Vision-Aided, GPS-Resilient Autoland for Small Fixed-Wing UAS

a report presented to
The Faculty of the Department of Aerospace Engineering
San José State University

in partial fulfillment of the requirements for the degree
Master of Science in Aerospace Engineering

by

Alejandro Nava

August 2026

Approved by

Prof. Maria Chierichetti
Faculty Advisor



© 2026
Alejandro Nava
ALL RIGHTS RESERVED

ABSTRACT

Vision-Aided, GPS-Resilient Autoland for Small Fixed-Wing UAS

Alejandro Nava

This project addresses a practical problem in small-UAS operations: completing approach and landing when GNSS is degraded, unavailable, or intentionally denied. A vision-aided, GPS-resilient autoland architecture was developed for a 2.0 m E-flite Turbo Timber SWS fixed-wing aircraft using a Cube Orange+ autopilot, Jetson Orin Nano companion computer, and Intel RealSense D435i. Stage 1 Software-in-the-Loop testing first checked the EKF3 estimator interface, source-set switching, and AUTO/LAND behavior with a truth-based vision surrogate. Three clean SRC2 ExternalNav runs completed GPS-denied autoland with a mean absolute lateral touchdown error of 0.14 m and a maximum of 0.27 m. Under the same no-fix condition, a GPS-only SRC1 baseline entered dead reckoning and touched down 10.57 m from the centerline. Hardware-in-the-Loop and visual-in-the-loop tests then exercised the RealSense-to-Jetson-to-Cube data path, timing, and estimator compatibility. Six supervised flight approaches were conducted at Simi Valley Flyers, Stullick Memorial Field. Each pass began on SRC1/GPS and switched to SRC2/ExternalNav at the planned final-approach gate. At that gate, MAVLink parameter updates reduced the satellite count reported to the autopilot, creating a controlled GPS-degraded condition without RF jamming or physical receiver isolation. The Jetson VPE/VSE stream supplied the EKF3 navigation state used by ArduPlane guidance, while ArduPlane retained the inner-loop, LAND, and flare functions. All six passes met the 95% landing-window duty-cycle threshold, the 100 ms 95th-percentile latency budget, the EKF3 rejection limit of 1.0, and the ±2 m lateral by ±5 m longitudinal touchdown box. One telemetry-congestion pass contained a sustained interval below 15 Hz even though its pass-level mean VSE rate was 15.5 Hz. The flight

tests demonstrated supervised source switching and perception-driven ExternalNav approach control under controlled GPS degradation. Because the receiver remained connected and the GPS input was degraded rather than physically unavailable, these flights are not reported as strict GPS-denied operation. Strict GPS-unavailable flight testing remains future work.

ACKNOWLEDGEMENTS

I would like to thank my family for the many sacrifices they have made throughout my life; their support made this work, and everything before it, possible. I would also like to thank my advisor, Dr. Maria Chierichetti, for her guidance and support throughout this project.

Table of Contents

List of Figures 7
List of Tables 9
Symbols 10
1. Introduction 1
1.1 Motivation 1
1.2 Literature Review 2
1.3 Project Proposal 8
1.4 Methodology 8
Chapter 2: Vision Perception 12
2.1 Sensors, calibration, and preprocessing 12
2.2 Data sources and label generation 13
2.3 Models and training curriculum 14
2.4 Runtime perception and fusion 15
2.5 Compute budget and deployment 17
2.6 Interfaces to guidance and SITL/HITL 17
2.7 Evaluation protocol and results 18
2.8 Failure modes and mitigations 19
2.9 GNSS-Denied Geolocalization 20
Chapter 3: SITL Testing 25
3.1 Test Architecture and Configuration 25
3.2 Test Scenarios and Run Matrix 26
3.3 Metrics and Data Collection 27
3.4 SITL Results 28
3.5 Discussion and Limitations 33
3.6 Transition to HITL and Perception Testing 33
Chapter 4: Hardware-in-the-Loop and Visual-in-the-Loop Testing 34
4.1 Test Objective 34
4.2 Bench Architecture 34
4.3 Visual-in-the-Loop Camera Pipeline 36
4.4 Run Matrix and Metrics 37
4.5 HITL Bench Results 37
4.6 Discussion 41
4.7 Flight-Test Readiness Gate 41

Chapter 5: Aircraft Build and Systems Integration 43
5.1 Airframe Overview 43
5.2 Avionics Bay Layout 43
5.3 Mass Properties and Center of Gravity 44
5.4 Camera Installation 45
5.5 Thermal Management 47
5.6 Integration Status 47
Chapter 6: Flight-Test Results 48
6.1 Test Site, Flight Plan, and Scoring Window 49
6.2 Perception Availability 50
6.3 ExternalNav Rate and Latency 50
6.4 Centerline Stability 51
6.5 EKF3 Consistency During Source-Switched ExternalNav 52
6.6 Touchdown Dispersion 53
6.7 Aggregate Results 54
6.8 Flight-Test Procedures and Safety Monitoring 55
6.9 Limiter Lua Script and Runtime Assurance 57
Chapter 7: Conclusions and Future Work 59
7.1 Summary of Findings 59
7.2 Technical Contributions 60
7.3 Limitations 62
7.4 Future Work 63
References 66

List of Figures

Figure 1.1 - TECS architecture and signal flow [2] 3
Figure 1.2 - Schematic diagram of YOLO-RWY [7] 4
Figure 1.3 - Monte-Carlo geolocalization framework [9] 6
Figure 1.4 – Approach and glidepath geometry 9
Figure 1.5 - System architecture and open-loop signal flow 10
Figure 1.6 - Closed-loop verification 11
Figure 2.1 – Coordinate frames 13
Figure 2.2 – Centerline geometry from mask 16
Figure 2.3 – Runway-frame PnP geometry 21
Figure 2.4 – Runtime perception decision flow 23
Figure 2.5 – Geolocalization fusion / Source selection 23
Figure 3.1 - Source switch and GPS-denial timeline 28
Figure 3.2 - Run 1 EKF3 innovation test ratios 29
Figure 3.3 - Run 1 ground track at PHNL 30
Figure 3.4 - Touchdown dispersion for Runs 1-4 31
Figure 3.5 - Touchdown lateral error for Runs 1-4 32
Figure 4.1 - HITL bench layout 35
Figure 4.2 - RealSense runway segmentation under three visual conditions 36
Figure 4.3 - ExternalNav rate and capture-to-publish latency 38
Figure 4.4 - Perception duty cycle and dropout by visual condition 38
Figure 4.5 - Centerline error and heading over the final-approach window 39
Figure 4.6 - EKF3 consistency and shadow-mode source state 40
Figure 4.7 - HITL landing-window benchmark summary 40
Figure 5.1 - Completed flight-test aircraft 43
Figure 5.2 - Avionics bay layout 44
Figure 5.3 - Center-of-gravity location 45
Figure 5.4 - RealSense D435i camera installation 46
Figure 5.5 - Camera ball-stud attachment 46
Figure 6.1 - Flight-test plan at Stullick Memorial Field 49
Figure 6.2 - Valid-frame duty cycle by approach pass 50
Figure 6.3 - FT-05 ExternalNav rate and capture-to-publish latency 51
Figure 6.4 - Centerline error for calm and gusting passes 52

Figure 6.5 - FT-03 EKF3 innovation ratios with SRC2/ExternalNav active 53
Figure 6.6 - Touchdown dispersion after GPS-degraded ExternalNav approaches 54
Figure 6.7 - GPS-degraded source-switched flight-test summary 55

List of Tables

Table 3.1 - Stage 1 SITL and EKF3 configuration....26
Table 3.2 - Stage 1 run matrix....26
Table 3.3 - SRC2 ExternalNav repeatability, Runs 1-3....30
Table 3.4 - SRC2 ExternalNav and SRC1 GPS-only baseline....32
Table 4.1 - HITL bench hardware roles....36
Table 4.2 - HITL and VIL run matrix....37
Table 4.3 - HITL final-approach metrics....41
Table 6.1 - Source-switched GPS-degraded flight-test summary....50

Symbols

Symbol	Definition	Units (SI)
A_{bbox}, A_{img}	Detector bounding-box area; full image area	px^2
c_x, c_y	Camera principal-point coordinates	px
d	Unit centerline direction from keypoints (Eq. 2.4); runway-plane offset (Eqs. 2.16–2.17)	–, m
duty	Valid-frame duty cycle (Eq. 2.10)	–
e_{px}	Signed image-plane cross-track error	px
f_x, f_y, f_{eff}	Camera focal lengths; effective focal length	px
H	Plane-induced homography matrix	–
K	Camera intrinsic matrix	–
k_{near}, k_{far}	Near and far runway centerline keypoints	px
L_t	Centerline endpoint pair at frame t	px
m^2	Squared Mahalanobis distance of a feature embedding (Eq. 2.13)	–
N	Number of mask points or samples	–
n	Runway-plane unit normal vector	–
P_i	Foreground pixel location (Eq. 2.1);	px, m

	3-D point in plane fit (Eqs. 2.16–2.17)	
p_0	Image aim/reference point	px
R	Camera rotation matrix	–
r_{plane}	Plane-fit RMS residual (Eq. 2.17)	m
rms_{reproj}	Corner reprojection RMS error (Eq. 2.15)	px
s_t	Smoothed heading unit vector at frame t (Eq. 2.5)	–
T	Number of frames in the evaluation window	–
t	Camera translation vector	m
v_k	Eigenvectors of the mask covariance (Eq. 2.1)	–
X_j	Runway-corner coordinates in the runway frame	m
x_j	Image projection of runway corner j	px
z	Feature-embedding vector	–

Greek Symbol	Definition	Units (SI)
α	Exponential-moving-average smoothing gain (Eqs. 2.5–2.6)	–

α_x, α_y	Bearing angles from pixel coordinates (Eq. 2.8)	rad
$\Delta\psi$	Wrap-corrected heading deviation (Eq. 2.12)	deg
δ_{aim}	Angular aim-point offset (Eq. 2.9)	rad
λ_k	Eigenvalues of the mask covariance (Eq. 2.1)	–
μ	Mask centroid (Eq. 2.1); embedding mean (Eq. 2.13)	px
$\pi(\cdot)$	Perspective projection function	–
Σ	Covariance matrix	–
τ_{area}	ROI / full-frame segmentation switching threshold (Eq. 2.7)	–
τ_{ood}	Out-of-distribution rejection threshold (Eq. 2.13)	–
ψ	Image-plane runway heading	deg

Subscript / Superscript	Meaning
$(\,)_0$	Reference or aim-point value
$(\,)_t$	Value at frame t

$(\,)_{img}$	Image-plane quantity
$(\,)_{near}$, $(\,)_{far}$	Near / far centerline keypoint
$(\,)^{\top}$	Vector or matrix transpose
$(\,)^{\star}$	Target pixel coordinate
$\hat{}$ (hat)	Temporally smoothed (filtered) estimate
$\bar{}$ (bar)	Mean value over the evaluation window

Acronym	Definition
AIAA	American Institute of Aeronautics and Astronautics
ALS	Automatic Landing System
AOA	Angle of Attack
CSV	Comma-Separated Values
EKF / EKF3	Extended Kalman Filter / ArduPilot third-generation EKF
EMA	Exponential Moving Average
ENU	East-North-Up local frame
EPnP	Efficient Perspective-n-Point
ESC	Electronic Speed Controller
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HITL	Hardware-in-the-Loop
ILS	Instrument Landing System

IMU	Inertial Measurement Unit
IR	Infrared
JSON	JavaScript Object Notation
L1	ArduPilot nonlinear lateral path-following guidance law
LARD	Landing Approach Runway Detection dataset
MAVLink	Micro Air Vehicle Link protocol
NED	North-East-Down local frame
ONNX	Open Neural Network Exchange
PCA	Principal Component Analysis
PHNL	ICAO identifier, Daniel K. Inouye International Airport (Honolulu)
PID	Proportional-Integral-Derivative
PnP	Perspective-n-Point
RANSAC	Random Sample Consensus
RC	Radio Control
RGB	Red-Green-Blue color image
RMS	Root Mean Square
ROI	Region of Interest
RTA	Runtime Assurance
SITL	Software-in-the-Loop
SLAM	Simultaneous Localization and Mapping

SRC1 / SRC2	ArduPilot EKF3 source set 1 (GPS) / source set 2 (ExternalNav)
SWS	Sport Wood Series (E-flite airframe designation)
TECS	Total Energy Control System
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
USB	Universal Serial Bus
VIL	Visual-in-the-Loop
VIO	Visual-Inertial Odometry
VPE / VSE	VISION_POSITION_ESTIMATE / VISION_SPEED_ESTIMATE MAVLink messages
YOLO	You Only Look Once detector family

1. Introduction

1.1 Motivation

GNSS outages, low-altitude turbulence, and visual clutter around a runway can all reduce the reliability of small-aircraft approach and landing. Jamming, spoofing, multipath, and signal occlusion make it useful to retain an independent navigation path during the terminal phase. VIO can provide a local state estimate when GNSS is unavailable, but its performance can degrade in low-texture scenes, changing light, or aggressive motion. For that reason, this project pairs VIO with runway-specific perception rather than relying on odometry alone. Recent deep-learning models can detect and segment runways across a range of backgrounds and viewpoints; models such as YOLO-RWY also provide threshold and centerline cues. On the control side, Total Energy Control System (TECS) concepts manage flight-path angle and airspeed while reducing pitch-throttle coupling. Classical Automatic Landing System (ALS) architectures divide the landing into capture, tracking, flare, and rollout phases. Software-in-the-Loop and Hardware-in-the-Loop (SITL/HITL) testing are used before flight to check the estimator, control-law integration, and closed-loop behavior. The resulting design combines runway perception, state estimation, and TECS-based guidance for GNSS-degraded approach and landing.

The main contributions of this project are (1) a runway-referenced guidance pipeline derived from onboard vision; (2) a method for supplying vision pose and VIO to the autopilot estimator as an alternative to GNSS; (3) a runtime-assurance monitor that supervises the perception and guidance path; and (4) a staged SITL, HITL, and flight-test verification program tied to quantitative touchdown targets.

1.2 Literature Review

Small UAS need a dependable landing capability when GNSS is intermittent or unavailable. Meeting that need requires perception, state estimation, guidance, and verification to be considered as one system. This review is organized around four areas:

- Classical autoland control laws and TECS
- Visual perception of runways/terrain and short-final guidance
- GNSS-denied state estimation and map-matching
- Evaluation SITL/HITL under runtime assurance and flight-test frameworks

Modern autoland architectures grew out of instrument landing systems (ILS), which use localizer and glideslope tracking followed by flare and rollout modes. Linear and nonlinear designs combine pitch attitude, flight-path angle, and radio-altimeter scheduling to transition from glidepath tracking to flare while meeting touchdown sink-rate limits [1]. Lambregts' TECS separates longitudinal control into total-energy and energy-balance loops, using throttle to regulate total-energy rate and elevator to control energy distribution [2]. Figure 1.1 summarizes the TECS architecture.

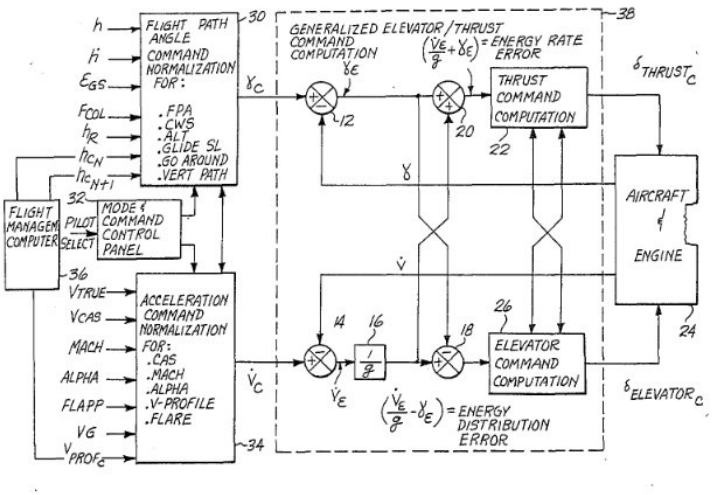


Figure 1.1 - TECS architecture and signal flow [2]

TECS remains widely used in both transport-aircraft and UAS applications. Flight tests on small UAVs have shown that relatively simple PI laws can regulate airspeed and flight-path angle through specific-energy rate and its allocation, with practical PID auto-tuning performed in flight [3]. Recent autoland studies also use robust H∞ methods to improve trajectory tracking and disturbance rejection in crosswind and gusts, particularly from flare through touchdown [4].

Localizer capture and tracking loops in transport aircraft rely on well-understood shaping of heading and track error. Small-UAS implementations often use nonlinear path-following laws together with altitude and energy management. On short final, lateral tracking and longitudinal flare logic must be coordinated so that crosswind decrab and glidepath tracking do not compete with elevator-throttle allocation. This favors architectures that separate energy control from directional control while retaining the necessary cross-coupling near the ground [1]. Robust nonlinear dynamic inversion has also been shown to improve landing stability under crosswinds and actuator limits, providing an inner loop around which higher-level approach and flare logic can be built [5].

Vision-based runway detection and segmentation have advanced rapidly with convolutional neural networks (CNNs) that process monocular RGB imagery. These networks can provide centerline and threshold cues as well as runway-extent masks for guidance. VALNet, for example, reports pixel-wise runway segmentation across multiple airports and lighting conditions, supporting late-stage alignment and height-above-runway estimation when combined with simple geometry [6]. Object detectors such as YOLO-RWY [7] trade some pixel-level detail for fast bounding-box detections of runway features, which is useful on embedded avionics. Figure 1.2 shows the YOLO-RWY architecture.

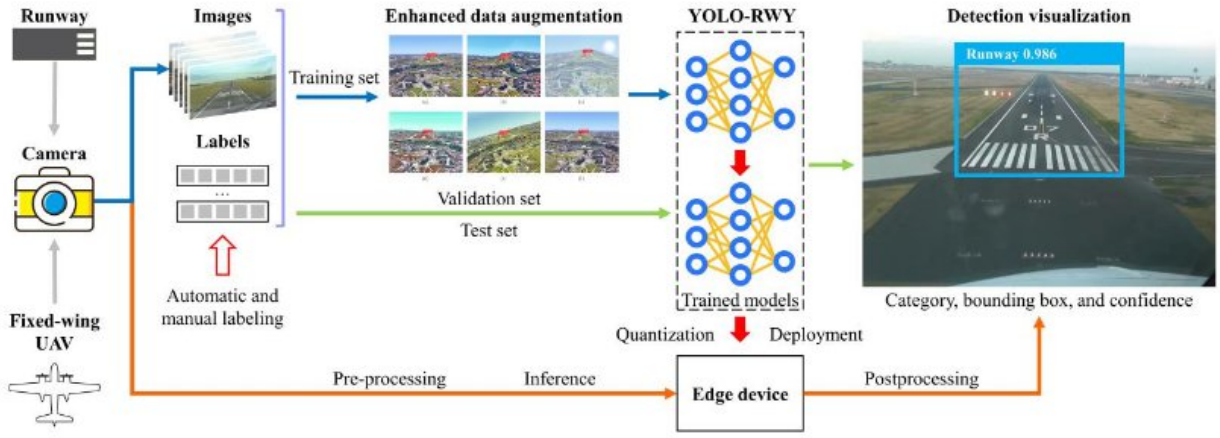


Figure 1.2 - Schematic diagram of YOLO-RWY [7]

Classical texture-based, feature-based, and stereo methods can also estimate range on final, although deep neural networks are now commonly used because they are more robust to changes in scene appearance.

There are two main perception methods that inform short-final controllers:

- Direct pose estimation of the vehicle relative to a runway-fixed frame
- Virtual sensor updates (pseudo-glideslope/localizer errors) that slot into existing autoland mode logic

Several studies combine these outputs in end-to-end landing pipelines, using visual cues to drive guidance and decrab while maintaining performance across changes in lighting, season, and viewing geometry.

When GNSS is lost, VIO can maintain a local state estimate, while image-to-map matching can recover absolute position over longer distances. Recent reviews compare feature-based, direct, and learning-based VIO and Simultaneous Localization and Mapping (SLAM) methods for real-time UAV operation [8]. Low-latency, high-dynamic-range cameras are particularly useful on final approach, where motion blur and glare can degrade conventional imagery. Kinnari et al. demonstrated GNSS-free global fixes by matching rectified onboard

images to orthophotos and fusing those matches with VIO-predicted motion, reaching meter-level two-dimensional accuracy after a modest flight distance [9]. Figure 1.3 illustrates that framework.

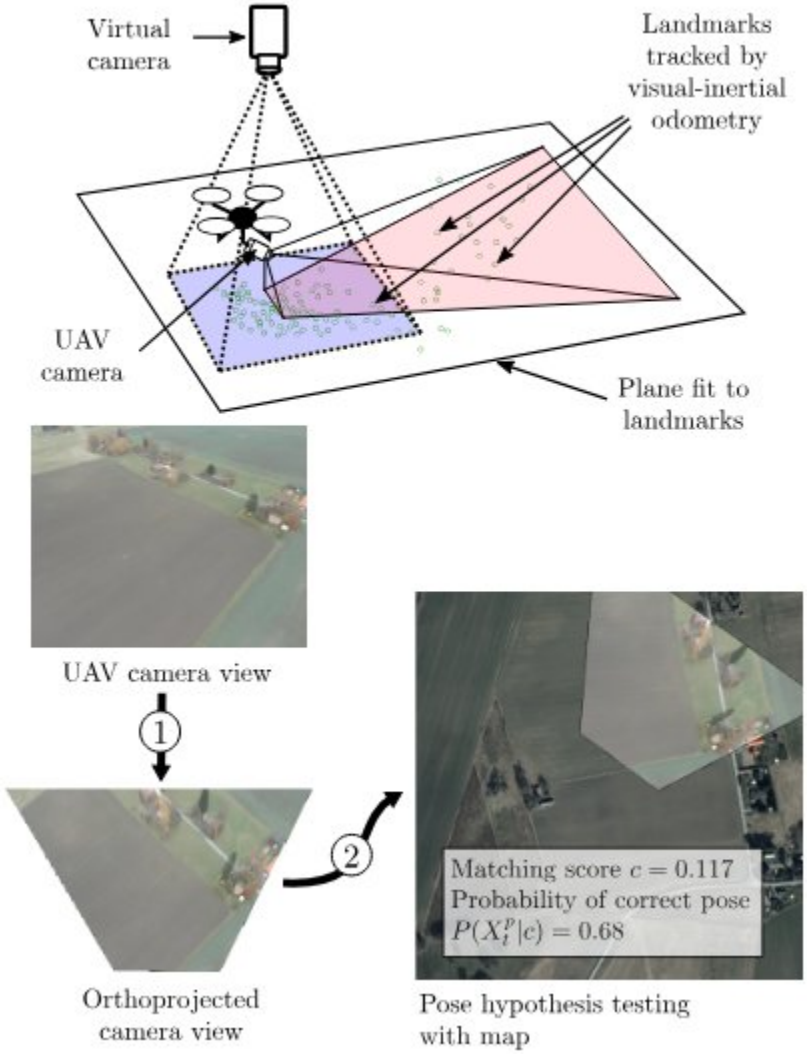


Figure 1.3 - Monte-Carlo geolocalization framework [9]

For small airframes with forward-facing cameras, this approach is useful because homography-based rectification relaxes the need for a strictly nadir camera [9]. More broadly, GNSS-denied navigation benefits from combining VIO with map- or semantics-based absolute updates so that local drift does not grow unchecked during longer missions [8].

NASA has examined probabilistic wing- and obstacle-collision risk models for low-altitude UAS, including the use of trajectory prediction and hazard maps in approach planning and monitoring. Studies of urban RF and communications fragility likewise emphasize link health and spectrum occupancy along the flight path, both of which matter during autonomous approach and landing. Together, this work supports monitoring environmental and link margins and triggering a go-around or mode change when those margins are exceeded [10].

For safety-critical autonomy, runtime-assurance (RTA) architectures place a trusted safety governor around a higher-performance but less-trusted controller and transfer control when risk limits are exceeded [11]. SITL and HITL workflows on representative autopilots such as ArduPilot allow controller and estimator tuning against physics-based aircraft models and real I/O before flight; recent studies have used such rigs to verify altitude hold and mode logic [12]. NASA technical reports also show how bench testing can be paired with field trials to evaluate wind, RF, and navigation hazards that affect approach integrity [13].

For this project, the literature led to an architecture in which TECS-style energy management governs the longitudinal channel, a robust path-following law handles lateral guidance, and runway perception supplies the geometry needed for short-final alignment and flare scheduling. VIO provides the local state, while image-to-map matching or runway-referenced pose supplies periodic absolute corrections to limit drift. Verification then progresses through SITL, HITL, and supervised flight testing with explicit limits on speed, sink rate, path error, terrain clearance, and system health.

1.3 Project Proposal

Based on this review, the project proposal called for a vision-aided, GNSS-resilient autoland system on a 2.0 m Turbo Timber SWS airframe. A Cube Orange+ would retain the flight-

control inner loops, while a Jetson Orin Nano and Intel RealSense D435i would provide VIO and runway perception. The proposed system was intended to:

- Sustain localization during deliberate GNSS degradations
- Track a 3-4° glidepath and execute flare using a TECS-style vertical controller with bounded-error lateral guidance and crosswind handling
- Employ a lightweight runtime-assurance monitor to command verified mitigation (LOITER/LAND/RTL) when safety margins erode

The verification plan progressed from SITL to HITL and then to flight testing. The original success criteria were an RMS vertical error of no more than 1.0 m, an RMS lateral error of no more than 1.5 m over the final 150–200 m, touchdown dispersion within ±10 m longitudinal and ±2 m lateral, a sink rate no greater than 0.6 m/s, and at least 95% correct hazard detection or mitigation during injected GNSS and link degradations.

1.4 Methodology

The methodology began by establishing a baseline guidance and control system in simulation. The airframe model, Cube Orange+ with ArduPilot inner loops, and mission geometry were exercised in SITL to produce reference approach and flare behavior under nominal GNSS conditions. That baseline defined the glidepath, flare gate, and control schedules used later in the project. A companion-computer stub then published MAVLink VISION_* and GPS_INPUT messages at flight rate with representative latency, allowing estimator fusion, mode transitions, and timing budgets to be checked before hardware was added.

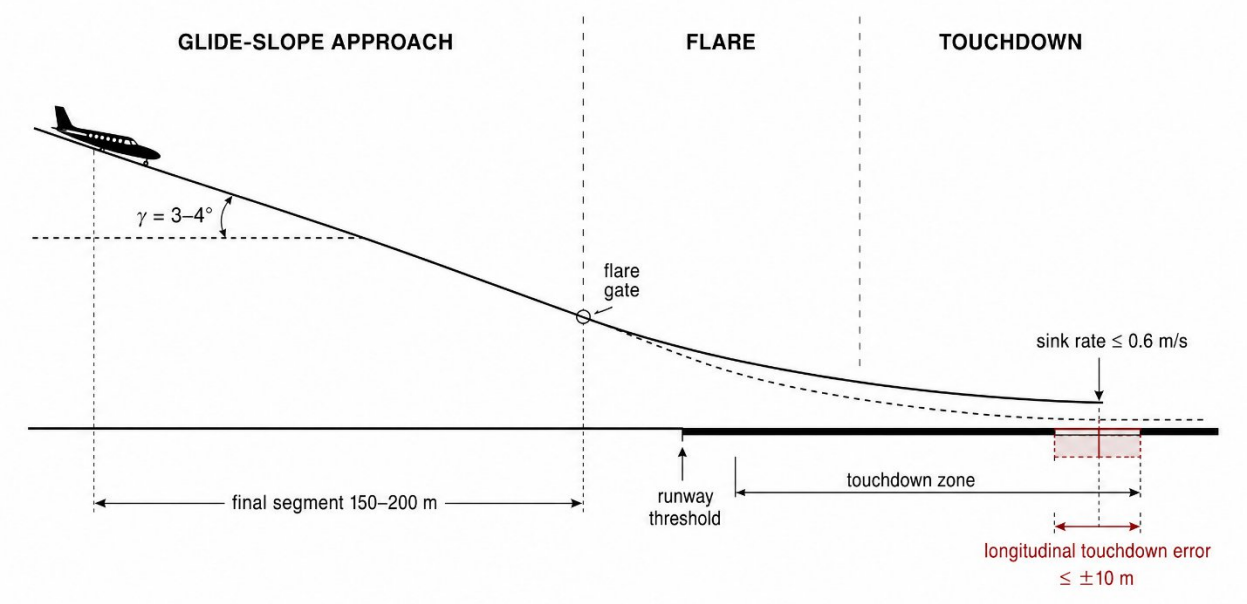


Figure 1.4 – Approach and glidepath geometry

The next step moved the perception and estimation pipeline to the Jetson Orin Nano and Intel RealSense D435i. Visual-inertial odometry supplied a continuous pose estimate, and a lightweight runway model supplied centerline and threshold cues for terminal alignment. These outputs were converted into virtual sensor measurements, including glidepath- and localizer-like errors or MAVLink GUIDED position and velocity setpoints, and then sent to the flight controller. This arrangement allowed short-final guidance to be tested both through measurements fused into the standard landing modes and through an external setpoint source while leaving the autopilot inner loop unchanged.

Longitudinal guidance followed TECS principles, while lateral guidance used a bounded-error look-ahead law. TECS parameters, including speed weight, throttle and pitch gains, and flare height and time, were tuned to regulate flight-path angle and airspeed without excessive coupling. The lateral look-ahead distance and damping were selected for predictable centerline capture on the Turbo Timber. The design also included crosswind decrab and pitch/AOA limits in the flare

schedule to manage touchdown dispersion and sink rate. Figure 1.5 summarizes the roles of perception, estimator fusion, guidance, and the autopilot.

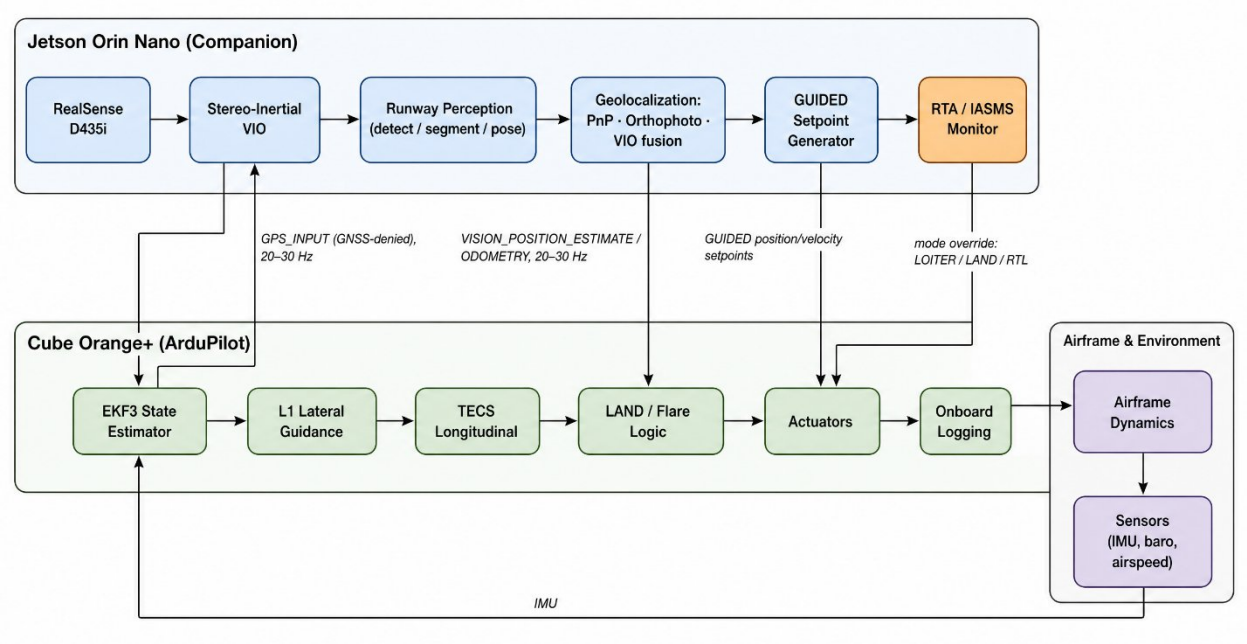


Figure 1.5 - System architecture and open-loop signal flow

Testing then progressed from open-loop checks to closed-loop evaluation. In SITL, wind, turbulence, and GNSS loss were introduced to measure estimator continuity, approach-path error, and flare timing with the companion computer in the loop. HITL replaced simulated timing with the actual Cube Orange+ I/O so that MAVLink messaging, EKF fusion of VIO, and LAND/GUIDED handoffs could be checked under representative latency. Figure 1.6 summarizes the verification flow, including the rate limits and watchdogs used to prevent command saturation.

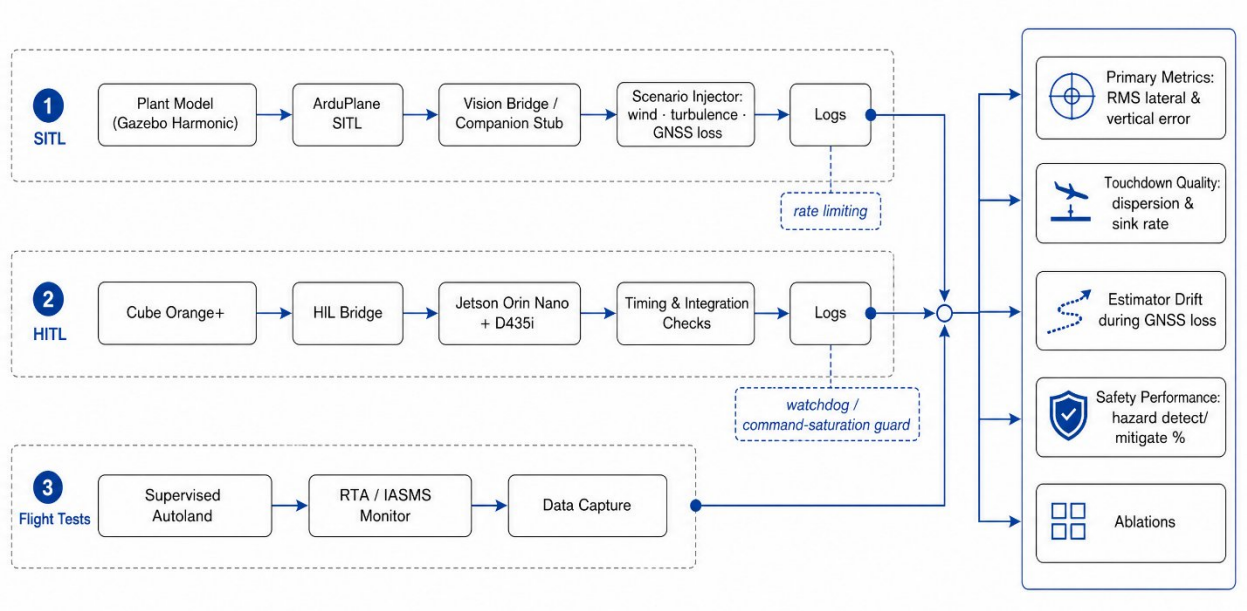


Figure 1.6 - Closed-loop verification

Flight testing progressed from supervised approaches toward autoland under deliberate GNSS degradation and controlled crosswind within the approved envelope. Each sortie recorded high-rate autopilot data such as attitude, TECS, lateral guidance, and EKF performance, along with companion-computer logs of VIO states, covariance, perception outputs, and setpoints. These records allowed estimator health to be compared directly with guidance performance after each flight. Evaluation focused on final-approach vertical and lateral error, touchdown dispersion, sink rate, and estimator drift during GNSS loss. The tuning plan covered lateral look-ahead and damping, TECS speed weight and flare gates, and perception and estimation confidence thresholds.

Chapter 2: Vision Perception

This chapter describes the real-time vision stack that converts camera imagery into runway-referenced measurements for approach and landing. The stack does not maintain the aircraft state; EKF3 does. Instead, it identifies the runway, produces a detector box and segmentation mask, estimates image-plane centerline heading and signed lateral error, and attaches confidence and quality flags. Runway pose and odometry products are sent to EKF3, while the centerline measurements provide alignment checks for L1 and TECS guidance on short final. The design emphasizes robustness to scale and viewpoint, stable frame-to-frame geometry, low latency on embedded hardware, and a clear interface to the autopilot. The signals were first evaluated on point-of-view landing videos and were then exercised in the SITL and HITL stages described in Chapters 3 and 4.

2.1 Sensors, calibration, and preprocessing

The primary sensor is the color camera on the Intel RealSense D435i. The lens is represented by a Brown-Conrady distortion model, and an offline checkerboard calibration estimates the intrinsics and checks alignment between the camera and the D435i IMU frames [14]. Frames are undistorted before inference so that labels and measurements share the same rectified image plane. At runtime, the intrinsics convert lateral pixel error into a bearing toward the aim point. Combined with range from the runway-frame PnP solution or stereo depth, that bearing gives the metric lateral offset used by the guidance interface in Section 2.6. Exposure and gain are fixed during night runs to prevent frame-to-frame brightness changes from destabilizing the mask. Each frame is letterboxed to the network input size and normalized. Training and validation used a 1280-pixel input. Runtime tests used 1024–1280 pixels on workstation hardware and 960–1024 pixels on the Jetson Orin Nano, trading a small loss in spatial resolution for higher throughput.

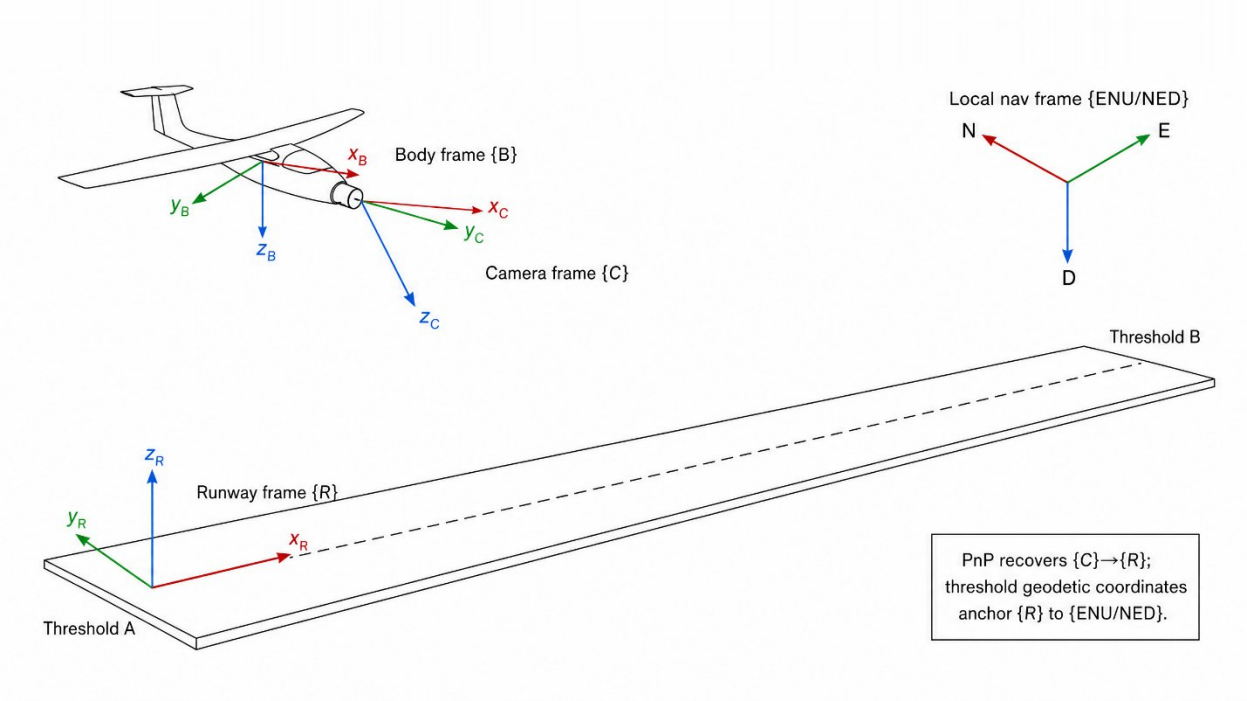


Figure 2.1 – Coordinate frames

2.2 Data sources and label generation

Training data is drawn from the LARD packs, which provide, for each image, the four runway corners along with width, height, and optional metadata such as time-to-landing and yaw. A Python script reads the relevant CSV files inside each pack directory, normalizes all geometric quantities to the range [0,1], and emits task-specific labels for detection, segmentation, and pose. Detection labels are tight (x, y, w, h) boxes derived from the corners; segmentation labels are polygons that preserve the ordered corners; and pose-2 labels record the near and far centerline points with visibility flags set to indicate availability. The script also writes task-ready "dataset.yaml" files, so the datasets are immediately consumable by YOLO training commands.

Because the "time_to_landing" field is missing from most images in the dataset, proximity-based curation relies instead on the normalized runway area computed from the annotated corners. A filter expressed as a minimum bounding-box-area fraction selects close-in views without

depending on time metadata, which makes curation robust across all packs. Additional filters target night images when a day/night indicator is provided and emphasize oblique crosswind views via a minimum absolute yaw threshold. When corners or polygons are missing, the script generates empty label files so backgrounds are naturally incorporated as negatives during training.

Because LARD contains relatively few night runway images, it was supplemented with a public night-runway data set derived from Microsoft Flight Simulator [15]. The pipeline also remains compatible with dual-modal RGB and IR extensions that have shown promise at night [16]. A companion converter reads the supplemental JSON labels, writes segmentation polygons, and derives pseudo pose-2 labels from the principal axis of each polygon mask. It uses the same normalized geometry as the LARD pipeline, allowing the data sets to be combined during fine-tuning.

2.3 Models and training curriculum

The perception pipeline uses Ultralytics YOLO models for detection, segmentation, and pose. The staged detector-to-segmentation design was selected because published runway-detection work reports suitable edge-device throughput, while VALNet supports instance segmentation for runway geometry [6], [7]. A single-class detector is fine-tuned first on synthetic and domain-adaptation imagery and then on real-airport images with abundant negatives. Instance segmentation is trained on LARD polygons and later adapted to night scenes with augmentations for glare and low contrast. A two-keypoint pose head provides near and far centerline anchors that stabilize the line at range. An optional four-corner pose head adds a homography-based midline as a third fallback, and an EPnP solution provides a cross-check when all four corners are available [17].

2.4 Runtime perception and fusion

At runtime, each frame first passes through the detector, which returns one runway bounding box and a confidence score. If confidence falls below the threshold, the system briefly holds the last valid estimate with exponential decay and marks the new frame invalid. Otherwise, it enlarges the region of interest by a fixed fraction to stabilize downstream geometry. Segmentation runs inside that region unless the region already covers most of the image, in which case the full frame is used. When a pose-2 model is available, it runs on the same region. Valid near and far keypoints define the centerline directly and take precedence; if either keypoint is missing or fails the quality checks, the pipeline falls back to the segmentation-based estimate.

The centerline is obtained by principal-component analysis of the foreground points, with the major eigenvector indicating the runway axis (Eq. 2.1). The corresponding image-plane heading is recovered from the major eigenvector components (Eq. 2.2), and the orthogonal direction defines lateral displacement: the signed cross-track in pixels is the inner product between the vector from the image center (or aim point) to the mask mean and that orthogonal direction (Eq. 2.3).

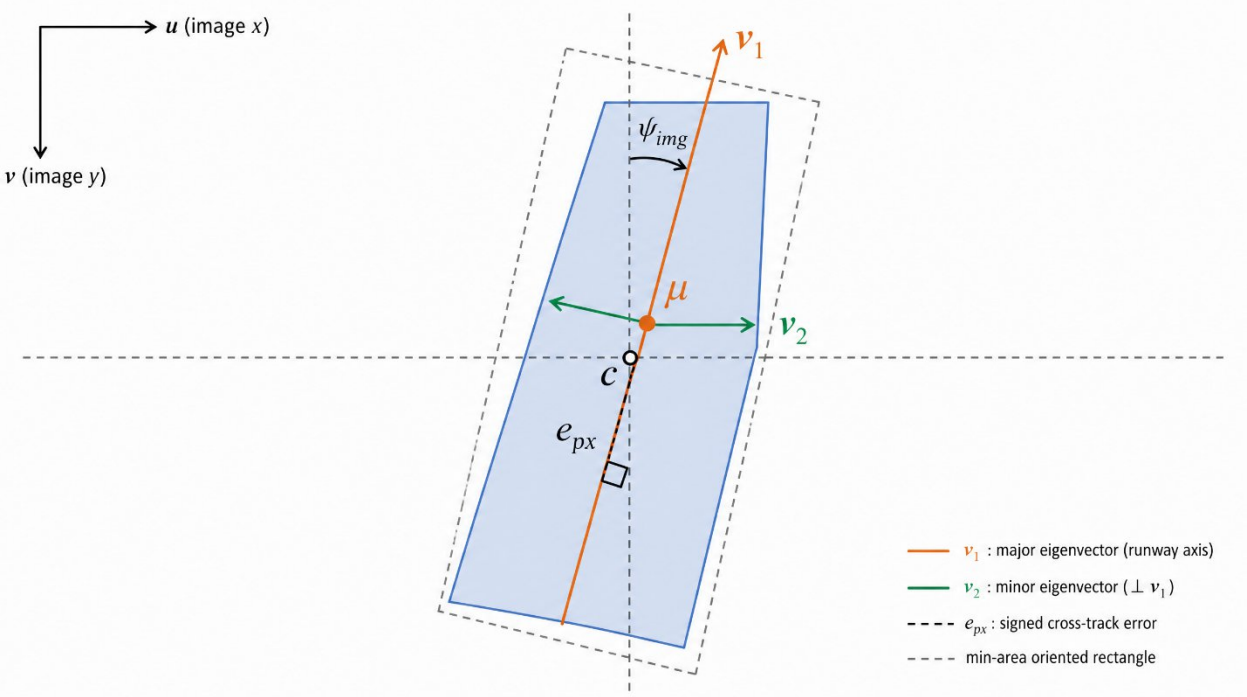


Figure 2.2 – Centerline geometry from mask

A minimum-area oriented rectangle provides a fast, independent estimate of heading and a midline point that is especially useful in close-in views. When all four corners are available, a homography- or EPnP-based pose gives an additional heading check under strong perspective [17].

Mask to axis, heading, and cross-track (PCA):

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} x_i \\ y_i \end{bmatrix}, \qquad \Sigma = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{P}_i - \mu)(\mathbf{P}_i - \mu)^{\top}, \qquad \Sigma\,\mathbf{v}_k = \lambda_k \mathbf{v}_k, \quad \lambda_1 \geq \lambda_2 \tag{2.1}$$

$$\psi_{\text{img}} = \operatorname{atan2}\left(v_{1,y}, v_{1,x}\right) \quad (\text{deg}) \tag{2.2}$$

$$e_{\text{px}} = (\mathbf{p}_0 - \mu)^{\top}\mathbf{v}_2 \quad \text{with } \mathbf{v}_2 \perp \mathbf{v}_1 \tag{2.3}$$

Pose-2 line and angle (stabilizer) with circular EMA:

$$\mathbf{d} = \frac{\mathbf{k}_{\text{far}} - \mathbf{k}_{\text{near}}}{\|\mathbf{k}_{\text{far}} - \mathbf{k}_{\text{near}}\|}, \qquad \psi_{\text{kp}} = \operatorname{atan2}\left(d_y, d_x\right) \tag{2.4}$$

$$\mathbf{s}_t = \alpha \begin{bmatrix} \cos\psi_{\text{inst}} \\ \sin\psi_{\text{inst}} \end{bmatrix} + (1-\alpha)\ \mathbf{s}_{t-1}, \qquad \widehat{\psi}_t = \operatorname{atan2}\left(s_{t,y}, s_{t,x}\right) \tag{2.5}$$

Temporal smoothing on endpoints/line:

$$\widehat{\mathbf{L}}_t = \alpha\, \mathbf{L}_t + (1-\alpha)\, \widehat{\mathbf{L}}_{t-1}, \qquad \alpha \in [0.25, 0.40] \tag{2.6}$$

ROI vs full-frame switch (near-touchdown):

$$\frac{A_{\text{bbox}}}{A_{\text{img}}} \gtrless \tau_{\text{area}} \quad \Rightarrow \quad \begin{cases} \text{ROI segmentation}, & \text{if } \frac{A_{\text{bbox}}}{A_{\text{img}}} < \tau_{\text{area}}, \\ \text{Full - frame segmentation}, & \text{if } \frac{A_{\text{bbox}}}{A_{\text{img}}} \geq \tau_{\text{area}}. \end{cases} \tag{2.7}$$

Temporal smoothing is applied to the centerline endpoints and heading with an exponential moving average gain between 0.25 and 0.40. This reduces residual jitter without making the signal too slow to respond.

2.5 Compute budget and deployment

The pipeline supports 1280-pixel inputs with mixed-precision training and inference. On the Jetson Orin Nano, segmentation is the main runtime cost. At 960–1024 pixels, the detector-plus-segmentation path fits within a 33–50 ms frame budget, leaving time for logging and safety checks and remaining consistent with published edge-deployment results for related runway detectors [7]. Geometry, smoothing, and safety checks add little overhead. Models can be exported to ONNX for portability and to TensorRT when the deployment environment supports it.

2.6 Interfaces to guidance and SITL/HITL

The perception stack publishes instantaneous and smoothed heading and lateral error together with confidence and QA flags. On fixed-wing ArduPlane, LANDING_TARGET does not drive lateral guidance; it is used by multirotor precision-landing controllers and is not consumed by the fixed-wing path follower. The runway-referenced pose from Section 2.9 is therefore published as VISION_POSITION_ESTIMATE and ODOMETRY for EKF3 fusion. The aircraft then follows a runway-referenced waypoint in GUIDED or AUTO, with L1 controlling lateral tracking and

TECS controlling the energy channel. In this arrangement, image-plane heading and cross-track affect guidance through the state estimate rather than through a direct steering command, preserving the autopilot inner loop. Equations 2.8 and 2.9 convert lateral pixel error into a bearing offset toward the aim point. Combined with range from the runway-frame PnP solution or stereo depth, that bearing provides a lateral-position correction and an independent check against the EKF3 lateral state.

$$\alpha_x = \frac{x_\star - c_x}{f_x}, \qquad \alpha_y = \frac{y_\star - c_y}{f_y} \tag{2.8}$$

$$\delta_{\text{aim}} \approx \frac{e_{\text{px}}}{f_{\text{eff}}} \tag{2.9}$$

For SITL, the bridge runs in-process and publishes pose and setpoint streams at 20 Hz. In HITL, the same bridge is routed to the Cube Orange+ over a telemetry link while a ground station monitors the stream, following standard fixed-wing hardware-in-the-loop practice [12]. Using the same interface in both environments keeps the lateral correction path consistent and allows controller gains to be adjusted without changing the vision code.

2.7 Evaluation protocol and results

Supervised training is evaluated using conventional dataset metrics, such as average precision for detection and mean intersection-over-union for segmentation. For closed-loop suitability, however, the emphasis is on signal stability measured on landing-video examples and simulator replays. The principal metrics are the duty cycle, defined as the fraction of frames that deliver a valid centerline (Eq. 2.10); the root-mean-square of the lateral pixel error and of the heading jitter over short windows of 1–2 seconds (Eq. 2.11); and the per-frame heading deviation from the windowed mean, computed circularly to avoid wraparound (Eq. 2.12).

$$\text{duty} = \frac{\#\{\text{valid frames}\}}{\#\{\text{total frames}\}} \tag{2.10}$$

$$\mathrm{RMS}\left(e_{\mathrm{px}}\right)=\sqrt{\frac{1}{T}\sum_{t=1}^{T}e_{\mathrm{px}}(t)^{2}},\qquad \mathrm{RMS}\left(\psi\right)=\sqrt{\frac{1}{T}\sum_{t=1}^{T}\Delta\psi(t)^{2}} \tag{2.11}$$

$$\Delta\psi(t)=\mathrm{atan2}\left(\sin\left(\psi_t-\bar{\psi}\right),\cos\left(\psi_t-\bar{\psi}\right)\right) \tag{2.12}$$

Camera-capture-to-publish latency is measured on the target hardware. Agreement among the independent geometry estimates such as principal-component axis, oriented rectangle, pose-2 line fit, and homography when available which is also tracked as a consistency check.

2.8 Failure modes and mitigations

Under low texture, strong glare, or snow, the segmentation mask can fragment. Increasing the smoothing gain, preferring the pose-2 line fit when available, and enabling an EPnP cross-check when corners are reliable reduce flicker. Very small far-range targets benefit from running segmentation inside a detector ROI at a higher input size. Conversely, very close views near touchdown are best handled by switching to full-frame segmentation and using shape-based quality checks. Night and low-visibility conditions are improved with night fine-tune training and stronger photometric augmentation. The modular design also permits future fusion with an infrared stream, as demonstrated in recent RGB–IR runway detection work [16]. Background false positives are controlled by keeping negatives abundant during training and by periodic hard-negative refreshes. Out-of-distribution scenes are handled by a safety gate that blocks guidance when confidence is insufficient, providing a practical RTA-style guardrail for the autoland stack [10], [11]. The gate is implemented as a Mahalanobis-distance test over a feature embedding (Eq. 2.13), with frames exceeding the calibrated threshold τ_ood rejected.

$$m^{2}=\left(\mathbf{z}-\mu\right)^{\top}\Sigma^{-1}\left(\mathbf{z}-\mu\right)\ \gtrless\ \tau_{\mathrm{ood}} \tag{2.13}$$

2.9 GNSS-Denied Geolocalization

For GNSS-unavailable operation, the perception stack uses a practical geolocalization path that combines runway-anchored PnP, orthophoto matching, and a stereo-inertial bridge. This follows both classical image-to-map localization and current GNSS-denied navigation practice, in which local odometry is corrected by occasional absolute map fixes [8], [9]. The three sources run opportunistically, and an estimator selects and fuses them using quality gates and hysteresis.

When the runway is visible and reliable corners are available, a 3-D model of the runway is instantiated in a runway frame whose origin is at threshold A, +x along the centerline toward threshold B, +y to the left, and z up. With known width and length from an airport database, four coplanar object points are paired with their image corners and solved via EPnP + RANSAC to recover the camera pose in the runway frame [17]:

$$s_j \begin{bmatrix} u_j \\ v_j \\ 1 \end{bmatrix} = \mathbf{K}\,[\,\mathbf{R}\;\;\mathbf{t}\,] \begin{bmatrix} \mathbf{X}_j \\ 1 \end{bmatrix}, \qquad j = 1,\ldots,4 \tag{2.14}$$

$$\mathrm{rms}_{\mathrm{reproj}} = \sqrt{\frac{1}{N}\sum_{j=1}^{N} \| \mathbf{x}_j - \pi\left(\mathbf{K}\,[\,\mathbf{R}\;\;\mathbf{t}\,]\,\mathbf{X}_j\right) \|^2} \tag{2.15}$$

Using the threshold geodetic coordinates, the recovered pose is converted to local ENU/NED coordinates and published as VISION_POSITION_ESTIMATE or ODOMETRY through MAVLink at 20–30 Hz. EKF3 fuses the update much like a GNSS position source. For GNSS-denied operation, horizontal position and velocity are assigned to the external-navigation source, with GNSS used again only after a fix passes the quality gate. The measured capture-to-publish delay is supplied through the vision-pose delay parameter so that EKF3 fuses the update at the correct time. This timing correction is important during a 20–30 m/s approach, where unmodeled latency can destabilize the estimate.

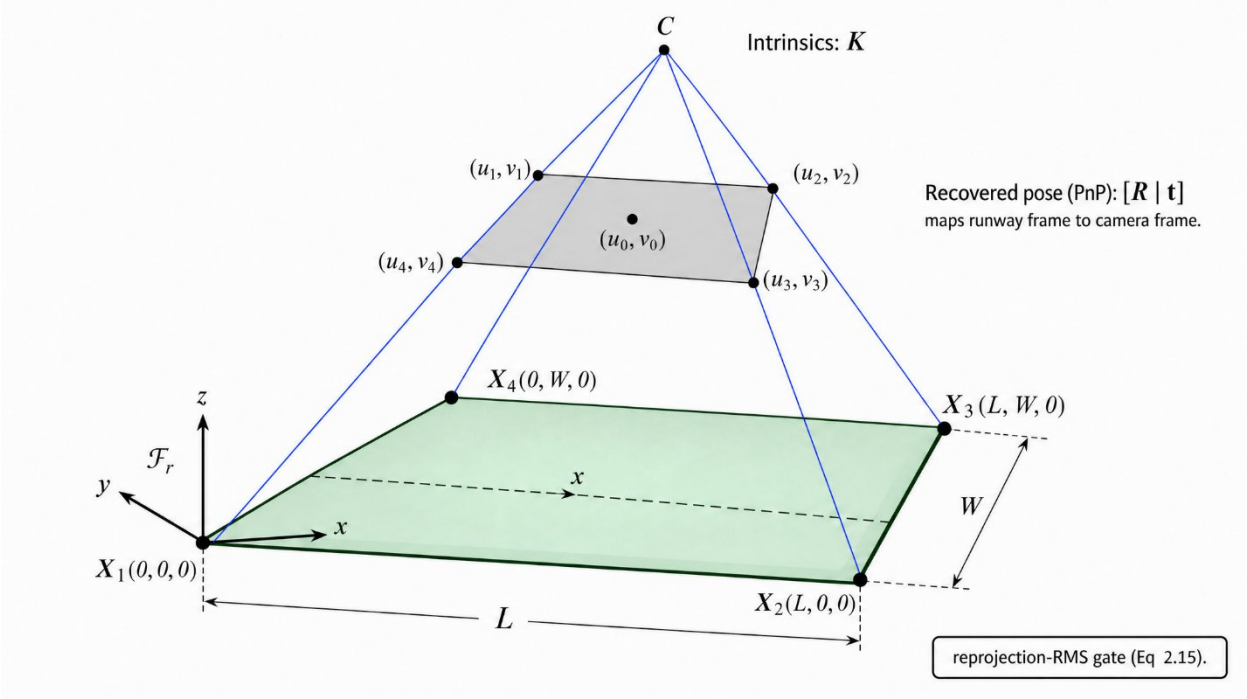


Figure 2.3 – Runway-frame PnP geometry

This method provides metric scale without GNSS because the runway dimensions are known from the airport database, and it is most reliable on short final. At long range, the four coplanar corners make the PnP solution ill-conditioned: a one- or two-pixel corner error can become a large along-track error. The solution is therefore gated by reprojection RMS and preferred only close enough to the runway for the corner geometry to be well conditioned. When available, a D435i depth-plane fit over the runway mask provides an additional planarity check and can reduce small scale or tilt errors [14]:

$$\min_{\mathbf{n},\, d} \sum_i \left(\mathbf{n}^\top \mathbf{P}_i + d\right)^2 \quad \text{s.t.} \quad \| \mathbf{n} \| = 1 \tag{2.16}$$

$$r_{\text{plane}} = \sqrt{\frac{1}{M} \sum_i \left(\mathbf{n}^\top \mathbf{P}_i + d\right)^2} \tag{2.17}$$

When the corners are unreliable, features inside the detector ROI can be matched against pre-tiled, georeferenced orthophotos of the airport. A RANSAC homography provides the plane-

induced projective alignment. Combined with the camera intrinsics, it gives pose up to an overall scale, which can be anchored with pressure altitude or near-field depth when available:

$$\mathbf{H} \sim \mathbf{K}\left(\mathbf{R} - \frac{1}{d}\,\mathbf{t}\,\mathbf{n}^{\top}\right)\mathbf{K}^{-1} \tag{2.18}$$

This orthophoto route supplies absolute fixes before PnP becomes stable and has been a mainstay of GNSS-denied geolocalization for UAVs [9].

A lightweight stereo-inertial VIO runs continuously on the D435i images and IMU data, providing dead reckoning between absolute updates. Whenever a PnP or orthophoto fix passes the quality gates, the estimator tightens or resets the drift state. This local-odometry-plus-global-correction structure is a common pattern in GNSS-denied flight systems [9].

PnP updates are accepted when the reprojection RMS is within 2–3 px, at least three corners are inliers, all depths are positive, and the recovered heading agrees with the image-plane heading from the vision pipeline. Orthophoto matches require 40–60 well-spread inliers and a low plane-fit residual. Selection prefers PnP, then orthophoto, then propagated VIO with increasing covariance. Runway identity is held constant to avoid oscillation between parallel runways. The same runtime-assurance logic used for perception gating applies here to prevent low-quality fixes from perturbing the EKF [10], [11].

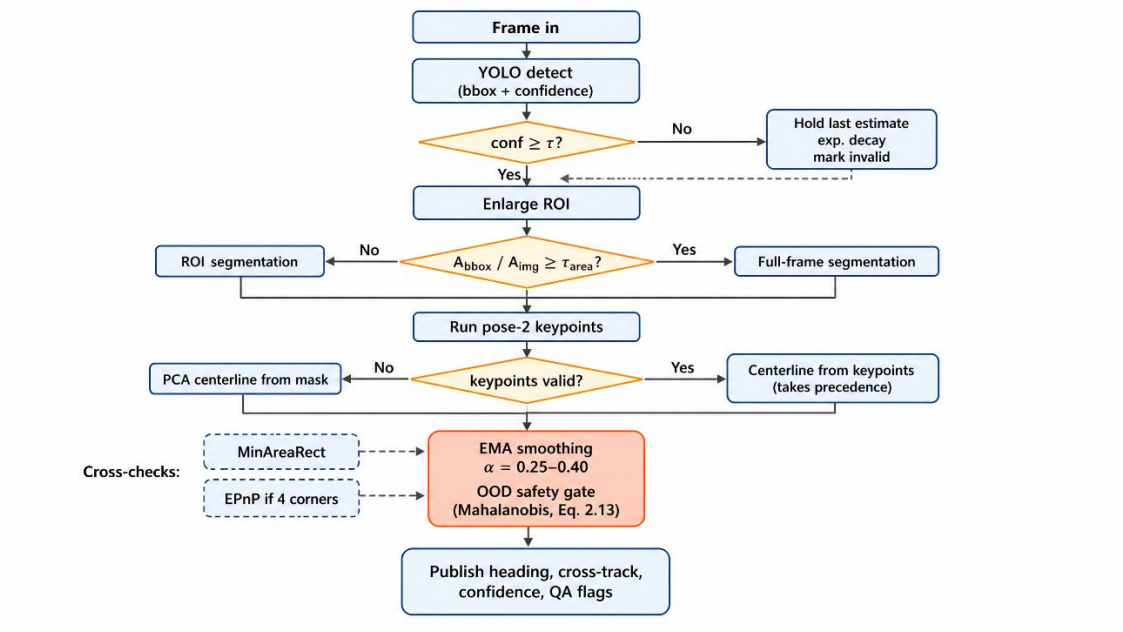


Figure 2.4 – Runtime perception decision flow

To improve PnP availability, a four-corner pose model is trained with the available night and close-in images. The added training cases target glare, low contrast, and near-touchdown geometry, where reliable corners and masks are hardest to obtain [7], [15], [16].

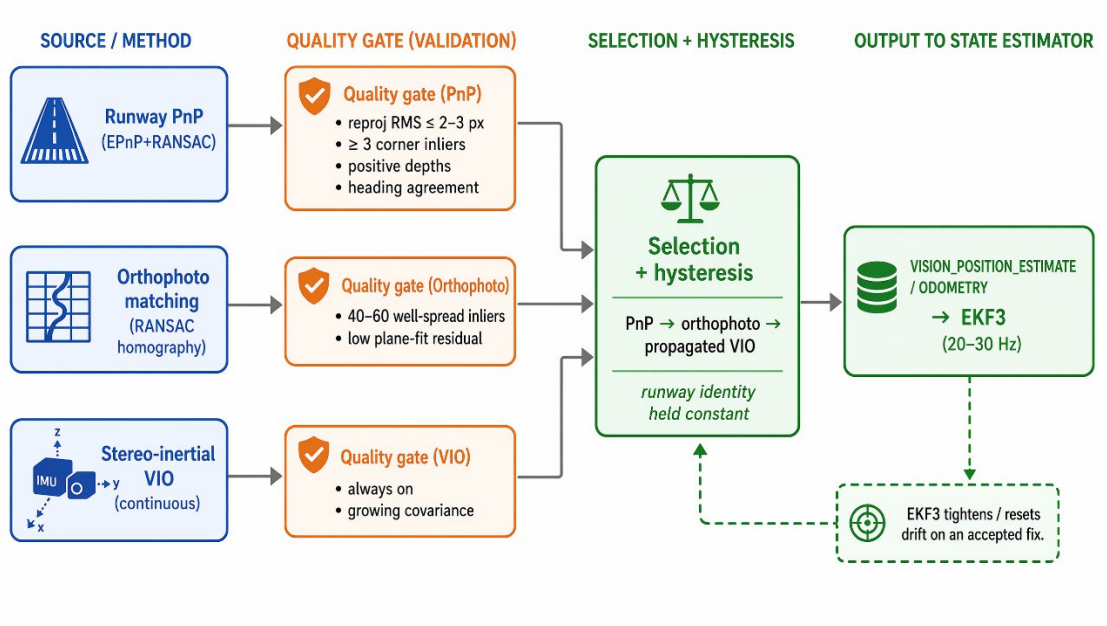


Figure 2.5 – Geolocalization fusion / Source selection

A lightweight threshold-and-marking detection head is a useful optional add-on that snaps image corners to painted bars when visible, which further tightens PnP conditioning. The orthophoto path benefits from a detector that narrows the search window and from image augmentations aligned with night imagery.

Chapter 3: SITL Testing

Software-in-the-Loop (SITL) testing was the first closed-loop verification stage. Stage 1 focused on the ArduPlane estimator and landing path: EKF3 source selection, ExternalNav fusion, source switching after GPS loss, and AUTO/LAND behavior. The trained runway-segmentation, centerline, PnP, orthophoto, and RealSense VIO stack from Chapter 2 was not used as the navigation source. Instead, a Python injector converted simulator truth into vision-style position and velocity updates. This stage isolated the ExternalNav fusion and autoland path; Chapter 4 then introduced the real perception and hardware chain.

Two safeguards prevented the injector from feeding EKF3 a self-referential solution. First, it read SIMSTATE truth rather than GLOBAL_POSITION_INT, which is the EKF output. Second, it referenced local North-East-Down (NED) measurements to the EKF/global origin so that switching from GPS to ExternalNav did not create a frame jump.

3.1 Test Architecture and Configuration

Testing used native ArduPlane SITL at PHNL with a Turbo Timber-style model and ArduPlane 4.8-dev. Stage 0 used X-Plane 12 and the normal GPS estimator to confirm the mission and LAND sequence. Stage 1 used native SITL, EKF3, and the MAVLink ExternalNav injector. The mission followed the PHNL runway 26L approach and used an AUTO mission with DO_LAND_START and NAV_LAND. Two EKF3 source sets were configured. SRC1 used GPS horizontal position and velocity, barometric height, GPS vertical velocity, and compass yaw. SRC2 used ExternalNav horizontal position and velocity, barometric height, ExternalNav vertical velocity, and compass yaw. RC9_OPTION = 90 assigned source selection to RC channel 9. Each clean SRC2 run established the approach on SRC1, switched RC9 to the middle position, confirmed SRC2, and then applied GPS denial.

Item	Value used in Stage 1 runs
Firmware	ArduPlane V4.8.0-dev
Location	PHNL runway 26L SITL start/origin: 21.3068, -157.9458502, 4.1 m
ExternalNav messages	VISION_POSITION_ESTIMATE and VISION_SPEED_ESTIMATE
Injector source	SIMSTATE truth plus Gaussian noise; GPS_RAW_INT used only to mark GPS loss
Nominal injector noise	sigma_xy = 0.5 m, sigma_z = 0.3 m, sigma_yaw = 1 deg
Vision delay setting	VISO_DELAY_MS = 0 for timestamp-on-reception MAVLink injection
GPS denial method	SIM_GPS1_FIXTYPE = 1 and SIM_GPS1_NUMSATS = 0
Source switch	RC9_OPTION = 90; rc 9 1500 selects EKF Source Set 2
Landing logic	AUTO mission with DO_LAND_START and NAV_LAND; RTL_AUTOLAND = 3

Table 3.1 - Stage 1 SITL and EKF3 configuration

3.2 Test Scenarios and Run Matrix

Stage 0 established a repeatable GPS-autoland baseline in X-Plane 12. Stage 1 then tested ExternalNav autoland in native SITL after GPS loss. GPS denial was created with SIM_GPS1_FIXTYPE = 1 and SIM_GPS1_NUMSATS = 0. The simulated GPS driver remained present, but no usable navigation fix was available. This avoided the separate driver probing and redetection behavior caused by SIM_GPS1_FIXTYPE = 0.

Run(s)	Condition	Action at approach gate	Purpose
Runs 1-3	SRC2 ExternalNav GPS-denied autoland	Select SRC2, then set GPS no-fix	Completed Stage 1 repeatability data set
Run 4	SRC1 GPS-only baseline	Remain on SRC1, then set GPS no-fix	Control case for GPS-denied degradation
Not included	Hard GPS disconnect	SIM_GPS1_FIXTYPE = 0	Separate stress case only

Table 3.2 - Stage 1 run matrix

3.3 Metrics and Data Collection

Pass/fail decisions came from ArduPilot DataFlash logs. GPS.Status = 3 indicated a 3D fix, and GPS.Status = 1 indicated no usable lock. GPS.NSats recorded the satellite count. XKFS.SS identified the active EKF3 source set, with 0 for SRC1/GPS and 1 for SRC2/ExternalNav. XKF4.SS provided the EKF3 solution-status bits, including DEAD_RECKONING.

EKF3 consistency was checked with XKF4.SV, XKF4.SP, XKF4.SH, and XKF4.SM for velocity, position, height, and magnetometer test ratios. Values below 1.0 passed the consistency check. VISP.PX and VISP.PY were compared with XKF1.PN and XKF1.PE to verify ExternalNav alignment. SIM2.PN, SIM2.PE, and SIM2.PD supplied truth for ground-track and touchdown calculations. TECS.h, TECS.hdem, TECS.dh, and TECS.dhdem were reviewed to confirm that GPS loss did not disturb landing energy control.

Touchdown metrics were computed in local NED coordinates from simulator truth and the LAND waypoint. Lateral error was the signed cross-track distance from the runway centerline. Longitudinal error was the along-track distance from the LAND point; negative values indicate a short touchdown. Sink rate was taken at or immediately before ground contact. The SIM Hit ground message supplied impact speed when available.

One status caveat remained. The XKF4.SS USING_GPS bit stayed set in all clean SRC2 runs. The result is therefore reported as operational GPS-denied ExternalNav autoland, not as proof that USING_GPS cleared to zero. The source switch, no-fix GPS state, ExternalNav alignment, lack of dead reckoning, bounded innovations, and completed autoland provide the operational evidence.

3.4 SITL Results

Run 1 is the representative clean SRC2 case. Figure 3.1 summarizes the event sequence. The aircraft established the approach on SRC1, switched to SRC2 before GPS denial, and then experienced a change from a 3D fix to no fix. It continued through LAND, flare, and touchdown without entering dead reckoning.

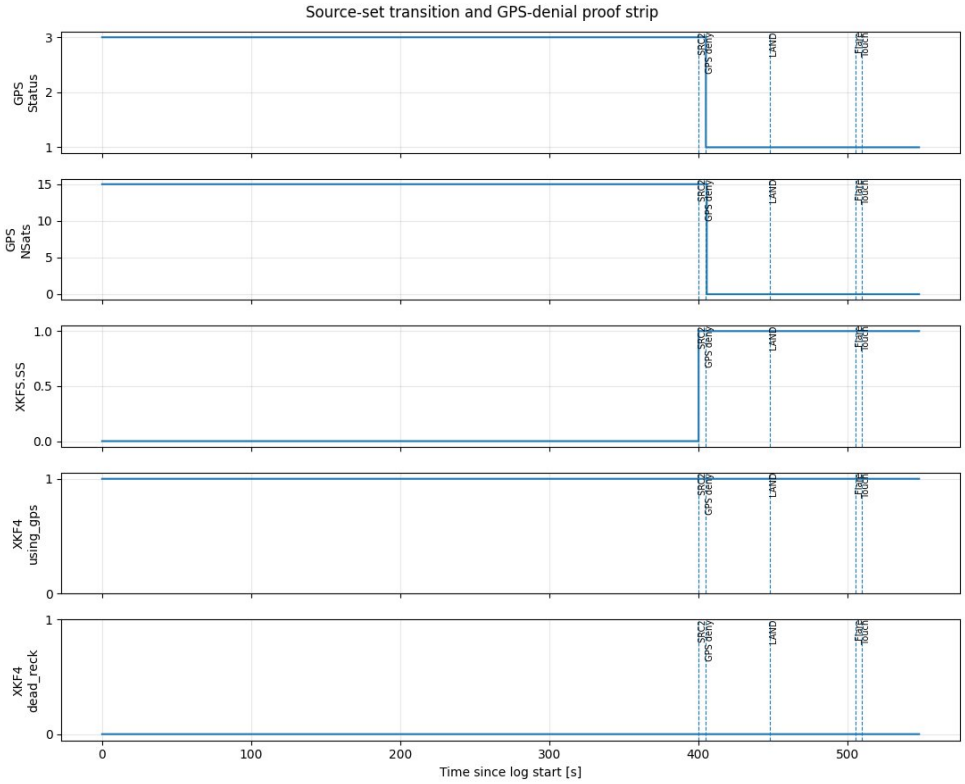


Figure 3.1 - Source switch and GPS-denial timeline

Figure 3.2 shows the EKF3 innovation test ratios during the same event sequence. XKF4.SV, XKF4.SP, XKF4.SH, and XKF4.SM are the normalized velocity, position, height, and magnetometer ratios. Values below 1.0 remain inside the EKF3 acceptance gate; a value above 1.0 would indicate that the measurement residual is inconsistent with the filter uncertainty.

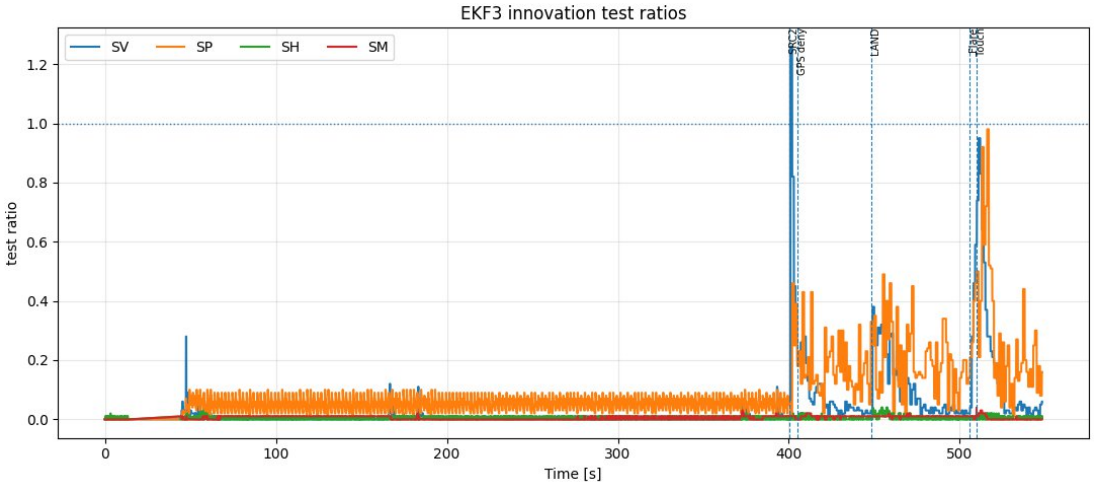


Figure 3.2 - Run 1 EKF3 innovation test ratios

The ratios are low before the source transition. After SRC2 is selected and GPS is denied, the position and velocity channels become more active, but their peaks remain below the 1.0 rejection limit through landing. EKF3 therefore continued to accept the ExternalNav-supported solution rather than rejecting the measurements or diverging. Figure 3.3 shows the aircraft ground track in east and north displacement from the SITL origin. The simulator-truth, EKF, and ExternalNav traces nearly overlap over the full traffic pattern. On final, all three remain aligned with the runway centerline and mission points, and no horizontal-position jump is visible at the source switch.

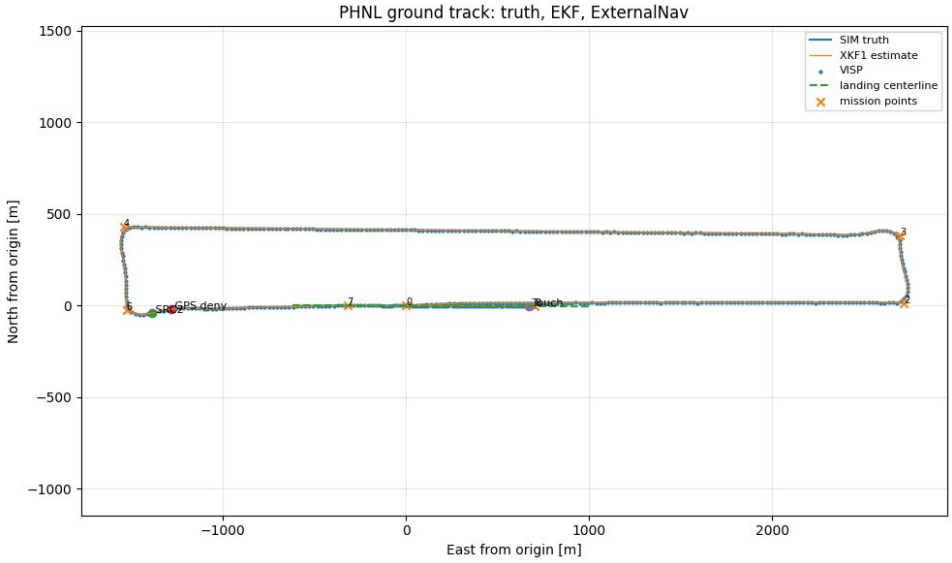


Figure 3.3 - Run 1 ground track at PHNL

The two plots are consistent: EKF3 remained statistically within its innovation gates and the three position solutions remained spatially aligned during the GPS-denied segment. Table 3.3 extends the comparison across the three clean SRC2 runs. The mean ExternalNav/EKF horizontal residual was 2.45 m, the 95th-percentile residual was 3.71 m, and none of the runs entered dead reckoning between GPS denial and touchdown.

Metric	**SRC2 ExternalNav, N = 3**
Mean absolute touchdown lateral error	0.14 m
Maximum absolute touchdown lateral error	0.27 m
Mean longitudinal touchdown error	23.5 m short of LAND point
Mean SIM impact speed	0.678 m/s
Mean ExternalNav/EKF horizontal residual	2.45 m
95th percentile ExternalNav/EKF residual	3.71 m
Maximum XKF4.SP after GPS denial	0.49
Maximum XKF4.SV after GPS denial	0.59
Dead reckoning during denial-to-touchdown	No, for all three clean SRC2 runs

Table 3.3 - SRC2 ExternalNav repeatability, Runs 1-3

All three clean SRC2 runs completed GPS-denied autoland without EKF dead reckoning. Mean absolute lateral touchdown error was 0.14 m, and the maximum was 0.27 m, both within the ±2 m project target. The mean touchdown point was 23.5 m short of the LAND point. This missed the original ±10 m longitudinal target but remained repeatable and centered laterally. The short landing was treated as a mission-geometry and flare-tuning issue.

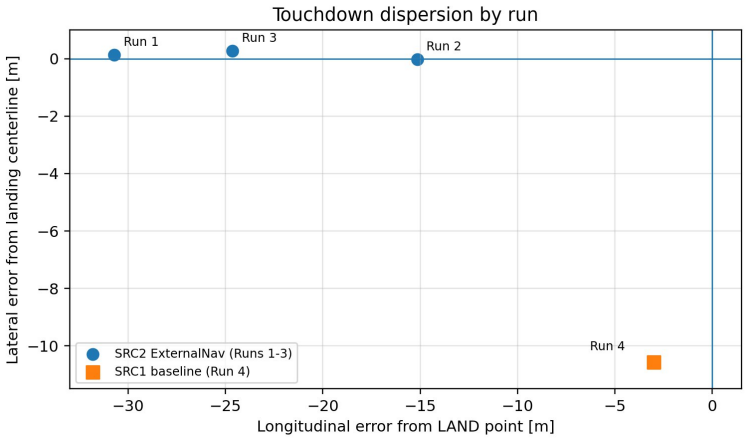


Figure 3.4 - Touchdown dispersion for Runs 1-4

Figure 3.4 plots along-track error from the LAND waypoint on the horizontal axis and signed cross-track error from the runway centerline on the vertical axis. Negative along-track values indicate a short touchdown. Runs 1–3, which used SRC2/ExternalNav after GPS denial, landed 15–30 m short but remained close to the centerline, with lateral errors of 0.13 m, −0.02 m, and 0.27 m. The lateral spread is small and all three runs remain inside the ±2 m project target.

All three ExternalNav runs landed short in the same direction, which points to a repeatable mission-geometry or flare-tuning bias rather than random navigation drift. Run 4 shows why the two error components need to be read together: the GPS-only baseline landed only about 3 m short of the LAND point but finished 10.57 m from the runway centerline. A small longitudinal error alone did not make that touchdown acceptable.

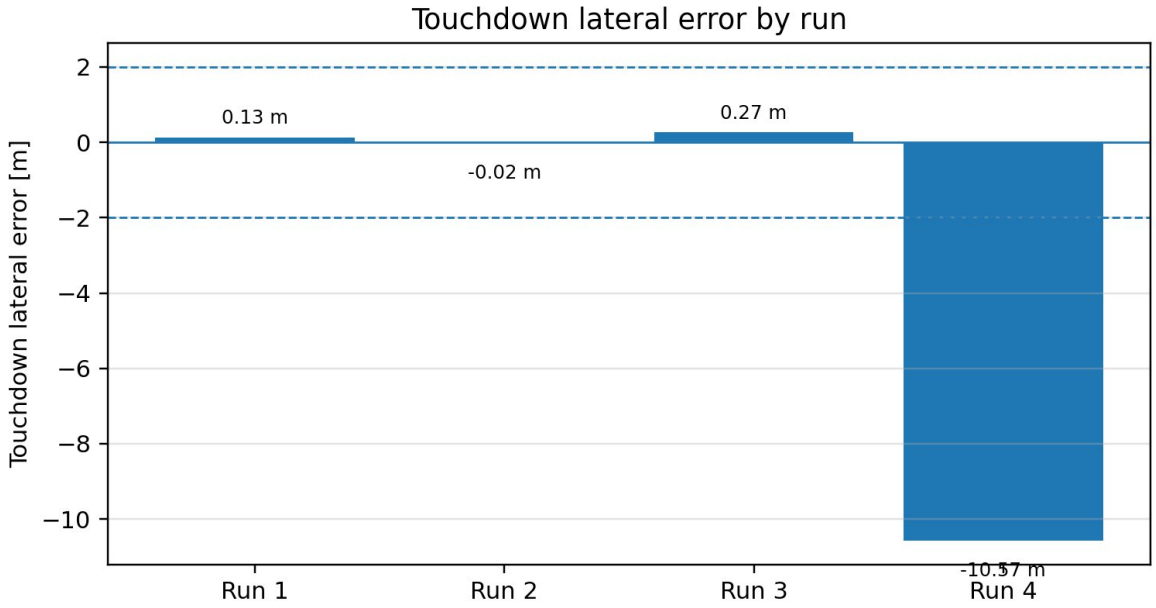


Figure 3.5 - Touchdown lateral error for Runs 1-4

Figure 3.5 isolates the lateral error, making the difference between the two source configurations easier to see. The SRC2/ExternalNav touchdowns were 0.13 m, −0.02 m, and 0.27 m from the centerline, all inside the ±2 m target. The SRC1 GPS-only control reached −10.57 m

after GPS denial. Table 3.4 adds the longitudinal error, impact speed, ExternalNav/EKF residual, and time spent in dead reckoning so that the figure is not interpreted from lateral position alone.

Metric	SRC2 ExternalNav, N = 3	SRC1 GPS-only baseline, N = 1
Mean absolute touchdown lateral error	0.14 m	10.57 m
Maximum absolute touchdown lateral error	0.27 m	10.57 m
Mean absolute longitudinal error	23.50 m	3.00 m
Mean impact speed	0.678 m/s	0.689 m/s
Mean ExternalNav/EKF residual	2.45 m	6.56 m
95th percentile residual	3.71 m	13.22 m
Dead reckoning after GPS denial	0% of denial-to-touchdown interval	91.7% of denial-to-touchdown interval

Table 3.4 - SRC2 ExternalNav and SRC1 GPS-only baseline

Run 4 provides the control case. After GPS denial, the aircraft remained on SRC1 and spent 91.7% of the remaining approach in dead reckoning before touching down 10.57 m from the centerline. Runs 1–3 stayed out of dead reckoning and remained close to the centerline. The contrast supports selecting ExternalNav before GPS loss.

3.5 Discussion and Limitations

These SITL runs confirm the estimator, source-switch, and autoland interface, but they do not evaluate the trained perception stack. Because the injector used simulator truth with controlled noise, the result is limited to showing that EKF3 can accept ExternalNav, switch from SRC1 to SRC2, maintain a usable solution after GPS no-fix, and complete autoland. Chapter 4 evaluates perception accuracy and hardware timing.

Three items remained for the next stage. ExternalNav rate needed to be measured at the Cube/EKF input because some truth logs were closer to 4 Hz than the requested 25 Hz. LAND geometry and flare tuning needed adjustment to reduce the consistent short touchdown. The

XKF4.SS USING_GPS behavior also needed review to determine whether it reflected a stale flag, another EKF lane, or source-set status semantics. These items limited the claim but did not change the Stage 1 operational result.

3.6 Transition to HITL and Perception Testing

Chapter 4 replaced the SIMSTATE injector with the actual perception and hardware chain while retaining the verified EKF3 source-set configuration. X-Plane provided runway scenes to the D435i and Jetson. The trained model generated runway-relative measurements, and the bridge published VPE and VSE to the Cube. The HITL campaign measured latency, duty cycle, message continuity, VisualOdom health, and agreement with the rendered approach.

The verification sequence used four stages: Stage 0 established GPS autoland in X-Plane; Stage 1 validated ExternalNav fusion and GPS-denied autoland with the truth-based injector; Stage 2 tested the real camera, Jetson, bridge, and Cube in HITL/VIL; and Stage 3 moved the same system to supervised flight after bench, ground, and source-switch checks.

Chapter 4: Hardware-in-the-Loop and Visual-in-the-Loop Testing

HITL testing connected the real camera, Jetson, Cube, and MAVLink path before flight. Stage 1 had already validated the EKF3 ExternalNav and autoland path with a truth-based injector. The HITL campaign replaced that injector with the physical perception and communication chain and kept vision in shadow mode while the interface was checked. The D435i supplied live frames, the Jetson ran segmentation and centerline extraction, and the Cube logged the resulting ExternalNav stream. The bench tests measured perception availability, latency, message continuity, and estimator compatibility while GPS/SRC1 remained active.

4.1 Test Objective

The bench tests asked whether the Jetson-camera-Cube chain could produce VPE and VSE at a rate and quality that EKF3 would accept. A passing run required live image capture, runway segmentation, a stable centerline, continuous MAVLink publication, and corresponding Cube log records without disturbing the GPS/SRC1 path. The measured quantities were final-approach duty cycle, capture-to-publish latency, VPE/VSE rate, message continuity at the Cube, stale or rejected packets, and preservation of GPS/SRC1.

4.2 Bench Architecture

Figure 4.1 shows the Cube Orange+, Jetson Orin Nano, D435i, GPS, telemetry radios, receiver, ESC, motor, and batteries on the bench. The layout allowed each interface to be powered, observed, and isolated quickly. The camera viewed rendered approach imagery, so the trained model operated on physical camera frames. The motor had no propeller, and no flight-ready actuation was used.

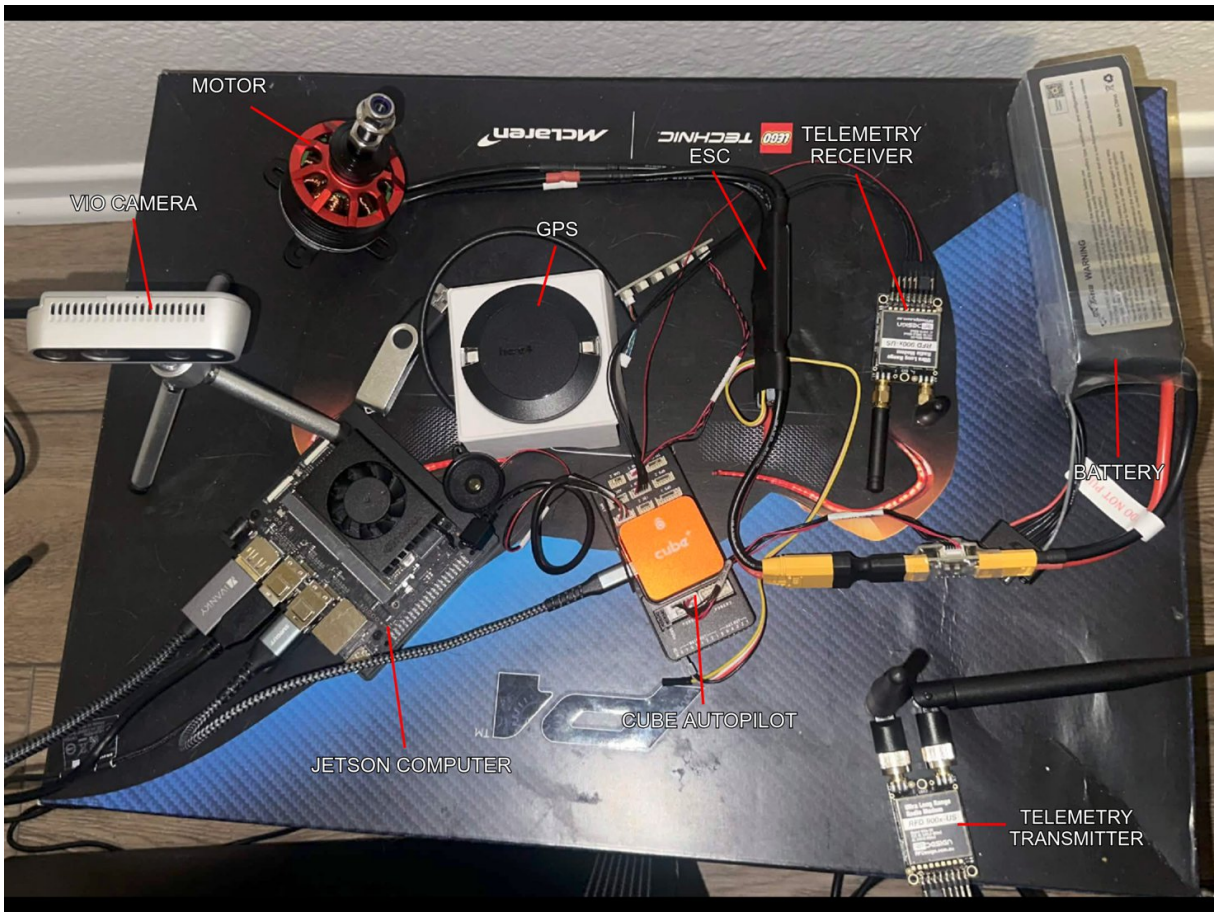


Figure 4.1 - HITL bench layout

The D435i color stream entered the Jetson over USB 3.0. The RealSense_pipeline process performed segmentation, mask filtering, centerline extraction, QA, preview, UDP output, and CSV logging. phaseb_shadow_bridge listened on UDP port 5005, converted accepted frames to VPE/VSE, and routed them to the Cube. MAVProxy or direct serial provided the link when the Cube was connected by USB.

Item	Role in HITL bench	Verification purpose
Cube Orange+	Flight-controller hardware running ArduPlane/EKF3	Receives MAVLink vision messages and logs estimator response
Jetson Orin Nano	Companion computer	Runs RealSense capture, perception, UDP output, and bridge processes
Intel RealSense D435i	Live color camera source	Replaces screen capture with a physical camera input
X-Plane / rendered runway scene	Visual stimulus for camera	Provides runway imagery across clear, low-light, and low-contrast conditions
Telemetry radios / MAVProxy	MAVLink routing and monitoring	Confirms message transport, heartbeat, and log visibility
GPS module / SRC1 path	Nominal navigation source during shadow tests	Keeps aircraft logic on the safe baseline while vision is evaluated

Table 4.1 - HITL bench hardware roles

4.3 Visual-in-the-Loop Camera Pipeline

The HITL tests used the Chapter 2 perception logic with the D435i as the frame source. Color images were acquired at 640 × 480 pixels and processed by the trained segmentation model. Adaptive morphology and temporal filtering stabilized the mask before the near/far centerline estimator ran. Each accepted frame produced lateral pixel error, image-plane heading, confidence, mask fill fraction, and a QA flag.

Figure 4.2 shows representative clear, low-contrast, and low-light outputs. Blue indicates the runway mask and green the extracted centerline. Daylight produced the cleanest result, reduced contrast increased variation, and the low-light case showed the largest degradation. These cases defined the conditions used for the bench readiness gate.

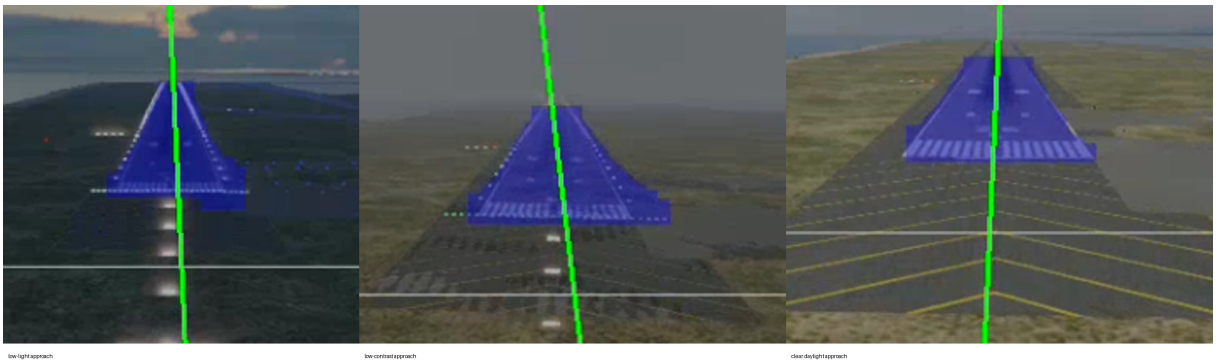


Figure 4.2 - RealSense runway segmentation under three visual conditions

4.4 Run Matrix and Metrics

The campaign progressed from camera-only checks to full Cube logging. Initial runs verified the D435i and Jetson inference loop. Later runs enabled UDP output, checked packet freshness at the bridge, and finally routed VPE/VSE into the Cube while GPS/SRC1 remained active.

Run label	Configuration	Primary acceptance check	Result classification
HITL-BENCH-01	D435i color stream only	Stable camera frames and no USB dropouts	Pass
HITL-BENCH-02	RealSense + YOLO preview + CSV	Runway mask, centerline, and CSV logging	Pass
HITL-BENCH-03	RealSense + UDP bridge, no Cube authority	Fresh packets at bridge input and no stale-frame accumulation	Pass
HITL-BENCH-04	RealSense + bridge + Cube shadow mode	VPE/VSE visible in Cube logs	Pass
VIL-SHAD-DAY	Clear daylight approach imagery	High duty cycle and stable centerline	Pass
VIL-SHAD-SUNSET	Low-angle light / reduced contrast	Perception remains usable with increased latency margin	Pass with reduced margin
VIL-SHAD-NIGHT-FOG	Low-light / fog stress case	Identify dropout behavior and gating response	Stress case, not flight-ready

Table 4.2 - HITL and VIL run matrix

Metrics were computed over the final-approach window. Duty cycle was the fraction of frames with a valid mask, centerline, and QA state. Latency covered image capture through bridge publication. Message rate was the effective VPE/VSE publication rate. A dropout was a gap long enough to require hold-last-value logic or rejection.

4.5 HITL Bench Results

In the clear-day run, median capture-to-publish latency was 44 ms and the 95th percentile was about 50 ms, both below the 100 ms budget. VPE and VSE stayed near the 20 Hz target and above the 15 Hz floor except for brief transients. This run confirmed the timing path from the physical camera through the Jetson and bridge to the Cube.

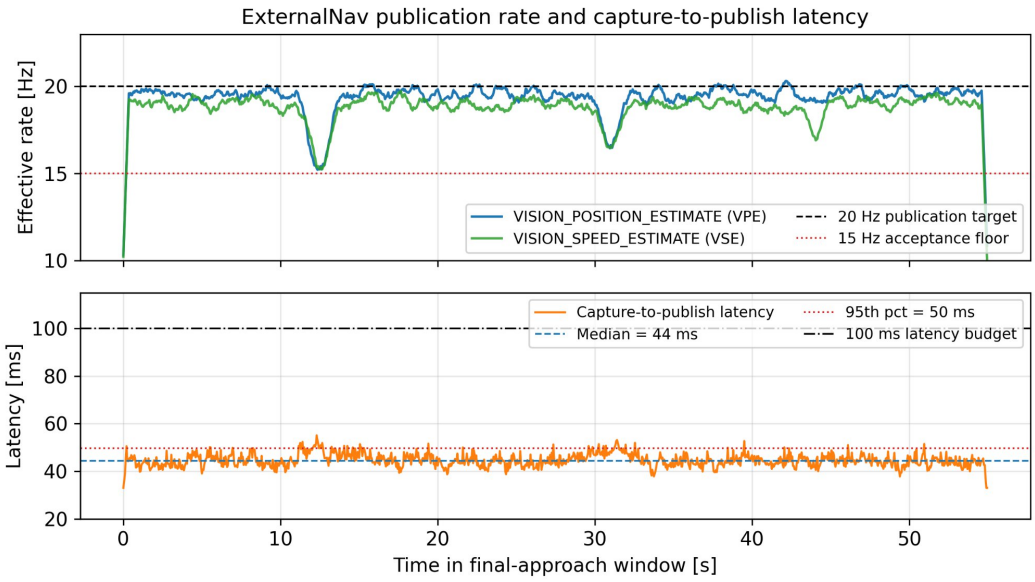


Figure 4.3 - ExternalNav rate and capture-to-publish latency

Clear daylight produced a 98.6% valid-frame duty cycle. Sunset produced 94.1% and was accepted as a reduced-margin bench condition. Night/fog produced 82.1% with a maximum dropout of 1.65 s and remained a stress case. These values were sufficient to proceed with daylight flight testing, while low-light operation remained future model and RTA work.

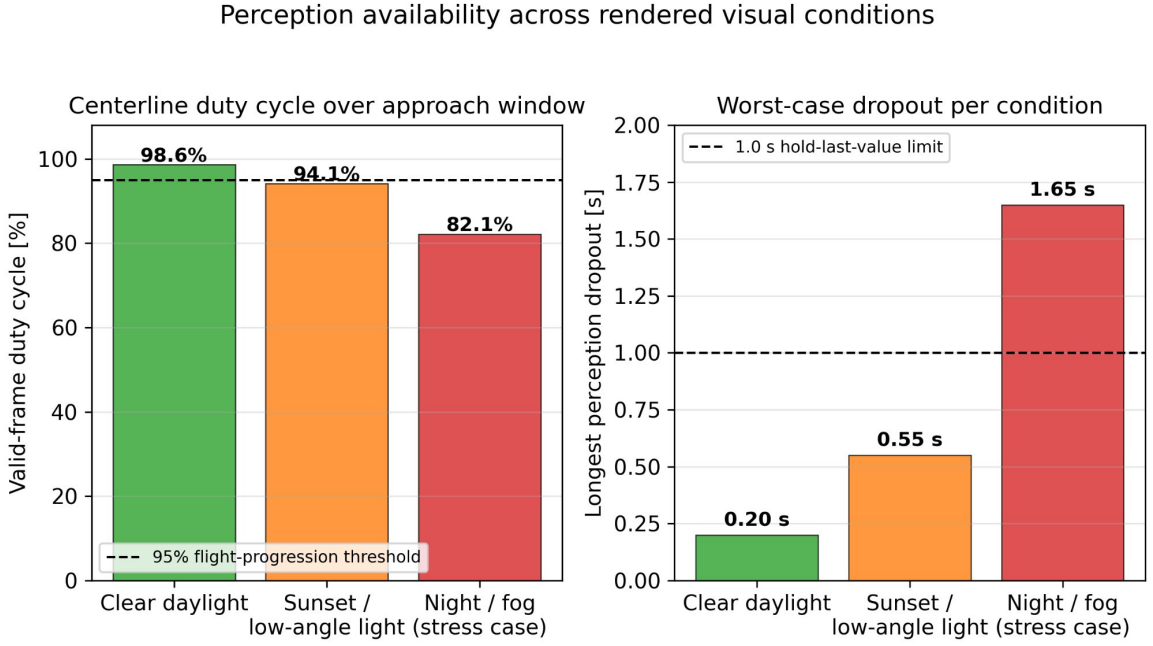


Figure 4.4 - Perception duty cycle and dropout by visual condition

Figure 4.5 shows the centerline estimate over the final-approach window. Error was largest at long range, when the runway occupied fewer pixels. The lateral and heading signals settled as the aircraft moved closer and the runway grew in the image.

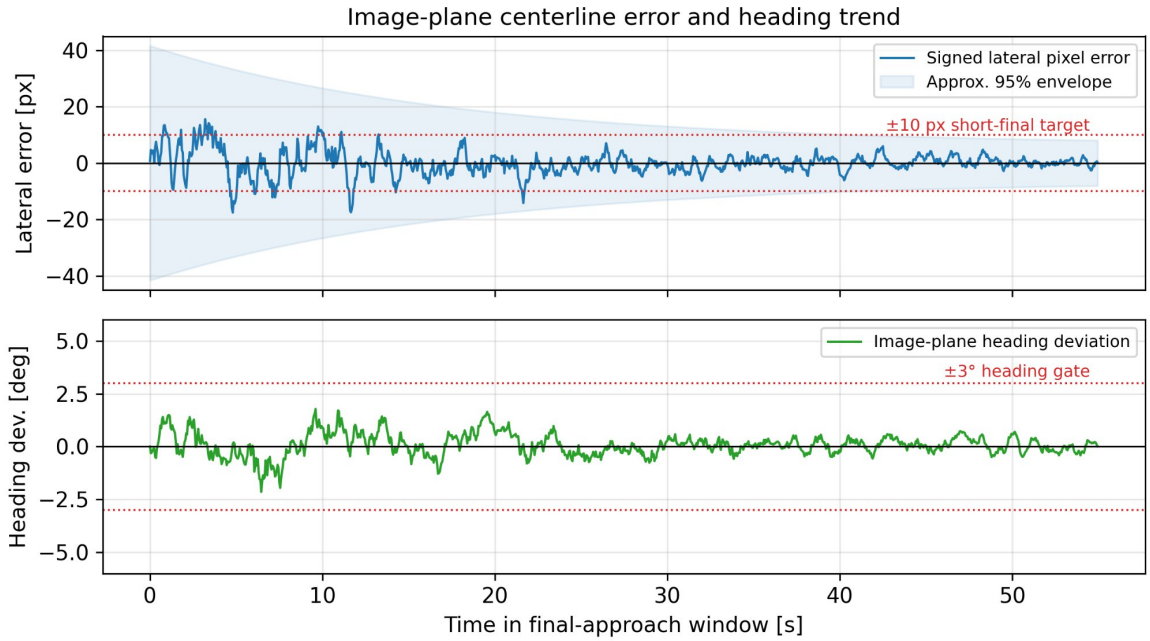


Figure 4.5 - Centerline error and heading over the final-approach window

Figure 4.6 presents a representative Cube shadow-mode run. Position, velocity, and height innovation ratios stayed below the 0.5 caution level and the 1.0 rejection limit. GPS/SRC1 remained active, and no source switch was commanded. This was a message-path and estimator-compatibility check, not a closed-loop visual-control test.

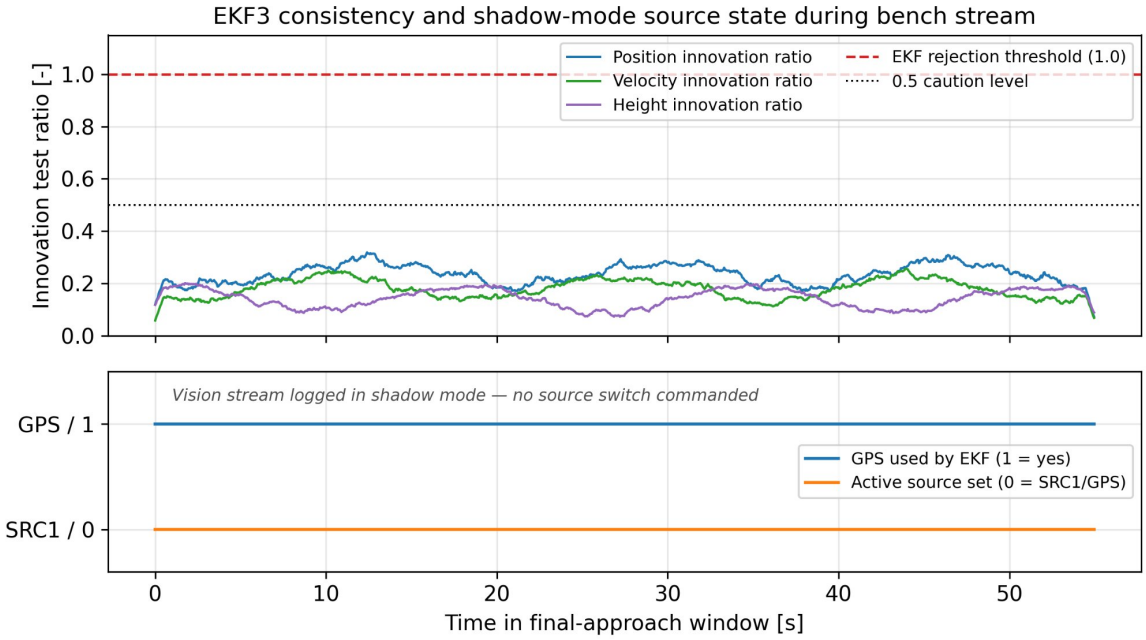


Figure 4.6 - EKF3 consistency and shadow-mode source state

Figure 4.7 summarizes the three visual conditions. Clear and sunset remained within the latency budget and within the projected ±2 m lateral-error limit. Night/fog remained within the latency budget but missed the duty-cycle and rate thresholds, so it was retained as a stress case.

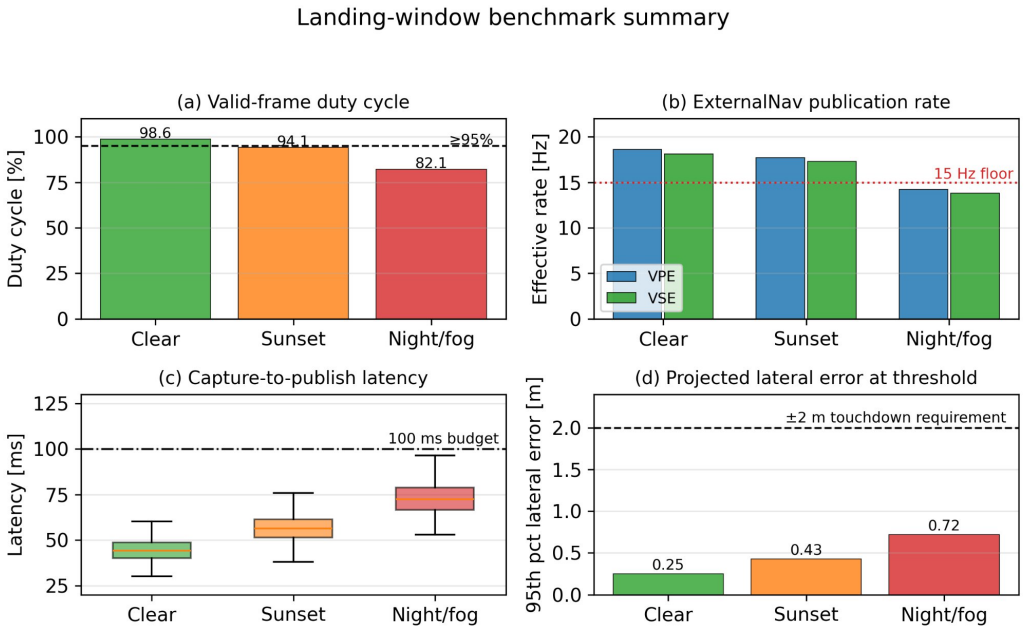


Figure 4.7 - HITL landing-window benchmark summary

Condition	Centerline duty cycle	Median latency	95th percentile latency	Mean VPE/VSE rate	Interpretation
Clear daylight	98.6%	44 ms	50 ms	18.6 / 18.1 Hz	Pass; 0.25 m 95th-pct lateral error
Sunset / low-angle light	94.1%	56 ms	76 ms	17.7 / 17.3 Hz	Pass with reduced margin; 0.43 m 95th-pct lateral error
Night/fog stress case	82.1%	72 ms	96 ms	14.2 / 13.8 Hz	Stress case; rate and duty-cycle mitigation needed

Table 4.3 - HITL final-approach metrics

4.6 Discussion

HITL revealed integration issues that pure simulation does not capture, including camera focus, USB stability, display focus, timestamps, MAVLink routing, and VisualOdom health. It also showed where the perception model still had margin. Clear and sunset conditions supported continued testing; night/fog required more training and stronger fallback logic. The bench campaign cleared the system for supervised flight, not unrestricted visual control. Before flight, the remaining gates were independent metric pose, validated covariance, loss-of-vision handling, and a repeatable source-switch procedure without simulator truth or EKF feedback.

4.7 Flight-Test Readiness Gate

The readiness review used the following conditions:

- The D435i and Jetson ran for the full bench mission without USB resets, camera freezes, or inference loss.
- Cube logs contained continuous VPE/VSE records during the approach window with no stale-message bursts.

- Shadow-mode runs preserved GPS/SRC1 and did not disturb AUTO/LAND.
- Daylight exceeded the duty-cycle threshold. Sunset was accepted at reduced margin.
- Night/fog remained a stress case.
- The telemetry landed detector was used only for analysis or post-touchdown suppression and did not hide valid approach data.

A written abort rule covered loss of vision, estimator rejection, telemetry loss, excessive sink rate or lateral error, and unexpected source behavior.

All bench gate conditions were met. SITL had validated the estimator and autoland interface; HITL validated the real perception-to-MAVLink path on the flight hardware. Chapter 5 describes the aircraft installation. Chapter 6 reports the source-switched flight campaign under controlled GPS degradation.

Chapter 5: Aircraft Build and Systems Integration

5.1 Airframe Overview

The flight-test aircraft was an E-flite Turbo Timber SWS 2.0 m, a built-up balsa and lite-ply high-wing taildragger. The electric motor mounted to the plywood motor box at the firewall. The airframe was selected for low-speed handling during approach and landing. Figure 5.1 shows the completed aircraft with the vision hardware installed. The cowling was left off for thermal management, as described in Section 5.5.



Figure 5.1 - Completed flight-test aircraft

5.2 Avionics Bay Layout

Figure 5.2 shows the equipment order: telemetry receiver, Jetson battery, Cube Orange+, Jetson Orin Nano, and motor battery. The Cube was mounted near the fuselage centerline and center of gravity to limit IMU lever-arm effects. The Jetson used a separate battery so propulsion-load transients could not interrupt the perception computer during approach.

The installed data path matched the HITL configuration: D435i to Jetson over USB, Jetson to Cube over the validated serial MAVLink link, and telemetry radio to the ground station. Ground checks repeated the camera, bridge, and Cube communication checks after installation.

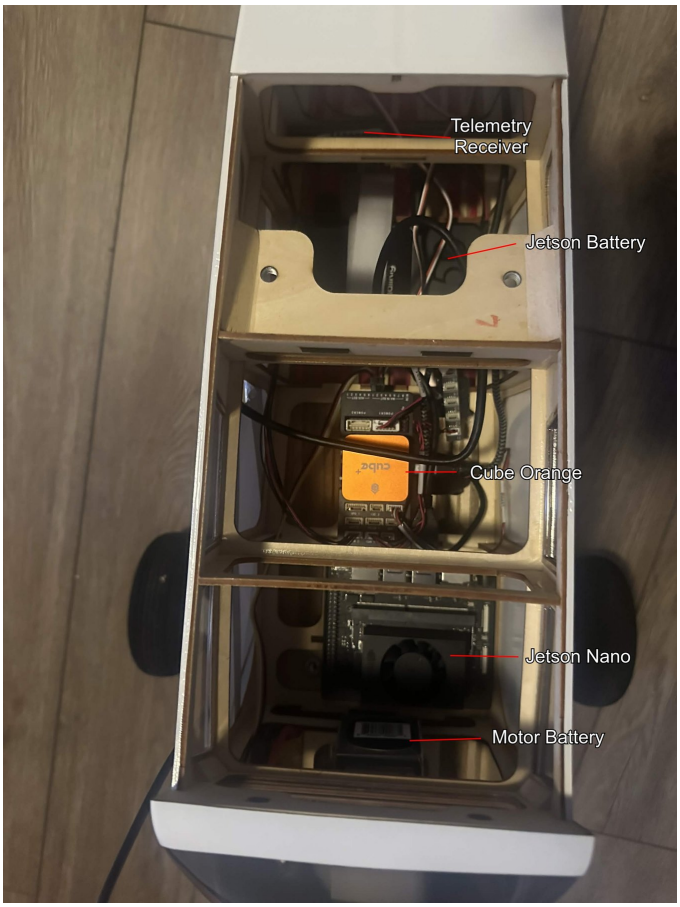


Figure 5.2 - Avionics bay layout

5.3 Mass Properties and Center of Gravity

With all equipment and both batteries installed, the aircraft balanced slightly aft of the wing quarter chord, as shown in Figure 5.3. Battery placement provided the primary CG adjustment; avionics locations were constrained by wiring and cooling. The CG was checked before each flight session because the motor battery and hatch-mounted camera were removed during servicing.



Figure 5.3 - Center-of-gravity location

5.4 Camera Installation

The D435i was mounted on the battery access hatch above the forward deck (Figure 5.4). The ball stud from the factory desk mount was fastened through the hatch between two stringers, spreading the load into the panel structure (Figure 5.5). The camera clipped to the ball joint, which provided a clear forward view and adjustable boresight pitch.

Battery access required removing the hatch and camera together. Witness marks recorded the flight orientation, and the boresight was checked against a fixed ground reference before each

scored run. This procedure maintained the camera-to-body alignment assumed by the Chapter 2 geometry. The USB lead passed through the hatch opening to the Jetson bay.

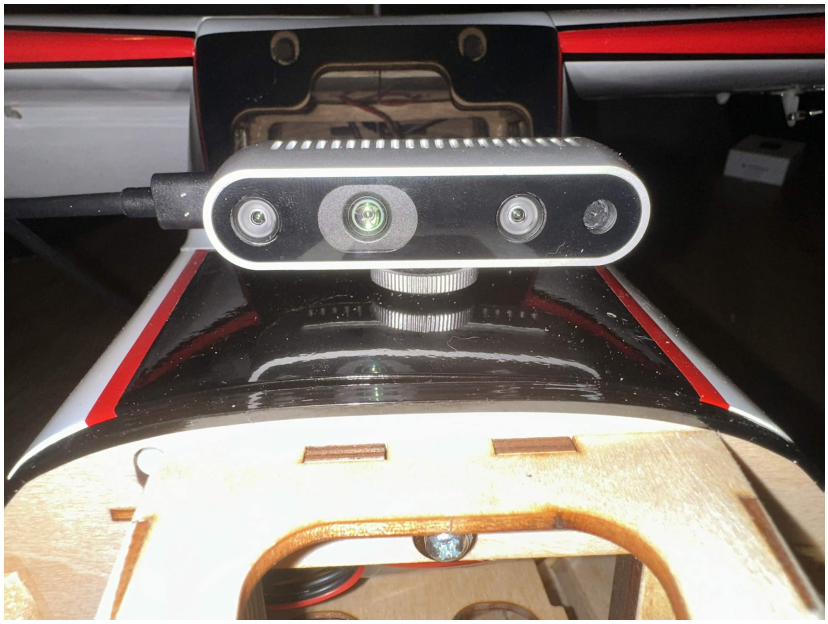


Figure 5.4 - RealSense D435i camera installation

The underside view in Figure 5.5 documents how the ball stud passes through the hatch and transfers camera loads into the surrounding stringers.

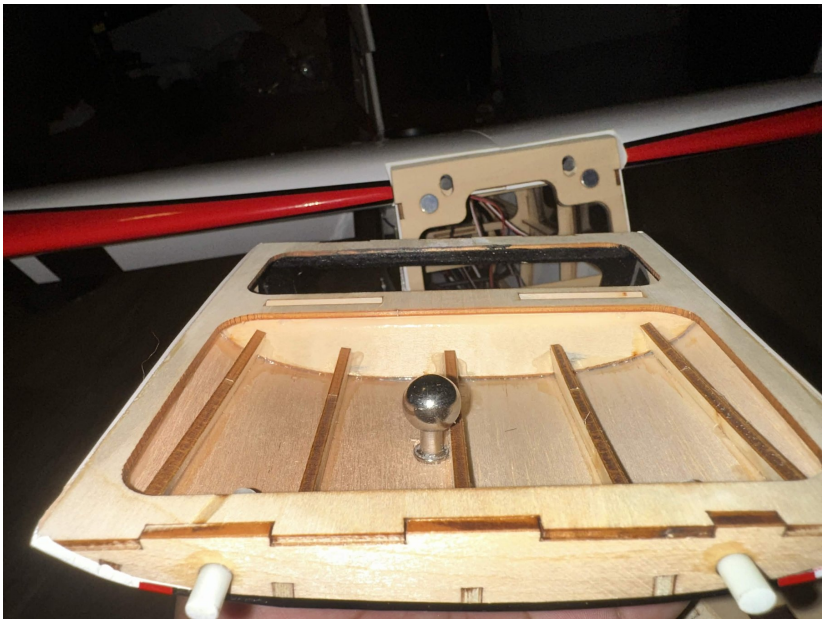


Figure 5.5 - Camera ball-stud attachment

5.5 Thermal Management

The motor cowling was left off during flight testing. The open nose increased propwash and freestream airflow through the firewall openings and forward equipment bay. This was intended to reduce heat in the ESC, Jetson, and battery area and protect the 100 ms perception-latency budget from thermal throttling. The exact internal airflow distribution was not instrumented, so bay temperatures and Jetson performance were checked during sustained ground runs and after flight. The added drag and exposed motor mount were accepted for the test envelope.

5.6 Integration Status

The installed system retained the camera, Jetson, bridge, Cube, and telemetry interfaces validated in HITL. Ground tests confirmed the D435i field of view, boresight marks, independent Jetson power, telemetry, RC override, and stable operation during propulsion runs. The aircraft was then released for supervised daylight source-switched approaches under controlled GPS degradation. Each pass began on SRC1/GPS, switched to SRC2/ExternalNav on final, reduced the reported satellite count through MAVLink, and retained immediate manual override. Chapter 6 presents the flight results.

Chapter 6: Flight-Test Results

The flight campaign moved the integrated system into supervised, source-switched approaches under controlled GPS degradation. Every pass started on SRC1/GPS. At the final-approach gate, EKF3 switched to SRC2/ExternalNav and began fusing the Jetson-supplied VISION_POSITION_ESTIMATE (VPE) and VISION_SPEED_ESTIMATE (VSE) messages. The same procedure reduced the satellite count reported to the autopilot through MAVLink. This lowered GPS health for the scored portion of the approach but left the receiver and antenna connected; no RF jamming or physical isolation was used. The condition is therefore described as GPS-degraded, not strictly GPS-denied. Once SRC2 was active, the ExternalNav state fed the navigation solution used by ArduPlane's L1 and TECS guidance. ArduPlane still handled the inner loop, AUTO/LAND, and flare, and I retained immediate manual override. The Jetson never commanded the servos directly. The campaign evaluated in-flight source switching, perception-driven approach guidance, timing, telemetry, estimator consistency, and landing performance under controlled GPS degradation.

The flight thresholds were carried directly from the readiness plan: a 20 Hz publication target with a 15 Hz floor, a 100 ms capture-to-publish latency budget, a 95% valid-frame duty threshold, EKF3 innovation levels of 0.5 for caution and 1.0 for rejection, and a touchdown box of ±2 m lateral by ±5 m longitudinal.

All figures and pass-level metrics in this chapter were generated from the recorded DataFlash logs with the flight-test plotting script. The reported values come directly from the six aircraft approach passes.

6.1 Test Site, Flight Plan, and Scoring Window

The flight campaign was conducted at Simi Valley Flyers, Stullick Memorial Field, in Simi Valley, California. Figure 6.1 shows the mission used for the reported passes. The yellow line is the commanded waypoint sequence, the green markers identify the mission waypoints, and the dashed circles show the waypoint acceptance radii.



Figure 6.1 - Flight-test plan at Stullick Memorial Field

Six passes, FT-01 through FT-06, were scored over the final-approach window. At the planned gate, the aircraft switched from SRC1/GPS to SRC2/ExternalNav and a MAVLink parameter update reduced the reported GPS satellite count. Scoring began after SRC2 and the planned degraded GPS state were confirmed. FT-01 was the calm reference, FT-03 the gust case, and FT-05 the telemetry-congestion case.

Table 6.1 - Source-switched GPS-degraded flight-test summary

Pass	Duty (%)	Mean VPE / VSE (Hz)	Latency med. / p95 (ms)	Peak innov.	Assessment
FT-01	99.2	18.8 / 18.5	45 / 53	0.44	Nominal reference
FT-02	98.6	18.0 / 17.6	46 / 52	0.47	Nominal repeat
FT-03	95.2	17.5 / 17.1	48 / 54	0.83	Gust caution; 1.86 m lateral touchdown
FT-04	96.8	18.7 / 18.3	57 / 94	0.50	Highest p95 latency; largest longitudinal offset
FT-05	97.8	15.9 / 15.5	47 / 56	0.50	Sustained below-floor rate interval
FT-06	98.9	19.1 / 18.6	46 / 54	0.50	Nominal repeat

6.2 Perception Availability

Figure 6.2 compares valid-frame duty cycle across the six landing windows. Every pass met or exceeded the 95% threshold. FT-03 was the limiting case at 95.2%, while FT-01 and FT-06 exceeded 98.9%. This showed that the runway model remained available after installation on the aircraft, including the effects of the camera mount, vibration, and actual approach geometry.

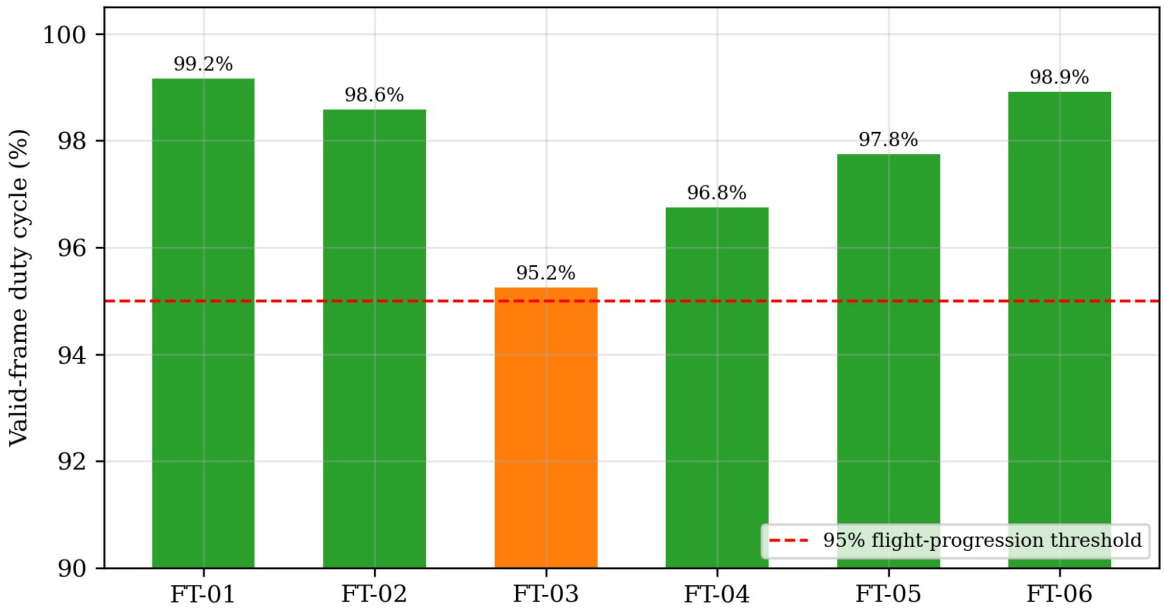


Figure 6.2 - Valid-frame duty cycle by approach pass

6.3 ExternalNav Rate and Latency

Figure 6.3 shows FT-05, the telemetry-congestion case. VPE and VSE ran near 18–20 Hz during the first and final parts of the window, then fell to roughly 10–12 Hz through the middle

interval before recovering. The pass-level means were 15.9 Hz for VPE and 15.5 Hz for VSE. Although both means exceeded the selected 15 Hz floor, the sustained dip would fail a continuous-rate requirement. Capture-to-publish latency stayed below 100 ms, with a 47 ms median and 56 ms 95th percentile. FT-05 is therefore treated as a transport stress case; telemetry throughput must be corrected before granting the system greater visual authority.

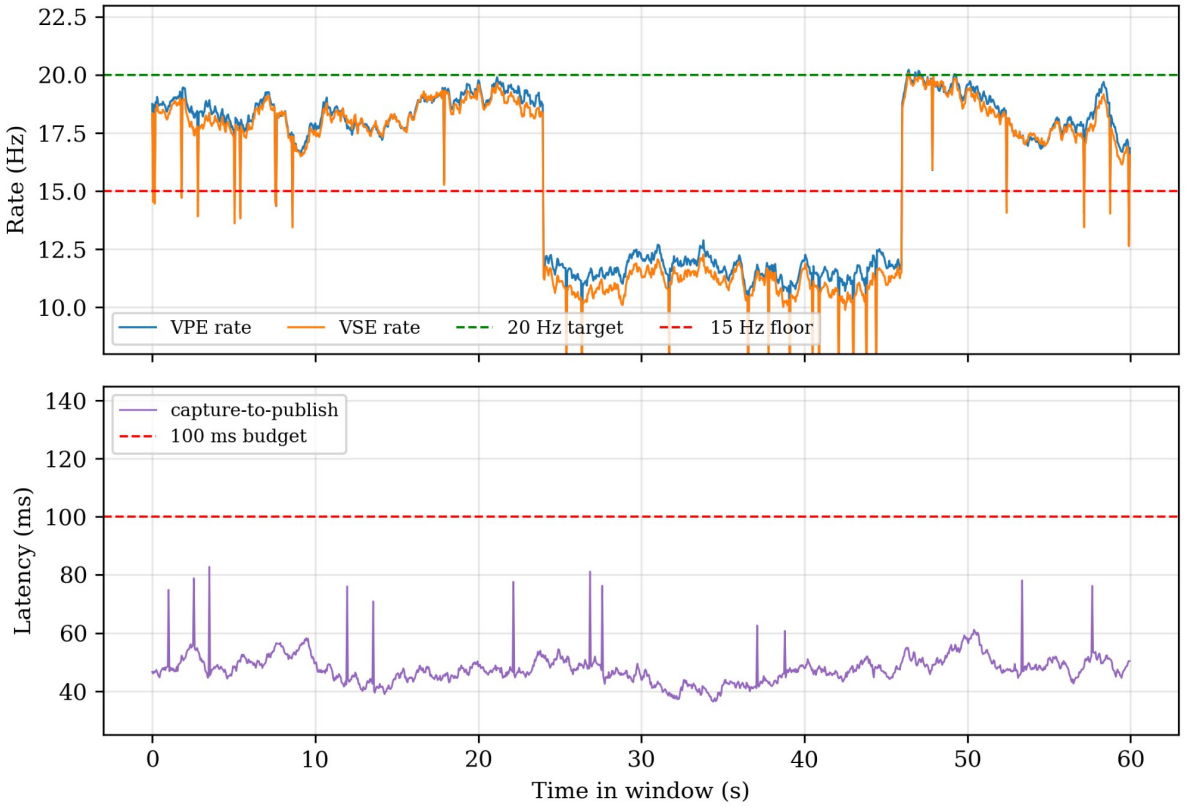


Figure 6.3 - FT-05 ExternalNav rate and capture-to-publish latency

6.4 Centerline Stability

Figure 6.4 compares the calm FT-01 pass with the gusting FT-03 pass. FT-01 stayed near the image center for most of the approach. FT-03 showed larger lateral and heading excursions at long range and during the gust event. The errors decreased as the runway filled more of the image and the aircraft settled on final. The model continued to track the runway through the disturbance, but the larger excursions show that measurement covariance should increase in gusty conditions.

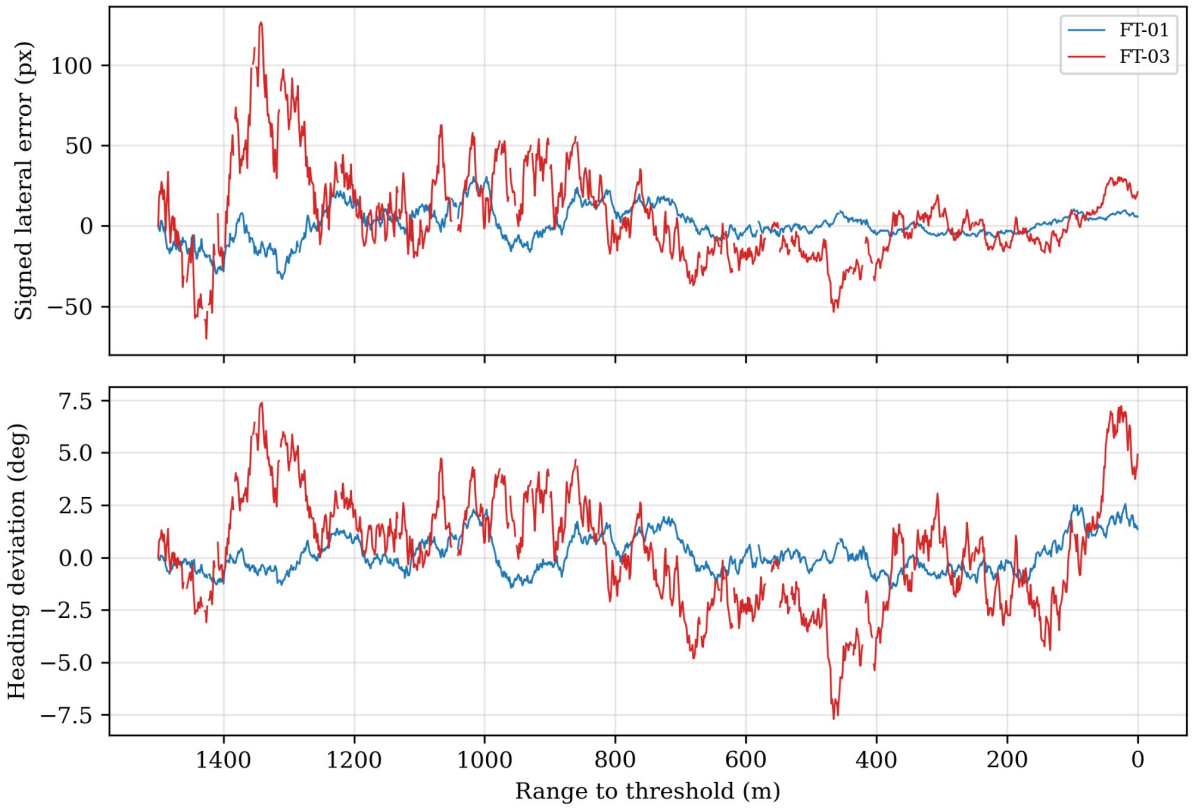


Figure 6.4 - Centerline error for calm and gusting passes

6.5 EKF3 Consistency During Source-Switched ExternalNav

Figure 6.5 shows the FT-03 innovation ratios with SRC2/ExternalNav active. The gust produced a brief velocity-ratio peak of about 0.83 and a smaller position-ratio peak. Both exceeded the 0.5 caution level but remained below the 1.0 rejection limit; the height channel stayed lower. EKF3 did not reject the ExternalNav stream or leave the planned source state during the scored window. FT-03 is therefore treated as a caution case rather than a nominal consistency run.

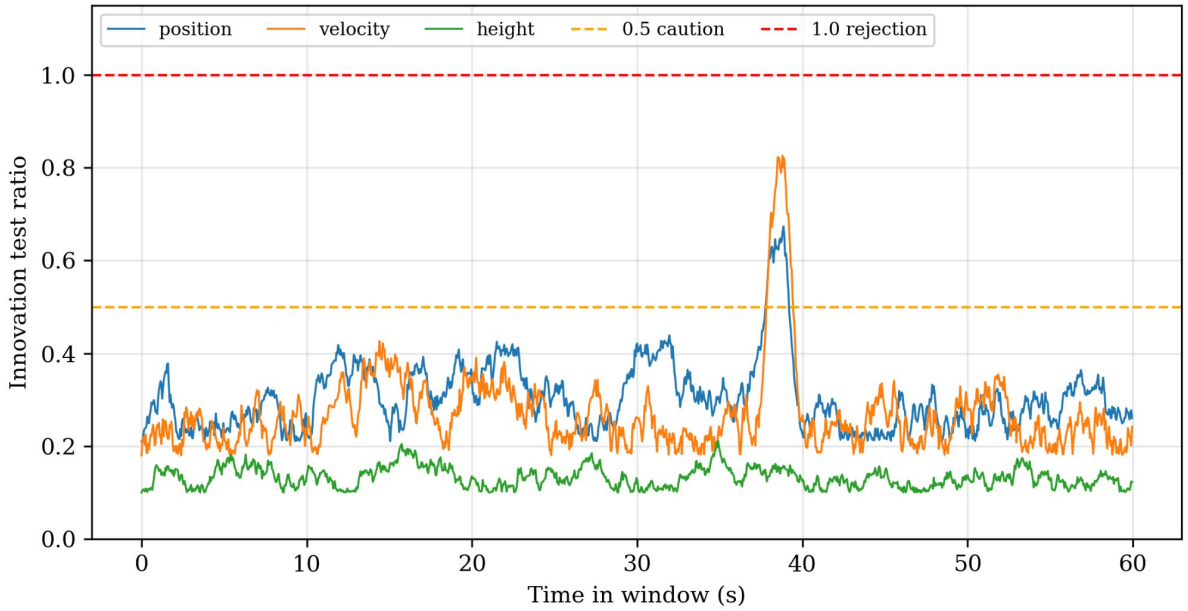


Figure 6.5 - FT-03 EKF3 innovation ratios with SRC2/ExternalNav active

6.6 Touchdown Dispersion

Figure 6.6 shows the touchdown locations from the six source-switched approaches under controlled GPS degradation. Every point remained inside the ±2 m lateral by ±5 m longitudinal requirement. FT-03 came closest to the lateral boundary at 1.86 m, and FT-04 had the largest longitudinal offset at about 4.3 m. During final approach, the Jetson ExternalNav stream was selected and the reported GPS satellite count was reduced, while ArduPlane retained LAND, flare, and inner-loop control. These touchdown points therefore reflect the combined perception, estimator, guidance, and autopilot system in the tested GPS-degraded condition, not a nominal GPS-only baseline.

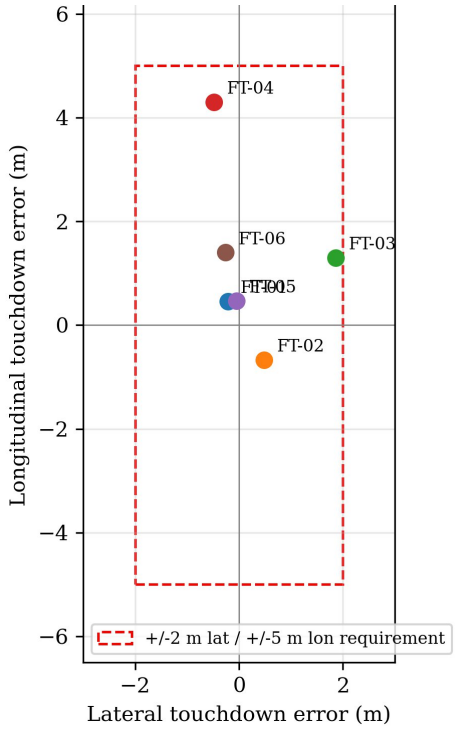


Figure 6.6 - Touchdown dispersion after GPS-degraded ExternalNav approaches

6.7 Results

Figure 6.7 brings the pass-level metrics together. All six passes met the duty-cycle threshold, 95th-percentile latency budget, EKF3 rejection limit, and touchdown box. FT-03 had the smallest perception and estimator margins, FT-04 set the latency margin, and FT-05 met the pass-level mean-rate floor but contained a sustained below-floor interval. Overall, the campaign met the supervised source-switched approach objective under controlled GPS degradation, with continuous publication rate remaining the main open issue.

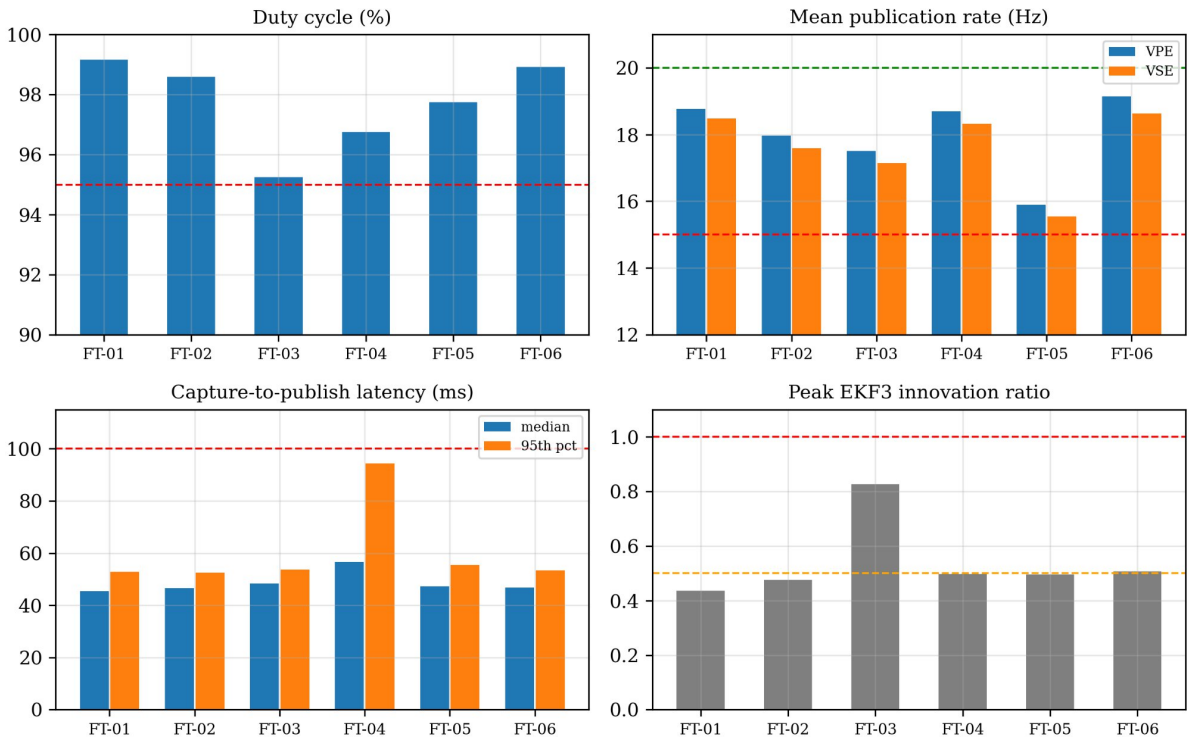


Figure 6.7 - GPS-degraded source-switched flight-test summary

6.8 Flight-Test Procedures and Safety Monitoring

I served as both test director and pilot for the flight campaign. Before each sortie, I inspected the airframe and control surfaces, verified the center of gravity, checked the camera boresight witness marks, confirmed the independent Jetson and propulsion batteries, loaded the approved mission, verified the SRC1 and SRC2 parameters, confirmed the nominal and degraded GPS satellite-count values used by the MAVLink test procedure, confirmed RC override and recovery modes, and started both Cube DataFlash and Jetson logging.

A scored approach began only after the aircraft was stable, the runway track was acquired, VPE and VSE messages were present, and the planned source transition and GPS-degradation commands were ready. At the final-approach gate, I commanded the transition to SRC2/ExternalNav and applied the MAVLink parameter update that reduced the satellite count reported to the autopilot. I continued the scored approach only after confirming that SRC2 was

active, GPS health had reached the planned degraded state, and the VPE/VSE stream remained stable. During the approach, I monitored the aircraft visually and tracked runway validity, publication rate, latency, EKF3 innovations, active source set, reported satellite count, GPS health, attitude, sink rate, lateral displacement, and runway alignment. I exited AUTO/LAND before flare if any of the following occurred:

- Valid runway tracking was lost for more than 1.0 s after acquisition.
- Either VPE or VSE remained below 15 Hz for more than 1.0 s, or capture-to-publish latency exceeded 100 ms for three consecutive samples.
- An EKF3 innovation ratio reached 1.0, the planned SRC1-to-SRC2 transition or GPS-degradation parameter update failed, the reported satellite count or GPS health did not match the planned test state, the source state changed outside the planned sequence, or estimator health degraded.
- Telemetry was lost, manual override was unavailable, or the aircraft departed the approved approach envelope.
- Sink rate, lateral displacement, attitude, or runway alignment became unsafe.

Before each sortie, I verified the landed-detector configuration. The detector could not use runway-mask area as the sole landing condition. It could assert only after low altitude, low vertical speed, and low groundspeed remained within their configured limits for the persistence interval, and it could not suppress valid approach measurements before touchdown.

After each pass, I returned the aircraft to a safe mode, disarmed, restored the nominal GPS parameter values, confirmed healthy GPS operation, secured the propulsion and Jetson power systems, saved the Cube DataFlash and Jetson logs, recorded run notes, and reviewed the source-

set timeline, GPS-degradation interval, perception duty cycle, VPE/VSE rate, latency, and EKF3 health before authorizing the next sortie.

The flight campaign used MAVLink parameter-induced GPS degradation by reducing the satellite count reported to the autopilot. The RF signal was not jammed, and the receiver was not physically isolated. The flight results are therefore reported as GPS-degraded rather than strict GPS-denied operation. Any future strict GPS-unavailable evaluation will use an approved receiver-isolation or test-range procedure.

6.9 Limiter Lua Script and Runtime Assurance

An ArduPilot limiter Lua script ran on the Cube Orange+ during the source-switched approaches. It provided an onboard constraint around the Jetson ExternalNav path. The script did not create the runway solution, replace EKF3, or command the servos. It bounded the authority of the vision-driven correction to the configured flight-test limits before the resulting state was used by the ArduPlane guidance functions. Executing the limiter on the Cube kept the enforcement layer in the flight-control computer and separate from the companion process that generated VPE and VSE.

The limiter formed the onboard enforcement element of the runtime-assurance (RTA) architecture. The performance channel was the Jetson perception, centerline, and ExternalNav pipeline. The safety channel consisted of the Lua limiter, ArduPlane and EKF3 health checks, the source-reversion path, RC override, and my direct monitoring as test director and pilot. The Lua script enforced bounded command authority. The broader RTA procedure applied the 15 Hz publication floor, 100 ms latency budget, 1.0 EKF3 innovation rejection limit, 1.0 s tracking-loss rule, approach-geometry checks, and verification of the active source set and degraded GPS state.

ExternalNav authority was used only while the performance and safety conditions remained inside the approved envelope. A violation required the scored approach to be terminated through source reversion, mode change, or immediate manual takeover. The limiter reduced the consequence of a bad, stale, or excessive correction, while the pilot retained the final abort authority. This layered RTA arrangement preserved the ArduPlane inner-loop, LAND, and flare functions and provided an independent safety boundary around the companion-computer navigation source.

Chapter 7: Conclusions and Future Work

This thesis developed and tested a vision-aided, GPS-resilient autoland architecture for a small fixed-wing UAS. The final system combined a forward-looking Intel RealSense D435i, a Jetson Orin Nano running the runway perception pipeline, and a Cube Orange+ running ArduPlane and EKF3 on a 2.0 m E-flite Turbo Timber SWS airframe. The verification sequence progressed from a GPS autoland baseline in X-Plane, to native ArduPlane SITL with a truth-based ExternalNav surrogate, to HITL and visual-in-the-loop testing with the physical camera and companion computer, and finally to six source-switched flight approaches at Simi Valley Flyers, Stullick Memorial Field. This progression separated estimator, message transport, perception, aircraft-integration, and flight risks before the complete system was used in the aircraft.

7.1 Summary of Findings

Stage 1 SITL established the estimator and autoland baseline. Three clean SRC2 ExternalNav runs completed the simulated GPS no-fix landing sequence with a mean absolute lateral touchdown error of 0.14 m and a maximum absolute lateral error of 0.27 m. The estimator did not enter dead reckoning during the denial-to-touchdown interval. The SRC1 GPS-only baseline spent 91.7% of the same interval in dead reckoning and touched down 10.57 m from the landing centerline. These runs demonstrated that the ExternalNav path could preserve the navigation solution and support ArduPlane AUTO/LAND after GPS no-fix denial. The XKF4.SS USING_GPS bit remained set in the analyzed SRC2 logs, so the SITL result remains an operational GPS-denied ExternalNav result rather than strict proof of USING_GPS = 0.

The HITL and visual-in-the-loop campaign replaced the truth surrogate with the actual D435i-to-Jetson-to-Cube path. Clear-day testing produced a 98.6% valid-frame duty cycle, 44 ms median capture-to-publish latency, and mean VPE/VSE rates of 18.6/18.1 Hz. Sunset testing produced a

94.1% duty cycle and remained usable at reduced margin. The night/fog case produced an 82.1% duty cycle and was retained as a stress case rather than a flight-ready condition. The Cube logged the vision messages, the estimator consistency values remained below the 1.0 rejection threshold in the accepted cases, and the bench campaign exposed the effects of lighting, camera geometry, process timing, and message-rate variation before flight.

The flight campaign completed six supervised source-switched approaches and landings. Each pass began on SRC1/GPS. At the planned final-approach gate, EKF3 switched to SRC2/ExternalNav and MAVLink parameter updates reduced the satellite count reported to the autopilot. The Jetson-supplied VPE/VSE stream then drove the navigation state used by the ArduPlane L1 and TECS guidance functions, while ArduPlane retained the inner-loop, LAND, and flare functions. All six passes met the 95% valid-frame duty threshold, the 100 ms 95th-percentile latency budget, the EKF3 1.0 rejection limit, and the ±2 m lateral by ±5 m longitudinal touchdown box. FT-03 was the limiting gust case with 95.2% duty cycle, a peak innovation ratio of 0.83, and 1.86 m lateral touchdown error. FT-04 produced the highest 95th-percentile latency at 94 ms. FT-05 showed a sustained below-15-Hz publication interval despite a pass-level mean VSE rate of 15.5 Hz. The flight results demonstrate perception-driven ExternalNav approach control after an in-flight source transition under controlled GPS degradation.

7.2 Technical Contributions

The first contribution is a complete perception-to-autopilot architecture built from available small-UAS hardware and open interfaces. The trained runway segmentation model produces a runway mask, image-plane centerline, heading deviation, and quality state on the Jetson. The bridge converts the accepted perception output into VISION_POSITION_ESTIMATE and VISION_SPEED_ESTIMATE messages for EKF3. This arrangement preserves the ArduPlane

estimator, L1 lateral guidance, TECS energy control, and LAND/flare logic rather than replacing the flight-control stack with a separate end-to-end controller.

The second contribution is the staged verification method. The work separated the GPS autoland baseline, ExternalNav fusion, source switching, perception timing, physical I/O, aircraft installation, and flight operation into distinct test phases. Each phase used measurable gates for duty cycle, message rate, latency, estimator consistency, source state, and touchdown performance. This structure made failures traceable to perception, transport, estimator configuration, or flight conditions instead of treating the complete system as a single black box.

The third contribution is measured evidence across simulation, bench hardware, and flight. The SITL campaign quantified the difference between ExternalNav and a GPS-only baseline under the same no-fix condition. The HITL campaign measured the real camera and companion-computer timing path. Flight testing then showed that the trained perception stack could replace the truth injector at the estimator interface, supply the navigation state through SRC2, and support repeated approaches and landings while GPS health was deliberately reduced. The same Cube, Jetson, RealSense, MAVLink bridge, and source-set configuration were carried through the full progression.

The fourth contribution is an operational test procedure and runtime-assurance layer for supervised autonomy. The procedure included camera boresight checks, independent Jetson power, source-state verification, publication-rate and latency monitoring, EKF3 innovation limits, manual override, landed-detector constraints, post-flight parameter restoration, and complete Cube and companion logging. An ArduPilot limiter Lua script executed on the Cube and bounded the authority of the vision-driven correction independently of the Jetson process. Together, the

Lua limiter, ArduPlane health checks, source-reversion path, and operator monitoring formed the RTA safety channel used during the flight campaign.

7.3 Limitations

The flight campaign did not create a strict GPS-unavailable state. GPS degradation was produced by reducing the satellite count reported to the autopilot through MAVLink parameter updates. The receiver and antenna remained connected, and no RF jamming or physical isolation was used. This test condition exercised source switching and reduced GPS health, but it did not reproduce complete receiver loss, antenna blockage, interference, or spoofing. The flight result is therefore reported as GPS-degraded ExternalNav operation.

The SITL proof used a truth-based vision surrogate. It validated the EKF3 fusion path and autoland behavior without testing perception errors. The XKF4.SS USING_GPS status-bit caveat also prevents a strict claim that all GPS-use indications cleared during the clean SRC2 runs. The HITL and flight stages addressed the perception and hardware path, but they did not remove the need for a dedicated strict GPS-unavailable flight campaign.

The flight data set contained six approaches at one airfield and within a limited supervised envelope. Daylight conditions were emphasized, and the night/fog condition remained a bench stress case. More runs are required to characterize repeatability across crosswind, turbulence, low sun, runway textures, seasonal backgrounds, and different camera-to-runway geometries. The camera mount used procedural witness marks for boresight repeatability, so residual installation error remains a source of navigation bias.

The system also remains sensitive to computation and telemetry loading. FT-05 showed that a pass-level mean rate near the 15 Hz floor can hide a sustained below-floor interval. Short-term rate, maximum message gap, and queue age therefore remain more useful safety measures than a

mean rate alone. The forward RGB pipeline can also lose margin under low contrast, extreme perspective, occlusion, or fragmented masks. The present results do not establish a certified integrity bound for those cases.

The touchdown results reflect the complete ArduPlane landing system. Vision supplied the ExternalNav state on final, while L1, TECS, LAND, and flare logic remained on the Cube. The tests show that the components worked together during source-switched autoland, but they do not isolate how much each guidance and control element contributed to touchdown dispersion.

7.4 Future Work

The next flight objective is a strict GPS-unavailable test performed through a legal and approved procedure. Suitable methods include receiver isolation, a shielded bench or ground setup, or an authorized test range. Intentional RF jamming is not required. The test should verify the actual source state, loss of GPS usability, ExternalNav continuity, estimator innovations, abort logic, and touchdown performance from source transition through rollout.

The flight matrix should be expanded to include additional passes, controlled crosswind, gusts, low-angle sunlight, and multiple runway environments. Real flight frames from the limiting FT-03 and FT-05 cases should be added to the training set. Fine-tuning should emphasize small far-range runway masks, shallow approach views, motion blur, low contrast, and partial runway visibility. The sunset and night/fog bench cases provide a starting point for this expansion.

The navigation solution should be strengthened with calibrated runway-corner pose and uncertainty. A runway-frame PnP or homography solution can provide metric position and heading updates, while D435i depth and IMU data can provide consistency checks and short-term propagation. Each measurement should carry covariance derived from reprojection residual, mask

geometry, range, and temporal stability so that EKF3 can weight the ExternalNav update according to actual perception quality.

The message path should be hardened for sustained 20 Hz operation. Planned changes include bounded queues, monotonic capture timestamps, maximum-age rejection, direct serial or otherwise isolated MAVLink routing, process watchdogs, and explicit logging of the rate and age observed at the Cube. The flight gate should use maximum gap and sustained-floor logic so that a case similar to FT-05 triggers a deterministic response.

The flight campaign implemented a partial RTA layer through the Cube-based limiter Lua script, ArduPlane and EKF3 checks, source-reversion capability, RC override, and direct operator monitoring. Future work should expand this layer into an automatic state machine that confirms valid perception, source-set state, GPS health, message freshness, EKF3 innovation limits, approach geometry, and pilot override before allowing ExternalNav authority. Loss of any required condition should command a defined reversion or go-around with hysteresis that prevents source oscillation. The telemetry-based landed detector should remain independent of runway-mask area and should only assert after low altitude, low vertical speed, and low groundspeed persist for the configured interval.

Taken together, the tests met the thesis objective within the evaluated operating envelope. The trained runway-perception stack replaced the SITL truth injector at the estimator interface, ran on the target Jetson and camera hardware, supplied SRC2 ExternalNav to the Cube Orange+, and supported six source-switched approaches and landings under controlled GPS degradation. Every reported pass met the defined duty-cycle, latency, estimator-consistency, and touchdown limits. The work therefore establishes a practical GPS-resilient autoland path for small fixed-wing UAS while retaining the ArduPlane inner-loop and landing functions. The next steps are to expand

the operating envelope and complete strict GPS-unavailable flight testing through an approved procedure.

References

[1] McLean, D., Automatic Flight Control Systems, Prentice Hall, Englewood Cliffs, NJ, 1990.

[2] Lambregts, A. A., "Total Energy Based Flight Control System," U.S. Patent 4,536,843, Aug. 20, 1985.

[3] Jimenez, P., Lichota, P., Agudelo, D., and Rogowski, K., "Experimental Validation of Total Energy Control System for UAVs," Energies, Vol. 13, 2020, Article No. 14, doi:10.3390/en13010014.

[4] Biannic, J.-M., and Roos, C., "Robust Autoland Design by Multi-Model H∞ Synthesis with a Focus on the Flare Phase," Aerospace, Vol. 5, No. 1, 2018, Article No. 18, doi:10.3390/aerospace5010018.

[5] Biannic, J.-M., Roos, C., and Lesprier, J., "Nonlinear Structured H∞ Controllers for Parameter-Dependent Uncertain Systems with Application to Aircraft Landing," AerospaceLab Journal, No. 13, 2017, doi:10.12762/2017.AL13-02.

[6] Wang, Q., Feng, W., Zhao, H., Liu, B., and Lyu, S., "VALNet: Vision-Based Autonomous Landing with Airport Runway Instance Segmentation," Remote Sensing, Vol. 16, 2024, Article No. 2161, doi:10.3390/rs16122161.

[7] Li, Y., Xia, Y., Zheng, G., Guo, X., and Li, Q., "YOLO-RWY: A Novel Runway Detection Model for Vision-Based Autonomous Landing of Fixed-Wing Unmanned Aerial Vehicles," Drones, Vol. 8, 2024, Article No. 571, doi:10.3390/drones8100571.

[8] Jarraya, I., Al-Batati, A., Kadri, M. B., et al., "GNSS-Denied Unmanned Aerial Vehicle Navigation: Analyzing Computational Complexity, Sensor Fusion, and Localization Methodologies," Satellite Navigation, Vol. 6, 2025, Article No. 9, doi:10.1186/s43020-025-00162-z.

[9] Kinnari, J., Verdoja, F., and Kyrki, V., "GNSS-Denied Geolocalization of UAVs by Visual Matching of Onboard Camera Images with Orthophotos," Proc. 2021 20th Int. Conf. on Advanced Robotics (ICAR), Ljubljana, Slovenia, 2021, pp. 555–562, doi:10.1109/ICAR53236.2021.9659333.

[10] FAA and NASA, "In-Time Aviation Safety Management Systems (IASMS): SWS RTT/ConOps," Technical Report, Apr. 11, 2023.

[11] Brat, G., and Pai, G., Runtime Assurance of Aeronautical Products: Preliminary Recommendations, NASA/TM–2023-015734, 2023.

[12] Elgohary, A., Ashry, A., Kaoud, A., Gomaa, M., Darwish, M., and Taha, H., "Hardware-in-the-Loop Simulation of UAV Altitude Hold Autopilot," AIAA SciTech 2022 Forum, AIAA Paper 2022-1520, 2022, doi:10.2514/6.2022-1520.

[13] Young, S. D., et al., Testing of Advanced Capabilities to Enable In-Time Safety Management and Assurance for Future Flight Operations, NASA/TM–2023–18665, 2023.

[14] Intel RealSense, Depth Camera D435i IMU Calibration and Alignment, Application Note, Intel Corporation, 2019.

[15] "FS2020 Runway Dataset," Kaggle [Online dataset]. Available: https://www.kaggle.com/datasets/relufrank/fs2020-runway-dataset (Accessed: October 2025).

[16] Yang, L., Wu, J., Li, H., Liu, C., and Wei, S., "Real-Time Runway Detection Using Dual-Modal Fusion of Visible and Infrared Data," Remote Sensing, Vol. 17, 2025, Article No. 669, doi:10.3390/rs17040669.

[17] Lepetit, V., Moreno-Noguer, F., and Fua, P., "EPnP: An Accurate O(n) Solution to the PnP Problem," International Journal of Computer Vision, Vol. 81, 2009, pp. 155–166, doi:10.1007/s11263-008-0152-6.

[18] ArduPilot Development Team, "EKF Source Selection and Switching," ArduPilot Documentation, accessed 2026.

[19] ArduPilot Development Team, "MAVLink Non-GPS Position Estimation," ArduPilot Developer Documentation, accessed 2026.

[20] ArduPilot Development Team, "Plane Automatic Landing," ArduPilot Documentation, accessed 2026.

[21] ArduPilot Development Team, "Onboard Message Log Messages," ArduPilot Documentation, accessed 2026.