Heisenpub

Application of Formal Methods to Verification of UAS Detect-and-Avoid Algorithms Against Separation and Collision Avoidance Requirements

Prakash Shrestha[†],

[†] Tribhuvan University, Department of Mechanical Engineering,
Kirtipur Road, Kathmandu 44618, Nepal

Dipesh Karki[‡],

[‡] Pokhara Institute of Technology, Department of Mechanical Engineering,
Lakeside Road, Pokhara 33700, Nepal

Abstract. Unmanned aircraft systems operating in shared airspace are required to maintain adequate separation from other aircraft and to avoid collisions under a wide range of encounter conditions. Detect-and-avoid algorithms have emerged as a primary means for enabling such operations, but their correctness is often assessed predominantly via simulation, Monte Carlo analysis, or flight testing, which may not fully cover rare but safety-critical scenarios. Formal methods provide a complementary approach by enabling mathematically rigorous reasoning about all behaviors of models of detect-and-avoid algorithms under explicitly characterized assumptions on sensing, guidance, and surrounding traffic. This paper examines the application of model checking, theorem proving, and barrier-certificate-based analysis to verify detect-and-avoid algorithms against separation and collision avoidance requirements representative of regulated airspace integration. A modeling framework is considered in which ownship, intruder dynamics, and detect-and-avoid logic are expressed as a hybrid system subject to bounded disturbances and sensing imperfections, and requirements are encoded as temporal and set-invariance properties over the relative state. The discussion emphasizes the construction of sound abstractions, the treatment of continuous and discrete decision layers, and the explicit accounting of approximation errors and conservatism. A conceptual case study is outlined to illustrate how the framework can be instantiated to obtain formal assurance arguments for representative detect-and-avoid designs without relying solely on empirical coverage. The aim is to clarify conditions under which formal methods can provide meaningful guarantees and highlight modeling choices that materially influence verification outcomes.

1. Introduction

Integration of unmanned aircraft systems into non-segregated airspace introduces complex technical, procedural, and verification challenges that collectively define the emerging landscape of autonomous airspace operations [1]. The central requirement driving this integration is the assurance of safety through robust detect-and-avoid capabilities that

maintain compatibility with established separation minima and midair collision risk objectives. The detect-and-avoid function is responsible for ensuring that unmanned aircraft maintain sufficient distance from other cooperative and non-cooperative aircraft, while simultaneously satisfying mission objectives and regulatory constraints. It operates in an environment characterized by uncertainty in sensing, communication, and maneuver execution. The detect-and-avoid system must therefore reconcile continuous physical dynamics with discrete decision processes, producing advisories that are both dynamically feasible and compliant with the underlying operational rules of air traffic management.

Detect-and-avoid algorithms typically integrate several tightly coupled components, including sensing, state estimation, trajectory prediction, conflict detection, and maneuver generation. Sensing may involve cooperative surveillance technologies such as ADS-B or non-cooperative radar and electro-optical sensors, each introducing distinct types of noise and latency [2]. Estimation filters fuse these data streams to obtain an estimate of the relative state between ownship and nearby traffic. Trajectory prediction then projects this relative motion into the future, under assumptions about both ownship and intruder dynamics. Conflict detection evaluates these predicted trajectories to determine whether a potential violation of well-clear separation could occur, typically within a specified lookahead time horizon. If a potential conflict is detected, the maneuver generation component computes one or more advisories, such as turn, climb, or descent commands, designed to restore or maintain safe separation. Each of these steps is influenced by uncertainty, making the detect-and-avoid algorithm a stochastic and hybrid system in both mathematical and operational terms.

Because detect-and-avoid systems mediate safety-critical decisions, it is essential to analyze their behavior beyond nominal performance [3]. Real-world operations expose the algorithm to rare but hazardous conditions, such as high closure rates, atypical intruder trajectories, or delayed and degraded measurements. In such conditions, even small deviations in estimated positions or velocities can lead to misclassification of conflicts or delayed issuance of avoidance maneuvers. Furthermore, human supervision or autopilot execution introduces additional layers of uncertainty. Human operators may interpret advisories with delay or variability, while autopilot interfaces may limit achievable maneuver aggressiveness. Traditional validation techniques, including flight tests and Monte Carlo simulations, are insufficient for proving correctness across the full range of encounter geometries. These methods, while indispensable, are limited to sampled subsets of possible encounters and cannot guarantee that untested configurations will not lead to unsafe outcomes.

Formal methods provide a complementary approach by constructing mathematical models that represent all possible behaviors of the detect-and-avoid system under specified assumptions [4]. These models enable exhaustive reasoning about whether the safety propertiestypically expressed as separation or collision avoidance invariantshold for all admissible trajectories of the hybrid system. A hybrid system framework naturally accommodates both the continuous motion of aircraft and the discrete logic of the detect-and-avoid algorithm. In this formulation, the system state includes ownship and intruder kinematics,

discrete logic modes corresponding to different algorithmic phases, and representations of advisory issuance and pilot or autopilot compliance. The systems evolution is described by a combination of differential or difference equations for continuous dynamics and guarded transitions for discrete events.

The core verification question is whether, for every possible evolution of this hybrid systemgiven bounded uncertainties in sensing, control response, and intruder behaviorthe separation and collision avoidance requirements are respected. The separation requirement typically ensures that horizontal and vertical distances between aircraft remain above defined minima, while the collision avoidance requirement ensures that even in worst-case conditions, aircraft trajectories do not enter specified near-midair-collision zones [5]. Addressing this verification problem involves several technical challenges. First, the operational requirements must be expressed in a formal language that can be interpreted by verification tools. Second, the continuous dynamics of aircraft motion must be abstracted into structures that can be handled by model checking or reachability analysis, which typically require finite or finitely branching representations. Third, approximation errors introduced by discretization or abstraction must be bounded so that the resulting verification results are sound, meaning that any guarantee of safety in the abstract model implies safety in the original continuous system.

Temporal logic, invariant set formulations, and game-theoretic methods form the mathematical foundation for expressing and analyzing these properties. Temporal logic allows requirements to be expressed as statements about sequences of system states over time. For example, a safety property might assert that the relative distance between aircraft always remains greater than a specified threshold, which in linear temporal logic is written as the invariant operator applied to the safe condition [6]. More complex properties can express requirements on timing and ordering, such as ensuring that if a conflict is detected, an alert is issued within a bounded number of time steps, or that a resolution maneuver begins before a certain lookahead time elapses. These temporal constraints are crucial because delays in detection or advisory generation directly affect safety margins.

Invariant set formulations provide a geometric interpretation of safety. The relative state of the ownship and intruder can be represented as a point in a multi-dimensional space defined by their positions, velocities, and headings. Within this space, the unsafe region corresponds to combinations of states that violate separation or collision avoidance requirements. The safe set is the complement of this region, and the detect-and-avoid algorithms task is to ensure that the system state remains within this safe set for all time [7]. Formal verification then reduces to proving that the reachable set of states, given all possible inputs and disturbances, never intersects the unsafe region. Because exact computation of reachable sets is generally infeasible for nonlinear systems, conservative overapproximations are used. These overapproximations must be tight enough to avoid false alarms but conservative enough to guarantee soundness.

Game-theoretic characterizations are particularly relevant because detect-and-avoid interactions inherently involve two agentsownship and intruderwhose actions influence each

others outcomes. From a formal perspective, this can be represented as a differential game in which ownship seeks to maintain separation while the intruder may act in ways that challenge it, subject to bounded capabilities. The verification question then becomes whether a winning strategy exists for ownship that ensures avoidance of unsafe states regardless of intruder actions within the allowed bounds [8]. The solution of such games yields safe sets and corresponding control strategies that can be compared against or used to synthesize detect-and-avoid algorithms. Game-theoretic reasoning thus provides both a theoretical limit on achievable safety and a benchmark for assessing whether a given detect-and-avoid algorithm realizes or approximates an optimal avoidance strategy.

Implementing these formal techniques requires systematic abstraction of continuous dynamics into discrete structures suitable for automated reasoning. The relative motion between aircraft can be discretized in position, velocity, and heading space, with transitions approximating the effect of bounded control and disturbance inputs. The detect-and-avoid logic can be modeled as a finite automaton with states representing algorithmic modes such as monitoring, alerting, and resolution, and transitions triggered by threshold crossings or timing conditions. The combined hybrid automaton captures both physical and logical evolution. Model checking tools can then exhaustively explore all reachable sequences of states to verify whether the safety properties expressed in temporal logic hold. [9]

However, abstraction introduces approximation error. If the discretization is too coarse, it may overlook important transitions or incorrectly merge states that differ in safety-critical ways. Conversely, excessively fine discretization can lead to state-space explosion, rendering verification computationally intractable. Therefore, bounding and managing abstraction error is essential. Techniques such as simulation relations, bisimulation approximations, and reachability refinements are employed to ensure that safety proofs derived from abstract models remain valid for the continuous system. Additionally, margins are applied to safety thresholds to compensate for these errors, ensuring that even if the abstract model slightly underestimates the set of reachable states, the true system remains safe. [10]

An additional dimension of complexity arises from the stochastic nature of sensing and estimation. Sensor noise, communication delays, and intermittent data losses produce uncertainty in the estimated relative state. This uncertainty must be explicitly modeled, often by defining bounded estimation error sets or stochastic disturbance models. The detect-and-avoid algorithms decisions are then verified not on the true state but on the estimated state, with formal guarantees required to hold under all estimation errors consistent with the defined bounds. Such robust verification ensures that safety is maintained even when measurements are degraded.

The formal modeling process also incorporates compliance and human-in-the-loop considerations. When advisories are issued, they are executed either by an autopilot or by a remote pilot [11]. Execution dynamics introduce latency and variability in the response. These effects are represented as additional hybrid transitions or delays within the model,

ensuring that the verification accounts for the total system behavior rather than the detect-and-avoid logic in isolation. This holistic approach prevents overestimation of safety margins that might occur if idealized execution were assumed.

Ultimately, the integration of formal methods into detect-and-avoid analysis reframes the safety question from one of statistical adequacy to one of mathematical correctness under explicit assumptions. Instead of relying on empirical evidence that a large but finite number of simulations show no conflicts, formal verification seeks to demonstrate that no conflict is possible for any trajectory consistent with the model. The scope of the guarantee is therefore tied directly to the fidelity and completeness of the model [12]. While formal methods cannot eliminate all uncertaintysince models necessarily simplify realitythey provide a transparent and repeatable framework for assessing algorithmic safety and identifying conditions under which guarantees hold or fail.

detect-and-avoid verification through formal methods is grounded in the hybrid systems paradigm, leveraging temporal logic, invariants, and game theory to provide exhaustive analysis of safety properties. The process involves constructing mathematical abstractions that capture all relevant interactions among sensing, estimation, dynamics, and decision logic, while bounding approximation errors to maintain soundness. This perspective complements traditional simulation-based approaches by enabling systematic reasoning about worst-case behaviors, adversarial trajectories, and timing effects that are difficult to capture empirically. Through this formalization, detect-and-avoid verification moves from empirical validation toward provable assurance, supporting the safe integration of unmanned aircraft systems into complex, shared airspace environments.

The application of formal methods to this domain must reconcile several challenges [13]. The continuous dynamics of aircraft motion, including turn-rate-limited maneuvers, climb and descent constraints, and wind disturbances, lead to high-dimensional and nonlinear models that are not directly tractable for exhaustive verification. Detect-and-avoid logic often embeds internal timers, state-dependent thresholds, and mode switching, which must be faithfully captured as discrete transitions to avoid spurious behavior during abstraction. Sensing and communication delays can produce subtle timing interactions that affect whether maneuvers remain feasible when alerts are issued. Moreover, the environment is open, with uncertain numbers and types of intruders, varying equipage levels, and differing rules of right-of-way, which motivates modeling choices that approximate these factors without asserting complete coverage of all aviation procedures.

This paper considers a verification perspective in which detect-and-avoid algorithms are modeled as components within a hybrid automaton, and formal methods tools are used to establish that all reachable trajectories under specified assumptions avoid defined loss-of-well-clear and near midair collision conditions. The presentation focuses on the construction of mathematically precise requirements, development of dynamic and stochastic encounter models that are compatible with formal reasoning, establishment of abstraction and discretization techniques for the hybrid system, and deployment of model checking, reachability analysis, and barrier certificate approaches. The objective is not to claim full

certification sufficiency but to articulate structures within which formal reasoning can support systematic assurance of detect-and-avoid functions and reveal sensitivities to modeling assumptions. [14]

Table 1. Representative Parameters for Ownship and Intruder Dynamics Models

Parameter	Symbol	Typical Value	Units	Description
Ownship true airspeed	V_o	55	m/s	Nominal constant airspeed of unmanned aircraft
Intruder true airspeed	V_i	60	m/s	Nominal airspeed of intruder aircraft
Turn rate limit	$\omega_{\max}$	0.05	rad/s	Maximum allowable heading rate
Climb rate limit	$\dot{h}_{\max}$	5	m/s	Vertical rate constraint for both aircraft

Table 2. Abstraction Granularity for Hybrid-State Discretization

State Dimension	Range	Resolution	Number of Intervals	Approximation Type
Relative range	[0, 2000] m	50 m	40	Uniform
Relative bearing	[0, 2π] rad	$\pi/16$	32	Angular partition
Relative altitude	[-400, 400] m	25 m	32	Symmetric vertical bins
Relative speed	[0, 80] m/s	5 m/s	16	Linear

Table 3. Key Safety and Separation Thresholds Used in Verification

Requirement	Symbol	Threshold Value	Units	Application Context
Horizontal well-clear boundary	d_{wc}	500	m	Minimum lateral separation
Vertical well-clear boundary	h_{wc}	120	m	Minimum vertical separation
Horizontal NMAC limit	d_{nmac}	150	m	Near midair collision bound
Vertical NMAC limit	h_{nmac}	30	m	Collision-critical vertical separation

Table 4. Comparison of Formal Verification Approaches

Method	Continuous Dynamics Support	Discrete Logic Modeling	Com
Model Checking	Limited (abstracted)	Exact	High (st
Reachability Analysis	Full	Partial (hybrid transitions)	Mo
Barrier Certificates	Continuous analytic	Approximate discrete transitions	
Simulation-based Falsification	Approximate	Exact logic execution	

Table 5. Environmental and Sensor Modeling Parameters

Component	Symbol	Nominal Value	Units	Description
Sensor update period	T_s	1	s	Sampling interval for surveillance updates
Measurement noise (position)	σ_p	5	m	One-sigma position uncertainty
Measurement noise (velocity)	σ_v	0.5	m/s	One-sigma velocity uncertainty
Communication latency	τ_c	0.4	s	Delay between advisory issue and executio

Table 6. State-Space Partitioning for Finite-State Abstraction

Dimension	Symbol	Interval Count	Step Size	Boundaries
Horizontal distance	d	40	50 m	[0, 2000]
Vertical separation	h	32	25 m	[-400, 400]
Relative bearing	θ	32	$\pi/16$ rad	$[0, 2\pi]$
Relative speed	v_r	16	5 m/s	[0, 80]

Table 7. Representative Temporal Logic Specifications for Verification

Specification ID	Property Type	Expression Form	Time Horizon	Objective
P1	Safety Invariant	$\square\neg S_{\text{nmac}}$	Infinite	Prevent near midair collision
P2	Well-Clear Maintenance	$\square\neg S_{\text{wc}}$	120 s	Maintain well-clear boundary
P3	Timely Alert	$\square(\mathsf{conflict} \rightarrow \Diamond_{[0,5]}\mathsf{alert})$	5 s	Alert delay constraint
P4	Resolution Recovery	$\square(\mathsf{alert} \rightarrow \Diamond_{[0,15]}\mathsf{safe})$	15 s	Guarantee post-alert safety

Table 8. Barrier Function Coefficient Examples for Polynomial Certificates

Coefficient Index	Symbol	Value Range	Dimension	Interpretation
1	a_1	[0.1, 0.5]	m^{-2}	Horizontal distance weighting
2	a_2	[0.05, 0.2]	s/m	Velocity coupling term
3	a_3	[0.01, 0.08]	m^{-1}	Vertical contribution
4	a_4	[0.002, 0.005]	s^{-2}	Time-to-closest-approach term

2. Operational Context and Detect-and-Avoid Requirements

Detect-and-avoid functions for unmanned aircraft are typically defined with respect to an operational context that includes the category of airspace, performance characteristics of

Table 9. Comparative Verification Results under Varying Intruder Turn Rates

Maximum Turn Rate	Verified Safety (Yes/No)	Safe Initial Volume (%)	Analysis Time (s)	Method Used
0.03 rad/s	Yes	96.2%	430	Reachability
0.05 rad/s	Yes	89.4%	515	Barrier Certificate
0.07 rad/s	No	72.8%	498	Model Checking
0.10 rad/s	No	55.3%	465	Model Checking

Table 10. Summary of Assumptions and Their Impact on Formal Guarantee Strength

Assumption	Modeled As	Relaxation Effect	Verification Impact	Guarantee Type
Perfect sensing	Deterministic	Minor	Reduced computation time	Strong (absolute)
Bounded sensor noise	Interval uncertainty	Moderate	Increased abstraction size	Conservative
Unbounded intruder turns	Adversarial input	Major	Potential violation found	Weak or conditional
Bounded latency	Temporal offset	Moderate	Delay constraints tighten	Conditional robust

the unmanned aircraft, expected intruder behavior, and allowable responsibilities assigned to automation versus human operators. For formal verification, this contextual information must be distilled into assumptions on encounter geometries, closure rates, altitude and heading change capabilities, and the feasible response envelope for resolution maneuvers. A detect-and-avoid algorithm is then evaluated with respect to a set of requirements that specify safe separation criteria, acceptable probabilities or frequencies of loss-of-well-clear conditions within the assumed environment, and constraints on generated maneuvers such as adherence to flight envelope limits and coordination rules.

A common operational abstraction considers one ownship, indexed by o, and one or more intruders, indexed by i. The position of each aircraft is described in a Cartesian frame aligned with a local tangent plane, with ownship position $p_o(t)$ and intruder position $p_i(t)$. The relative position is $r_i(t) = p_i(t) - p_o(t)$, and associated relative velocity $v_i(t)$. A generic separation requirement can be expressed in terms of a protected zone around ownship, such as a cylinder composed of horizontal and vertical components. Loss of well clear occurs if

both the horizontal and vertical components of separation fall below prescribed thresholds [15]. To support formal verification, the well-clear boundary is represented as a subset S_{wc} of the relative state space, and safety is expressed as the requirement that trajectories remain outside a more restrictive set S_{nmac} associated with near midair collision conditions.

Operational requirements encompass both preventive and corrective aspects. A preventive requirement may specify that, for encounters starting outside a given horizon region and under compliant intruder behavior, the detect-and-avoid logic shall provide alerts and corresponding guidance such that separation is maintained. A corrective requirement may acknowledge that certain initial conditions are already incompatible with strict prevention, in which case the objective becomes avoiding collision and minimizing severity within feasible maneuver limits. For formal modeling, such distinctions are represented by partitioning the initial state space according to whether the detect-and-avoid logic is required to guarantee well-clear preservation or only collision avoidance, and by encoding maneuvers as constraints on control inputs available to the ownship.

To capture algorithmic behavior, the detect-and-avoid system is represented by internal modes associated with surveillance, conflict detection, alerting, and guidance. Each mode is governed by guards and invariants that depend on the estimated relative state, time-to-closest-approach metrics, and advisory histories. For instance, a conflict detection requirement can be formulated so that, if an encounter is predicted to cross the well-clear boundary within a specified lookahead time under assumed intruder dynamics, the system shall enter an alerting mode within a bounded response time [16]. Subsequent guidance modes generate maneuvers that are required to be both operationally compatible and dynamically feasible. These elements are embedded into the hybrid system model used for verification, making explicit the timing and logical dependencies that influence whether separation and collision avoidance requirements are satisfied.

An additional consideration arises from the presence of multiple intruders and potential interactions among resolution maneuvers. In the formal framework, this is treated by considering either an aggregated worst-case intruder model that captures the most critical relative trajectories, or by modeling a finite but potentially large set of intruder aircraft with pairwise interaction constraints. The resulting state space grows combinatorially, so abstractions that conservatively project multi-intruder behavior into equivalent worst-case single-intruder scenarios are often introduced. Such abstractions must be designed so that if safety is verified under the abstraction, it holds for the original multi-intruder system [17] [18]. This requirement shapes the mathematical characterization of detect-and-avoid requirements, emphasizing monotonicity and compositional arguments that permit tractable yet sound verification.

3. Dynamic and Stochastic Modeling of UAS Encounters

The basis for formal verification of detect-and-avoid algorithms is a dynamic model that captures the evolution of ownship and intruder states under both commanded maneuvers

and disturbances. A commonly used representation for ownship dynamics in a local horizontal plane is a kinematic model with bounded turn rate and speed variations. Let $x_o(t)$ denote the ownship state, including position and velocity components. A discrete-time approximation over step Δt can be written as

$$x_o(k+1) = f_o\big(x_o(k), u_o(k)\big),$$

where $u_o(k)$ represents lateral and vertical guidance inputs selected either by the detect-and-avoid algorithm or by an autopilot subject to its commands. Intruder dynamics are modeled similarly as

$$x_i(k+1) = f_i\big(x_i(k), u_i(k), w_i(k)\big),$$

with $w_i(k)$ representing exogenous variations or uncertainties. The relative dynamics for a given intruder are then [19]

$$x_r(k+1) = F\big(x_r(k), u_o(k), u_i(k), w_i(k)\big),$$

where $x_r(k)$ encodes relative position and velocity. This formulation emphasizes that detect-and-avoid decisions effectively shape $u_o(k)$ in response to evolving estimates of $x_r(k)$ and assumptions on intruder controls.

To describe a range of realistic encounters suitable for verification, the intruder input $u_i(k)$ and disturbance $w_i(k)$ are not fixed but constrained within sets that encode admissible behaviors. For example, one may specify bounds on heading change rates, climb gradients, and speed variations, capturing both nominal and maneuvering traffic. In a deterministic worst-case framework, $u_i(k)$ is treated as an adversarial input seeking to challenge separation, subject only to these bounds. In a stochastic framework, encounter models are defined as Markov processes over the intruder state space, with transition kernels that reflect traffic distributions and maneuver likelihoods. Formal verification in the latter setting can involve probabilistic temporal logics, but even in purely probabilistic analyses, it is often operationally relevant to also consider worst-case trajectories that lie in the support of the stochastic model.

Sensing and estimation enter the dynamic model through measurement processes and filters that infer the relative state. Let $z(k)$ denote the surveillance measurement, which may include ownship navigation data and intruder tracks derived from cooperative transponders or noncooperative sensors. The estimator is represented as

$$\hat{x}_r(k+1) = G\big(\hat{x}_r(k), z(k+1)\big),$$

and detect-and-avoid logic bases its decisions on $\hat{x}_r(k)$. Estimation errors introduce additional uncertainty that must be accounted for when guaranteeing satisfaction of separation requirements [20]. A conservative approach is to characterize the estimation error $e(k) = x_r(k) - \hat{x}_r(k)$ as belonging to a bounded set for all times of interest, leading to robust verification conditions that quantify margins necessary in alerting thresholds and maneuver guidance to compensate for state uncertainty.

The combination of continuous dynamics, bounded but potentially adversarial disturbances, estimation errors, and logical switching of detect-and-avoid modes leads to a hybrid

system representation. This system can be viewed as

$$H = (X, Q, U, W, T),$$

where X is the continuous state space, Q is the finite set of modes capturing algorithm states, U is the set of ownship control inputs admissible in each mode, W is the set of environment inputs, and T defines continuous and discrete transitions. Reachable states of H include all possible values of (x_r, q) obtained under all admissible inputs and disturbances. Formal verification aims to show that the reachable set does not intersect undesirable regions, such as near midair collision sets, under specified initial conditions and assumptions. The fidelity of this model is critical: if the model omits relevant dynamics or underestimates disturbances, the verification results may not reflect operational behavior, whereas overly conservative modeling can render verification inconclusive by admitting trajectories that are not physically realizable. [21]

To incorporate stochastic encounter models into a framework compatible with formal methods, one can represent the intruder as a stochastic process on a finite or countable state abstraction, with transitions constrained by performance envelopes and operational rules. The detect-and-avoid algorithm is then analyzed either under probabilistic temporal logic, where properties concern probabilities of unsafe events, or through an overapproximation that treats all states in the support of the process as possible. Both approaches depend on defining encounter sets that meaningfully approximate real traffic situations while preserving mathematical tractability. The modeling choices made here directly influence the interpretation of verification outcomes, as they determine whether guarantees are absolute within stated bounds or conditioned on probabilistic assumptions about intruder behavior.

4. Formal Specification of Separation and Collision Avoidance Requirements

Formal verification requires that separation and collision avoidance requirements be expressed as precise properties over the trajectories of the hybrid system representing the detect-and-avoid scenario. A basic safety requirement demands that the relative state remain outside a near midair collision set S_{nmac}. Let $d(t)$ denote the Euclidean norm of horizontal relative position and $h(t)$ denote vertical separation. One can define [22]

$$S_{\text{nmac}} = \{x_r : d < d_{\text{n}},\ |h| < h_{\text{n}}\},$$

with constants d_{n} and h_{n} representing collision thresholds. The collision avoidance requirement is captured by the temporal property that along all trajectories,

$$\square \neg S_{\text{nmac}},$$

meaning the system state never enters S_{nmac}. A more conservative separation requirement uses a larger protected zone S_{wc}, defined analogously with thresholds d_{w} and h_{w}. The corresponding well-clear preservation property is

$$\square \neg S_{\text{wc}},$$

for all trajectories that start outside S_{wc} and satisfy specified environment assumptions.

Detect-and-avoid algorithms often have finite detection and response delays. To incorporate these, temporal operators with time bounds can be used. For example, suppose that if a conflict is predicted within a lookahead interval, an alert must be issued within a maximum response delay and a resolution maneuver initiated. Let $\mathsf{conflict}$ be a predicate indicating that, based on $\hat{x}_r(k)$, there exists a trajectory within the intruder admissible set that would enter S_{wc} within a finite horizon. Let alert capture that the detect-and-avoid system is in an alerting or resolution mode. A requirement can then be written informally as: whenever $\mathsf{conflict}$ holds, alert must be activated within a specified number of steps. In temporal logic notation, with $\Diamond_{[0,N]}$ denoting the eventually operator bounded by N,

$$\square\big(\mathsf{conflict} \rightarrow \Diamond_{[0,N]}\mathsf{alert}\big).$$

Separation and collision avoidance are then considered jointly as properties constraining both when alerts are issued and whether the resulting maneuvers successfully avoid entry into S_{wc} or S_{nmac}.

Hybrid invariants and barrier functions provide alternative formulations [23]. Define a continuous function $B : X \rightarrow \mathbb{R}$ such that $B(x_r) \geq 0$ for all states considered safe and $B(x_r) < 0$ for all states in S_{nmac}. If the dynamics and control policies guarantee that along trajectories starting with $B(x_r(0)) \geq 0$, the function $B(x_r(t))$ never becomes negative, then collision avoidance is assured. For discrete-time dynamics, one can require that for all admissible ownship controls and all environment disturbances,

$$B\big(x_r(k+1)\big) - B\big(x_r(k)\big) \geq -\alpha B\big(x_r(k)\big),$$

for some nonnegative constant α, whenever $B(x_r(k)) \geq 0$. This inequality ensures that trajectories do not cross the barrier into unsafe states. Detect-and-avoid logic is encoded in how $u_o(k)$ is selected to maintain the barrier condition in the face of intruder behavior and disturbances constrained by the environment model.

The coexistence of preventive and corrective requirements can also be captured by partitioning the initial condition space into regions with distinct formal obligations. Let X_0^{wc} denote initial states from which well-clear preservation is required and feasible, and X_0^{ca} denote states from which only collision avoidance is required under physical constraints. Formal conditions for membership in these sets can be expressed using backward reachability. For instance, define a set of states from which there exists a detect-and-avoid policy that keeps trajectories outside S_{wc} under all admissible intruder actions, and let this set be K_{wc}. Then one can require that operationally allowed initial states with no preexisting conflicts belong to K_{wc}. States outside K_{wc} but still outside S_{nmac} may be assigned collision avoidance obligations. These constructions provide a bridge between intuitive requirements and mathematically checkable properties.

When considering probabilistic requirements, one may specify that the probability of entering S_{nmac} under a stochastic encounter model remains below a given threshold. This

can be expressed in probabilistic temporal logic as a property such as [24]

$$P_{\leq\gamma}\left[\Diamond S_{\text{nmac}}\right],$$

with γ representing an acceptable bound under the modeling assumptions. However, to maintain compatibility with deterministic worst-case guarantees desired for certification-oriented arguments, the analysis can emphasize that such probabilistic properties are evaluated with respect to explicitly defined traffic models and are complemented by deterministic invariance results obtained under bounded disturbance assumptions. The combination of temporal logic specifications, invariants, and barrier-based conditions thus forms a vocabulary for rigorously encoding detect-and-avoid requirements in a form suitable for formal methods tools.

5. Formal Verification Architecture and Algorithms for Detect-and-Avoid Logic

The hybrid nature of detect-and-avoid systems motivates a verification architecture that separates continuous dynamics from discrete decision logic while preserving soundness. A structured approach begins by constructing a finite-state or symbolic abstraction of the continuous state space that captures all behaviors relevant for satisfaction or violation of the requirements. For relative motion in a plane, one may discretize distance and bearing into regions and approximate velocity ranges into finite sets. Let the abstract state be denoted by s, and define a transition system [25]

$$\mathcal{T} = (S, S_0, \rightarrow, L),$$

where S is a finite set of abstract states, S_0 represents possible initial states, $\rightarrow$ is a transition relation derived from the continuous dynamics and control limits, and L is a labeling function that maps states to atomic propositions indicating, for instance, whether they are safe, in conflict, or violating separation. The abstraction is constructed so that for any concrete state consistent with an abstract state, its successors under admissible inputs are contained within the union of successors of that abstract state in $\mathcal{T}$. This overapproximation ensures that if a safety property holds on the abstract model, it also holds on the original hybrid system.

The detect-and-avoid algorithm is incorporated into this framework as a decision function that, based on the current labeled abstract state and internal logic mode, selects a set of permissible maneuvers. The closed-loop abstract system becomes a product of the abstraction and the algorithm logic. Safety properties expressed in temporal logic can then be checked using standard model checking procedures on the finite-state system. If the model checker reports that the property holds, the overapproximation guarantees that no violations exist in the concrete system under the modeling assumptions. If a counterexample is found, it is necessary to determine whether it corresponds to a realizable trajectory; if not, the abstraction is refined, for example by partitioning states or tightening transition bounds, and the verification is repeated. [26]

For higher fidelity dynamics or when the state dimension precludes explicit finite abstraction, set-based reachability analysis can be employed. In this approach, sets of continuous states are propagated over time under the combined effect of dynamics, control decisions, and disturbances. For linear or suitably constrained nonlinear dynamics, convex overapproximations such as polytopes or zonotopes can be used. Let X_k denote an overapproximation of the set of possible relative states at step k. Given a model of detect-and-avoid decisions, one computes

$$X_{k+1} \supseteq \Phi\big(X_k\big),$$

where Φ represents the successor operation under all admissible inputs and disturbances. Safety is established if X_k remains outside S_{nmac} and, when applicable, outside S_{wc} for all k within the horizon of interest. For unbounded time properties, fixed-point computations are used to approximate the maximal invariant subsets that avoid unsafe regions. The detect-and-avoid logic influences Φ because the set of applied maneuvers depends on alerting thresholds and resolution rules, which are encoded as constraints on control actions consistent with each region in the state space. [27]

Barrier certificate methods provide another mechanism suited to continuous and hybrid systems. Given candidate barrier functions, possibly obtained through optimization or systematic search, one verifies inequalities that guarantee safety. For polynomial dynamics, sum-of-squares techniques can be applied to establish that a function $B(x_r)$ satisfies the necessary conditions across the state space. In the detect-and-avoid setting, the barrier can be shaped to align with well-clear boundaries, with detect-and-avoid advisories interpreted as strategies that maintain the state in the region where the barrier conditions hold. When discrete logic is present, separate barrier functions can be defined for each mode, together with conditions on transitions ensuring that barrier nonnegativity is preserved. The resulting certificates yield concise proofs of safety that can be mechanically checked.

An important aspect of the verification architecture is the explicit handling of numerical and modeling approximations [28]. Discretization of state and time, truncation of encounter spaces, and linearization of dynamics introduce discrepancies between the analysis model and the underlying physical system. To maintain soundness, margins are introduced in the form of tightened thresholds for safety sets and enlarged disturbance bounds. For instance, if numerical analysis can only guarantee that the true distance is at least a certain value minus an error bound, then the verified separation minima are adjusted accordingly. This leads to conditions where satisfaction of the formal property implies that the original requirements are met with a quantifiable margin. Conversely, if analysis reveals that no barrier or reachable set invariant exists under conservative assumptions, this indicates sensitivity of the detect-and-avoid logic to those assumptions and suggests directions for algorithmic adjustments.

Scalability remains a challenge, especially with multiple intruders and complex detection logic [29]. Compositional techniques address this by decomposing verification into smaller problems, such as analyzing pairwise interactions and then using assumptions about mutual independence or structural monotonicity to reason about the combined system. Another

direction uses game-theoretic formulations in which ownship and intruder are modeled as players in a differential game. Collision avoidance requirements correspond to winning conditions for ownship under constrained strategies. In simplified settings, one can derive analytic characterizations of winning sets of initial states from which ownship has strategies to avoid unsafe regions. Detect-and-avoid logic is then evaluated on whether its advisory structure implements or approximates such strategies within the available maneuver envelope.

6. Case Study Style Analysis and Sensitivity to Modeling Assumptions

To illustrate the implications of the formal methods framework, consider a notional detect-and-avoid algorithm that issues horizontal maneuver advisories based on predicted time to violation of a cylindrical well-clear volume. The algorithm monitors an estimate of relative position and velocity and computes projected miss distance under straight-line extrapolation of relative motion [30]. If predicted separation falls below prescribed thresholds within a given lookahead time, the system issues a turn advisory intended to increase horizontal miss distance beyond the protected boundary. Verification focuses on whether, under bounded rates of intruder maneuver and specified ownship performance, the advisories are sufficient to prevent entry into the near midair collision set.

The hybrid system model for this scenario includes modes representing no-conflict monitoring, conflict detection, and resolution. In the no-conflict mode, ownship follows its nominal trajectory subject to commanded waypoints. In the conflict detection mode, relative state estimates trigger a transition to resolution when projected miss distance falls below a threshold. In resolution mode, ownship executes a coordinated turn with bounded rate until the predicted separation satisfies a termination condition [31]. Using a finite abstraction, the relative distance is partitioned into radial bands, and the relative bearing and heading difference are partitioned into sectors. For each abstract state, one determines whether the algorithm would remain in monitoring mode, escalate to resolution, or exit resolution. Transitions among abstract states are computed by propagating representative continuous states under ownship and intruder maneuvers within their admissible sets.

Model checking is applied on this finite transition system with the safety property that no path reaches states labeled as near midair collision. If the property holds under a set of assumptions on maximum intruder turn rates and speed changes, then, by construction of the abstraction, the original algorithm is safe under these assumptions, subject to approximation margins. If a counterexample path is returned, its realizability is examined in the continuous model. Realizable counterexamples expose sequences of intruder maneuvers and detect-and-avoid decisions that lead to loss of safety, often revealing corner cases such as delayed alerts, oscillatory advisories, or interactions between vertical and horizontal maneuvers [32]. Unrealizable counterexamples motivate refinement of the abstraction by narrowing state partitions or recalculating transition bounds.

Sensitivity analysis arises naturally by varying the parameters used in the verification model. For instance, tightening the assumed bound on intruder turn rate may enlarge

the set of initial conditions from which safety can be guaranteed, while loosening it can produce violations. Similarly, increasing estimation error bounds requires earlier alerts and more conservative advisories to maintain safety; verification may indicate that, beyond certain error magnitudes, no advisory strategy of the considered type suffices to satisfy the requirements. Analyzing these trends helps identify which aspects of the detect-and-avoid system, such as sensor accuracy, communication latency, or maneuver authority, most strongly influence the possibility of obtaining formal guarantees.

Barrier certificate methods can be instantiated for the same algorithm by searching for functions that separate safe initial states from unsafe regions while being compatible with the closed-loop dynamics under advisory rules [33]. For example, one may propose a barrier that depends on projected time to closest point of approach and relative geometry, and then use optimization-based methods to verify that this barrier is nondecreasing along all permissible trajectories under the detect-and-avoid advisories. If successful, this yields a compact certificate that safety holds for a specified class of encounters. If no barrier of the chosen form is found, it does not directly imply unsafety but suggests that either the algorithm or the certificate structure may require adaptation. Thus, the combination of reachability-based and certificate-based analyses provides complementary views on algorithm robustness.

When multiple intruders are introduced, the verification model becomes more complex, but similar principles apply. One approach considers pairwise conflicts independently and assumes that the detect-and-avoid algorithm resolves each without interference [34]. If this assumption is not structurally valid, more expressive models consider the coupled effect of maneuvers. Abstracting such interactions can again involve conservative overapproximations that treat the worst-case alignment of intruder trajectories. The analysis may show that specific rule sets or coordination protocols are needed to avoid conflicting advisories in dense traffic conditions. By encoding these coordination rules formally and examining their effect on reachable sets, the verification process highlights requirements on higher-level traffic management needed to support detect-and-avoid safety.

Overall, the case study style examination underscores that formal verification outcomes depend strongly on modeling assumptions, chosen abstraction granularity, and verification technique. Under appropriately stated bounds on dynamics and sensing, detect-and-avoid algorithms can be shown to satisfy separation and collision avoidance properties within those bounds. Relaxing assumptions or introducing additional uncertainties may lead to the absence of conclusive guarantees, indicating conditions where algorithm redesign, enhanced sensing, or procedural mitigations are necessary to achieve desired safety margins [35]. The technical structure provided by formal methods allows such conclusions to be articulated in a mathematically explicit manner.

7. Conclusion

Application of formal methods to the verification of unmanned aircraft detect-and-avoid algorithms enables a structured and exhaustive approach to reasoning about safety properties under explicitly defined modeling assumptions. By representing ownship and intruder dynamics, sensing and estimation mechanisms, and detect-and-avoid decision logic within a unified hybrid system framework, these methods enable rigorous examination of all admissible system evolutions. Unlike simulation-based approaches that depend on enumerated encounter sets, formal verification explores the entire reachable space of behaviors consistent with defined assumptions, thus encompassing scenarios that may be overlooked by empirical sampling. The separation and collision avoidance requirements, when encoded as temporal logic specifications, invariant set constraints, and barrier conditions, can be verified mathematically to hold across all modeled trajectories. This capability offers a systematic means of establishing algorithmic correctness, quantifying safety margins, and identifying failure regions under specified uncertainties. [36]

Within this analytical framework, several methodological approaches serve complementary purposes. Finite-state abstractions enable the reduction of continuous flight dynamics to discrete transition systems suitable for model checking, allowing automated exploration of all reachable configurations. Set-based reachability computations propagate overapproximations of the hybrid state space through time, identifying invariant regions that guarantee separation or revealing conditions under which the system may approach unsafe boundaries. Barrier certificates, by contrast, provide functional witnesses that mathematically separate safe and unsafe regions, ensuring that trajectories remain within allowable domains if the inequalities governing the barrier function hold. Each of these methods involves approximations, discretizations, and conservatism that must be managed carefully to preserve soundness. Margins introduced to account for these approximations ensure that conclusions drawn from abstract models remain valid for the true continuous dynamics. This disciplined handling of uncertainty and error bounds differentiates formal verification from heuristic or statistical validation methods, where guarantees cannot extend beyond tested scenarios. [37]

Verification outcomes are inherently sensitive to assumptions embedded in the system and environment models. The dynamic behavior of intruders, the geometric configurations of encounters, and the response capabilities of ownship all influence the reachability structure of the system. For example, tighter constraints on intruder maneuverability simplify the verification problem by reducing the range of admissible relative trajectories, potentially enabling formal proofs of safety. Conversely, when intruder dynamics are weakly constrained or include abrupt changes in velocity or heading, the reachable space expands, making it more difficult to demonstrate invariant separation. Similarly, assumptions regarding sensor performance, measurement latency, and communication reliability significantly affect the guarantees obtainable through formal methods. A detect-and-avoid algorithm may be provably safe under perfect sensing but fail to meet requirements when

realistic latency or degradation models are introduced [38]. Verification results must therefore be interpreted in the context of these underlying assumptions, recognizing that safety proofs are conditional on the fidelity and completeness of the modeled environment.

The influence of ownship maneuver authority further constrains achievable guarantees. Algorithms that generate advisories assuming idealized aircraft response may fail to satisfy formal safety properties when control actuation limits or rate constraints are applied. Formal analysis can reveal these discrepancies by demonstrating that, under the true bounded control set, reachable states include conditions leading to violation of well-clear separation or near midair collision zones. Such findings inform not only algorithmic refinement but also operational policies, indicating, for instance, that certain speed or climb-rate capabilities are prerequisites for compliance with defined safety margins. Thus, formal verification serves as a diagnostic tool for identifying the interplay between algorithmic design, vehicle performance, and operational requirements. [39]

In cases where uncertainties and disturbances exceed manageable bounds, formal analysis may conclude that existing detect-and-avoid logic cannot guarantee compliance with safety requirements. These results are not failures of the verification process but valuable indicators of model or design insufficiency. They provide a mathematically grounded basis for revising algorithms, tightening operational envelopes, or improving sensor fidelity. In this sense, formal methods do not merely certify correctness but delineate the boundaries within which correctness can be assured. When properly contextualized, these outcomes contribute to a comprehensive understanding of system behavior under extreme or unmodeled conditions, supporting risk-informed decision-making in system development and regulatory evaluation.

Integrating formal methods into detect-and-avoid development processes from early stages offers tangible benefits. Encoding candidate algorithms and requirements in formal languages suitable for model checking or reachability analysis exposes logical ambiguities and hidden assumptions before implementation [40]. For instance, an algorithm that relies on implicit timing relationships between conflict detection and advisory generation can be formally analyzed to confirm whether these relationships always hold under bounded delays. Similarly, reachability analysis can reveal whether thresholds for alert generation provide sufficient temporal margin for maneuver execution given modeled response dynamics. Early detection of such issues reduces redesign cost and increases confidence in subsequent validation phases.

The artifacts produced through formal verification complement traditional validation data. In addition to simulation-based evidence, the verification process yields algebraic certificates, logical proofs, and explicit invariant descriptions that collectively substantiate compliance with safety requirements. These artifacts are reproducible, transparent, and subject to independent verification, making them valuable for certification and assurance documentation [41]. Moreover, the mathematical structure of formal results enables clear traceability: each verified property is linked to the assumptions, models, and parameters under which it holds. This traceability supports structured argumentation frameworks

in which assurance claims are decomposed into verifiable subclaims supported by formal evidence.

Despite their rigor, formal methods are not a substitute for empirical validation. The fidelity of formal results depends on the representativeness of the model, and models inevitably simplify reality. Nevertheless, the combination of formal verification and empirical testing provides complementary coverage: formal analysis ensures correctness within defined bounds, while experiments confirm that the assumptions underlying those bounds are met in operational conditions. Together, these approaches yield a more comprehensive assurance basis than either can achieve alone. Formal methods thus contribute to a layered verification strategy, where mathematical reasoning guards against logical and computational errors, and experimental testing validates environmental and implementation fidelity. [42]

Future progress in this domain depends on advances in scalability, model expressiveness, and automation. Scalable abstractions that can handle high-dimensional hybrid systems with complex logic transitions are essential for verifying realistic detect-and-avoid algorithms that include multiple interacting subsystems. Richer encounter models incorporating variable traffic densities, multi-intruder interactions, and environmental effects such as wind or turbulence must be represented in ways that remain analyzable within current verification frameworks. Furthermore, toolchains that link engineering-level modelsexpressed in simulation or software design environmentsto formal verification engines will reduce the overhead of constructing and maintaining formal representations. Such integrations can enable continuous verification throughout development, providing ongoing feedback as algorithms evolve.

Another promising direction involves the synthesis of detect-and-avoid logic directly from formally specified requirements [43]. Instead of verifying existing algorithms post hoc, synthesis methods generate algorithms that are guaranteed by construction to satisfy safety properties within the given assumptions. Although computationally demanding, these approaches eliminate ambiguity between design intent and implementation, producing controllers that are provably correct by design. In the context of unmanned aircraft systems, synthesis may eventually support adaptive detect-and-avoid logic that dynamically adjusts conservatism levels based on current uncertainty and performance conditions while maintaining formal safety guarantees.

The application of formal methods to detect-and-avoid verification contributes to a rigorous understanding of system safety under uncertainty. It provides a structured mechanism for quantifying the boundaries of algorithmic validity and for identifying trade-offs between operational flexibility and provable assurance. While not all aspects of certification can be reduced to formal analysis, the explicit and mathematical nature of the evidence produced by these methods strengthens the overall safety case. As computational tools mature and modeling practices become standardized, formal verification will increasingly serve as a cornerstone of the assurance process for unmanned aircraft integration into shared airspace, bridging the gap between theoretical guarantees and practical operational safety [44].

References

[1] I. Japins, S. Kodors, and S. Emale, "Integration of unmanned aerial vehicles flying beyond visual line of sight into air traffic," *HUMAN. ENVIRONMENT. TECHNOLOGIES. Proceedings of the Students International Scientific and Practical Conference*, no. 24, pp. 53–58, Apr. 22, 2020. DOI: 10.17770/het2020.24.6750

[2] J. F. Falorca, J. P. N. D. Miraldes, and J. Lanzinha, "New trends in visual inspection of buildings and structures: Study for the use of drones," *Open Engineering*, vol. 11, no. 1, pp. 734–743, Jan. 1, 2021. DOI: 10.1515/eng-2021-0071

[3] J. L. Sanchez-Lopez, J. Pestana, P. de la Puente, and P. Campoy, "A reliable open-source system architecture for the fast designing and prototyping of autonomous multi-uav systems: Simulation and experimentation," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 779–797, Oct. 23, 2015. DOI: 10.1007/s10846-015-0288-x

[4] M. Kratky and J. Farlik, "Countering uavs the mover of research in military technology," *Defence Science Journal*, vol. 68, no. 5, pp. 460–466, Sep. 12, 2018. DOI: 10.14429/dsj.68.12442

[5] T. Mastelic, J. Lorincz, I. Ivandic, and M. Boban, "Aerial imagery based on commercial flights as remote sensing platform," *Sensors (Basel, Switzerland)*, vol. 20, no. 6, pp. 1658–, Mar. 17, 2020. DOI: 10.3390/s20061658

[6] F. Nilsen et al., "Pc-2 winter process cruise (wpc): Cruise report," *The Nansen Legacy Report Series*, no. 26, Nov. 22, 2021. DOI: 10.7557/nlrs.6324

[7] A. Milosavljevi, "Automated processing of remote sensing imagery using deep semantic segmentation: A building footprint extraction case," *ISPRS International Journal of Geo-Information*, vol. 9, no. 8, pp. 486–, Aug. 11, 2020. DOI: 10.3390/ijgi9080486

[8] J. B. Babaan, J. P. Ballori, A. M. Tamondong, R. V. Ramos, and P. M. Ostrea, "Estimation of pm_{2.5} vertical distribution using customized uav and mobile sensors in brgy. up campus, diliman, quezon city," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-4/W9, pp. 89–103, Oct. 26, 2018. DOI: 10.5194/isprs-archives-xlii-4-w9-89-2018

[9] P. Álvares, L. Silva, and N. Magaia, "Blockchain-based solutions for uav-assisted connected vehicle networks in smart cities: A review, open issues, and future perspectives," *Telecom*, vol. 2, no. 1, pp. 108–140, Mar. 12, 2021. DOI: 10.3390/telecom2010008

[10] M. J. B. Lotinga, C. Ramos-Romero, N. Green, and A. J. Torija, "Noise from unconventional aircraft: A review of current measurement techniques, psychoacoustics, metrics and regulation," *Current Pollution Reports*, vol. 9, no. 4, pp. 724–745, Dec. 7, 2023. DOI: 10.1007/s40726-023-00285-4

[11] A. Zolich et al., "Survey on communication and networks for autonomous marine systems," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 3, pp. 789–813, Apr. 21, 2018. DOI: 10.1007/s10846-018-0833-5

[12] A. Prait, D. Bereikien, and N. ilinsk, "Regulation of unmanned aerial systems and related privacy issues in lithuania," *Baltic Journal of Law & Politics*, vol. 10, no. 2, pp. 107–132, Dec. 1, 2017. DOI: 10.1515/bjlp-2017-0014

[13] M. Schootman et al., "Emerging technologies to measure neighborhood conditions in public health: Implications for interventions and next steps.," *International journal of health geographics*, vol. 15, no. 1, pp. 20–20, Jun. 23, 2016. DOI: 10.1186/s12942-016-0050-z

[14] R. Martorana, P. Capizzi, A. Pisciotta, S. Scudero, and C. Bottari, "An overview of geophysical techniques and their potential suitability for archaeological studies," *Heritage*, vol. 6, no. 3, pp. 2886–2927, Mar. 9, 2023. DOI: 10.3390/heritage6030154

[15] P. McDowall and H. J. Lynch, "Ultra-fine scale spatially-integrated mapping of habitat and occupancy using structure-from-motion.," *PloS one*, vol. 12, no. 1, pp. 0 166 773–, Jan. 11, 2017. DOI: 10.1371/journal.pone.0166773

[16] Z. Sándor, "Challenges caused by the unmanned aerial vehicle in the air traffic management," *Periodica Polytechnica Transportation Engineering*, vol. 47, no. 2, pp. 96–105, Dec. 21, 2017. DOI: 10.3311/pptr.11204

[17] A. L. Diaz et al., "The bathy-drone: An autonomous unmanned drone-tethered sonar system," *Drones*, vol. 6, no. 8, pp. 220–220, Aug. 22, 2022. DOI: 10.3390/drones6080220

[18] A. C. Canolla, M. B. Jamoom, and B. Pervan, "Interactive multiple model hazard states prediction for unmanned aircraft systems (uas) detect and avoid (daa)," in *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, 2018, p. 2011.

[19] J. Castro, F. O. Borges, A. Cid, M. I. Laborde, R. Rosa, and H. C. Pearson, "Assessing the behavioural responses of small cetaceans to unmanned aerial vehicles," *Remote Sensing*, vol. 13, no. 1, pp. 156–, Jan. 5, 2021. DOI: 10.3390/rs13010156

[20] M. Behjati, R. Nordin, M. A. Zulkifley, and N. F. Abdullah, "3d global path planning optimization for cellular-connected uavs under link reliability constraint.," *Sensors (Basel, Switzerland)*, vol. 22, no. 22, pp. 8957–8957, Nov. 19, 2022. DOI: 10.3390/s22228957

[21] L. G. Torres, S. L. Nieukirk, L. S. Lemos, and T. E. Chandler, "Drone up! quantifying whale behavior from a new perspective improves observational capacity," *Frontiers in Marine Science*, vol. 5, Sep. 10, 2018. DOI: 10.3389/fmars.2018.00319

[22] W. A. Reid and I. M. Albayati, "Design of an unmanned aircraft system for high-altitude 1 kw fuel cell power system.," *Aerospace Systems*, vol. 4, no. 4, pp. 353–363, Oct. 22, 2021. DOI: 10.1007/s42401-021-00101-1

[23] L. Bretschneider et al., "Messbarmulticopter and instrumentation for air quality research," *Atmosphere*, vol. 13, no. 4, pp. 629–629, Apr. 15, 2022. DOI: 10.3390/atmos13040629

[24] M. B. Jamoom, A. Canolla, B. Pervan, and M. Joerger, "Unmanned aircraft system sense and avoid integrity: Intruder linear accelerations and analysis," *Journal of Aerospace Information Systems*, vol. 14, no. 1, pp. 53–67, 2017.

[25] M. Doukari, S. Katsanevakis, N. Soulakellis, and K. Topouzelis, "The effect of environmental conditions on the quality of uas orthophoto-maps in the coastal environment," *ISPRS International Journal of Geo-Information*, vol. 10, no. 1, pp. 18–, Jan. 6, 2021. DOI: 10.3390/ijgi10010018

[26] S. Yldz, S. Kvrak, and G. Arslan, "Using drone technologies for construction project management: A narrative review," *Journal of Construction Engineering, Management & Innovation*, vol. 4, no. 4, pp. 229–244, Dec. 31, 2021. DOI: 10.31462/jcemi.2021.04229244

[27] N. Sookram, D. Ramsewak, and S. Singh, "The conceptualization of an unmanned aerial system (uas) shipshore delivery service for the maritime industry of trinidad," *Drones*, vol. 5, no. 3, pp. 76–, Aug. 6, 2021. DOI: 10.3390/drones5030076

[28] E. Stott, R. Williams, and T. Hoey, "Ground control point distribution for accurate kilometre-scale topographic mapping using an rtk-gnss unmanned aerial vehicle and sfm photogrammetry," *Drones*, vol. 4, no. 3, pp. 55–, Sep. 8, 2020. DOI: 10.3390/drones4030055

[29] F. Mechan, Z. Bartonicek, D. Malone, and R. S. Lees, "Unmanned aerial vehicles for surveillance and control of vectors of malaria and other vector-borne diseases.," *Malaria journal*, vol. 22, no. 1, pp. 23–, Jan. 20, 2023. DOI: 10.1186/s12936-022-04414-0

[30] G. Zogopoulos-Papaliakos, G. C. Karras, and K. J. Kyriakopoulos, "A fault-tolerant control scheme for fixed-wing uavs with flight envelope awareness," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 2, pp. 1–33, May 27, 2021. DOI: 10.1007/s10846-021-01393-3

[31] A. C. Canolla, M. B. Jamoom, and B. Pervan, "Unmanned aircraft systems detect and avoid sensor hybrid estimation error analysis," in *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, p. 4384.

[32] C. E. Lin, P.-C. Shao, and Y.-Y. Lin, "System operation of regional utm in taiwan," *Aerospace*, vol. 7, no. 5, pp. 65–, May 25, 2020. DOI: 10.3390/aerospace7050065

[33] C. D. Johnson, M. E. Miller, C. F. Rusnock, and D. R. Jacques, "Applying control abstraction to the design of humanagent teams," *Systems*, vol. 8, no. 2, pp. 10–, Apr. 12, 2020. DOI: 10.3390/systems8020010

[34] S. Samaras et al., "Deep learning on multi sensor data for counter uav applications-a systematic review.," *Sensors (Basel, Switzerland)*, vol. 19, no. 22, pp. 4837–, Nov. 6, 2019. DOI: 10.3390/s19224837

[35] E. Husson, H. Reese, and F. Ecke, "Combining spectral data and a dsm from uas-images for improved classification of non-submerged aquatic vegetation," *Remote Sensing*, vol. 9, no. 3, pp. 247–, Mar. 7, 2017. DOI: 10.3390/rs9030247

[36] A. Jenal, G. Bareth, A. Bolten, C. Kneer, I. Weber, and J. Bongartz, "Development of a vnir/swir multispectral imaging system for vegetation monitoring with unmanned aerial vehicles.," *Sensors (Basel, Switzerland)*, vol. 19, no. 24, pp. 5507–, Dec. 13, 2019. DOI: 10.3390/s19245507

[37] R. S. Stansbury, M. A. Vyas, and T. A. Wilson, "A survey of uas technologies for command, control, and communication (c3)," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1, pp. 61–78, Jul. 22, 2008. DOI: 10.1007/s10846-008-9261-2

[38] S. Hashemi and R. Botez, "Lyapunov-based robust adaptive configuration of the uas-s4 flight dynamics fuzzy controller," *The Aeronautical Journal*, vol. 126, no. 1301, pp. 1187–1209, Feb. 11, 2022. DOI: 10.1017/aer.2022.2

[39] S. Sanz-Martos et al., "Drone applications for emergency and urgent care: A systematic review.," *Prehospital and disaster medicine*, vol. 37, no. 4, pp. 502–508, Jun. 9, 2022. DOI: 10.1017/s1049023x22000887

[40] P. Ramos, V. Vargas, N.-E. Zergainoh, and R. Velazco, "Assessing the static and dynamic sensitivity of a commercial off-the-shelf multicore processor for noncritical avionic applications," *Journal of Nanotechnology*, vol. 2018, pp. 1–8, Jul. 8, 2018. DOI: 10.1155/2018/2926392

[41] T. Merz and S. Chapman, "Autonomous unmanned helicopter system for remote sensing missions in unknown environments," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-1/C22, pp. 143–148, Sep. 6, 2012. DOI: 10.5194/isprsarchives-xxxviii-1-c22-143-2011

[42] E. N. Polisar, "Slot machine warfare: China's campaign to undermine american military plans in the commonwealth of the northern mariana islands," *Journal of Advanced Military Studies*, vol. 11, no. 1, pp. 44–63, Jun. 16, 2020. DOI: 10.21140/mcuj.20201102002

[43] G. Verhoeven, "Are we there yet? a review and assessment of archaeological passive airborne optical imaging approaches in the light of landscape archaeology," *Geosciences*, vol. 7, no. 3, pp. 86–, Sep. 14, 2017. DOI: 10.3390/geosciences7030086

[44] J.-w. Tao, W.-c. Ji, and Q.-j. Fan, "An effective approach of collision avoidance for uav," *Journal of Intelligent & Robotic Systems*, vol. 108, no. 2, Jun. 3, 2023. DOI: 10.1007/s10846-023-01869-4