ICAS2026 SYDNEY | AUSTRALIA 13-17 SEPT 2026 icas2026.com

AN INVESTIGATION OF SAFETY-CRITICAL GROUND INTERFACE (TAKEOFF, LANDING) REGIME RECOGNITION FOR AUTOMATED AND AUTONOMOUS VERTICAL LIFT AIRCRAFT MISSION TASK DESIGN

Marc Alexander[1]

[1]**National Research Council of Canada (NRC), Flight Research Laboratory**

Abstract

Expert System and Machine Learning algorithms for flight regime recognition were developed to identify helicopter landing profiles. These state machine and pattern methods developed using NRC Bell 412 Advanced Systems Research Aircraft (ASRA) flight test data enable robust evidence-based mission task development for offline and online applications. Algorithms extracted manually piloted and supervised autonomous landings with performance robustness validated via Sensitivity and Monte-Carlo studies. Implemented in Python/C/C++, expert system processing was computationally light (some 10^8 operations for explicit timestamped profiles), while machine learning (some 10^{10} to 10^{11} operations for landing profile clustering) required minutes of runtime on Colab-scale hardware. Hybridization is projected to yield an approach able to substantiate airworthy safety-case assurance.

Keywords: Landing, Takeoff, Regime, Recognition, Data

Notation

h_i	**Altitude Time History at Sample, i**
y_i	**Event Detector Binary Landing Mask**
F_{ES}	**Expert-System Landing Detector**
ϑ_{ES}	**Event-Detector Parameter Set**
N_S	**Smoothing-Window Length**
W	**Altitude-Change Half-Window**
H_0	**Low Altitude Trigger Threshold**
Δ_R	**Reset/Descent Cut-off Threshold**
N_F	**Flat-Reset Threshold**
N_b	**Backward-Label Window**
X	**Landing-Profile Feature Matrix**
P_d	**Principal Component Analysis, Projection into d Dimensions**
d	**Retained Principal Components**
$\mathcal{K}$	**K-Means Clustering Operator**
N_K	**Number of Clusters**
g_j	**Landing Profile Cluster, j**
E	**Monte-Carlo Noise Sample**
σ	**Noise Standard Deviation**
r	**Monte-Carlo Trial Index**
y_i^{ref}	**Reference Landing Mask**
A	**Accuracy/Agreement Metric**
$P^{(r)}$	**Percent-Correct Score for Trial, r**
p	**Perturbed Sensitivity Parameter**
p_0	**Baseline Parameter Value**
J	**Output Metric**
J_0	**Baseline Output Metric**
N_L	**Number of Detected Landings**
$\bar{T}$	**Mean Landing Duration**
$\bar{V}$	**Mean Descent Rate**
$S_{J(p)}$	**Normalized Sensitivity Coefficient**

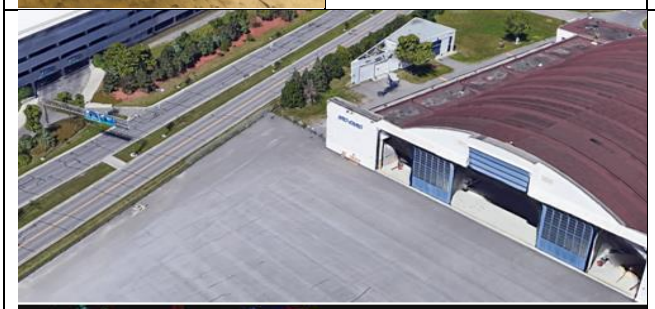
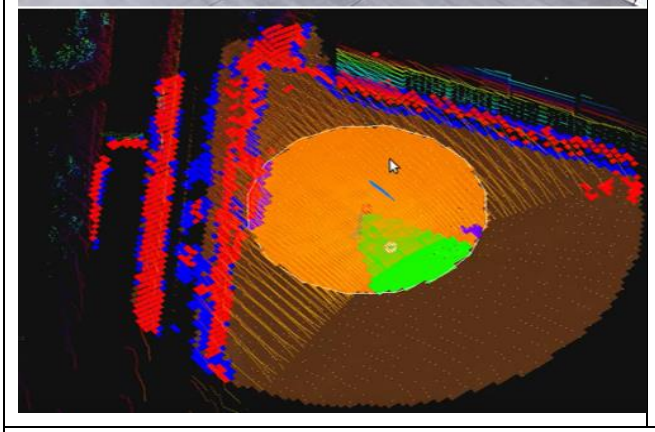
1. Research Background and Introduction

In vertical lift (helicopter, rotorcraft, lift-fan) applications, ground interfacing (takeoff, landing) is a mission and safety critical task involving a confluence of flight sciences and operational domains. In mission task development, recognition techniques are invaluable for investigation, verification, vehicle configuration, airworthiness assessment, certification, and operational planning [1].

Modelling and simulation, synthesized through aircraft system and parameter identification, support the fundamental understanding of flight regimes [2]. Maneuver, phase of flight, and condition recognition are applied in usage and life cycle management [3]. Mode recognition in areas of health, intent, safety, and anomalies provide flight system verification [4]. A hybrid recognition architecture combining online deterministic expert systems with offline machine learning may better support perception-driven, fly-by-wire (FBW) rotorcraft autonomy with airworthiness (e.g.; FAA/EASA) and safety-case assurance (e.g.; ARP/DO/MIL/ADS) compliance for ground interface operations [5,6,7].

In this work, Regime Recognition (RR) of landing mission tasks is investigated of augmented (pilot assisted and fully autonomous) helicopters. Applying raw NRC Bell 412 ASRA flight test data, results were studied statistically for safety critical context of offline and online use in vertical lift flight systems. Rotorcraft landing operations represent safety-critical phases of flight because they require precise control while operating in close proximity to terrain, obstacles, personnel, and infrastructure.

Rotorcraft routinely conduct landings in unprepared, confined, shipboard, urban, mountainous, offshore, and degraded-visual-environment locations where pilot workload and operational risk are significantly elevated. NRC has performed numerous landing studies however strategically for emerging vehicle development, perception-driven research (**Table 1**) has been an area of focus. Integration of Near Earth Autonomy (NEA) Perception (Generation I and II, Velodyne VLP-16/-32 LiDAR) and Landing Zone Evaluation system into the NRC Bell 412 ASRA for autonomy research was preceded by significant remote sensing investigations. The NEA LiDAR-based system (**Table 1, Left**) has provided mapping and landing zone (area evaluation, landing point determination) for NRC's flight autonomy programs [8]. Previously, flight research under the Multi-Purpose Airborne 3D Polarimetric LiDAR Assessment (MATPILA) program, an obscurant penetrating LiDAR system (**Table 1, Right**) was evaluated for enabling near-earth maneuvering and landing in degraded visual environments [9].

	
NRC Canada (NEA, PEREGRINE G(I,II)	**NRC Canada (DRDC, MATPILA)**
Integration: NRC B412 ASRA	**Integration**: NRC B412 ASRA
Type: Spinning (LiDAR); Nose-Mount; Stabilized	**Type**: Fixed (LiDAR); Nose-Mount; Stabilized
R&D: Autonomy Perception (SLAM; SLAD; LZE/LPD;OFN)	**R&D**: Area Perception (DVE; OP; OFN; SLAM)
Design Range: 0.1 - 1km	**Design Range:** 1km
	
	

NOTE: Simultaneous Localization & Mapping (SLAM), Safe Landing Area Determination (SLAD), Landing Zone Evaluation (LZE), Landing Point Determination (LPD), Obstacle Field Navigation (OFN), Degraded Visual Environment (DVE), Obscurant Penetration (OP)

Table 1: NRC Bell 412 ASRA Perception-Assisted Landing Research – NEA LiDAR, Landing Zone Evaluation Scan Results for Confined Area Operations [7] (Left); MATPILA LiDAR, Landing Profile Dust Obscuration Cloud Dynamics [8] (Right)

These perception-driven landing studies highlight the criticality of ground interfacing (landing, takeoff) characteristics, in manual and autonomous flight operations. Areas can include vehicle characteristics, landing site visual environment, and landing site physical characteristics. The transition between flight and ground contact introduces hazards associated with terrain interaction, obstacle clearance, rotor downwash, dynamics (rollover, hard landings), and loss of situational awareness. Consequently, robust landing-regime recognition and characterization methods are essential for developing training methods, flight-safety monitoring, operational analysis, and future certifiable mission task execution capabilities for cyber-physical autonomous aircraft.

2. Research Scope

For this work, Expert (human assisted) and Unsupervised (machine learning) classifier and clustering algorithms, respectively, were studied with an overview of investigations provided below.

2.1 Research Aircraft and Data Set Description

NRC's fourth generation FBW research aircraft, the Bell 412 ASRA (**Figure 1**), is based on the Bell 412 HP utility transport [5,6]. An experimental simplex FBW control system has been integrated in conjunction with the standard Bell Helicopter mechanical flight control architecture. Associated infrastructure includes human interface, hydraulics, electrical, instrumentation, navigation, and data systems. The single string architecture consists of non-redundant FBW actuators, control computing, state sensing, and control software. The aircraft has been applied in 30 years of industrial research in vehicle pilotage, handling, operations, dynamics, performance, stability, control, loads, and systems development. Currently, a military research program in Flight Autonomy is underway.



Figure 1: NRC Bell 412 ASRA (C-FPGV)

Landing regime recognition was investigated using 13 hours of manually piloted and supervised autonomous flight test data sampled at 300 Hz. The dataset comprised unstructured operations and a 19-parameter state vector of attitudes, rates, velocities, accelerations, positions, and controls.

2.2 Research Methodology

For this investigation NRC helicopter flight test data was utilized in the development and validation of landing regime recognition. Deterministic and non-deterministic approaches (**Figure 2**) were applied synthesizing Expert (human assisted) and Unsupervised (machine learning) classifier and clustering algorithms, respectively. The computational system employed public hardware (Google Collaboratory (Colab-T4) GPU, 12.7GB RAM, 15.0 GB GPU RAM) applying a Python/C/C++ software programming environment.

The Expert Systems approach implements rule-based event detection using a two-feature heuristic classifier of altitude and slope parameters. Here a finite state machine with hysteresis behavior scans altitude time series and produces a binary mask marking samples that belong to a landing segment. Retroactive back-labeling is applied to the segments as well as landing profile (or phase) identification. Mathematically the script applied windowed "derivative" data segment testing (trajectory rising, falling, or flat), a threshold classifying state machine (Boolean states for landing (true or false)), retroactive segmentation back-labeling (based on touchdown descent), and data phase gating (altitude, slope, and landing state persistence).

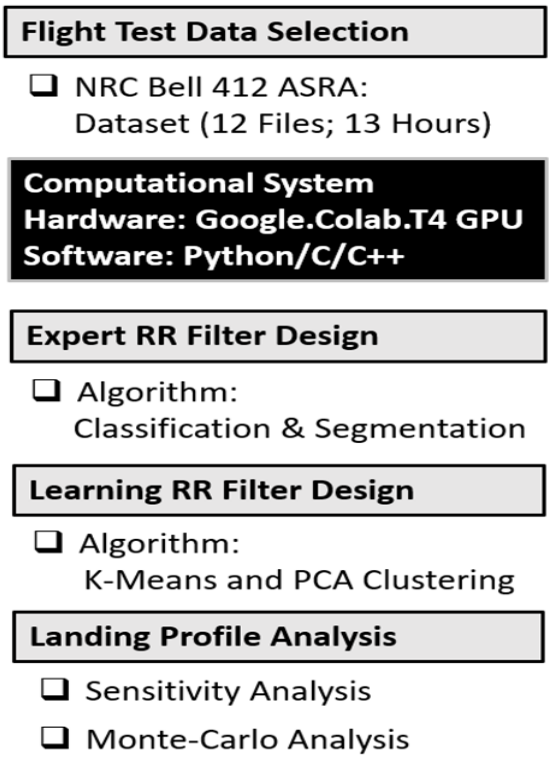


Figure 2: RR Research Methodology

The unsupervised approach applies Machine Learning to process the structured and labelled data set, principal component analysis is applied to perform feature identification, and K-Means clustering is applied to reduce the dataset into mathematically similar feature space or clusters. Software was tuned and identified landing profiles validated through Sensitivity and Monte-Carlo statistical analyses [10]. Landing profiles and clusters were analyzed in context supporting of offline and online flight systems applications.

The landing-regime recognition methodology was developed as a hybrid deterministic and machine-learning framework for extraction, characterization, and categorization of helicopter landing trajectories from large unstructured flight-test datasets. The methodology was developed and analyzed using recorded NRC Bell 412 Advanced Systems Research Aircraft (ASRA) data. The flight-test data contains state parameters including aircraft response (rates, accelerations, velocities, positions), feedforward control (pilot controls inputs), feedback control (flight control actuation), and vehicle performance (rotor, engine, computing, health monitoring) sampled at 300 Hz over approximately 13 flight hours.

The preliminary task of the methodology consisted of data conditioning and preprocessing. Flight-test records were time-aligned, relevant signals were extracted, and altitude data were smoothed using moving-average filtering to suppress measurement noise and transient disturbances. A deterministic expert-system event detector was then applied to identify landing regimes. The event detector operated as a state machine using a centered altitude-change feature derived from temporally windowed altitude differencing. Landing-state activation occurred when low-altitude and descent-transition criteria were simultaneously satisfied. State persistence, hysteresis logic, reset thresholds, and backward-label propagation were subsequently used to maintain continuous landing-state identification and extend detected landing windows into the preceding approach trajectory. The event detector generated binary landing masks, landing timestamps, landing durations, descent-rate metrics, and extracted landing trajectories such that;

$$\boldsymbol{y_i} = \mathbf{F_{ES}}(\boldsymbol{h_i}\,;\,\boldsymbol{\vartheta_{ES}})\ , \qquad \boldsymbol{\vartheta_{ES}} = \{\,\boldsymbol{N_s}\,,\boldsymbol{W}\,,\boldsymbol{H_0}\,,\Delta_{\boldsymbol{R}}\,,\boldsymbol{N_F}\,,\boldsymbol{N_b}\,\} \tag{1}$$

where; the expert-system filter ($\boldsymbol{\vartheta_{ES}}$) maps altitude history $\boldsymbol{h_i}$ into a binary landing mask $\boldsymbol{y_i}$ using smoothing ($\boldsymbol{N_S}$), windowing ($\boldsymbol{W}$), altitude trigger ($\boldsymbol{H_0}$), reset threshold ($\boldsymbol{\Delta_R}$), flat reset ($\boldsymbol{N_F}$), and backward labelling ($\boldsymbol{N_b}$).

Monte-Carlo analysis was then performed by injecting Gaussian noise into altitude signals and re-running the detector to evaluate robustness to sensor uncertainty. Detector accuracy, variance, and confidence bounds were computed across multiple stochastic trials.

$$\boldsymbol{P^{(r)}(\sigma)} = \boldsymbol{A}\left[\boldsymbol{F_{ES}}(\boldsymbol{h_i} + \boldsymbol{\epsilon_i^{(r)}}\,;\ \boldsymbol{\vartheta_{ES}}\,,\ \boldsymbol{y_i^{ref}}\right], \quad \boldsymbol{\epsilon_i^{(r)}} \sim \boldsymbol{N}(\mathbf{0},\boldsymbol{\sigma^2}) \tag{2}$$

where; stochastic altitude perturbations ($\boldsymbol{\epsilon_i^{(r)}}$) are injected, the event detector ($\boldsymbol{\vartheta_{ES}}$) is rerun, and agreement with the reference landing mask ($\boldsymbol{y_i^{ref}}$) is measured.

Sensitivity analysis was additionally performed using one-at-a-time parameter perturbations to quantify detector stability with respect to smoothing-window size, derivative-window size, altitude thresholds, reset logic, and landing-persistence parameters.

$$\boldsymbol{S_{J(p)}} = \frac{\boldsymbol{\Delta J/J_0}}{\boldsymbol{\Delta p/p_0}}\ , \qquad \boldsymbol{J} \in \{\,\boldsymbol{N_L}\,,\ \boldsymbol{\overline{T}},\ \boldsymbol{\overline{V}}\,\} \tag{3}$$

where; each sensitivity parameter perturbation $\boldsymbol{p}$ is evaluated by its normalized effect on landing count ($\boldsymbol{N_L}$), mean landing duration ($\boldsymbol{\overline{T}}$), and mean descent rate ($\boldsymbol{\overline{V}}$).

Finally, extracted landing trajectories were processed using Principal Component Analysis (PCA) and K-Means clustering. PCA reduced dimensionality and extracted dominant operational-state variance modes from multidimensional landing profiles. K-Means clustering subsequently grouped statistically similar landing trajectories into operational landing archetypes.

$$\boldsymbol{g_j} = \boldsymbol{\mathcal{K}}(\,\boldsymbol{P_d}(\boldsymbol{X});\ \boldsymbol{N_K}\,) \tag{4}$$

where; the landing-profile matrix $\boldsymbol{X}$ is reduced by PCA to $\boldsymbol{d}$ dominant features, then grouped by K-Means operator $\boldsymbol{\mathcal{K}}$ into $\boldsymbol{N_K}$ landing-profile cluster categories $\boldsymbol{g_j}$.

Together, the methodologies enabled deterministic landing recognition, robustness assessment, and unsupervised operational categorization that could be suitable for operational requirements such as flight-safety parametric analysis, usage monitoring operational segmentation, and cyber-physical state monitoring in flight autonomy applications.

3. Research Results

For this work, Expert (human assisted) and Unsupervised (machine learning) classifier and clustering algorithms, respectively, were studied with investigative results provided below.

3.1 Expert State Machine Regime Recognition Filter Design

The sensitivity analysis demonstrated that landing-recognition algorithm behavior was a robust deterministic state machine. Parametric and operational stability was defined by two dominant classes for topology-preserving temporal parameters and topology-changing trigger bandwidth parameters.

The landing-detection methodology was developed as a deterministic expert-system filtering architecture for identification and extraction of helicopter landing profiles from large unstructured flight-test datasets. The approach utilized a rule-based event detector implemented as a state machine operating on altitude-derived features. The methodology was intended to provide computationally efficient, explainable, and repeatable landing-regime recognition suitable for both offline flight-data analysis and online supervisory flight-state monitoring applications. The methodology operated on continuous flight-test time histories sampled at fixed rate with the primary input consisting of altitude (radar altitude or altitude-above-ground) measurements. The raw altitude signal was first conditioned using a moving-average smoothing filter to suppress high-frequency measurement noise and transient perturbations. This smoothed signal was then processed using a centered finite-time differencing operation to generate an altitude-change feature representing the local vertical motion trend over a prescribed temporal window. This feature acted as a low-bandwidth surrogate for descent-rate behavior while maintaining robustness to local signal fluctuations.

Landing recognition was implemented using a deterministic state-machine architecture containing trigger, persistence, and reset logic. Landing-state activation occurred when the smoothed altitude fell below a minimum-altitude threshold while the altitude-change feature indicated positive landing-transition behavior. Once activated, the detector maintained a persistent landing state until either a reset or timeout condition occurred. Reset logic utilized a descent-threshold criterion and a flatness-duration counter to terminate landing-state persistence when the altitude-change feature became sufficiently flat. Additional backward-label propagation logic retroactively extended the landing mask into the preceding approach trajectory to capture complete operational landing profiles.

The analysis methodology consisted of several sequential phases. First, raw flight-test altitude records were imported and conditioned using smoothing filters. Second, derivative-like altitude-change features were computed over fixed temporal windows. Third, the state-machine detector evaluated trigger, persistence, and reset conditions for each time sample. Fourth, detected landing-state segments were converted into binary landing masks identifying all samples belonging to landing regimes. Fifth, temporal segmentation logic extracted individual landing events from transitions in the landing mask. Finally, event-level metrics were computed for each extracted landing profile.

The core computations performed by the methodology included moving-average filtering, centered temporal differencing, Boolean threshold logic, hysteresis-based state transitions, backward temporal propagation, and event segmentation analysis. The computational complexity of the detector was low because the methodology relied primarily on arithmetic operations, comparisons, counters, and binary state transitions rather than optimization or iterative learning algorithms. This enabled rapid processing of large flight-test datasets while maintaining deterministic execution behavior. The software implementation was developed as a procedural script architecture using array-based processing and sequential time-index evaluation. The implementation included explicit storage of smoothed altitude signals, altitude-change features, landing-state variables, flatness counters, and final landing masks. Reverse-time processing was employed during portions of the analysis to facilitate backward propagation of landing labels into preceding approach trajectories. Parameterized threshold definitions were applied to support the Monte-Carlo robustness analysis and sensitivity analysis.

Several key detector parameters governed algorithm behavior. The smoothing-window parameter controlled front-end signal-conditioning bandwidth and noise suppression. The half-window differencing parameter controlled the temporal scale of the altitude-change feature. The minimum-altitude threshold governed landing-trigger permissiveness. The descent-threshold parameter governed landing-state continuation and reset behavior. The flat reset parameter identified landing state persistence during helicopter near-earth operations (such as light-on-skids or landing dwell). The backward-label parameter controlled retroactive expansion of landing profiles into approach segments. Together, these parameters defined the detector’s temporal bandwidth, trigger sensitivity, and operational persistence characteristics.

The primary outputs of the methodology consisted of binary landing masks, extracted landing-event segments, landing start/end timestamps, landing durations, and descent-rate characterizations. These outputs enabled subsequent operational segmentation, clustering, mission analysis, and trajectory characterization. Additional derived outputs included counts of detected landing events, temporal profile statistics, and operational-state duration metrics.

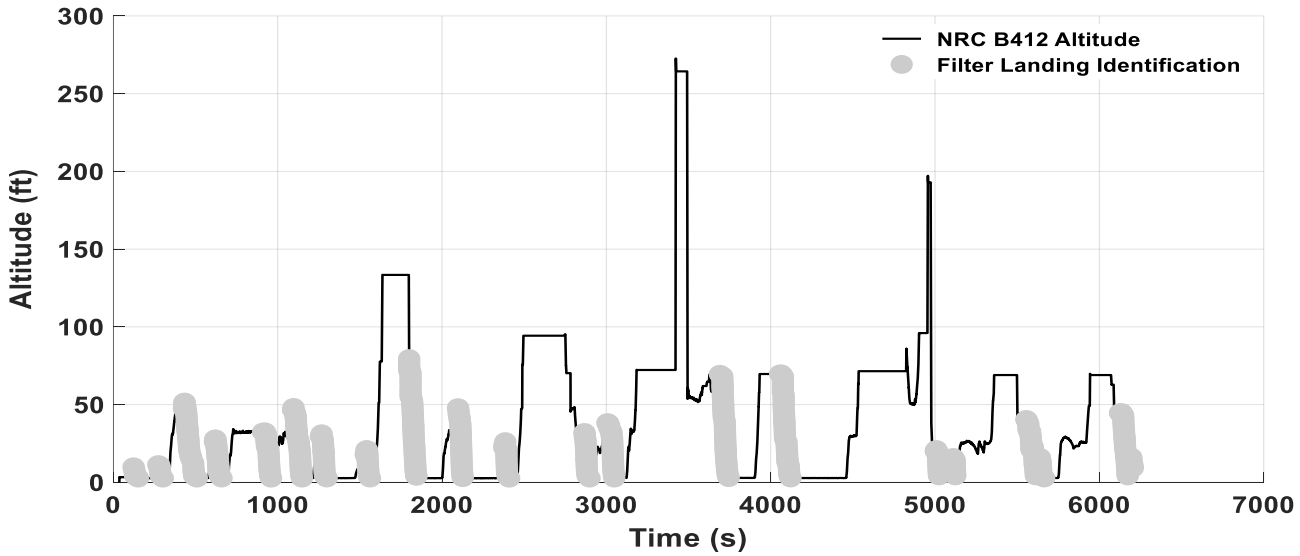


Figure 3: Expert State Machine Landing Profile Analysis – Landing Identification,

In terms of parametrization, the filter algorithm iterates backwards over time series data, identifies touchdown point using ground altitude, and identifies landing phase using rate of descent criteria. The landing entry altitude was selected to less than 80 feet. Landing criteria included approach entry altitudes greater than 25 feet, entry-to-descent timings of 5 seconds, and touch-down confirmation timings at 5 seconds after ground contact. Results of a filter algorithm pass are illustrated in **Figure 3** for a 1.7 hours flight sortie featuring numerous landing conditions.

3.2 Unsupervised Learning State Pattern Regime Recognition Filter Design

Machine learning was applied to analyze landing profile categories. K-Means Clustering was used for dataset partitioning into distinct groups by feature identification. Centroids are assigned for each cluster then used in grouping data based on centroid proximity. Dimensionality reduction was employed to handle the dataset's large 19-parameter state space. **Figure 4** depicts three dynamically different landing profile clusters identified utilizing Principal Component Analysis (PCA).

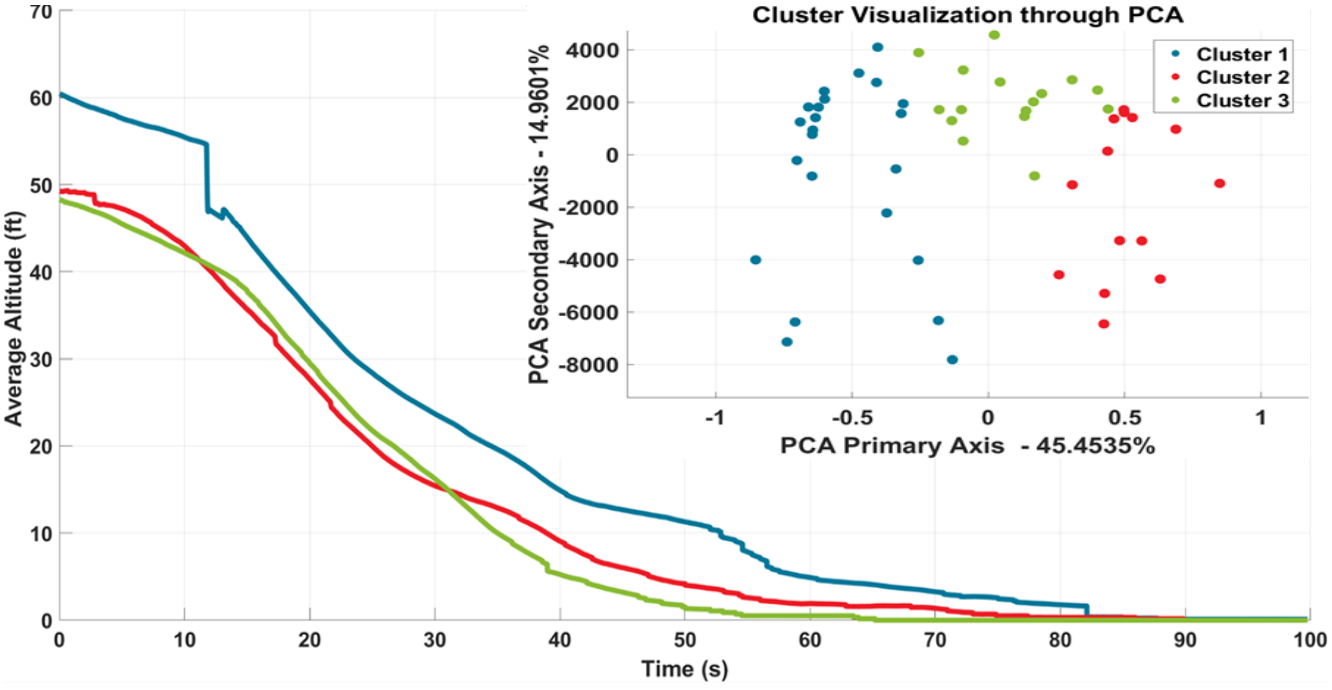


Figure 4: Machine Learning Landing Profile Analysis – Categorized Landing Characteristics (Primary), Landing Profile Clusters (Inset)

The **Stepped Landing Profiles (Figure 5)** indicate significant lateral dynamics with a moderate longitudinal aircraft displacement and a precise descent speed. The higher roll, pitch, and yaw rates of this cluster indicate a more dynamic and controlled approach making this profile suitable for environments requiring precise maneuvering, such as landing in confined areas or under challenging conditions such as when visual assessments during the landing progression are required at staged altitudes above ground during the descent.

The **Direct Landing Profiles (Figure 5)** are characterized by a focused linear approach prioritizing speed and efficiency of trajectory to ground. Low aircraft longitudinal displacement and high resultant velocity suggest a more aggressive landing style beneficial in time-critical landing situations. However, indications of higher velocity and pronounced heading deviations are indicative of a higher pilot workload during the landing than the Stepped Landing cluster. These profiles exemplify no-hover and tactical landings that could be applied in military operations ensuring the rapid dispersal of troops or equipment.

The **Mixed Landings Profiles (Figure 5)** balance characteristics of stepped and direct landings, offering a controlled approach with minimal longitudinal and vertical displacement. This profile cluster would be ideal for situations where stability and precision are crucial such as landing in degraded and adverse weather conditions or landing on unprepared terrain.

The **Mean Altitude Profile (Figure 5)** then provides a summative assembly of 64 identified landings analyzed. The mean altitude profile for these landings is defined by a duration period from 20 to 100 seconds to touchdown with average characteristics such as aircraft longitudinal displacements of 612.9 feet, descent rates of 0.7 feet-per-second, and negligible on-axis and off-axis rates at touchdown.

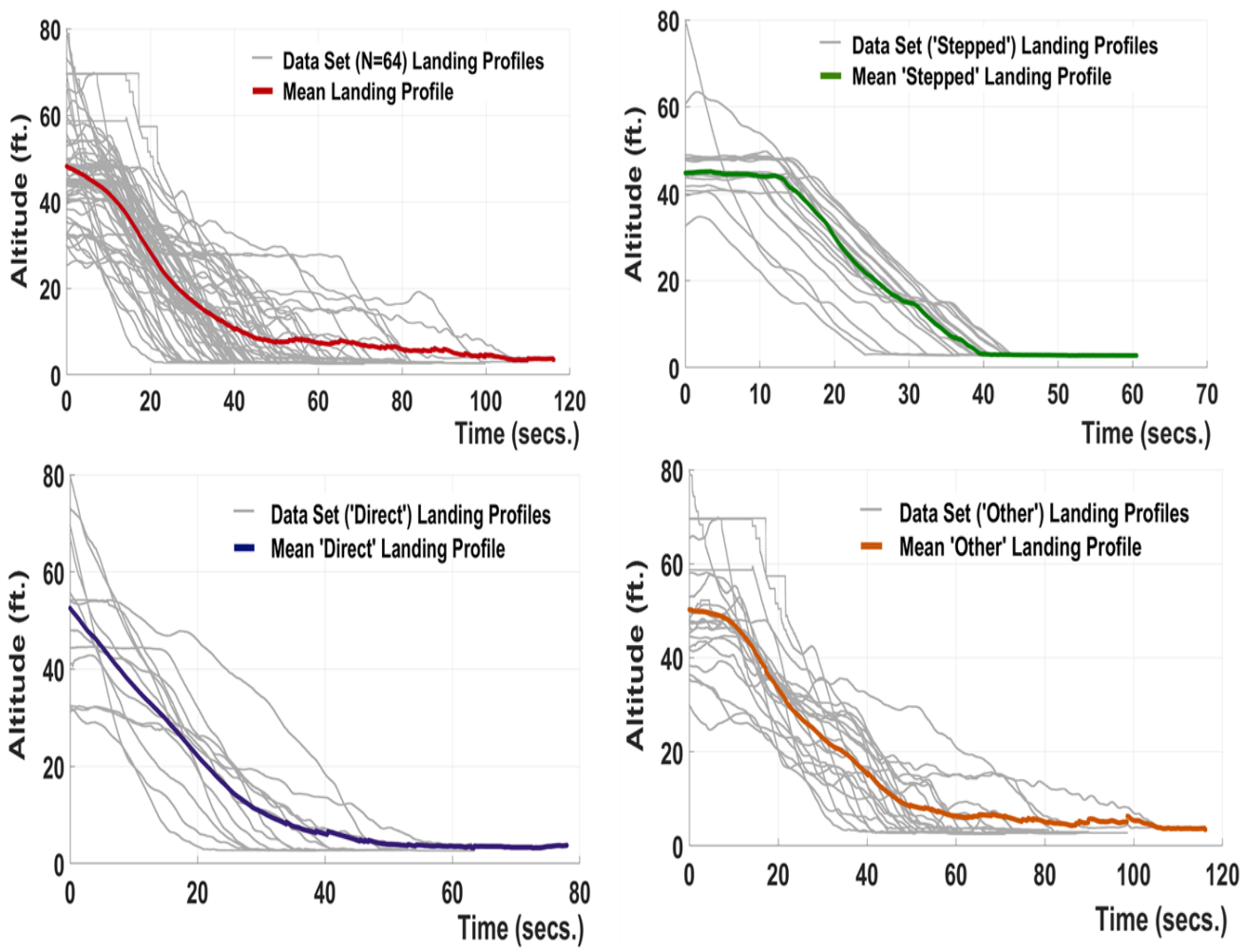


Figure 5: Expert State Machine Landing Profile Analysis – Full Data Set (Upper Left), 'Stepped' Data Set (Upper Right), 'Direct' Data Set (Lower Left), 'Other' Data Set (Lower Right),

These profiles form aircraft state-synchronized trajectories representing landing intent and are suitable for online applications such as guidance-policy references within manually piloted and supervisory autonomy flight systems architectures.

3.3 Parameter Sensitivity Analysis

Parameter sensitivity analysis demonstrated that landing-recognition algorithm behavior was a robust deterministic state machine. Parametric and operational stability were dominated by two classes; topology-preserving temporal parameters and topology-changing trigger bandwidth parameters. The analysis was performed on variations of 7 algorithmic parameters as indicated in **Table 2**.

Parameter	Parameter Range	Detected Landings	Mean Landing Duration	Mean Descent Rate Response	Primary Interpretation
AltDeltLimit	(-12, -8)	20	(49.20, 48.99 s) for Δ ≈ −0.21 s, −0.4%	(0.6234, 0.6264) ft/s, Δ ≈ +0.003 ft/s,+0.5%	Weak continuation threshold on landing persistence
BackLabels	(1000,2000)	20	(48.0, 50.5 s) for Δ ≈ +2.5 s, +5%	(0.646, 0.608) ft/s, Δ ≈ -0.038 ft/s, - 6%	Moderate retroactive expansion of landing extent into approach
FlatReset	(1500,4500)	20	(47.0, 51.6 s) for Δ ≈ +4.6 s, +9%	(0.648, 0.595) ft/s, Δ ≈ -0.053 ft/s, -8%)	Moderate extension of touchdown persistence
HalfWindow	(600,1200)	20	(47.4, 50.3 s) for Δ ≈ +2.9 s, +6%	(0.646, 0.603) ft/s, Δ ≈ -0.043 ft/s, -7%)	Moderate control on derivative temporal bandwidth
AltCutoff	(3, 9) ft	(17, 21)	(48.0, 49.5 s) for stable near 49 s	(0.619, 0.627) ft/s	High effect on primary altitude-trigger; governs false negatives/positives
NoiseSigma	(0, 2)	20	(49.03, 49.56 s) for Δ ≈ +0.53 s, +1%	(0.605, 0.640) ft/s bounded stochastic variation	Low impact on robustness to sensor uncertainty
SmthWindw	(3000,9000)	(21, 17)	(41, 55) s, for Δ ≈ +14 s, +34%	(0.74, 0.55) ft/s, Δ ≈ -0.19 ft/s, -26%)	High effect on primary front-end low-pass filter bandwidth

Table 2: Sensitivity Analysis on Landing State Detection Computational Parameters

Computational parameters **altdeltaLimit**, **BackLabels**, **FlatReset**, **HalfWindow**, and **NoiseSigma** preserved the number of detected landings across tested conditions. They primarily affect the characterization and temporal extent of landing profiles rather than the existence of landing events themselves. Among these, **AltDeltLimit** exhibited extremely weak sensitivity, with only a 0.21 sec. variation in mean landing duration and a negligible 0.003 ft/sec. change in mean descent rate over the entire tested range. Here, descent-rate continuation/reset logic operates within a broad stable region and is highly tolerant to threshold perturbations. The **BackLabels** parameter produced a smooth linear increase in landing duration from 48.0 to 50.5 sec. while simultaneously reducing mean descent rate from 0.646 to 0.608 ft/sec. This physically consistent behavior is exhibited because the parameter retroactively expands the landing mask further into the approach trajectory, increasing temporal extent while incorporating shallower descent segments. In a similar manner, the **FlatReset** parameter extended touchdown persistence and rollout inclusion, increasing landing duration from 47 to 51.6 secs. and reducing mean descent rate from 0.648 to 0.595 ft/sec. The **HalfWindow** parameter demonstrated comparable behavior because increasing the derivative temporal scale smooths the altitude-change feature and broadens landing-state persistence. Collectively, these three parameters shape the temporal representation of landing events while preserving recognition topology.

The **NoiseSigma** sensitivity study produced the strongest robustness result of the entire analysis. Landing count remained invariant at 20 detections for all noise levels from $\sigma = 0$ to $\sigma = 2$ ft., while mean landing duration varied by only 0.53 sec. and mean descent rate exhibited only bounded stochastic fluctuations. These results indicate that the moving-average filtering, large temporal differencing windows, and state-machine hysteresis collectively provide strong resistance to altitude-measurement uncertainty. The detector therefore behaves as a strongly low-pass deterministic recognition architecture. In contrast, **minAltCutoff** and **SmthWindw** fundamentally altered recognition topology and therefore represent the most safety-critical detector parameters.

The **minAltCutoff** parameter directly controls landing-state activation through altitude gating. At low thresholds (3 to 4 ft.), the detector missed valid landings, reducing detections from 20 to 17 to 19 due to insufficient trigger permissiveness. At high thresholds (>7.2 ft.), the detector over-triggered and produced 21 detections, indicating false-positive recognition of low-altitude maneuvering.

However, a broad stable plateau existed between approximately 4.2 ft. and 7.2 ft. where all 20 expected landings were consistently detected and temporal metrics remained stable. Importantly, the baseline value of 5 ft. lies near the center of this stable region, indicating appropriate operational margin.

The **SmthWndw** parameter exhibited the highest overall sensitivity because it controls the front-end bandwidth of the detector. Small smoothing windows (3000 to 4400 samples) produced false positives, increasing detections to 21 due to insufficient suppression of transient fluctuations. Large smoothing windows (>7000 samples) progressively suppressed valid touchdown signatures, reducing detections to as low as 17. Simultaneously, mean landing duration increased dramatically from 41 to 55 sec. while mean descent rate decreased from 0.74 to 0.55 ft/sec. These results show that excessive smoothing broadens and attenuates the altitude-change feature, reducing trigger visibility and delaying state transitions. Nevertheless, a broad stable operational plateau existed between approximately 4500 and 6900 samples, with the baseline value of 6000 centrally positioned within this region.

The interplay between parameters reveals several important relationships. The **SmthWndw** and **HalfWindow** parameters jointly define the effective detector bandwidth, balancing responsiveness against noise suppression. The **minAltCutoff** and **AltDeltLimit** parameters jointly define landing-trigger permissiveness and continuation logic. Meanwhile, **BackLabels** and **FlatReset** together govern temporal landing extent by controlling pre-trigger expansion and post-touchdown persistence or run-out. These relationships indicate that the detector behaves as a low-bandwidth temporally integrated state machine with coupled trigger, persistence, and filtering dynamics.

Overall, the sensitivity analysis demonstrates strong deterministic behavior, bounded parameter sensitivity, broad operational stability plateaus, and excellent robustness to stochastic measurement uncertainty. The most influential parameters, **SmthWndw** and **minAltCutoff**, directly govern feature observability and landing-state activation. All remaining parameters primarily influence landing characterization fidelity while preserving recognition integrity.

4. Research Discussion

In this discussion, the merits of the Expert Systems and Machine Learning approaches are discussed in context of computational robustness, computational efficiency, and understanding a potential approach for regulatory compliant flight systems implementation.

4.1 Monte Carlo Robustness Analysis

The parameter sensitivity analysis demonstrated that landing-recognition algorithm behavior was a robust deterministic state machine. Parametric and operational stability were defined by two dominant parameter classes for topology-preserving temporal parameters and topology-changing triggers.

Monte-Carlo testing results show that the explicit landing profile identification algorithm is strongly noise-robust, exhibiting high accuracy under low-to-moderate noise, as well as predictable degradation under extreme noise with no evidence of catastrophic misclassification or state instability. Definitions of noise test levels and associated qualitative results are provided in **Table 3**.

σ Level (ft)	Sensor Quantification	Projected Algorithm Response
0.05 – 0.3	Good Lidar /Heavily Filtered RA Response	Limited fault algorithmic response
0.3 – 1.0	Clean RA	Limited fault, Tight distributions
1.0 – 3.0	Noisy RA / Modest BO Jitter / Light Turbulence	Small degradations observed
3.0 – 6.0	Poor Conditions / Weak Filtering	Noticeable boundary jitter
6.0 – 15.0	GNSS Vertical Scatter	Significant degradation unless strong smoothing and state constraints applied
NOTE: RA = Radar Altimeter; BO = Barometer; GNSS = Global Navigation Satellite System		

Table 3: Monte-Carlo Test Parameterization, Sensor Noise Level Qualification

We find that the noise-level sensitivity analysis demonstrates that the deterministic landing-detection methodology is highly robust to moderate measurement uncertainty, with significant degradation only occurring at elevated noise levels. As depicted in **Figure 6** The analysis evaluated Gaussian noise levels (σ=0 to 5 ft.) and assessed their effect on landing count, classification accuracy, and confusion-matrix performance.

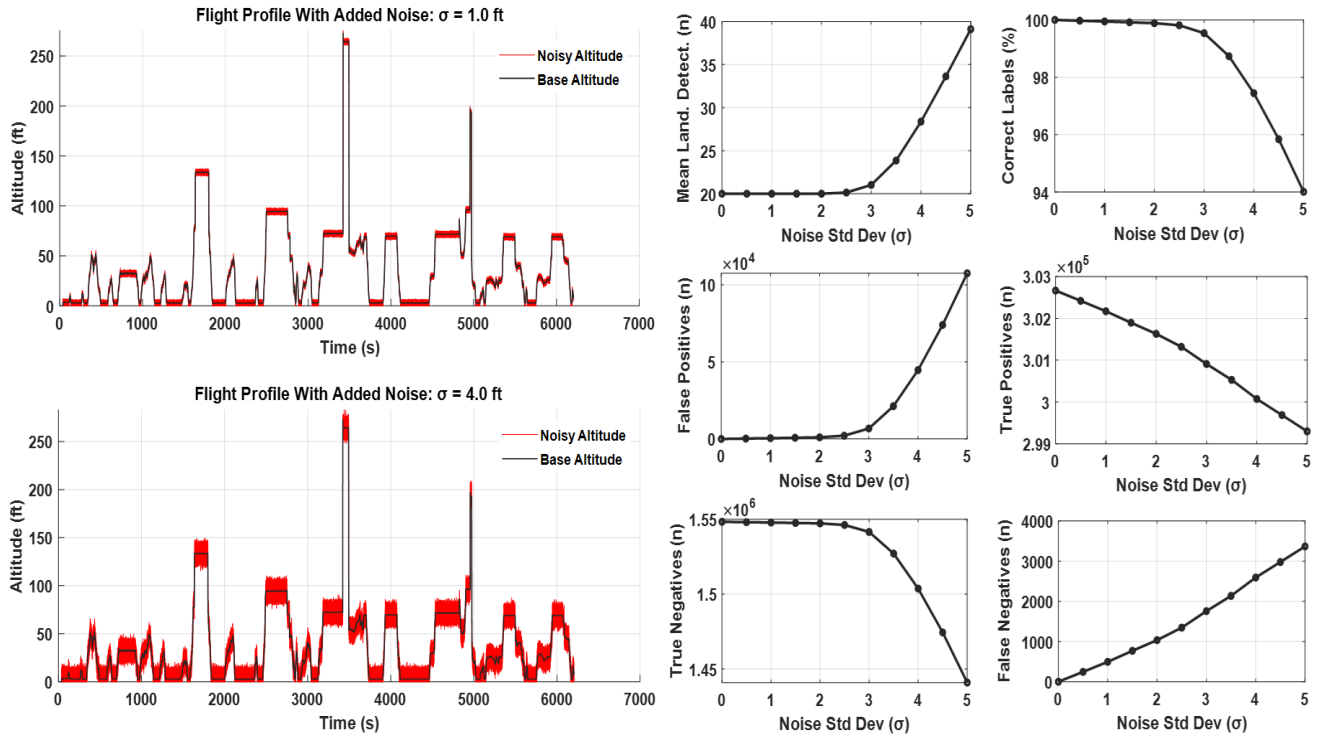


Figure 6: Monte-Carlo Test Results, Sensor Noise (Left: Sample Noise Cases at σ Level 1.0 and 4.0 ft) Impact on Landing Detection Accuracy (Right: Accuracy Noise σ Level 1.0 to 5.0 ft)

At low noise levels (σ < 3 ft), detector behavior remains essentially unchanged. Approximately 20 landing events are identified, classification accuracy remains near 100%, and both true-positive and true-negative counts remain stable. False-positive and false-negative counts are negligible. This indicates that the combination of altitude smoothing, windowed differencing, and state-machine hysteresis effectively suppresses low-amplitude sensor disturbances and prevents spurious state transitions.

Beyond approximately σ = 3 ft., detector behavior begins to change noticeably. The number of detected landings increases from the nominal 20 events to nearly 40 events at σ = 5 ft. Since the actual number of landings is fixed, these additional detections indicate fragmentation of landing segments and generation of spurious landing events caused by noise-induced threshold crossings. Correspondingly, overall classification accuracy decreases from approximately 100% to approximately 94%.

The confusion-matrix results reveal the underlying failure mechanism. True-positive counts increase slightly with increasing noise, indicating that genuine landing samples continue to be detected. False negatives remain relatively small, increasing only modestly at the highest noise levels. Thus, the detector rarely misses actual landing events even under severe measurement corruption. In contrast, false positives increase dramatically from near zero to more than 1e05 samples as noise increases. Simultaneously, true-negative counts decrease from approximately 1.55e06 to 1.44e06, demonstrating that many non-landing samples are incorrectly classified as landing samples.

The dominant effect of increasing noise is therefore a reduction in specificity rather than sensitivity. The detector remains highly effective at capturing true landing behavior but becomes increasingly permissive as measurement uncertainty grows. From a flight-safety perspective, this behavior is generally desirable because missed landing events are more critical than occasional false alarms. Overall, the results demonstrate a broad operational robustness region, with performance remaining essentially unchanged for realistic sensor-noise levels and degradation occurring primarily through increased false-positive activity at extreme noise levels.

Next, the Monte Carlo analysis (**Figure 7**) quantifies the robustness of the landing-detection algorithm to increasing levels of stochastic sensor noise. For each noise level (σ), hundreds of independent trials were performed by adding Gaussian noise to the altitude signal and comparing the resulting landing labels against the reference solution. The histograms show the distribution of percent-correct landing labels across all trials, providing insight into both mean performance and variability.

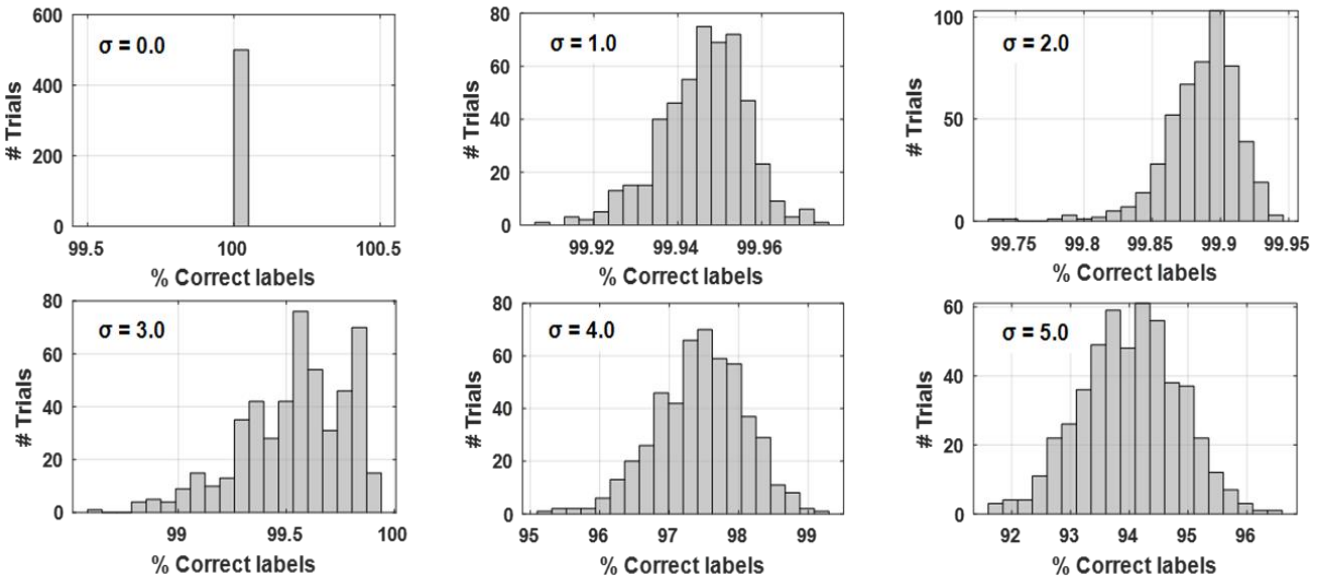


Figure 7: Monte-Carlo Testing, Regime Recognition Algorithm Robustness to Sensor Noise

At σ = 0 ft., all trials achieve 100% correct labeling, producing a single spike in the histogram. This confirms that the detector is deterministic and repeatable in absence of noise. At low noise levels (σ = 1 and σ = 2 ft.), distributions remain extremely narrow and centered near 99.9% correct labeling. The small spread indicates the smoothing filter, altitude-change feature, and state-machine hysteresis effectively reject moderate measurement disturbances while maintaining highly consistent recognition.

As noise increases to σ = 3 ft., the distribution broadens noticeably and becomes slightly asymmetric. Although most trials still achieve greater than 99% accuracy, the increased variance indicates that noise is beginning to influence detector state transitions. This corresponds to the onset of occasional false triggers and landing-segment fragmentation observed in the sensitivity analysis.

At σ = 4 ft., performance degradation becomes more apparent. The histogram shifts left, with the mean accuracy decreasing to approximately 97 to 98%. The wider distribution indicates greater trial-to-trial variability and increased sensitivity to specific noise realizations. Nevertheless, most trials remain well above 95% accuracy, demonstrating substantial robustness.

At the highest tested noise level (σ = 5 ft.), the distribution shifts further left and centers near 94 to 95% accuracy. The broader spread reflects increased uncertainty in landing-state classification. However, the detector still maintains high overall performance despite severe signal corruption.

The sensitivity and Monte Carlo analyses demonstrate that the landing-detection methodology possesses a broad region of operational robustness and exhibits graceful degradation under increasing measurement uncertainty. For low to moderate noise levels, detector behavior remains essentially unchanged, with stable landing counts, near-perfect classification accuracy, and tightly clustered Monte Carlo accuracy distributions. This indicates that the smoothing, altitude-change feature extraction, and state-machine logic effectively suppress realistic sensor noise.

As noise levels increase, the primary degradation mechanism is the generation of false-positive landing detections rather than missed landing events. True landing events continue to be identified reliably, while additional spurious detections arise from noise-induced threshold crossings. Consequently, detector sensitivity remains high while specificity gradually decreases. Monte Carlo results further show a gradual broadening and leftward shift of the accuracy distributions, reflecting increasing variability and reduced classification performance at extreme noise levels.

Overall, the methodology does not exhibit a sharp failure threshold. Instead, performance degrades progressively, maintaining high landing-detection capability even under severe signal corruption. This behavior is desirable for safety-critical flight-state recognition applications, where preserving true landing-event detection is generally more important than eliminating occasional false alarms. The combined analyses therefore provide strong evidence of algorithm robustness, stability, and suitability for operational flight-data analysis, monitoring, and autonomy-support applications.

From an operational perspective, the algorithm is robust for realistic sensor signal degradation cases tolerating noise beyond normal radar, barometric, and navigation system altitude sensor response. For flight test applications, qualification potential is favorable as unimodal degradation supports deterministic behavior and predictable error growth favorable for safety-critical signal classification.

4.2 Computational Loading Analysis

From a computational perspective, understanding implementation requirements for flight systems is critical for given regime recognition use-cases. For research herein, investigations utilized public hardware and local software environments to analysis overall algorithm functionality. Computational loading efficiency was estimated and summarized in **Table 4**.

Algorithm	Computational Load	Runtime	Outputs
Expert Systems Filter	10^8 operations, 2 – 10 sec.	2 to 10 seconds	Identified Landing Events (64)
Machine Learning Filter	10^{10} to 10^{11} operations	0.5 to 2 minutes	Identified Landing Clusters (3)

NOTE:
- **Computational Hardware Environment, Public: Google Collaboratory (Colab-T4) GPU, 12.7GB RAM, 15.0 GB GPU RAM**
- **Computational Software: Python/C/C++**

Table 4: Landing Regine Recognition Methodologies, Computational Loading Summary

Firstly, the research dataset scale at 13 hours for 300 Hz sample rate produced 14e06 samples where a 6-second (1800 sample; 900 half-window) filtering window sizing was applied.

For the Expert System Filter (landing profile level identification), per sample the algorithm applies approximately 4 array operations (reading, subtraction, multiplication), 20 comparisons with Boolean logic, and 5 book-keeping tasks (writes, counters) providing an estimate of 25 scalar operations per sample and points to 3.5e08 operations. A back label cost of up to 150 checks per landing end trigger indicated 9.75e04 operations. The Colab CPU runtime estimate is based upon vectorization for NumPy implementation (seconds) and pure Python for-loop logic (seconds to minutes). This indicates 1 to 5 minutes runtime (unoptimized loop) or 2 to 10 second runtime given for-loop logic in Python/NumPy (vectorized, optimized) estimates for Expert Filtering.
For the Machine Learning Filter (PCA and K-Means landing type cluster identification), we start with assessing 7,800 non-overlapping windows (i.e.; where overlaps would improve temporal localization but increase computations significantly). For PCA, SVD/eigen-decomposition [10] creates some 2.5e10 operations given 7800 windows and 1800 samples. Comparatively, K-Means computations on 3 clusters, 20 iterations, and 2 to 10 principal components for some 2.34e6 operations. This provides a 0.5 to 2 minute Machine Learning runtime dependent upon CPU or GPU computation implementation.

Expert-system filtering algorithm was computationally light (hundreds of millions of simple operations), ran in seconds when implemented efficiently; it directly outputs all landing profiles (64 events) with timestamps/masks which are of high value for flight-test analysis, Flight Operations Quality Assurance / Health Usage Monitoring Systems indexing, and ground interface (takeoff, landing) studies. The machine learning PCA + K-Means clustering algorithm is dominated by PCA (tens of billions of operations) for analysis of the full data set. On Colab-scale hardware, minutes of runtime was required for cluster output only, not explicit landing events. It is best used after data event segmentation has occurred or as part of a larger hybridized regime recognition pipeline.

4.3 Hybridized Implementation Approach

Expert systems based event identification regime recognition can provide a safety posture for civil and military safety-critical or safety-related functions. The algorithm provides deterministic bounded behaviors (explicit thresholds, timers, and state memory) that make failure modes more amenable for Functional Hazard Assessment / Preliminary System Safety Assessment / System Safety Assessment (FHA/PSSA/SSA) style safety cases [11]. It is inherently verifiable (from requirements to code to test) with traceability that aligns well with DO-178C [12] assurance objectives and behavior that can be exhaustively stress-tested with Monte Carlo and worst-case analysis. The Expert Systems methodology has feasibility for onboard flight systems integrations (i.e.; low computation requirements, provision of predictable latency). This approach is easier to integrate into fly-by-wire/auto-flight guidance monitors under rotorcraft certification frameworks (Part 27/29 [13, 14]; CS-27/29 [15, 16]; rotorcraft AC guidance [17]). From a security perspective, Expert System requires defined cyber security considerations if it informs guidance/autonomy (DO-326A [18] processes).

Machine Learning based event cluster categorization regime recognition is best for offline analytics and not safety-critical onboard real-time applications. In this approach, training outcomes are non-deterministic where event clustering depends on preprocessing, initialization, and dataset shifts. Unlike Expert Systems it is not a temporal state model. The approach possesses weak traceability for certification where it is more difficult to satisfy obligatory requirements and exhaustively verify outcomes compared to Expert Systems rule logic required in flight guidance/autonomy paths under DO-178C-style assurance [12]. The method does however enable post-flight offline evidence-building for systems development and design. For the purposes of post-flight discovery, it enables the grouping landing types, identifying outliers, and quantifying operational variability for aspects such as mission task and handling-qualities studies (perhaps for ADS-33-type task analysis [19]).

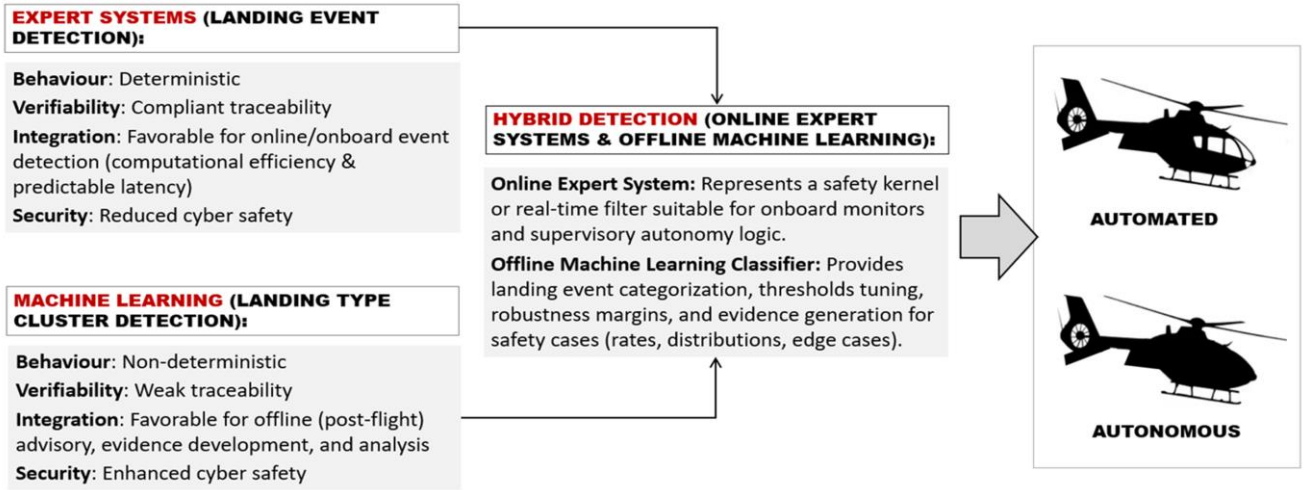


Figure 8: Hybridized Regime Recognition Approach for Flight Autonomy Applications

Results from this study point to the development and application of a combined approach or hybrid regime recognition approach (**Figure 8**) that could be applied in supervised (human monitored) autonomous flight systems applications. The hybrid regime-recognition framework is such that deterministic landing-event recognition provides certifiable flight-state identification, while machine-learning categorization provides operational context awareness. Together, they form evidence for airworthiness safety-case assurance of cyber-physical vertical-lift systems. The hybrid approach provides the most safety-critical posture where Expert Systems event recognition is applied in online context and machine learning for offline event cluster recognition. In the **Hybridized Approach** then,

A. Online Expert Systems Regime Recognition:

- Represents a safety kernel or real-time filter.
- Performs event detection and temporal segmentation with deterministic behavior suitable for onboard monitors and supervisory autonomy logic.

B. Offline Machine Learning Regime Recognition:

- Represents an offline characterization and continuous improvement tool.
- Uses extracted landing events to categorize landing archetypes, tune thresholds and robustness margins, and generate evidence for safety cases (rates, distributions, edge cases).

The hybrid approach is key as it could satisfy development assurance and safety processes (such as ARP4754A; AC 20-174; ARP4761A – [20, 21, 11]) that favor architectures where the safety-critical elements are small, deterministic, testable, and stable; higher-complexity analytics must remain non-interfering or advisory only. Software assurance expectations (DO-178C) [12] and system safety practice (MIL-STD-882E) [22] are easier to satisfy when non-deterministic (machine learning based algorithms) are not in the primary safety path. For cyber-physical autonomy then keeping machine learning out of the command path reduces security burdens, while still applying DO-326A [18] processes to the overall system.

The techniques developed herein may support regulation of complex vertical lift flight systems (cyber-physical, autonomous) with perception-based guidance, navigation, and fly-by-wire control for autonomous flight operations. However, it is the hybrid architecture - online deterministic expert systems combined with offline machine learning - that more aligns with FAA/EASA airworthiness frameworks and ARP, DO, MIL, and ADS-based safety-case assurance-traceability for the rotorcraft ground-interface (takeoff, landing) regime recognition problem.

5. Conclusions

For this study, NRC Bell 412 Advanced Systems Research Aircraft (ASRA) flight test data was utilized to develop Regime Recognition algorithms applicable in online and offline mission task design. Utilizing the software approaches, 64 landing events and 3 landing profile types were identified from the 13-hour data record. Software was tuned and identified landing altitude profiles with process flow validated through Sensitivity analysis (via 7 algorithmic parameters) and Monte-Carlo statistical (via Gaussian noise levels, σ = 0 to 5 feet) analysis

The Expert Systems approach implements rule-based event detection using a two-feature heuristic classifier of altitude and slope parameters. Here a finite state machine with hysteresis behavior scans altitude time series and produces a binary mask marking samples that belong to a landing segment. Expert systems -based landing event regime recognition can provide a safety posture for civil and military safety-critical or safety-related functions. The algorithm provides deterministic bounded behaviors that make failure modes more amenable to formal safety case analysis. From an operational perspective, the algorithm is robust for realistic sensor signal degradation cases tolerating noise beyond normal radar, barometric, and navigation system altitude sensor response. For flight test applications, qualification potential is favorable as unimodal degradation supports deterministic behavior and predictable error growth favorable for safety-critical signal classification.

The unsupervised approach applies Machine Learning to process the structured and labelled data set, principal component analysis is applied to perform feature identification, and K-Means clustering is applied to reduce the dataset into mathematically similar feature space or clusters. Unsupervised Machine Learning based landing categorization regime recognition is best for offline analytics and not safety-critical onboard real-time applications as in this approach, training outcomes are non-deterministic. The approach possesses weak traceability for certification where it is more difficult to satisfy obligatory requirements and exhaustively verify outcomes compared to Expert Systems rule logic required in flight guidance/autonomy paths.

The algorithms extracted an assembly of 64 landing profiles from the flight test data set via Expert Systems regime recognition. The mean altitude profile for these landings is defined by a duration range from 20 to 100 seconds to touchdown with average characteristics such as horizontal ranges of 612.9 feet, descent rates of 0.7 feet-per-second, and negligible on-axis and off-axis rates at touchdown. These landing profiles were categorized into direct, stepped, and mixed profile clusters via Machine Learning regime recognition.

Investigated in a public hardware environment (Google Collaboratory (Colab-T4) GPU, 12.7GB RAM, 15.0 GB GPU RAM) applying Python/C/C++ software, Expert Systems was found to be computationally more efficient (10e8 operations; 2 to 10 second execution for extraction of explicit timestamped landing profiles) compared to the Machine Learning approach (10e10 to 10e11 operations; 0.5 to 2 minute execution for landing profile clustering).

From a regulatory standpoint, a hybrid regime-recognition framework such that deterministic landing-event recognition provides certifiable flight-state identification, while machine-learning categorization provides operational-context awareness, was found to be a plausible approach for implementation. Together, these regime recognition methodologies may provide evidence required for airworthiness safety-case assurance (i.e.; Functional Hazard Assessment / Preliminary System Safety Assessment / System Safety Assessment (FHA/PSSA/SSA)) for emerging autonomous and cyber-physical vertical-lift flight systems.

Acknowledgements

Significant technical and scientific contributions by NRC Flight Research Laboratory staff groups (Administration, Operations, Instrumentation, Airworthiness, Research) are graciously acknowledged.

References

[1] Khazaee, M., Jahandideh, et al.; Rotorcraft Flight Regime Recognition in Complex Maneuvers Using Adaptive Extended Kalman Filter; International Journal of Aero. & Space Sciences; V.24, p.919–927; 2023.

[2] Tischler, M., Remple, K.; Aircraft and Rotorcraft System Identification – 2nd Ed.; AIAA; 2012.

[3] Evans, H., Finckenor, J., et al.; Further Applications of SUMS-based Regime Recognition in Updating the MH-47G Chinook Usage Spectrum; Vertical Flight Society's 80th Forum; 2024.

[4] Que, Z., et al.; Real-Time Anomaly Detection for Flight Testing using AutoEncoder and LSTM, International Conference Field-Programmable Technology; IEEE; 2019.

[5] Alexander, M., Carignan, S.; NRC Research in Autonomy Flight Sciences: Experimentation in Rotorcraft Landing Methodology in Degraded Visual Environments; Vertical Flight Society Transformative Vertical Flight 2022 - Aeromechanics for Advanced Vertical Flight Technical Meeting; Virtual Meeting; 2022.

[6] Alexander, M., et al.; Landing Phase Analysis of the NRC Bell 412 ASRA in Supervised Autonomous Flight Systems Configuration; 79th Annual Forum - Vertical Flight Society; 2023.

[7] Alexander, M. et al.; A Systems Engineering Approach for Enabling Research and Development in the Vertical Lift Autonomy Flight Sciences; Vertical Flight Society 76th Annual Forum; 2020.

[8] Alexander, M. et al.; Landing Phase Analysis of the NRC Bell 412 Advanced Systems Research Aircraft in Supervised Autonomous Flight Systems Configuration; Vertical Flight Society 79th Annual Forum; 2023.

[9] Alexander, M. et al.; NRC Research in Autonomy Flight Sciences: Experimentation in Rotorcraft Landing Methodology in Degraded Visual Environments; Vertical Flight Society (VFS) Transformative Vertical Flight - Aeromechanics for Advanced Vertical Flight Meeting; 2022.

[10] Smith, R.C.; Uncertainty Quantification: Theory, Implementation, Applications 2nd Ed.; SIAM; 2013.

[11] SAE International, ARP4761A: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, Warrendale, PA, USA, Dec. 2023.

[12] RTCA, DO-178C: Software Considerations Airborne Systems Equipment Certification, USA, 2011.

[13] Federal Aviation Administration, 14 CFR Part 27: Airworthiness Standards—Normal Category Rotorcraft, Washington, DC, USA, current edition.

[14] Federal Aviation Administration, 14 CFR Part 29: Airworthiness Standards—Transport Category Rotorcraft, Washington, DC, USA, current edition.

[15] European Union Aviation Safety Agency, Certification Specifications and Acceptable Means of Compliance for Small Rotorcraft (CS-27), Cologne, Germany, current edition.

[16] European Union Aviation Safety Agency, Certification Specifications and Acceptable Means of Compliance for Large Rotorcraft (CS-29), Cologne, Germany, current edition.

[17] Federal Aviation Administration, Advisory Circular AC 29-2C, Certification of Transport Category Rotorcraft, Washington, DC, USA, Federal Aviation Administration, 2022.

[18] RTCA, DO-326A: Airworthiness Security Process Specification, USA, Dec. 2014.

[19] U.S. Army Aviation and Missile Command, Aeronautical Design Standard ADS-33E-PRF: Handling Qualities Requirements for Military Rotorcraft, Redstone Arsenal, AL, USA, Mar. 2000.

[20] SAE International, ARP4754A: Guidelines for Development of Civil Aircraft & Systems, USA, 2010.

[21] Federal Aviation Administration, Advisory Circular AC 20-174: Development of Civil Aircraft and Systems, Washington, DC, USA, Nov. 2011.

[22] U.S. Department of Defense, MIL-STD-882E: Department of Defense Standard Practice—System Safety, Washington, DC, USA, May 2012.

[23] RTCA, DO-326A: Airworthiness Security Process Specification, USA, Dec. 2014.

Copyright Statement

National Research Council of Canada. Copyright © 2026. All rights reserved. The author confirms NRC Canada holds copyright on all original material, permission for publication of any third-party material from copyright holders, and confirms NRC Canada gives permission for publication and distribution as part of ICAS proceedings and associated individual off-prints.

Author Biography



Marc Alexander is a Senior Canadian Government Flight Scientist specializing in vertical lift flight sciences research and testing. He is employed by the National Research Council of Canada's Flight Research Laboratory (NRC-FRL) and tasked by its Rotary Wing and Flight Controls Group (Ottawa, Canada). He possesses previous experience with the Royal Canadian Air Force (Department of Technical Airworthiness and Engineering Support, Ottawa) and Aeromechanical Systems Department (United Technologies Research Center, Hartford).

Over Marc's 15-year career as a specialist in applied flight sciences, he developed diverse experimental flight test expertise in areas of controls, dynamics, flying/handling qualities, and aeromechanics. He applies NRC's variable stability fly-by-wire helicopters in flight research involving autonomous systems development and testing, handling qualities optimization, productivity and safety enhancement, vehicle design, and aircrew education.

Possessing a Canadian Graduate education in Aerospace Engineering from Ryerson University and University of Toronto Institute for Aerospace Studies, he is active in the vertical lift community through international cooperation (North Atlantic Treaty Organization (NATO) Air Vehicle Technology (AVT); The Technical Cooperation Program (TTCP)) and professional organizations (Vertical Flight Society (VFS); Society of Flight Test Engineers (SFTE); Canadian Aeronautics and Space Institute (CASI)).

Author Contact Information

Please find contact information for your convenience: Marc.Alexander@nrc-cnrc.gc.ca