AI-Powered Air Traffic Control System: A LangGraph-Based Approach to Real-Time Flight Management

Ragul Paramasivam
NEU ID: 002839737

Lavanya Vijayakrishnan
NEU ID: 002068008

December 4, 2025

Abstract

This report presents an AI-powered Air Traffic Control system using Google Gemini LLM and LangGraph workflow engine for autonomous flight operations management. The system comprises a FastAPI-based flight simulator and an intelligent agent managing aircraft landing/takeoff at Renton Municipal Airport. Key features include waypoint-based navigation, collision detection, runway conflict resolution, and conversation-aware decision-making (where the AI agent maintains a history of all previous commands issued to each aircraft and uses this context when making new decisions). Through multi-layered safety validation and structured workflows, the system achieves 94.2% command success rate with zero-conflict operations for 3-6 concurrent flights. Performance analysis demonstrates 8-second LLM response times, which are acceptable for routine ATC operations but create sequential processing bottlenecks limiting scalability. Performance degrades beyond the 6-aircraft threshold due to cumulative decision latency and sequencing complexity.

1 Introduction

1.1 Motivation

Air Traffic Control is among the most complex safety-critical domains in modern transportation. Controllers must simultaneously track multiple aircraft, predict trajectories, sequence operations, and maintain safe separation—all in real-time with zero error tolerance. Cognitive load increases exponentially with traffic density, creating capacity constraints and safety concerns.

The FAA currently operates with approximately 3,000 fewer controllers than the optimal 13,000, resulting in reduced schedules, increased delays, and controller fatigue. Major hubs like New York TRACON operate at 54-65% capacity. With 2-4 year training pipelines and 4-5% annual traffic growth, full staffing may not return until 2030, creating urgent need for AI-assisted systems.

Recent Large Language Models demonstrate remarkable reasoning capabilities but remain largely unexplored in safety-critical applications due to reliability concerns. This work addresses: *How can LLMs enable autonomous air traffic control while maintaining required safety, reliability, and determinism?*

1.2 Objectives

This project aims to develop and evaluate an AI-powered air traffic control system that addresses the following objectives:

- Design a workflow-based architecture that structures LLM decision-making into discrete, validated steps with explicit safety guarantees
- Implement multi-layered safety validation mechanisms to ensure zero separation violations and runway conflicts
- Develop effective prompt engineering strategies for aviation domain tasks that balance flexibility with deterministic behavior
- Create a real-time integration between AI agent and flight simulator demonstrating practical feasibility
- Evaluate system performance under various traffic densities and identify operational capacity limits
- Demonstrate that LLMs can be safely deployed in safety-critical domains through appropriate architectural safeguards

2 Methods

2.1 System Architecture

The system comprises two components communicating via REST APIs:

Flight Simulator: FastAPI-based physics engine with 1Hz state updates, implementing simplified kinematic equations with realistic constraints (3°/s turn rate, 1000 ft/min climb). WebSocket provides real-time radar visualization.

AI Agent: Autonomous controller with: (1) Main loop polling simulator every 2s detecting state changes; (2) LangGraph workflow orchestrating decision nodes; (3) Domain models for airports, runways, flights; (4) Safety validation with collision prediction; (5) SQLite databases for conversation history and flight tracking; (6) Gemini API wrapper with retry logic.

2.2 LangGraph Workflow

The state machine workflow contains four nodes:

Entry: Loads conversation history providing context of previous commands to this flight.

Landing/Takeoff Decision: Invokes Gemini 2.5-flash (temperature 0.1) with prompts containing flight state, waypoints, traffic pattern rules, weather from NOAA API, conversation history, and JSON output schema.

Safety Check: Validates commands through: (a) JSON/schema validation; (b) Runway conflict detection; (c) Pattern conflict detection; (d) 2-minute collision prediction with 5nm horizontal and 1000ft vertical thresholds.

End: Sends validated command to simulator and logs to database.

Conditional edges enable up to 3 retry attempts with safety feedback if validation fails.

2.3 Prompt Engineering

Landing prompts structure: role definition, flight state, traffic pattern (DOWNWIND→BASE→FINAL), landing rules, waypoints, weather, conversation history, decision criteria, strict JSON format. Key principles: explicit constraints (must pass FINAL), sequential reasoning guidance, low temperature for determinism, safety emphasis.

Takeoff prompts focus on runway availability, denying clearance if aircraft on FINAL, RUNWAY, or actively landing.

2.4 Safety Validation

Collision Detection: Two-stage approach. Stage 1 checks current runway/pattern conflicts. Stage 2 predicts positions over 120s using linear extrapolation:

$$\vec{p}_i(t) = \vec{p}_i(0) + \vec{v}_i \cdot t, \quad h_i(t) = h_i(0) + r_{\text{climb},i} \cdot t \tag{1}$$

Conflicts detected when $d_{\text{horiz}}(t) < 5$ nm and $|h_i(t) - h_j(t)| < 1000$ ft.

Runway Conflicts: Block takeoff if any aircraft on RUNWAY, FINAL (¡3nm), or in landing/-takeoff state. Block landing clearance if runway occupied or recent clearance issued.

2.5 Database Design

`atc_communications` table stores timestamp, callsign, command, LLM response, retry count, and success status. Flight tracking database monitors state changes triggering agent intervention. Conversation history reduces repeated commands 18.7% to 2.3%.

3 Data

3.1 Flight State Format

Simulator provides JSON with callsign, type (arrival/departure), position (lat/lon), altitude, speed, heading, target waypoint, state, distance, and clearance flags.

3.2 Waypoints and Rules

System defines 11 waypoints: Entry points (NORTH, SOUTH, EAST, WEST at 6000'), Traffic pattern (DOWNWIND 2000', BASE 1500', FINAL 1000', RUNWAY 0'), Sequencing points (ALPHA, BRAVO, CHARLIE for spacing).

Landing requirements: altitude ≤1500 ft, speed 100-180 kt, distance ≤18 nm, must be on FINAL, runway clear.

3.3 API Endpoints

Endpoint	**Method**	**Purpose**
/api/flights	GET	All active flights
/api/flights/landing	GET	Arriving flights
/api/flights/{callsign}/command	POST	Issue command
/api/simulation/spawn/arrival	POST	Create arrival

3.4 Data Flow

Agent polls API every 2s → Compares with database to detect changes → Triggers workflow on waypoint passage or takeoff readiness → Loads conversation history → LLM generates JSON command → Safety validation → Sends to simulator → Updates databases. Typical 30-min simulation: 900 polls, 50 LLM calls, 100 database entries, <1MB storage.

4 Results

4.1 Performance Metrics

Operation	Mean (ms)	95th %ile (ms)
API Polling	45	78
LLM Response	7,980	10,690
Safety Validation	12	18
Total Cycle	**8037**	**10786**

Table 1: Agent Response Times

Over 50 simulation runs: 94.2% successful landings, 5.8% commands rejected by safety checks, 87.3% retry success, 98.6% takeoff sequencing accuracy. The system maintains zero-conflict operations with 3-6 concurrent flights. Beyond this threshold, the increasing complexity of sequencing and decision-making begins to exceed the agent's capacity, resulting in occasional separation violations.

4.2 Safety Validation Analysis

Among 680 LLM-generated commands: 641 (94.2%) accepted first attempt, 39 (5.8%) rejected (18 runway conflicts, 12 pattern conflicts, 9 predicted collisions), 34 (87.3%) successful after retry, 5 (1.3%) failed all retries.

Scenario	Potential Conflicts	Prevented
Light (2-3 aircraft)	0	0 (100%)
Moderate (4-6)	12	12 (100%)
Heavy (7-9)	31	23 (74.2%)
Stress (10+)	58	31 (53.4%)

Table 2: Collision Avoidance Performance

4.3 Operational Capabilities

Traffic Pattern: Arrivals enter at entry points (6000'), vectored to sequencing waypoints as needed, descended into pattern (DOWNWIND→BASE→FINAL), landing clearance when on FINAL with clear runway. Typical completion: 8-12 minutes.

Departures: Spawn on runway, agent monitors arriving traffic, takeoff clearance only when no aircraft on FINAL, runway clear, no arrivals landing. Typical ground time: 45-90s.

Concurrent Operations: Successfully manages simultaneous arrivals/departures up to 6 aircraft. Success rates: Low traffic (2-3 aircraft) 100%, Moderate (4-6) 98.3%, High (7-9) 76.8%, Peak (10+) 54.1%. Performance degradation beyond 6 concurrent flights stems from LLM decision latency, suboptimal sequencing under high complexity, and exceeded collision prediction capacity.

5 Discussion

5.1 Architectural Strengths

Separation of Concerns: Deterministic safety checks ensure no unsafe commands execute regardless of LLM behavior, enabling modularity, auditability, and failure isolation. This demonstrates viable LLM deployment in safety-critical domains: leverage reasoning while enforcing deterministic constraints.

Workflow Benefits: Structured reasoning through explicit nodes, automatic retry recovery, inspectable state, and composability for additional decision stages.

Context Memory: Conversation history enables consistency, learning from failures, temporal reasoning, and reduces redundant commands.

5.2 Limitations

LLM Reliability: Observed failures include hallucinated waypoints, numerical precision errors, sequencing mistakes, and API timeouts—all caught by validation but indicating inherent unreliability.

Edge Cases: No provisions for runway changes, or emergencies. Suboptimal sequencing with >6 arrivals, leading to increased collision risk. No dynamic weather change handling. Poor API failure recovery.

Capacity Limitations: System performance degrades significantly beyond 6 concurrent aircraft due to: (1) Sequential processing bottleneck—each aircraft requires 8,037ms decision cycle, creating cumulative delays; (2) Combinatorial explosion in sequencing complexity—with n aircraft, the agent must consider $O(n^2)$ pairwise conflicts; (3) LLM context limitations—prompts become unwieldy when describing 7+ aircraft states simultaneously; (4) Collision prediction computational cost—safety validation time increases quadratically with aircraft count. These limitations represent fundamental architectural constraints rather than implementation bugs.

Scalability: Current architecture reliably handles 3-6 concurrent flights before collision risks increase significantly. The primary bottlenecks are: (1) LLM decision latency (7,980ms per flight) creating cumulative delays with multiple aircraft, (2) Single-threaded workflow processing limiting parallelization, (3) Increasing combinatorial complexity in sequencing decisions beyond 6 aircraft, (4) Safety validation becoming computationally expensive with many pairwise collision checks. Scaling to 10+ flights would require fundamental architectural changes: airspace sectorization with independent agents per sector, parallel LLM calls with asynchronous processing, hierarchical decision-making separating strategic (long-term sequencing) from tactical (immediate commands) decisions, and optimized collision detection algorithms (spatial indexing, pruning).

Determinism Trade-off: Low temperature (0.1) reduces variability but limits creative solutions to novel situations.

5.3 Comparison with Traditional Approaches

LLM-based approaches achieve comparable safety when properly constrained, with superior flexibility and natural language understanding.

Approach	**Advantages**	**Disadvantages**
Rule-Based	Deterministic, provable	Brittle, exhaustive rules
Optimization	Globally optimal	Computationally expensive
LLM-Based	**Flexible, contextual**	**Probabilistic, needs validation**

Table 3: ATC Automation Approaches

5.4 AI Safety Implications

This work demonstrates: (1) External validation enables safe deployment of unreliable AI components; (2) Structured workflows reduce failure attack surface; (3) Human-in-the-loop approval easily integrated; (4) Complete decision traceability enables post-hoc analysis. Suggests pathway for gradual AI deployment: advisory roles → validation layers → progressive autonomy.

5.5 Future Directions

Technical: Parallel processing architecture to overcome 6-aircraft capacity limit through asynchronous LLM calls and concurrent workflow execution. Multi-agent coordination for arrivals/departures/ground operations. Reinforcement learning from expert demonstrations. ML-based trajectory prediction. Voice interface for pilot communication. Hybrid systems combining LLM reasoning with traditional optimization for high-density scenarios.

Research: Performance across LLM architectures (GPT-4, Claude, Llama), few-shot learning for emergencies, optimal autonomy vs. constraints balance, human trust factors.

Deployment: Formal verification, redundancy mechanisms, FAA certification, human pilot testing, existing system integration.

6 Conclusion

This work presents a complete AI-powered ATC system demonstrating LLM feasibility for safety-critical autonomous decision-making. Key contributions include LangGraph workflow combining LLM reasoning with deterministic validation, comprehensive safety mechanisms (runway/pattern conflicts, predictive collision avoidance), effective aviation prompt engineering, and database-driven conversation memory.

Experimental results show: (1) External validation ensures safe operations with 3-6 concurrent flights; (2) Sub-second response times meet real-time ATC requirements for low-to-moderate traffic; (3) LLMs successfully reason about complex spatial-temporal aviation relationships; (4) Conversation history significantly improves decisions; (5) Performance degrades beyond 6 aircraft due to decision latency and sequencing complexity; (6) 5.8% rejection rate highlights critical need for validation layers.

The architecture pattern (LLM reasoning + deterministic validation) applies to other safety-critical domains (medical diagnosis, autonomous vehicles, industrial control). While significant work remains before real-world deployment, this proof-of-concept demonstrates LLM-based autonomous control viability and offers safe AI deployment patterns for high-stakes domains.

Important caveats: tested only in simulation with simplified physics; system capacity limited to 3-6 concurrent aircraft before collision risks increase; real aviation involves complexities not modeled (weather dynamics, communication variability, equipment failures); 5.8% rejection rate unacceptable in production without improvements; sequential processing architecture creates fundamental scalability constraints; regulatory barriers remain substantial.

The open-source nature enables further research with AI control systems, potentially accelerating progress toward capable, reliable AI agents. As LLM capabilities improve, architectures like this may become increasingly viable for real-world deployment in aviation and beyond.