# SIMULATION CAPABILITY FOR SPECIAL OPERATIONS IN NON-SEGREGATED AIRSPACE

ANA-MARIA BORDEI , CATALIN NAE

*Abstract*. This article surveys how simulation has been used to study special operations in non-segregated airspace, where civil, military and unmanned aircraft increasingly share the same volume. The operations of interest include the handling of non-cooperative unmanned aircraft, counter-UAS activity, military flights transiting civil airspace, and cooperative maneuvers such as air-to-air refueling. How this mixed traffic behaves, the workload it imposes on controllers, and the value of any new procedure or decision-aiding tool are empirical questions that cannot be answered safely, affordably, or repeatably in live flight. The discussion is organized around four threads: the role of simulation as a research instrument; the fast-time and real-time human-in-the-loop paradigms, together with the platforms that embody them (FACET, BlueSky, ESCAPE, en-route digital twins); the conflict-detection, conflict-resolution and detect-and-avoid methods that have been developed and validated computationally; and the human-factors evidence from controller-in-the-loop experiments. Drawing these threads together, the article argues that a dedicated, high-fidelity simulation capability is not a convenience but a precondition for safely integrating special operations into shared airspace, and it identifies where the literature remains thin, most notably the scarcity of platforms purpose-built for special operations and for non-cooperative traffic.

*Key words:* air traffic management; non-segregated airspace; special operations; scenario-based simulation; human-in-the-loop; fast-time simulation; conflict detection and resolution; detect-and-avoid; controller workload.

## 1. INTRODUCTION

In a non-segregated airspace, aircraft with very different operational profiles, such as commercial transport, general aviation, manned military aircraft, and unmanned aircraft systems (UAS), coexist within the same continuous volume rather than being separated into dedicated, reserved blocks [1–3]. This integration is the direct operational consequence of the Flexible Use of Airspace (FUA) concept [4], under which separation between dissimilar users must be assured tactically, through real-time procedures and tools, rather than structurally, through rigid airspace design. Within this shared volume, a growing class of complex

National Institute for Aerospace Research "Elie Carafoli" (INCAS), B-dul Iuliu Maniu 220, Bucharest, Romania

Ro. J. Techn. Sci. – Appl. Mechanics, Vol. 71, N° 1, P. 5–23, Bucharest, 2026

DOI:10.59277/RJTS-AM.2026.1.01

special operations must also be accommodated: the detection and handling of non-cooperative drones, counter-UAS activities, military transits through civil airspace, and time-critical cooperative maneuvers such as air-to-air refueling [5, 6]. These specialized missions place heavy demands on separation assurance and raise controller workload well beyond the levels of routine traffic.

ATM simulation capability is an essential enabler for validating the safe integration of special operations into non-segregated airspace. This is also considered as a key precondition for safely integrating special or unmanned operations into non-segregated airspace, because it lets regulators and operators validate interoperability, procedures, capacity, human factors, and safety before live deployment. Non-segregated airspace means the new operation must coexist with existing traffic, so the simulation has to show that separation, communications, and controller interaction remain safe and efficient. One source [7] explicitly says that integrated RPAS/ATC simulation can support ATM validations for human factors, capacity, and efficiency issues in non-segregated airspace. Another ICAO paper [8] states that full integration requires equivalent equipage, performance, and operator competency, and that approved UTM systems should be interoperable with existing ATM and demonstrably at an equivalent level of safety.

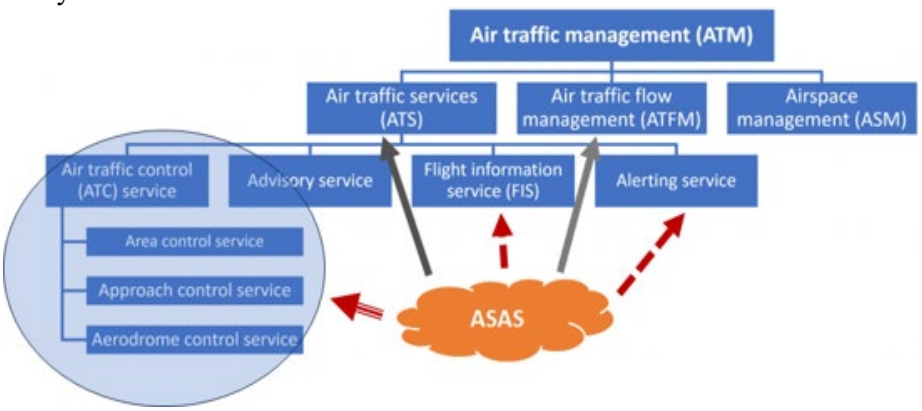


Fig. 1 – ASAS as a trustworthy machine reasoning simulator [9]

In practice, this means simulation is not the only requirement, but it is a necessary validation step for proving that the special operation can behave predictably and coexist with conventional aircraft without degrading safety. The literature also highlights detect-and-avoid capability, training, procedures, and ATC interoperability as part of that broader safety case. The critical questions surrounding these operations are fundamentally empirical rather than merely theoretical. It remains essential to understand how a highly heterogeneous traffic picture behaves as an interconnected whole, how human controllers cope with the compounding complexity, and whether candidate procedures or decision-aiding

tools provide effective operational support. Such questions cannot be answered confidently through analytical models alone, yet testing them in live airspace introduces unacceptable cost and safety risk. For over four decades, the research community has turned to simulation as the principal means of answering them [10–11].

The major conclusion from ASAS Project [9] and the complex setup as in Fig. 1 was that there is no single universal "ATM simulation fidelity standard" specific to non-segregated airspace. We advocate for a performance-based, human-in-the-loop approach that requires realistic traffic, controller interaction, communications, trajectories, and contingencies to be validated before operations are accepted. For RPAS or other special operations in shared airspace, the simulation should reproduce ATC interaction, separation assurance, traffic conflicts, voice/data-link behavior, latency, surveillance, and lost-link contingencies closely enough to evaluate controller workload and capacity effects. The SESAR work on real-time RPAS simulation [12] explicitly says the simulator should include a realistic RPAS operation, an ATM environment that integrates traffic and RPAS, and historical or forecast IFR traffic with the corresponding airspace structure.

In developing our new ASAS+ advanced simulator, a useful benchmark is whether the simulation can represent the same functional blocks and conceptual data flows as the real ATM environment, rather than just generating plausible aircraft tracks. A dedicated paper [13] describes using real-time voice communications, transponder and ADS-B behavior, datalink, satellite-induced latency, and controller working positions to assess workload and capacity impacts in mixed airspace. That implies fidelity must be high enough to support safety-case evidence, not merely visual realism.

In practical terms, the standard is: the simulator must be credible enough that regulators and ANSPs can use its outputs to judge separation safety, ATC workload, contingency procedures, and interoperability with existing ATM systems. If the scenario is about non-segregated airspace, the key test is whether the simulator can expose the special operation to realistic surrounding IFR/VFR traffic and realistic ATC procedures without hiding the operational edge cases that matter.

This paper surveys published academic and industrial research and reports on a new development towards ASAS+ system at INCAS. Its contribution is twofold: first, it provides a structured synthesis of a domain whose foundational research is currently scattered across the air traffic management (ATM), unmanned-systems, and human-factors literatures; second, it assembles a cohesive development strategy for a dedicated, high-fidelity new simulation capability (ASAS+) towards safely integrating special operations into non-segregated environments.

The remainder of the paper is organized as follows. Section 2 frames the core operational problem and Section 3 establishes why simulation is the instrument of choice. Section 4 surveys ASAS+ simulation paradigms and the platforms that

embody them, and Section 5 reviews the analytical methods validated through these simulations. Sections 6 and 7 then discuss the strategic importance of the capability and map the remaining gaps in the literature.

## 2. NON-SEGREGATED AIRSPACE AND SPECIAL OPERATIONS

Classical airspace management keeps incompatible activities apart by reserving areas from which other traffic is excluded. Segregation is safe but wasteful, since reserved volumes are unavailable even when the activity inside them is intermittent. The FUA concept replaces static partitions with a negotiated, rolling allocation in which airspace is treated as a single continuum [4]; the benefit is capacity, and the cost is that separation between dissimilar users must be assured tactically by air navigation procedures and supporting tools framed in standards such as ICAO Doc 4444 and Annex 11 [1–3]. The design and effect of those tools must themselves be studied before they are fielded, which is the first place simulation enters.

The pilot's historic duty to "see and avoid" [3] is itself limited: the foundational analysis by Australia's Bureau of Air Safety Investigation found that unalerted see-and-avoid is unsuitable as a primary means of separation, owing to human visual and attentional constraints [14]. With no pilot on board even this limited faculty is absent, which is why an engineered *detect-and-avoid* capability must replace it, and detect-and-avoid has accordingly become a research field of its own [15-16]. The integration of remotely piloted aircraft into non-segregated airspace under instrument flight rules is the subject of dedicated ICAO guidance [17]. In Europe the route to routine drone operation is the U-space framework (Commission Implementing Regulations (EU) 2021/664, 665, 666), which mandates services such as network identification, geo-awareness and traffic information [5, 18]; for military unmanned aircraft, NATO STANAG 4671 sets airworthiness requirements aimed expressly at operation in non-segregated airspace [6], with the civil-military accommodation of military RPAS in non-segregated (general air traffic) airspace addressed by joint EASA-EDA guidance [19]; and the wider SESAR program coordinates the validation of integration concepts at European scale [20], within the long-term roadmap set out in the European ATM Master Plan [21]. These frameworks assume cooperation: an aircraft that announces its identity and state. The *non-cooperative* drone, which does neither, is the hard residual case, and studying how a traffic picture and a controller cope with it belongs squarely in a simulator.

The main challenge for our ASAS+ development is that detect-and-avoid validation depends on both the encounter model and the human/operator model, and small errors in either can change the outcome substantially. In ATM simulations, you are not just testing sensors; you are testing how the DAA logic,

traffic environment, communications, and remote pilot or controller responses interact under realistic encounter conditions.

One difficulty is scenario realism: real near-miss encounters are rare, so validation often relies on artificially generated encounter sets that must still have realistic distributions of geometry, closure rates, and airspace context. Another is that the same DAA system can behave very differently under deterministic versus Monte Carlo runs, especially when sensor noise, latency, or pilot reaction variability are included.

A second challenge is modeling the remote pilot or controller correctly, because response delay, situational awareness, and compliance with resolution advisories can materially change whether the system appears safe. SESAR demonstration findings also suggest that normal operations may look manageable, but abnormal events and handover/communication edge cases increase workload and require specific procedures.

A third issue is fidelity of the ATM environment itself: the simulator must include surveillance, datalink, voice, transponder/ADS-B behavior, and sector handovers closely enough to expose integration failures rather than hide them. EUROCONTROL's validation platform framing also emphasizes that testing must account for airspace particularities and use representative encounter sets, because the system must be fit for a specific operational context, not just for generic safety.

The validation problem is less "does the DAA algorithm work in theory?" and more "does it still work when realistic traffic, sensors, latency, operator behavior, and airspace-specific encounter patterns are all combined?". A strong validation campaign therefore needs broad sensitivity analysis, representative scenarios, and human-in-the-loop simulation, not just algorithmic test cases.

Special operations bracket the spectrum of difficulty that integration imposes. At the adversarial, information-poor end sit the detection and handling of non-cooperative or maneuvering drones and counter-UAS tasks, where the target neither cooperates nor flies predictably [22]. At the cooperative but unforgiving end sit precise, time-critical maneuvers such as automated air-to-air refueling [23] and formation operations [24], where aircraft must arrive at the same point at the same time. Military operations transiting civil airspace combine elements of both. Because these operations push separation assurance, surveillance and controller workload to their limits, they are exactly the cases for which experimentation is most necessary and live testing least acceptable, and therefore the cases that most depend on simulation.

## 3. WHY SIMULATE: SIMULATION AS A RESEARCH INSTRUMENT

Studying mixed traffic and special operations in live airspace is unsafe, costly and unrepeatable; for the failure cases that matter most it is simply not permissible. The literature is consistent that simulation removes these constraints and, in doing

so, confers several distinct research advantages: **safety**, since hazardous encounters can be staged without consequence; **repeatability**, since a scenario can be run many times with one variable changed at a time, which is the basis of the open-data, open-source push for comparable ATM research [25]; **controllability and observability**, since traffic and events can be scripted exactly and every state measured, including quantities no live system exposes; and **access to the edges**, since rare, extreme and adversarial situations can be visited deliberately. Together these turn open operational questions into controlled experiments, organized as a repeatable research loop (Fig. 2): a scenario is designed, run, analyzed by embedded logic, observed and measured, then evaluated and refined.

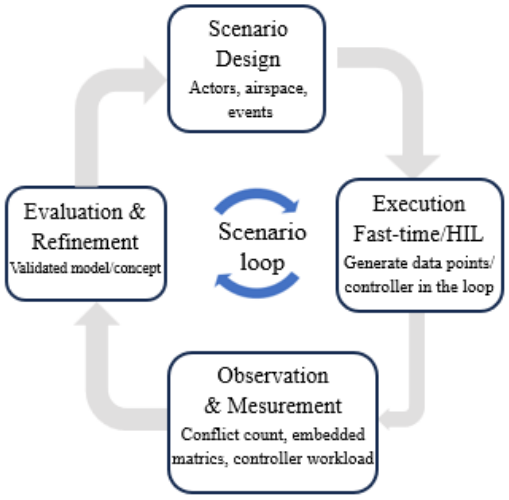


Fig. 2 – The scenario-research loop that recurs across the simulation literature.

Beyond research, the same capability underpins three further uses that the literature documents: **validation toward operational readiness**, since concepts are matured and demonstrated in representative environments before deployment under a structured validation methodology [26], the role NASA built FACET for [27] and that EUROCONTROL's real-time platform serves for controller-in-the-loop validation [28]; **regulatory and concept validation**, since the Free Flight concept was designed and validated through simulation and human-factors study [29], as later integration concepts have been within SESAR [30]; and **training** against reproducible, escalating situations. A platform credible for these uses must reach adequate fidelity: representative traffic, representative displays and, for human-factors work, rated operators. Readiness is conventionally expressed on the Technology Readiness Level scale, on which TRL 7 denotes a prototype demonstrated in an operational environment.

For an aviation ATM simulator in special-ops use, the most useful advanced logic is a multi-layer model that combines traffic flow, controller workload, security events, and contingency handling. Research and operational simulator platforms support dynamic sectorization, conflict detection/resolution, communications emulation, and scenario-based attack or anomaly injection, which maps well to special-ops training. The core simulation layers used are:

- Traffic layer: aircraft trajectories, routes, speeds, separation minima, runway sequencing, metering, and holding.
- Control layer: sector ownership, handoffs, clearances, workload, and coordination between ATC roles.
- Security layer: anomalous instructions, comms spoofing, cyber events, insider actions, denied access, and data corruption.
- Contingency layer: weather deviation, runway closure, radar loss, nav aid outage, and emergency priority handling.

The special-ops logic should prioritize threat response over nominal efficiency. That means we need explicit escalation states such as normal, suspicious, confirmed threat, degraded control, and recovery, with rule-based transitions that can override standard separation optimization. For realism, the simulator should also model how operators re-route traffic, isolate sectors, switch to backup comms, and preserve logs for post-event analysis.

A good architecture model that has already been introduced in ASAS project [0] is event-driven with rule precedence. For example:

- Ingest traffic and sensor state.
- Evaluate safety conflicts.
- Evaluate security/threat conditions.
- Resolve priority conflicts using a policy matrix.
- Emit controller, pilot, and system responses.

This structure matches large-scale ATM simulators that model workload, conflict detection, and tactical resolution while supporting scenario-driven evaluation. In ASAS+ we are interested in a rather simple state model and a compact state machine that include: AdvisoryAlert, TrafficDeconfliction, SecurityInvestigate, RestrictedOps, and Recovery as main modules. For severe cases we consider: ThreatDetected, SectorIsolation, AlternateRouting, BackupComms, and IncidentClosure.

The model became more realistic by adding timers, hysteresis, and confidence scores so the system does not overreact to one bad message. In practical implementation in a research simulator, the best pattern is to separate:

- deterministic flight physics and traffic rules,
- stochastic threat injection,
- controller workload estimation,

- and response policy scripts.

Separating simulator logic from the traffic layer using a scenario Domain Specific Language (DSL) is a highly effective architecture for flight simulation. It prevents tightly coupled code, allowing writing of testable, declarative scenarios that run entirely decoupled from SimConnect or other flight simulators. In ASAS+ we can turn this into a C++ architecture for a Prepar3D/SimConnect-style environment introducing scenario DSL for ATM logic.

## 4. SIMULATION PARADIGMS AND PLATFORMS

The literature divides simulation into two complementary paradigms. **Fast-time** simulation runs faster than real time with every component automated; by executing a scenario, or a swept family of scenarios, many times, it builds statistical confidence about outcomes such as conflict rates, capacity limits or the safety of a detect-and-avoid logic, and it is the case of Monte-Carlo safety assessment built on statistical encounter models and multi-agent risk methods [30, 31]. **Real-time, human-in-the-loop (HIL)** simulation runs at wall-clock speed with operators inside the loop and is the principal way to study human factors (controller workload, situational awareness and decision-making) under realistic operating conditions [28–29]. The two are complementary: fast-time sweeps identify the conditions worth examining, and HIL sessions probe those conditions with a human present. A platform that supports both can move a question from broad statistical screening to focused human-centered evaluation within one framework.

Several platforms recur across the literature and illustrate the design space. NASA's Future ATM Concepts Evaluation Tool (FACET) is a fast-time, system-wide tool able to model thousands of trajectories on a desktop, built to explore concepts such as airborne self-separation before field testing [27]; the agent-based Airspace Concept Evaluation System (ACES) plays a comparable nationwide role [32]. BlueSky, an open-source, open-data simulator from TU Delft, was created precisely to make ATM research reproducible and freely modifiable, addressing the recurring complaint that proprietary simulators are too costly to adapt [25].

*Table 1*

Representative air-traffic-management simulation platforms in the literature

| Platform | Mode | Origin | Primary focus | Ref |
|---|---|---|---|---|
| **FACET** | Fast-time | NASA Ames | System-wide NAS concept evaluation; CD&R; self-separation | [27] |
| **BlueSky** | Fast-time + real- | TU Delft (open | Reproducible, freely | [25] |

| | time | source) | modifiable ATM research | |
|---|---|---|---|---|
| **ESCAPE** | Real-time HIL | EUROCONTROL | Controller-in-the-loop concept validation | [28] |
| **En-route digital twin** | Fast-time + HIL | NATS / Turing / Exeter | Training and evaluating AI agents for ATC | [33] |
| **DAIDALUS** | Library (Monte-Carlo sim) | NASA Langley | Detect-and-avoid / well-clear for UAS | [15] |
| **ACES** | Fast-time (agent-based) | NASA | Nationwide NAS concept evaluation | [32] |

EUROCONTROL's ESCAPE is a scalable real-time platform used to validate operational concepts with rated controllers [28], while recent probabilistic digital twins of en-route airspace combine fast-time execution for training and benchmarking automated agents with HIL interfaces for assessment by qualified controllers [33]. Beyond these, dedicated digital-twin architectures are being proposed for the airspace of the future, coupling high-fidelity models with live data feeds [34]. Table 1 summarizes representative examples. A common pattern is the use of commercial or open flight simulators as the traffic core, connected through a data interface to the analysis logic [35–36].

Across these examples several research-grade properties recur, worth naming because together they form the literature's implicit specification for a useful simulation capability (Fig. 3).

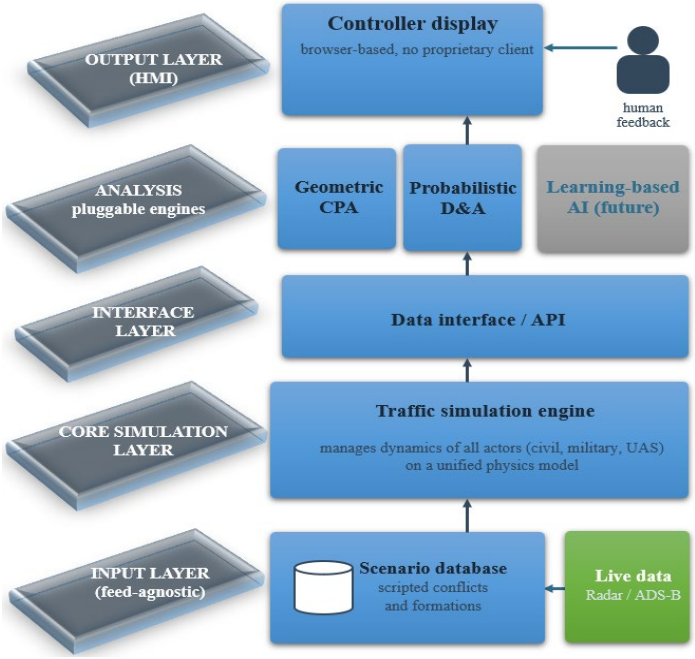


Fig. 3 – Anatomy of a research-grade ATM simulation platform.

A **reusable, shareable scenario library and open data** make experiments reproducible and comparable [25]; **replaceable analysis engines** let competing detection or decision-aiding methods be compared within an unchanged scenario; **feed-agnostic input** lets simulated traffic and live surveillance be interchanged; and a **low-friction interface** lets evaluators, including operational controllers, take part. Where these properties are present, a single platform can serve fast-time screening, HIL evaluation, validation and training; where they are absent, research fragments across incompatible tools.

Below is a clean architecture for a SimConnect-based ATM simulation environment originated from ASAS project [3] that fits the scenario DSL idea and keeps the simulator logic separate from the SimConnect transport layer. SimConnect is a client/server, asynchronous API, and Microsoft's SDK supports C++ out-of-process add-ons, with x64 as the recommended target for modern builds. Consequently, our architecture use four layers:

- SimConnectTransport: owns the connection, dispatch loop, event registration, and raw SimConnect calls.
- DomainCore: contains the ATM logic, rules engine, scenario state machine, and policy evaluation.
- ScenarioRuntime: parses the DSL and turns it into scheduled events, rules, and initial state.

- Adapters: translate between SimConnect data/events and the domain objects.

This separation keeps the ATM logic testable without the simulator and avoids mixing rule evaluation with SDK callbacks. In more details, we highlight the following characteristics and responsabilities:

1. Transport layer: Connect/disconnect to SimConnect; Register data definitions and client events; Handle DispatchProc / CallDispatch callback flow; Queue simulator updates from the domain layer. The key design points are: Keep SimConnect calls on one dedicated thread; Use a thread-safe command queue from domain logic to transport; Convert SDK data into plain C++ structs before handing it off.

2. Domain layer: Model airspace sectors, aircraft, routes, clearances, comms state, and security state; Run rule evaluation and escalation logic; Track confidence levels, timers, and state transitions; Emit high-level commands such as reroute, hold, close_sector, switch_comms. This layer should know nothing about SimConnect types.

3. Scenario runtime: Loads scenario DSL or JSON; Build initial conditions; Schedule timed/conditional events; Inject threats, outages, weather changes, and traffic reshaping; Validate scenario consistency before starting. This is where the special-ops logic lives, including threat escalation and recovery policies.

4. Adapter layer: Map SimConnect simvars to domain model fields; Map domain commands to SimConnect events or data writes; Handle aircraft identification, sector labeling, and controller handoff mapping. This is the translation boundary between simulator vocabulary and the basic rule vocabulary.

In particular, the reference data flow and the process loop is the following 5-step approach that matches the asynchronous SimConnect model described in the SDK documentation and common client implementations in Prepar3D:

- SimConnect delivers a snapshot or event.
- Transport converts it to SimStateSnapshot.
- Domain engine updates state and evaluates rules.
- Domain engine emits DomainCommand objects.
- Transport translates those into SimConnect actions.

One thread is used for SimConnect I/O and one for domain ticking. The transport thread should only do SDK work, while the domain thread handles rules, scheduling, and scenario evolution. Communication between them is via lock-free or mutex-protected queues, not direct shared mutable state. A practical pattern is the following:

- SimConnectTransport pushes snapshots into an input queue.

- DomainEngine consumes snapshots, produces commands.
- SimConnectTransport consumes commands and sends them out.

This makes the system deterministic and easier to debug under load. In the execution model for the DSL, each rule is compiled into one of these forms: time-based trigger; condition-based trigger; composite trigger with priorities; recovery action with cooldown. This is highly dependent on the anticipated usage. Example of requested execution states are: nominal, alert, sector isolation, alternate routing, backup comms, and recovery. The same engine can be reused for both routine ATM disruption scenarios and special-ops incidents.

## 5. SIMULATION NEEDS AND POTENTIAL IMPLEMENTATION

Conflict detection is the analytical core that every mixed-traffic scenario exercises, and the reference framework for the field remains the survey of Kuchar and Yang, which organized dozens of conflict-detection-and-resolution (CD&R) models around a small set of functional choices (Table 2) [10]. Later reviews extended the taxonomy to unmanned and mixed manned/unmanned aviation, where dense traffic and limited onboard authority spurred a distinct branch of methods [11, 37].

*Table 2*

Functional axes by which conflict-detection-and-resolution methods are classified

| **Functional axis** | **Principal options (Kuchar & Yang [10])** |
|---|---|
| **State dimensionality** | Horizontal / vertical / three-dimensional |
| **State propagation** | Nominal (constant-velocity) / worst-case / probabilistic |
| **Resolution philosophy** | Prescribed / optimized / force-field / manual |
| **Maneuver dimensions** | Speed / lateral / vertical / combined |
| **Multi-aircraft handling** | Pairwise / global |

Simulation has been the proving ground for all of the principal families: geometric/state-based methods, of which the closest-point-of-approach test is the simplest representative [38]; provably-safe maneuver synthesis via reachability and hybrid-systems analysis [39]; optimization-based and automated resolution evaluated under realistic demand [40]; and, more recently, learning-based approaches. No single family dominates; each is characterized, in the literature, by how it trades transparency, optimality, robustness to uncertainty and cost [10, 37].

Across these families, simulation studies have mapped the characteristic trade-offs rather than crowned a winner. Geometric, state-based tests, of which the closest-point-of-approach computation is the simplest, are closed-form,

deterministic and inexpensive, which makes them easy to embed and to audit, but they assume constant-velocity motion and set aside uncertainty and intent. Probabilistic methods propagate a distribution of future states and estimate a conflict probability, trading transparency for a more faithful treatment of wind, surveillance error and maneuver uncertainty; worst-case and reachability-based methods instead bound the set of reachable positions to provide guarantees; and optimization-based and, more recently, learning-based resolvers seek efficiency at the cost of explainability. The recurring contribution of simulation is to expose these trade-offs under controlled, repeatable traffic, replaying the same encounters through different logics and measuring conflict rate, false-alert rate and resolution cost [10, 37].

Because special operations often involve aircraft that do not cooperate, the detect-and-avoid problem (keeping a UAS "well clear" of other traffic without a pilot on board) is central. The literature's reference point is DAIDALUS, a detect-and-avoid alerting and guidance logic whose well-clear definition and algorithms were formally specified, verified, and validated against stressing cases jointly developed by air-force, laboratory and agency partners, and which informs the relevant minimum operational performance standards [15]. Its development relied heavily on Monte-Carlo simulation over large ensembles of encounters, drawing on statistical airspace encounter models built from recorded radar data [30], to characterize alerting performance. Such fast-time evidence is necessary but not sufficient: under accepted validation practice it bounds risk and screens candidate designs, while a credible safety case also requires human-in-the-loop evaluation and flight test [31, 26]. Airborne collision avoidance is governed by standards for TCAS and its successor ACAS X [41], with detect-and-avoid and ACAS Xu standards (DO-365/DO-386) extending the approach to unmanned aircraft [42], themselves developed and certified against extensive simulated encounter sets.

What ultimately determines whether a special operation can be managed safely is the human controller, and here only real-time HIL simulation gives valid answers. The Free Flight studies established the template: a concept and its cockpit and procedural design were validated through HIL simulation with explicit attention to safety and human factors [29]. The same approach underlies modern controller-in-the-loop validation on platforms such as ESCAPE [28] and the HIL assessment interfaces built into recent en-route digital twins, where qualified controllers observe, interrogate and intervene in agent-driven traffic [33]. The recurring finding is that it is cognitive complexity, not raw aircraft count, that sets the practical limit: controller workload is the dominant modeled determinant of sector capacity [43], and it is driven by traffic complexity and structure rather than density alone [44–46]. Heterogeneous, non-standard special-operations traffic is therefore disproportionately demanding, because it defeats the structure-based abstractions controllers rely on to manage complexity.

In preliminary development, a practical SimConnect usage is based on the following three steps:

- Registration: data definitions for aircraft position, altitude, speed, and AI object state, client events for custom control commands, notification groups for priority handling.
- Mapping domain commands to (either): SimConnect_TransmitClientEvent, data writes for sim objects, or custom client data channels, depending on the action.
- Testing strategy and building test tiers: unit tests for rule evaluation, replay tests for recorded SimConnect snapshots, integration tests against a running simulator.


Fig. 4 – ASAS validation exercise for special-ops.

A key advantage of this development is that the DSL and logic can be verified without a live simulator for every change. This was tested in a validation exercise for ASAS (Fig. 4). To achieve this, the system is split into three distinct layers:

- The Scenario DSL: Declarative, plain-text files (or a builder API in languages like Python or Rust) that define what happens in a scenario without worrying about how it is rendered or flown. Example: "Spawn aircraft at \(LROP\); wait for 5 minutes; trigger engine failure."
- The Abstract Simulation Engine: Pure logic and state machines that interpret the DSL. It tracks scenario state, evaluates triggers, and emits generic event commands (e.g., SetFuelValve(0), SetAltitude(1000)).
- The Transport Layer Adapter (e.g., SimConnect): A completely separate module that listens to the generic commands from the Engine and translates them into specific API calls (e.g., SimConnect_SetDataOnSimObject).

The main benefits of this approach can be summarized as follows:

- Backend Agnosticism: scenarios can be reused across different flight simulators or used for headless automated testing by swapping out the transport layer.
- Testability: we can mock the SimConnect layer to write robust unit and integration tests without ever booting up the simulator.
- Declarative Readability: complex mission designers can edit plain-text scenarios (using JSON, YAML, or a custom text-based syntax) rather than compiling rigid C++ code.

## 6. DISCUSSION: THE IMPORTANCE OF THE CAPABILITY, AND THE GAP

Drawing the threads together, the literature supports a single conclusion: for special operations in non-segregated airspace, the ability to simulate is a precondition for safe integration rather than an optional aid. The argument is cumulative. The decisive questions are empirical (Section 1); live experimentation on them is unsafe and unrepeatable (Section 3); every method on which these operations depend (conflict detection, detect-and-avoid, collision avoidance) was established and certified in simulation, not in the air (Section 5); and controller workload, the dominant modeled determinant of sector capacity, can only be measured with operators in a simulated loop (Section 5). A research community or an air navigation service provider without a credible simulation capability therefore cannot answer the questions integration poses, cannot validate the tools it proposes, and cannot train its controllers for the operations it will face. The capability is foundational infrastructure.

Purpose-built capability for special operations. The mature platforms surveyed (FACET, BlueSky, ESCAPE, en-route digital twins) target en-route and terminal civil traffic flow and self-separation; few are built specifically around special operations (counter-UAS, interception, military activity in civil airspace, refueling) in a non-segregated setting. Even the U-space framework, at present, largely keeps drones and manned aircraft apart rather than truly integrated [18].

Non-cooperative sensing fidelity. Most platforms are fed by cooperative state (simulated tracks or ADS-B). Studying genuinely non-cooperative traffic at high fidelity requires modeling primary-radar and electro-optical / infrared detection ahead of the analysis stage, which is still uncommon.

Coupling of paradigms. Fast-time and HIL studies are often reported separately; systematically coupling Monte-Carlo screening with selective HIL probing within one scenario framework remains the exception rather than the rule.

Statistical power and reproducibility of HIL. HIL sessions are expensive, so the number of runs and participants bounds the strength of human-factors conclusions; the open-data, open-source movement [25] addresses reproducibility for fast-time work but less so for human-in-the-loop results.

Transparency versus performance. Learning-based methods promise performance gains but at the cost of explainability; the literature has not settled how such methods should be validated in simulation for safety-critical, certifiable use.

## 7. FUTURE DEVELOPMENTS

The gaps above translate into a clear research agenda. Most directly, the field would benefit from simulation environments designed around special operations in non-segregated airspace, hosting mixed cooperative and non-cooperative actors, with realistic non-cooperative sensor models, and able to run the same scenario in fast-time for statistics and in real time with a controller in the loop. Such environments should adopt the research-grade properties the literature already identifies as valuable: open and reusable scenarios for reproducibility [25], replaceable analysis engines so that geometric, probabilistic and learning-based methods can be benchmarked side by side, feed-agnostic interfaces so that simulated and live data are interchangeable, and validation to a high readiness level with rated operators [28, 33]. Interoperability with the emerging operational ecosystem (U-space services and four-dimensional trajectory data [5, 20]) would let such a capability study integration not only as a simulation but as a participant in it.

The ASAS test-bed and its extension towards ASAS+, detailed in Sections 3–5, are a concrete step along exactly this agenda. Reported at TRL 7, the platform has already been exercised on the special operations highlighted here, a non-cooperative unmanned-aircraft interception and an air-to-air refueling rendezvous (Fig. 4), each flown with a certified pilot and controller through the full mission cycle. These exercises show that the research-grade properties and the special-operations focus identified in this review can be brought together within a single environment [47].

## 8. CONCLUSIONS

The integration of civil, military and unmanned traffic into a single non-segregated airspace, and the conduct of special operations within it, raise questions about safety, capacity, workload and tooling that are empirical and that cannot be answered safely or repeatably in live operations. This review has surveyed the literature's answer: simulation, in its complementary fast-time and human-in-the-loop forms, embodied in platforms from FACET and BlueSky to ESCAPE and en-route digital twins, is the instrument through which the methods of conflict detection, detect-and-avoid and collision avoidance were developed and validated, and through which controller workload, the dominant determinant of sector capacity, is measured. A dedicated, high-fidelity simulation capability is a precondition for the safe integration of special operations in non-segregated

airspace, not a convenience; and it has identified where the literature is still thin: purpose-built capability for special operations, non-cooperative sensing fidelity, the coupling of paradigms, and the validation of learning-based methods. Closing those gaps is the natural next step for any institution that intends to support such operations responsibly.

***Acknowledgements.*** This work was supported by the Romanian Ministry of Research, Innovation and Digitization, through the project CNCS – UEFISCDI, contract no. 15SOL(T15)/2024, project number PN-IV-P6-6.3-SOL-2024-2-0224, and the Nucleu Program, contract no. 36N/12.01.2023, project number PN 23 17 06 03.

*Received on June 15, 2026*

## REFERENCES

1. ICAO, *Procedures for Air Navigation Services and Air Traffic Management (PANS-ATM)*, Doc 4444, 16th ed., International Civil Aviation Organization, 2016.
2. ICAO, *Annex 11: Air Traffic Services*, 15th ed., International Civil Aviation Organization, 2018.
3. ICAO, *Annex 2: Rules of the Air*, International Civil Aviation Organization, 2018.
4. EUROCONTROL, *Specification for the Application of the Flexible Use of Airspace (FUA)*, EUROCONTROL-SPEC-0112, Ed. 1.1, 2009.
5. EASA, *U-space Regulatory Framework: Commission Implementing Regulations (EU) 2021/664, 2021/665 and 2021/666*, with associated AMC and GM, 2021.
6. NATO, *STANAG 4671: Unmanned Aerial Vehicle Systems Airworthiness Requirements (USAR)*, Edition 3, 2019.
7. FRESNO, L., ESCRIBANO, D., SÁNCHEZ-ESCALONILLA, P., SÁNCHEZ-PALOMO, L., *Human Factor Impact Assessment of RPAS Integration into Non-Segregated Airspace*, Proc. 6th SESAR Innovation Days (SID 2016), Delft, The Netherlands, 2016.
8. ICAO, *Unmanned Aircraft Systems Traffic Management (UTM) – A Common Framework with Core Principles for Global Harmonization*, Edition 4, International Civil Aviation Organization, 2021.
9. ASAS - System Simulator for Aviation Accident Analysis, under Romanian PN-IV-P6-6.3-SOL-2024-2-0224;
10. KUCHAR, J.K., YANG, L.C., *A Review of Conflict Detection and Resolution Modeling Methods*, IEEE Transactions on Intelligent Transportation Systems, **1**, *4*, pp. 179–189, 2000.
11. JENIE, Y.I., VAN KAMPEN, E., ELLERBROEK, J., HOEKSTRA, J.M., *Taxonomy of Conflict Detection and Resolution Approaches for Unmanned Aerial Vehicle in an Integrated Airspace*, IEEE Transactions on Intelligent Transportation Systems, **18**, *3*, pp. 558–567, 2016.
12. SESAR PJ19, *SESAR Human Performance Assessment Process V1 to V3 – including VLD*, Project Number 731765, H2020-SESAR-2019-1, EUROCONTROL, Edition 00.03.00. January 2020.
13. WANG, Z.; JIANG, P.; WANG, Z.; HAN, B.; LIANG, H.; AI, Y.; PAN, W., *Enhancing Air Traffic Control Communication Systems with Integrated Automatic Speech Recognition: Models*, *Applications and Performance Evaluation*, Sensors **24**, 4715, 2024, https://doi.org/10.3390/s24144715
14. BUREAU OF AIR SAFETY INVESTIGATION (BASI), *Limitations of the See-and-Avoid Principle*, BASI, Canberra, Australia, 1991.
15. MUÑOZ, C., NARKAWICZ, A., HAGEN, G., UPCHURCH, J., DUTLE, A., CONSIGLIO, M., CHAMBERLAIN, J., *DAIDALUS: Detect and Avoid Alerting Logic for Unmanned Systems*,

Proc. IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), Prague, Czech Republic, pp. 5A1-1-5A1-12, 2015.
16. McFADYEN, A., MEJIAS, L., *A Survey of Autonomous Vision-Based See and Avoid for Unmanned Aircraft Systems*, Progress in Aerospace Sciences, **80**, pp. 1–17, 2016.
17. ICAO, *Manual on Remotely Piloted Aircraft Systems (RPAS)*, Doc 10019, 1st ed., International Civil Aviation Organization, Montreal, 2015.
18. BARRADO, C., BOYERO, M., BRUCCULERI, L., FERRARA, G., HATELY, A., HULLAH, P., MARTIN-MARRERO, D., PASTOR, E., RUSHTON, A.P., VOLKERT, A., *U-Space Concept of Operations: A Key Enabler for Opening Airspace to Emerging Low-Altitude Operations*, Aerospace, **7**, *3*, art. 24, 2020.
19. EUROPEAN DEFENCE AGENCY (EDA), EASA, *Guidelines for the Accommodation of Military IFR MALE-type RPAS under General Air Traffic (GAT) in Airspace Classes A-C*, EDA UAS airspace-integration activity, 2019.
20. EUROCONTROL, SESAR JOINT UNDERTAKING, *SESAR Solutions Catalogue and the Digital European Sky Programme*, 2023.
21. SESAR JOINT UNDERTAKING, *European ATM Master Plan 2020: Digitalising Europe's Aviation Infrastructure*, Publications Office of the European Union, 2020.
22. WANG, J., LIU, Y., SONG, H., *Counter-Unmanned Aircraft System(s) (C-UAS): State of the Art, Challenges, and Future Trends*, IEEE Aerospace and Electronic Systems Magazine, **36**, *3*, pp. 4–29, 2021.
23. PARRY, J., HUBBARD, S., *Review of Sensor Technology to Support Automated Air-to-Air Refueling of a Probe Configured Uncrewed Aircraft*, Sensors, **23**, *2*, art. 995, 2023.
24. SEO, J., KIM, Y., KIM, S., TSOURDOS, A., *Collision Avoidance Strategies for Unmanned Aerial Vehicles in Formation Flight*, IEEE Transactions on Aerospace and Electronic Systems, **53**, *6*, pp. 2718–2734, 2017.
25. HOEKSTRA, J. M., ELLERBROEK, J., *BlueSky ATC Simulator Project: An Open Data and Open Source Approach*, Proc. 7th International Conference on Research in Air Transportation (ICRAT), Philadelphia, USA, 2016.
26. EUROCONTROL, *European Operational Concept Validation Methodology (E-OCVM), Version 3.0*, EUROCONTROL, 2010.
27. BILIMORIA, K. D., SRIDHAR, B., GRABBE, S. R., CHATTERJI, G. B., SHETH, K. S., *FACET: Future ATM Concepts Evaluation Tool*, Air Traffic Control Quarterly, **9**, *1*, pp. 1–20, 2001.
28. SEKINE, K., IWATA, D., BOUCHAUDON, P., TATSUKAWA, T., FUJII, K., TOMINAGA, K., ITOH, E., *Validating Flow-Based Arrival Management for En Route Airspace: Human-In-The-Loop Simulation Experiment with ESCAPE Light Simulator,* Aerospace*,* **11***, 11*, art. 866, 2024.
29. HOEKSTRA, J. M., VAN GENT, R. N. H. W., RUIGROK, R. C. J., *Designing for Safety: the Free Flight Air Traffic Management Concept*, Reliability Engineering & System Safety, **75**, *2*, pp. 215–232, 2002.
30. KOCHENDERFER, M.J., EDWARDS, M.W.M., ESPINDLE, L.P., KUCHAR, J.K., GRIFFITH, J.D., *Airspace Encounter Models for Estimating Collision Risk*, Journal of Guidance, Control, and Dynamics, **33**, *2*, pp. 487–499, 2010.
31. STROEVE, S.H., BLOM, H.A.P., BAKKER, G.J., *Systemic Accident Risk Assessment in Air Traffic by Monte Carlo Simulation*, Safety Science, **47**, *2*, pp. 238–249, 2009.
32. SWEET, D. N., MANIKONDA, V., ARONSON, J. S., ROTH, K., BLAKE, M., *Fast-Time NAS Simulation System for Analysis of Advanced ATM Concepts (ACES)*, AIAA Modeling and Simulation Technologies Conference and Exhibit, Monterey, USA, AIAA 2002-4593, 2002.
33. PEPPER, N., KEANE, A., HODGKIN, A., GOULD, D., HENDERSON, E., LAURITSEN, L., VLAHOS, C., DE ATH, G., EVERSON, R., CANNON, R., SIERRA-CASTRO, A., KORNA, J., CARVELL, B.J., THOMAS, M., *A Probabilistic Digital Twin of UK En Route*

*Airspace for Training and Evaluating AI Agents for Air Traffic Control*, AIAA SciTech Forum, 2026.
34. SOUANEF, T., AL-RUBAYE, S., TSOURDOS, A., AYO, S., PANAGIOTAKOPOULOS, D., *Digital Twin Development for the Airspace of the Future*, Drones, **7**, *7*, art. 484, 2023.
35. MICROSOFT, *SimConnect SDK Reference*, data interface between flight simulator and analysis software.
36. LOCKHEED MARTIN, *Prepar3D SDK Documentation*, data interface between flight simulator and analysis software.
37. RIBEIRO, M., ELLERBROEK, J., HOEKSTRA, J., *Review of Conflict Resolution Methods for Manned and Unmanned Aviation*, Aerospace, **7**, *6*, art. 79, 2020.
38. CHIANG, Y. J., KLOSOWSKI, J. T., LEE, C., MITCHELL, J. S. B., *Geometric Algorithms for Conflict Detection/Resolution in Air Traffic Management*, Proc. 36th IEEE Conference on Decision and Control (CDC), San Diego, USA, pp. 1835–1840, 1997.
39. TOMLIN, C., PAPPAS, G. J., SASTRY, S., *Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems*, IEEE Transactions on Automatic Control, **43**, *4*, pp. 509–521, 1998.
40. ERZBERGER, H., LAUDERDALE, T. A., CHU, Y.-C., *Automated Conflict Resolution, Arrival Management, and Weather Avoidance for Air Traffic Management*, Proc. IMechE, Part G: Journal of Aerospace Engineering, **226**, *8*, pp. 930–949, 2012.
41. RTCA, *DO-385: Minimum Operational Performance Standards (MOPS) for Airborne Collision Avoidance System Xa (ACAS Xa)*, RTCA Inc., Washington DC, 2018.
42. MANFREDI, G., JESTIN, Y., *An Introduction to ACAS Xu and the Challenges Ahead*, Proc. IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, USA, pp.1–9, 2016.
43. MAJUMDAR, A., OCHIENG, W. Y., *Factors Affecting Air Traffic Controller Workload: Multivariate Analysis Based on Simulation Modeling of Controller Workload*, Transportation Research Record, **1788**, *1,* pp. 58–69, 2002.
44. LAUDEMAN, I. V., SHELDEN, S. G., BRANSTROM, R., BRASIL, C. L., *Dynamic Density: An Air Traffic Management Metric*, NASA/TM-1998-112226, NASA Ames Research Center, 1998.
45. KOPARDEKAR, P., MAGYARITS, S., *Dynamic Density: Measuring and Predicting Sector Complexity [ATC]*, Proc. 21st IEEE/AIAA Digital Avionics Systems Conference (DASC), Irvine, USA, 2002.
46. HILBURN, B., *Cognitive Complexity in Air Traffic Control: A Literature Review*, EEC Note No. 04/04, EUROCONTROL Experimental Centre, 2004.
47. BORDEI, A.-M., DIȚEI, M., NICHIFOR, S.-E., IONEL-FAKHOURI, A. M., NAE, C., *Real-Time Conflict Detection and Intercept Planning for ATC Simulation*, 1st INCAS Spring Conference, Bucharest, Romania, 2026.