies.

3) *Representative Human-in-the-loop Interactions*: Frequency-specific, addressed radio traffic and confidence-gated actor behaviors approximate operational interactions.

4) *Logging and Timing*: Built-in logging records per-modality and end-to-end latencies for each advisory event.

5) *Deployment*: A lightweight Godot runtime with compact FastAPI services runs on a single workstation and benefits from optional GPU acceleration.

Fig. 3 presents a high-level overview of the *AIRHILT* architecture, while the following subsections provide a deeper exploration into the design, functionality, and implementation details of each layer and component.

B. Scenario Orchestration (L0)

The scenario execution in *AIRHILT* is orchestrated through a clearly structured, declarative Scenario JSON, providing researchers with full control over simulation initialization and runtime conditions. Each scenario specification encapsulates:

- Scene type and geometry**: Defines whether the scenario involves airport surface operations or en route airborne interactions, alongside geometric details for each actor's initial placement and orientation.
- Actor roster and initial states**: Specifies actors such as aircraft, ATCs, vehicles, and wildlife, assigning each with initial behavior states.

- Scripted ATC communication timeline**: Specifies structured, timestamped radio exchanges, including addressed clearances and instructions, with options to inject controlled noise artifacts.
- Camera configuration**: Determines the placement and orientation of cameras (e.g., aircraft-mounted and/or tower-based), along with sampling rates for visual perception components.
- Randomization seed**: Controls deterministic variants for reproducible small geometric and timing perturbations.

Upon ingestion of the Scenario JSON, the Director (Runner) automatically loads and initializes all scene assets and actors as defined in the scenario specification. It subsequently mounts synchronized communication and vision buses to maintain consistent alignment of audio (radio), visual (camera), and corresponding ground-truth data streams.

The environment explicitly supports two primary scene types selected for their operational relevance to aviation safety:

- Airport Surface Scene**: Inspired by Ronald Reagan Washington National Airport (DCA), featuring intersecting runways and high-complexity ground operations representative of common conflict scenarios.
- En route Airspace Scene**: Represents typical airborne interactions, including converging flight paths

and altitude-based conflict geometries.

C. Actors and Behavior Abstractions (L1)

The environment defines a set of actor types and structured behaviors that aim to closely represent operational interactions in aviation scenarios, balancing fidelity and computational efficiency. These models support pilot-in-the-loop interactions without the overhead of full-fidelity flight dynamics. Additionally, the environment was intentionally designed to simplify the addition of new actor types and behaviors, with clear guidance provided in the repository documentation.

- **Actor Types:** The environment currently models four primary actor classes:

- *Pilot/Aircraft*: Capable of executing named behaviors such as takeoff, cruise, landing, and responding to radio commands.
- *Air Traffic Controller (ATC)*: Issues scripted, addressed instructions and clearances, following structured timelines.
- *Vehicle Operators*: Simulate ground vehicles with pre-defined driving patterns (e.g., drive, stop).
- *Wildlife*: Incorporates animal actors exhibiting basic behaviors (walk, stand, fly), serving as hazard sources.

- **Control Inputs:** For all non-wildlife actors, addressed radio transmissions are parsed with a slot-based method that aims to approximate how real operators extract intent from speech (callsign, runway/altitude assignments, temporal instructions such as “hold short” or “cleared for takeoff”). Low-confidence or ambiguous inputs trigger structured clarification behavior at the actor layer.

- **Physics-based Motion:** Actor movements use computationally efficient, physics-informed models sufficient to maintain timing and geometric realism in conflict scenarios. Motion behaviors include timed vertical maneuvers (climbs and descents), and relatively simplified representations of landing (glide, flare, rollout) and takeoff (roll, rotate, climb-out) phases, among others. Detailed equations and parameter choices are provided in the repository.

D. Communications Loop: Radio, TTS, and ASR (L2A)

The communications subsystem of *AIRHILT* is structured to approximate operational aviation radio interactions while supporting controlled experimentation with pilot and ATC communications. This communication loop incorporates clearly defined interfaces for radio transmissions, TTS synthesis, and ASR, all implemented as interchangeable modules accessible through standardized REST/JSON endpoints:

- **Radio Bus:** The radio subsystem utilizes frequency-specific channels for addressed transmissions and includes optional overhearing capabilities. Transmission guarantees include ordered delivery, stable identifiers

for each radio turn, and precise emission timestamps (t_{tx}).

- **ASR Service (Interchangeable):** Pluggable ASR models transcribe received audio and emit finalization timestamps (t_{asr_out}) and optional confidences. The ASR runs in parallel to the pilot audio path by default; however, users may configure the ASR output as an intermediate step prior to pilot reception, depending on the assistive architecture and experimental setup being evaluated.
- **TTS Bridge (Interchangeable):** The environment integrates pluggable TTS services (e.g., XTTS-v2 [26]) that convert scripted text instructions into realistic audio. An optional augmentation layer adds configurable radio-channel effects, including noise and distortion, with precise control over signal-to-noise ratio (SNR) and mixing parameters.

To support latency evaluation, the logger records the timestamps needed to compute per-module and end-to-end timings (e.g., t_{tx} , t_{asr_out} , augmentation/SNR settings, and stable transmission IDs).

E. Camera and Vision Pipeline (L2B)

The visual perception subsystem of *AIRHILT* provides configurable, synchronized camera feeds designed to replicate representative visual data streams. This pipeline maintains clearly defined and stable REST/JSON interfaces for seamless module interchangeability, supporting systematic experimentation and reproducibility:

- **Camera Feeds:** The environment includes multiple predefined camera perspectives such as runway views, aircraft nose and tail cameras, and tower-based views. These feeds are captured at configurable sampling rates, with optional per-actor sampling control for targeted experimentation.
- **Multiplexer:** A configurable multiplexer mirrors camera feeds to off-screen viewports, enabling simultaneous data capture without disrupting primary simulation rendering performance.
- **Ego Mask:** Per-camera visibility masks are implemented to suppress model self-detections.
- **Detection Service (Interchangeable):** Visual detection services operate as independently swappable modules accessible via REST APIs. They receive images (alongside optional ego masks), timestamps (ts_ms), and camera identifiers ($camera_id$), and return structured detections in JSON. Each detection object includes classification labels, confidence scores, and bounding box coordinates.
- **Ground Truth Data:** Each actor is assigned a unique color in dedicated viewport renders. From these per-frame masks we derive presence labels and ideal (axis-aligned) bounding boxes, and log them alongside class IDs for analysis and visualization.

F. Multimodal Assistant Interface (B): Inputs, Decision Engine (Interchangeable), and Outputs

The multimodal assistant exposes a well-specified interface that defines inputs, outputs, and the pluggable decision module. This decouples environment integration from any particular algorithm and allows alternative implementations to be swapped with minimal code changes elsewhere.

- **Inputs:** The assistant receives synchronized multimodal data in three categories:
 - *Radio Transcripts*: Structured sequences of communications with timestamps ($t_{\text{ts_ms}}$), speaker IDs, transcript text, frequency, unique turn IDs, optional *over-hearing/subscription* flags (to receive nearby actors' traffic), addressed callsigns, and confidences when available.
 - *Vision Detections*: Lists of detection results from visual perception, each containing frame timestamps ($t_{\text{ts_ms}}$), associated camera identifiers, detected object classifications, confidence scores, and bounding box coordinates.
 - *ADS-B and Flight Context*: Data slices summarizing ownership operational states, expected and cleared runway information, and positional and velocity tracks.
- **Decision Engine (Interchangeable):** Decision-making modules are integrated via standardized HTTP/JSON interfaces, allowing simple substitution between the different options of rule-based logic, LLM-based reasoning, or other hybrid methods. Inputs are posted as structured requests, and modules return standardized advisory objects (message, severity, optional recommendations, metadata).
- **Outputs:** The output schema is user-configurable. Advisories could include concise text, a severity level (INFO/ADVISORY/CAUTION/WARNING), optional recommendations, and supporting metadata for traceability and debugging.
- **Advisory Delivery:** Advisory objects are synthesized into audible alerts via the previously described TTS subsystem and delivered through the appropriate radio channel.

G. Timebase and Latency Accounting (C)

We instrument all subsystems with a shared monotonic timebase and log start/end events for each module. For radio/ASR we record the transmission time t_{tx} and ASR finalization t_{asr} . For vision we record frame exposure end t_{frame} and detector completion t_{vision} . For ADS-B we record ingest $t_{\text{adsb,in}}$ and post-processor output $t_{\text{adsb,out}}$. The decision engine records when all required inputs are available t_{ready} and when an advisory is produced t_{dec} (this includes LLM inference when used). The audio path records the first audible sample delivered to the radio bus t_{tts} . Each scenario annotates the conflict-window opening time t_{conflict} .

From these timestamps we compute per-module and

end-to-end timings during post-processing:

$$\text{ASR latency} = t_{\text{asr}} - t_{\text{tx}}, \quad (1)$$

$$\text{Vision latency} = t_{\text{vision}} - t_{\text{frame}}, \quad (2)$$

$$\text{ADS-B latency} = t_{\text{adsb,out}} - t_{\text{adsb,in}}, \quad (3)$$

$$\text{Decision latency} = t_{\text{dec}} - t_{\text{ready}}, \quad (4)$$

$$\text{TTS latency} = t_{\text{tts}} - t_{\text{dec}}, \quad (5)$$

$$\text{Time to first warning} = t_{\text{tts}} - t_{\text{conflict}}. \quad (6)$$

Figure 3C illustrates the latencies and warning intervals.

V. EVALUATION SCENARIOS

Six primary scenario families were developed to evaluate diverse, realistic aviation conflicts across terminal and en route airspace, illustrated in Figure 4. We focus exclusively on human-, procedural-, and communication-driven conflicts, whereas mechanical failures are out of scope in the current release.

In terminal airspace (airport surface operations), three scenario families assess runway incursions and occupancy conflicts:

- **S01A (Runway Overlap)** evaluates conflicts such as miscommunication during runway clearances, including bad readbacks, missed cancellation transmissions, and misaddressed instructions.
- **S01B (Vehicle Runway Incursions)** involves ground vehicle incursions due to delayed or dropped HOLD commands, misaddressing, or intentional noncompliance.
- **S01C (Wildlife Runway Incursions)** addresses wildlife presence hazards, particularly delayed wildlife warnings or failures to detect wildlife incursions.

In en route airspace (sky domain), three scenario families cover airspace geometry and coordination conflicts:

- **S02A (Geometric Airspace Conflicts)** evaluates situations like in-trail closing, head-on trajectories, and vertical separation violations.
- **S02B (Airborne Emergency Coordination)** assesses emergency situations such as bird strikes and engine-out drift-down scenarios requiring precise coordination.
- **S02C (Uncoordinated Intruders)** focuses on encounters with non-cooperative aircraft lacking ADS-B/TCAS coordination, requiring purely visual detection.

The environment was deliberately designed to simplify the creation of these scenarios, providing a structured and intuitive JSON-based configuration. A comprehensive guide to scenario creation, detailing workflows, parameters, and extensions, is included in our public repository.

VI. REFERENCE PIPELINE

To demonstrate *AIRHILT*, we implemented a reference pipeline and tested it on **S01A—Runway Overlap**. For instance, in the *bad readback accepted* variant, an arrival is cleared to land runway 01, the pilot incorrectly reads back runway 19, the tower replies “roger,” and later a departure is cleared for takeoff on runway 01, creating an occupancy

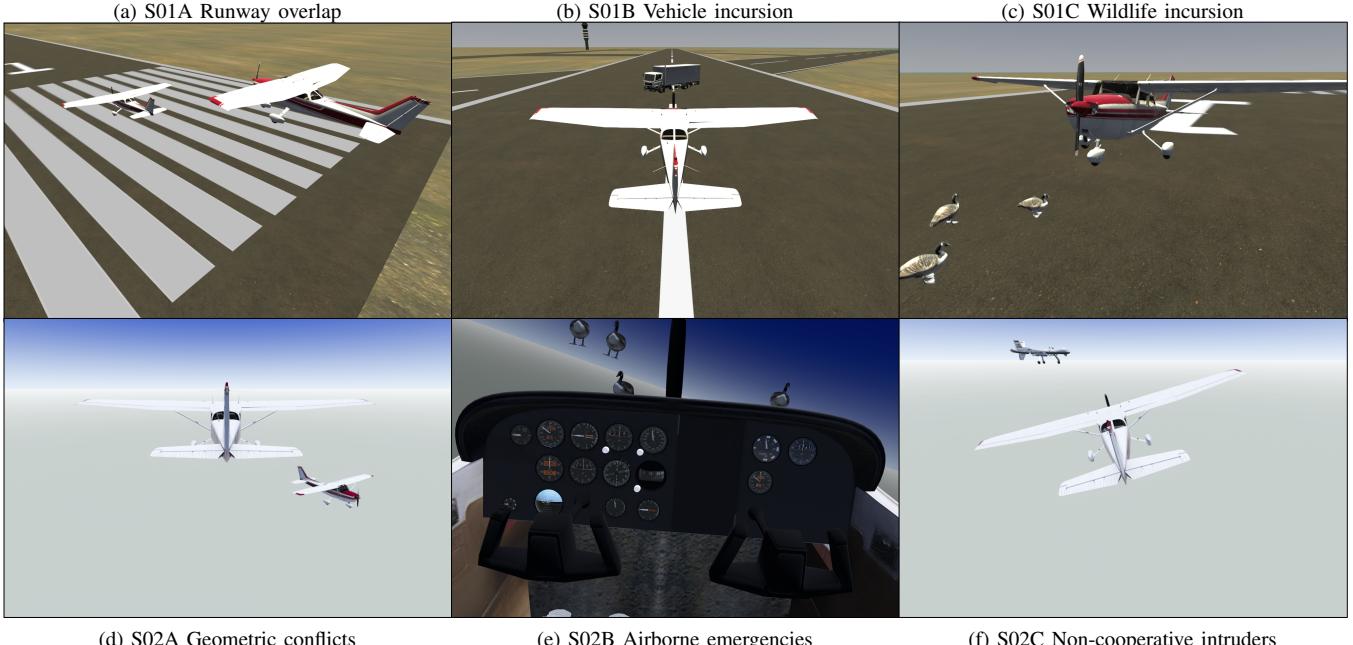


Fig. 4. Representative examples from the six scenario families used for evaluation. Top row shows terminal-area scenarios (runway overlaps S01A, vehicle incursions S01B, wildlife incursions S01C). Bottom row shows en route scenarios (geometric conflicts S02A, airborne emergencies S02B, non-cooperative intruder encounters S02C).

conflict. Other S01A conflict types exercised in the suite include: *cancel takeoff not received* (tower cancels the departure but the intended aircraft never hears it), *misaddressed takeoff clearance* (clearance spoken with the wrong callsign, departure accepts), and *tight timing overlap* (both clearances valid, but spacing is insufficient).

A. Models and Interfaces

Table I summarizes the modules and essential information regarding the models used in the reference pipeline.

B. Decision Logic

Algorithm 1 outlines how the decision engine integrates information from the different modalities to determine when and what warnings to raise.

C. Preliminary Results

We evaluated **S01A** across 10 runs per conflict type with randomized visibility and SimuGAN SNR settings to stress both the vision and radio paths. Averages across these runs were: time to first warning $t_{\text{ts}} - t_{\text{conflict}}$ **7.66 s**; ASR latency $t_{\text{asr}} - t_{\text{tx}}$ **5.88 s**; vision latency $t_{\text{vision}} - t_{\text{frame}}$ **0.415 s**; and **TTS synthesis/delivery** $t_{\text{ts}} - t_{\text{dec}}$ **0.9 s**. The first vision detection typically occurred at $\sim 125 \text{ m}$ range.

In-depth results across all scenario families are available in the public repository.

VII. CONCLUDING REMARKS

AIRHILT provides an open-source and flexible environment that facilitates experimentation with different multi-modal assistive systems in aviation.

Algorithm 1: Decision engine for S01A family (parallel modality logic)

Inputs: Radio turns \mathcal{A} ; vision detections \mathcal{V} ;

ADS-B/roster \mathcal{B} ; thresholds $\tau_{\text{ASR}}, \tau_{\text{vis}}$

Outputs: Advisory object $\langle \text{type}, \text{severity}, \text{text}, \text{evidence} \rangle$

- 1 **Modality updates (run in parallel):** (i) **ASR parse:** normalize runway tokens; compute *slot_conf*.
(ii) **Vision occupancy:** multi-cam corroboration (K) or conf-persistence; produce activity/occupancy flags.
(iii) **ADS-B slice:** compute TTG and determine nearby traffic.
 - 2 **Guard:** if $\text{slot_conf} < \tau_{\text{ASR}}$ then **request clarification**.
 - 3 **On any stream update, evaluate ladder gates:**
 - a) readback mismatch + activity $\Rightarrow \text{CAUTION}$;
 - b) occupancy + ($\text{TTG} \leq 8 \text{ s}$ or arrival context) $\Rightarrow \text{WARNING}$;
 - c) recipient ambiguity + activity $\Rightarrow \text{CAUTION}$;
 - 4 **If none fired (evidence fallback):** score $S \leftarrow 0.50W_V + 0.35W_A + 0.15W_C$; if $S \geq 0.75 \Rightarrow \text{CAUTION}$; if $S \geq 0.50 \Rightarrow \text{ADVISORY}$
 - 5 **Compose advisory text** from rules (optionally reformulated by GPT-OSS-20B) and **deliver via TTS**
 - 6 **Return** advisory object with evidence (radio IDs, camera IDs, TTG, rules triggered)
-

TABLE I

REFERENCE PIPELINE MODULES, TRAINING DATA, AND KEY I/O AND BEHAVIOR. RADIO-PATH REALISM VIA SIMUGAN (APPROX. 4.5 H TARTANAVIATION ATC) OR DSP, APPLIED ONLY TO ATC–PILOT COMMS (WAV/TTS); ADVISORY TTS NOT NOISE-AUGMENTED. SEE THE PUBLIC REPOSITORY FOR DETAILS.

Subsystem	Model / version	Training / fine-tune	Key I/O and behavior
ASR	Whisper (OpenAI), fine-tuned	ATC-focused fine-tune (large-v2) [16]	Inputs: speaker, frequency. Output: transcript and t_{ASR} . Notes: runway token canonicalization; confidence gate $\geq \tau_{\text{ASR}}$.
TTS (advisories)	Coqui XTTS-v2 [26]	—	Advisory audio (no noise augmentation).
Radio noise (ATC–pilot)	SimuGAN (learned RF/VHF); DSP fallback (SNR)	approx. 4.5 h TartanAviation ATC (for SimuGAN) [16]	Applied only to ATC and pilot received audio. Controls: SNR, wet mix, profile.
Vision	Ultralytics YOLOv10 [27]; ego-masking; tiled inference	Class filter (airplane, truck, bird)	Outputs: detections and t_{vision} . Multi-camera corroboration (K) or confidence persistence.
ADS-B / flight context	Parser + roster filters (airport map)	—	Outputs: runway expectations, occupancy, and tracks.
Decision engine	Rules + GPT-OSS-20B (natural language generation, NLG only)	—	Rule ladder: occupancy + time-to-go (TTG); see Alg. 1 for details.
Output object	Advisory object + delivery	—	Fields: severity, message, recipients, evidence; speak only if $\geq \text{SPEAK_MIN_LEVEL}$.

A. Limitations

Current limitations of *AIRHILT* include a sim-to-real gap, particularly within the vision detection pipeline, where simulation artifacts can lead to discrepancies from real-world performance. Human-factors validation is also limited, as we have not yet conducted controlled pilot and ATC studies to quantify workload, situational awareness, and operator acceptance of the multimodal assistant.

B. Avenues for Future Work

Future efforts will focus on simplifying environment setup procedures and improving the clarity and flexibility of the provided interfaces, aiming to reduce onboarding time and support rapid experimentation. Additionally, we plan to conduct simulator studies involving pilots and controllers to improve the fidelity and realism of modeling human-in-the-loop interactions within *AIRHILT*.

REFERENCES

- [1] Godot Engine, “Godot Engine — Free and open-source 2D and 3D game engine.”
- [2] P. S. Della Rocco and P. S. Della Rocco, “The Role of Shift Work and Fatigue in Air Traffic Control Operational Errors and Incidents,” Tech. Rep. DOT/FAA/AM-99/2, Federal Aviation Administration, Office of Aviation Medicine, Washington, D.C., Jan. 1999.
- [3] M. R. Rosekind, E. E. Flynn-Evans, and C. A. Czeisler, “Assessing Fatigue Risk in FAA Air Traffic Operations,” tech. rep., U.S. Federal Aviation Administration (FAA), Apr. 2024.
- [4] S. Schier, T. Rambau, F. Timmermann, I. Metz, and T. H. Stelkens-Kobsch, “Designing the Tower Control Research Environment of the Future,” in *Proceedings of the German Aerospace Congress (Deutscher Luft- und Raumfahrtkongress, DLRK2013)*, (Stuttgart, Germany), Sept. 2013.
- [5] P. G. Manske and S. L. Schier, “Visual Scanning in an Air Traffic Control Tower – A Simulation Study,” *Procedia Manufacturing*, vol. 3, pp. 3274–3279, Jan. 2015.
- [6] I. Chaisit, E.-M. Hubbard, and P. Lepper, “Enhancing Public Acceptance of the Advanced Air Mobility (AAM) Ecosystem Technology: Exploring Conceptual Theoretical Frameworks and Regulatory Support,” in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 324–330, June 2024. ISSN: 2575-7296.
- [7] W. N. Kaliardos, J. Holbrook, and A. Hobbs, “Identifying Human Factors Research for Unmanned Aircraft Systems and Advanced Air Mobility,” in *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, pp. 1–9, Sept. 2024. ISSN: 2155-7209.
- [8] International Civil Aviation Organization, “Procedures for Air Navigation Services — Air Traffic Management (PANS-ATM),” Tech. Rep. Doc 4444, International Civil Aviation Organization (ICAO), Montréal, Canad, 2016.
- [9] Federal Aviation Administration, “Air Traffic Control,” Tech. Rep. JO 7110.65BB, Federal Aviation Administration (FAA), Sept. 2024.
- [10] Federal Aviation Administration, “Hazardous Wildlife Attractants On or Near Airports,” Tech. Rep. AC 150/5200-33C, Federal Aviation Administration (FAA).
- [11] Federal Aviation Administration, “ACAS Training Program,” Tech. Rep. AC 90-120, Federal Aviation Administration (FAA).
- [12] International Air Transport Association, “ACAS (TCAS) Guide,” tech. rep., International Air Transport Association (IATA).
- [13] H. Helmke and O. Ohneiser, eds., *Automatic Speech Recognition and Understanding in Air Traffic Management*. MDPI, Mar. 2024.
- [14] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavy, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” Dec. 2022. arXiv:2212.04356 [eess].
- [15] J. van Doorn, J. Sun, J. Hoekstra, P. Jonk, and V. de Vries, “Whisper-ATC: Open Models for Air Traffic Control Automatic Speech Recognition with Accuracy,” 2024.
- [16] O. Garib, O. J. Pinon Fischer, and D. N. Mavris, “SimuGAN-Whisper-ATC: Generative Noise Injection for Improved Automatic Speech Recognition in Air-Traffic Control,” Manuscript in preparation. Accepted for publication at AIAA SCITECH 2026.
- [17] J. L. van Doorn, *Applying Large-Scale Weakly Supervised Automatic Speech Recognition to Air Traffic Control*. PhD thesis, Delft University of Technology, Delft, Netherlands, Dec. 2023.
- [18] W. Li, J. Liu, and H. Mei, “Lightweight convolutional neural network for aircraft small target real-time detection in Airport videos in complex scenes,” *Scientific Reports*, vol. 12, p. 14474, Aug. 2022.
- [19] M. Radovic, O. Adarkwa, and Q. Wang, “Object Recognition in Aerial Images Using Convolutional Neural Networks,” *Journal of Imaging*, vol. 3, p. 21, June 2017.
- [20] T. R. Pratama, J. Sembiring, and K. Ummah, “System for Tracking

- Aircraft Ground Movement Utilizing Artificial Intelligence,” *E3S Web of Conferences*, vol. 477, p. 00009, 2024.
- [21] A. Singh, S. Kumar, and D. Choudhury, “Deep Learning Driven Real-Time Airspace Monitoring Using Satellite Imagery,” *Scalable Computing: Practice and Experience*, vol. 25, Oct. 2024.
 - [22] W. Alshaibani, M. Helvacı, I. Shayea, and H. Mohamad, “Airplane Detection Based on Mask Region Convolution Neural Network,” *ArXiv*, Aug. 2021.
 - [23] H. Kim and K. Lee, “Air Traffic Prediction as a Video Prediction Problem Using Convolutional LSTM and Autoencoder,” *Aerospace*, vol. 8, p. 301, Oct. 2021.
 - [24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” May 2016. arXiv:1506.02640 [cs].
 - [25] “TartanAviation: Image, Speech, and ADS-B Trajectory Datasets for Terminal Airspace Operations,” tech. rep., Mar. 2025. arXiv:2403.03372 [cs].
 - [26] “Coqui TTS — A deep learning toolkit for Text-to-Speech.”
 - [27] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, “YOLOv10: Real-Time End-to-End Object Detection,” Oct. 2024.