Collision Avoidance Ontologies for UAS

Finley Holt

2025-12-25

Table of contents

1 Collision Avoidance Ontologies for UAS

Type: Domain Application Ontology Review **Focus**: Ontology-Based Collision Avoidance Systems for Unmanned Aircraft **Relevance**: Safety-critical autonomous decision-making for UAV operations

1.1 Overview

Collision avoidance represents one of the most critical safety challenges for autonomous UAS integration into civilian airspace. Ontology-based approaches to collision avoidance leverage formal knowledge representation to enable UAVs to make real-time decisions about separation distances, conflict detection, and resolution maneuvers while maintaining regulatory compliance.

Unlike purely reactive systems (e.g., simple distance thresholds) or black-box learned approaches (e.g., deep reinforcement learning without constraints), ontology-driven collision avoidance systems combine explicit safety rules, spatial reasoning, and semantic understanding of flight contexts to produce explainable, verifiable autonomous behaviors.

1.2 Decision Impact for Flyby-F11

1.2.1 ADOPT - High Confidence

Collision Avoidance as Canonical Problem - Collision avoidance provides a well-scoped, safety-critical test case for SUMO-constrained RL - Clear success/failure metrics (safety violations vs. mission completion) - Regulatory alignment ensures practical applicability for defense/civilian airspace integration - **Action**: Implement collision avoidance as primary validation scenario in simulation

Spatial Reasoning Ontology - Formal spatial relations (`TooClose`, `CollisionCourse`, `SafeSeparation`) enable verifiable safety constraints - Proven approach in UAV domain (reduces collision rates vs. unconstrained learning) - Directly maps to sensor inputs (vision, depth, ADS-B) through perception pipeline - **Action**: Develop spatial relations ontology as core component of ONTOLOGY_FOUNDATION.md

Safety Constraint Encoding via SWRL Rules - Declarative rule encoding enables formal verification and certification prospects - Explainable decision-making critical for MCTSSA collaboration and defense applications - Runtime constraint checking provides safety net for RL policy exploration - **Action**: Encode minimum separation distances, geofence boundaries, and emergency protocols as SWRL rules

1.2.2 CONSIDER - Needs Validation

Reasoning Latency for Real-Time Control - OWL reasoners (Pellet, HermiT) may not achieve 10+ Hz reasoning cycles required for UAV control - **Risk**: Inference latency could degrade flight performance or introduce control delays - **Mitigation Options**: - Custom RETE-based rule engine trading completeness for speed - Hybrid architecture: ontology for high-level planning (1 Hz), fast reactive controller for low-level stabilization (100+ Hz) - Pre-compiled rule lookup tables for common scenarios - **Action**: Benchmark reasoner performance on Jetson Orin NX under flight-representative workloads

Coverage vs. Specificity Tradeoff - Overly general ontologies lack precision for safety-critical edge cases - Overly specific ontologies become brittle, fail on novel scenarios not anticipated during design - **Risk**: Ontology may miss critical safety scenarios or over-constrain RL policy - **Mitigation Options**: - Iterative refinement through extensive simulation testing - Fallback to conservative behaviors when ontology encounters unknown situations - Hybrid learning: RL discovers edge cases, engineers formalize them into ontology - **Action**: Establish ontology validation protocol with adversarial scenario generation

1.2.3 AVOID - Evidence Against

Overly General Ontologies - Generic upper-level ontologies without domain-specific grounding lack actionable constraints - Collision avoidance requires precise numerical thresholds (separation distances, velocities), not just abstract concepts - **Evidence**: Systems relying only on high-level concepts without quantitative constraints fail to prevent safety violations - **Implication**: SUMO provides foundation, but domain-specific collision avoidance ontology must include concrete safety parameters

Purely Reactive Collision Avoidance - Simple distance thresholds without contextual reasoning lead to overly conservative or dangerous behaviors - Reactive systems cannot anticipate future conflicts (e.g., intersecting trajectories that are currently distant) - **Evidence**: Reactive approaches in UAV-ON benchmark exhibit poor task completion vs. ontology-guided planning - **Implication**: Ontology must encode predictive spatial reasoning (trajectory intersection, time-to-collision), not just instantaneous geometry

1.2.4 INVESTIGATE - Open Questions

Sensor Fusion Integration with Ontological Concepts - How do we map noisy, probabilistic sensor data (vision detections, depth estimates) to crisp ontological assertions? - Example: Vision system detects object with 75% confidence at 8m ± 2m range. Does this satisfy `SafeSeparation(UAV, Object)` if threshold is 10m? - **Research Direction**: Probabilistic extensions to OWL (e.g., PR-OWL) or Bayesian logic integration - **Near-Term Approach**: Conservative interpretation (treat uncertain detections as worst-case for safety rules)

Uncertainty in Ontological Relations - Spatial relations (`TooClose`, `Approaching`) depend on sensor accuracy and prediction horizon - Sensor noise, GPS drift, and wind disturbances introduce uncertainty into geometric calculations - **Research Direction**: Fuzzy logic extensions, probabilistic reasoning over ontological assertions - **Near-Term Approach**: Safety margins in constraint thresholds (e.g., 10m separation becomes 12m with 2m uncertainty buffer)

Multi-Agent Coordination with Heterogeneous Ontologies - Different UAVs may have different ontological models (varying safety constraints, capabilities) - How do UAVs negotiate conflict resolution when their ontologies differ? - **Research Direction**: Ontology alignment protocols, semantic negotiation for multi-agent systems - **Near-Term Approach**: Single-agent validation first; defer multi-agent scenarios to later development phases

Ontology Evolution During Runtime - Can UAVs learn new spatial relations or refine safety constraints from experience? - How do we validate dynamically updated ontologies without compromising safety? - **Research Direction**: Safe ontology learning, formal verification of runtime modifications - **Near-Term Approach**: Static ontology validated offline; RL learns within fixed constraint framework

---

1.2.5 Safety Constraint Examples (SWRL Rules)

Minimum Horizontal Separation

```
Obstacle(?o)  UAV(?u)  horizontalDistance(?u, ?o, ?d)  swrlb:lessThan(?d, 10.0)
→ SafetyViolation(?u)  ViolationType(?u, "MinimumSeparationBreach")
```

If any obstacle is within 10m horizontal distance, assert safety violation.

Collision Course Detection

```
Aircraft(?a1)  Aircraft(?a2)  trajectory(?a1, ?t1)  trajectory(?a2, ?t2)
 trajectoryIntersection(?t1, ?t2, ?intersection)  timeToIntersection(?a1, ?intersection, ?tti
 timeToIntersection(?a2, ?intersection, ?tti2)  swrlb:lessThan(?tti1, 30.0)  swrlb:lessThan(?
→ CollisionCourse(?a1, ?a2)  RequiresAvoidance(?a1)
```

If two aircraft trajectories intersect and both will reach intersection within 30 seconds, assert collision course.

Geofence Constraint

```
UAV(?u)  position(?u, ?p)  Geofence(?g)  insideBoundary(?p, ?g, false)
→ GeofenceViolation(?u)  EmergencyReturnToGeofence(?u)
```

If UAV position is outside geofence boundary, trigger emergency return protocol.

Flight Phase Constraint (Landing Approach)

```
UAV(?u)  flightPhase(?u, "LandingApproach")  Obstacle(?o)  inFlightPath(?o, ?u, true)
 verticalDistance(?u, ?o, ?vd)  swrlb:lessThan(?vd, 15.0)
→ LandingAbort(?u)  ExecuteGoAround(?u)
```

During landing approach, if obstacle is in flight path with less than 15m vertical clearance, abort landing.

Right-of-Way Rule (FAA Part 107)

```
UAV(?u1)  MannedAircraft(?a)  proximityZone(?u1, ?a, ?zone)  swrlb:lessThan(?zone, 500.0)
→ YieldRightOfWay(?u1, ?a)  AvoidanceManeuver(?u1, "Descend")
```

If manned aircraft is within 500m proximity zone, UAV must yield right-of-way and descend.

1.2.6 Canonical Problem Definition

Problem Statement: Autonomous waypoint navigation with dynamic obstacle avoidance in GPS-denied environment

Mission Scenario: - Start Position: [0, 0, 10] (10m altitude) - Goal Waypoint: [100, 100, 10] (100m north, 100m east) - Dynamic Obstacles: 3 simulated aircraft on crossing trajectories - Static Obstacles: 2 simulated buildings (vertical obstacles) - Environment: Urban simulation in Gazebo (GPS-denied via visual-inertial odometry only)

Safety Constraints (Ontologically Encoded): 1. Minimum 10m horizontal separation from dynamic obstacles 2. Minimum 5m vertical separation from static obstacles 3. Geofence boundary:

±150m from start position (hard constraint) 4. Maximum bank angle: 30° during avoidance maneuvers 5. Maximum velocity: 8 m/s

Success Criteria: - **Primary**: Reach goal waypoint within 3m position tolerance - **Critical**: Zero safety constraint violations during flight - **Secondary**: Complete mission within 90 seconds (baseline straight-line flight: 60s)

Evaluation Metrics: 1. **Safety Performance**: - Safety violation count (zero tolerance for deployment) - Minimum separation distance achieved (margin above constraints) - Time spent in warning zones (5-10m proximity)

2. **Mission Efficiency**:
   - Task completion rate (binary: reached goal or not)
   - Mission time (seconds from start to goal)
   - Path length ratio (actual path / optimal straight-line path)
3. **Control Quality**:
   - Maximum jerk (smoothness of avoidance maneuvers)
   - Control input variance (stability metric)
   - Energy consumption (integral of thrust commands)
4. **Explainability**:
   - Number of ontological rules triggered during flight
   - Semantic trace completeness (can every decision be explained via ontology?)
   - Constraint violation prediction lead time (how early does ontology detect impending violations?)

Baseline Comparisons: - **Unconstrained RL**: Pure DQN/PPO policy without ontological constraints (expect 30%+ collision rate per UAV-ON benchmark) - **Pure Reactive**: Simple distance-threshold avoidance (expect poor task completion due to over-conservative behavior) - **SUMO-Constrained RL**: Our approach (target: >95% task completion, zero safety violations)

Simulation Platform: Gazebo with PX4 SITL, ROS 2 Humble, MAVSDK interface **Target Hardware Validation**: Flyby F-11 with Jetson Orin NX 16GB (50 TOPS)

1.3 Methodology

1.3.1 Ontological Framework for Collision Avoidance

Effective collision avoidance ontologies typically encompass several interconnected knowledge domains:

1.3.1.1 1. Spatial Representation and Reasoning

- **Geometric Relations**: Formal definitions of spatial relationships (above, below, approaching, receding, intersecting trajectories)
- **Distance Metrics**: Ontological concepts for safe separation distances (horizontal, vertical, 3D Euclidean)
- **Velocity Vectors**: Representation of relative motion and predicted future positions
- **Conflict Zones**: Spatial regions defined by regulatory requirements and safety margins

1.3.1.2 2. Aircraft State and Capabilities

- **Flight Phase Ontology**: Takeoff, cruise, landing, loitering, emergency maneuvers

- **Performance Constraints**: Maximum bank angle, climb/descent rates, turn radius, acceleration limits
- **Sensor Coverage**: Field of view, detection range, occlusion reasoning
- **Actuator Response**: Latency between decision and control surface actuation

1.3.1.3 3. Regulatory and Safety Rules

- **Separation Standards**: Minimum safe distances based on aircraft categories and airspace class
- **Right-of-Way Rules**: Formal encoding of aviation regulations (e.g., FAA Part 107, Visual Flight Rules)
- **No-Fly Zones**: Airspace restrictions (airports, military zones, temporary flight restrictions)
- **Emergency Protocols**: Loss of separation, system failures, degraded sensor conditions

1.3.1.4 4. Conflict Detection and Resolution

- **Threat Assessment**: Classification of encounters (imminent collision, near miss, safe passage)
- **Resolution Maneuvers**: Ontological definitions of avoidance actions (climb, descend, turn left/right, speed adjustment, hover)
- **Maneuver Selection Criteria**: Safety priority, fuel efficiency, mission continuity, regulatory compliance
- **Coordination Protocols**: Multi-agent coordination when multiple UAVs must deconflict

1.3.2 Integration with Autonomous Systems

Collision avoidance ontologies connect to autonomous flight systems through several mechanisms:

Perception Pipeline Integration: - Sensor data (vision, radar, ADS-B) feeds semantic object detection - Detected objects classified ontologically (aircraft, obstacle, terrain) - Spatial relationships inferred from sensor observations

Decision-Making Architecture: - Ontology reasoner evaluates current state against safety rules - Conflicts identified through logical inference - Valid resolution actions filtered based on aircraft capabilities and context

Action Selection: - Constrained action space provided to motion planner or RL policy - Ontology ensures only safe, compliant maneuvers are executable - Explainability through semantic rule tracing

1.4 Key Findings

1.4.1 Advantages of Ontology-Based Collision Avoidance

1. **Formal Verifiability**: Safety rules encoded declaratively can be formally verified against specifications
2. **Explainability**: Decisions traceable to specific ontological rules and inferences
3. **Regulatory Alignment**: Direct encoding of aviation regulations ensures compliance by construction
4. **Adaptability**: New rules and contexts can be added without retraining models
5. **Hybrid Integration**: Ontology provides safety envelope within which learning-based optimization occurs

1.4.2 Challenges and Limitations

1. **Computational Complexity**: Real-time reasoning over large ontologies requires efficient reasoners
2. **Incomplete Knowledge**: Open-world scenarios may present situations not anticipated in ontology
3. **Sensor Uncertainty**: Ontological reasoning assumes accurate perception (must handle noisy inputs)
4. **Dynamic Environments**: Fast-moving threats require low-latency inference cycles
5. **Coverage vs. Specificity Trade-off**: Overly general ontologies lack precision; overly specific ontologies lack generalization

1.4.3 Empirical Performance Considerations

While specific quantitative results depend on system implementation, ontology-constrained collision avoidance systems typically demonstrate:

- **Reduced Safety Violations**: Hard constraints prevent rule-breaking behaviors that pure learning approaches exhibit (cf. UAV-ON benchmark's 30%+ collision rate for unconstrained LLM-based navigation)
- **Improved Certification Prospects**: Formal rule encoding facilitates regulatory approval processes
- **Graceful Degradation**: Systems can fall back to conservative ontological rules when learning components fail
- **Interpretable Failures**: When collisions occur despite ontology constraints, failure modes are diagnosable

1.5 Relevance to Flyby F-11 Autonomy

Collision avoidance ontologies are **CRITICALLY RELEVANT** to our SUMO-constrained RL approach:

1.5.1 1. Safety-Critical Constraint Encoding

Our ontology framework must encode: - Minimum safe distances from obstacles (static and dynamic) - Geofence boundaries (hard spatial constraints) - Flight phase-specific collision avoidance behaviors (e.g., more conservative during landing) - Emergency maneuver protocols

1.5.2 2. RL Integration Strategy

The collision avoidance domain demonstrates how ontology can structure RL: - **State Abstraction**: Semantic categories (threat_level: imminent, warning, safe) instead of raw sensor values - **Action Filtering**: Ontology eliminates unsafe actions from RL policy's action space - **Reward Shaping**: Ontological violations contribute negative rewards proportional to safety criticality - **Hierarchical Decomposition**: High-level ontological mission planning, low-level RL trajectory optimization

1.5.3 3. Explainability and Verification

For MCTSSA collaboration and potential defense applications: - Decisions must be auditable (ontology provides semantic trace) - Safety arguments must be demonstrable (formal verification of ontological rules) - Failure analysis must be interpretable (which ontological constraints were active during incident)

1.5.4 4. Domain Vocabulary for ONTOLOGY_FOUNDATION.md

Collision avoidance provides concrete examples for populating our domain vocabulary:

Spatial Relations: - `TooClose(UAV, Obstacle)` - distance below safety threshold - `CollisionCourse(UAV, Aircraft)` - intersecting predicted trajectories - `SafeSeparation(UAV, Object)` - compliance with separation standards

Processes: - `AvoidanceManeuver` - parent class for evasive actions - `EmergencyClimb`, `EmergencyDescent`, `EvasiveTurn` - specific maneuver types - `ConflictResolution` - multi-agent coordination process

States: - `ThreatLevel` - semantic categorization of collision risk - `AvoidanceMode` - system state when actively avoiding collision - `SafetyMarginViolation` - constraint breach indicator

1.5.5 5. Canonical Problem Definition

A natural canonical problem for our system:

Mission Objective: Navigate from waypoint A to waypoint B while avoiding dynamic obstacles (simulated aircraft, moving ground vehicles)

Constraints: - Maintain minimum 10m horizontal separation from obstacles - Maintain minimum 5m vertical separation from terrain - Do not exceed 30-degree bank angle during avoidance maneuvers - Complete mission within battery reserve limits

Evaluation Metrics: - Task completion rate (reached waypoint B) - Safety violation count (separation distance breaches) - Maneuver smoothness (jerk, control input variance) - Mission time efficiency

1.6 Research Directions

1.6.1 Open Questions for Our Implementation

1. **Ontology Granularity**: How detailed should collision avoidance rules be? (trade-off: specificity vs. reasoner performance)
2. **Sensor Fusion Integration**: How do we semantically fuse vision-based obstacle detection with depth sensing?
3. **Uncertainty Handling**: How do we extend ontological spatial relations to handle probabilistic detections?
4. **Real-Time Performance**: Can OWL reasoners operate at 10+ Hz for UAV control loops, or do we need custom inference engines?
5. **Learning vs. Rules Balance**: Which aspects should be learned (optimal avoidance trajectories) vs. hard-coded (minimum separation distances)?

1.6.2 Integration with Existing Work

From our literature review context:

- **Survey of Ontology-Enabled Robot Autonomy** (Paper 03) provides reasoning engine recommendations and dependability frameworks
- **UAV-ON Benchmark** (Paper 05) demonstrates that unconstrained learning approaches have critical safety gaps (30%+ collision rate)
- **Knowledge vs. Data-Driven Survey** (Paper 06) explicitly recommends hybrid approaches for safety-critical domains

Our collision avoidance ontology should: - Build on SUMO upper-level ontology (per Adam Pease guidance) - Integrate with ROS 2 perception pipeline (sensor data → ontological concepts) - Constrain RL policy (provide safe action space boundaries) - Enable runtime verification (monitor compliance during flight)

1.7 Technical Notes

1.7.1 Potential Ontology Implementation Stack

Knowledge Representation: OWL 2 (Web Ontology Language) - Description Logic expressivity: SROIQ(D) for complex spatial/temporal reasoning - Supports property chains for transitive spatial relations - Datatype properties for numerical constraints (distances, velocities)

Reasoning Engine: Options to evaluate - **Pellet**: Complete OWL DL reasoner, good for offline verification - **HermiT**: Hypertableau-based, faster for complex ontologies - **Custom Rule Engine**: RETE-based for real-time performance (may sacrifice completeness for speed)

ROS 2 Integration: - `ontology_interface` package: Bridges sensor topics to ontological concepts - `safety_monitor` node: Runs reasoner at 10 Hz, publishes constraint violations - `constrained_policy` node: RL policy node that queries ontology for valid actions

1.7.2 Performance Considerations for Jetson Orin NX

- **Reasoning Latency**: Target <100ms per reasoning cycle for real-time control
- **Memory Footprint**: 16GB allows loading full ontology + neural policy in unified memory
- **GPU Offload**: Vision-based obstacle detection runs on CUDA cores, reasoner on CPU cores

1.8 References for Further Investigation

Suggested Search Topics: 1. "OWL ontology collision avoidance UAV" 2. "Semantic spatial reasoning autonomous aircraft" 3. "Description logic real-time robotics" 4. "Ontology-constrained motion planning drones" 5. "Formal verification UAV safety constraints"

Related Standards: - RTCA DO-365: Minimum Operational Performance Standards for Detect and Avoid (DAA) - ASTM F3442: Standard Specification for Detect and Avoid Systems - FAA Part 107: Small UAS regulations (operational rules that must be encoded)

1.9 Summary

Collision avoidance ontologies provide a proven framework for encoding safety-critical knowledge in autonomous UAS. By formally representing spatial relationships, regulatory rules, and resolution maneuvers, these ontologies enable verifiable, explainable autonomous decision-making. For our Flyby F-11 project, collision avoidance serves as both a canonical problem domain and a design pattern for integrating SUMO-based ontological constraints with reinforcement learning policies.

The key insight is that ontology defines the safety envelope (what must never happen), while RL optimizes within that envelope (how to accomplish missions efficiently).