Comparative Analysis of State Estimation Robustness: Evaluating Dead Reckoning, Simple Filtering, and EKF under Sensor Failure for sUAS Flight Safety

Philippe Kruettli
Ann and H.J. Smead Department of Aerospace Engineering Sciences
University of Colorado Boulder
Boulder, USA
philippe.kruettli@colorado.edu

***Abstract*—As small Uncrewed Aircraft Systems (sUAS) transition from line-of-sight operations to complex autonomous missions, the reliability of state estimation becomes paramount. This paper investigates the comparative performance and robustness of three estimation strategies — Basic Dead Reckoning, a Simple Estimator (low-pass filtered sensor inversion), and an Extended Kalman Filter (EKF) — within a simulated sUAS environment. While modern estimation techniques provide superior accuracy under nominal conditions, their behavior during sudden sensor degradation or even sensor loss of signal remains a critical failure point. This study utilizes the 6-DOF nonlinear simulation environment developed in ASEN 5128 to evaluate how each estimator handles injected sensor failures, specifically GPS signal loss and IMU bias drift. The unique contribution of this research is a quantitative analysis of the "Time-to-Violation" (TTV) of a predefined safe flight corridor. By profiling the divergence rates of each algorithm post-failure, this paper maps the relationship between estimator complexity and the available buffer time for emergency software interrupts or manual intervention. It evaluates and demonstrates the trade-offs between computational overhead and safety margins, providing a framework for designing fault-tolerant navigation software in resource-constrained aircraft systems. Results demonstrate that while the EKF provides superior state stability under nominal conditions, its safety margin advantage narrows significantly during GNSS spoofing, where both the EKF and Simple Estimator exhibit similar Time-to-Violation profiles. This suggests that the added computational complexity of probabilistic filters may not always be justified by the resulting TTV buffer in resource-constrained environments.**

***Index Terms*—sUAS, Dead Reckoning, Low-Pass Filter, Sensor Inversion, Extended Kalman Filter, State Estimation, Sensor Failure, Flight Safety.**

I. Introduction

The operational paradigm for small Uncrewed Aircraft Systems (sUAS) is shifting from simple line-of-sight operations toward high-consequence autonomous missions where maintaining a safe flight envelope during sensor degradation is crucial. Understanding failure modes is essential for developing reliable, fault-tolerant navigation software and integrity of the state estimation represents a critical single point of failure for flight safety.

This research utilizes a 6-DOF nonlinear simulation to answer: *How does estimator complexity affect robustness and time-to-violation (TTV) during sensor degradation or blackout?*. Following expert feedback, the study first frames Dead Reckoning as a baseline that diverges rapidly due to hardware limitations before comparing it to a "SimpleEstimator" (sensor inversion) and a full EKF within a simulated sUAS environment.

A. Background and Motivation

Different sources of error can limit the ability of sensors to pick up correct signals, especially in contested or unpredictable environments. The necessity for robust estimation is critical in three primary scenarios:

- **Malicious Interference**: Intentional jamming or spoofing of GPS signals can provide "truth" data that is fundamentally flawed.
- **Environmental Unavailability**: Signal masking in "urban canyons" or deep valleys, as well as spacecraft occultation, leads to extended GNSS blackouts.
- **Long-Duration Fatigue**: Hardware susceptibility to "soft failures," such as IMU bias drift or thermal noise, slowly degrades the solution.

While the Extended Kalman Filter (EKF) is an industry standard, its reliance on linear approximations and Gaussian noise makes it susceptible to rapid state drift and filter divergence in non-nominal scenarios like sensor blackouts.

For either a human pilot or the flight controller to intervene, a metric to define the time available for intervention is needed. This study introduces the "Time-to-Violation" (TTV) metric. TTV quantifies the operational window available before the aircraft exits a predefined safe flight corridor.

B. Related Work

The challenge of maintaining integrity in a state estimation system for sUAS is well-documented in literature, primarily focusing on the trade-off between algorithm complexity and robustness.

1) Foundations of State Estimation in sUAS: State estimation for small aircraft typically relies on the fusion of low-cost Micro-Electro-Mechanical Systems (MEMS) and Global Navigation Satellite Systems (GNSS). The Extended Kalman Filter (EKF) remains the industry standard due to its ability to handle nonlinear dynamics by linearizing around the current state estimate [1], [2]. However, as noted by Groves, "cheap" MEMS sensors are subject to significant stochastic noise and deterministic biases that complicate simple integration [3]. While the EKF provides optimal estimation under Gaussian noise assumptions, it is susceptible to "filter smearing" when sensors provide non-nominal or biased data.

2) Navigation in Contested and Degraded Environments: Recent research has shifted toward navigation in GNSS-denied or "contested" environments, where primary truth data is compromised or lost. Kissai and Smith analyzed dead reckoning performance in polar regions, illustrating that without the corrective feedback loop of a GNSS signal, the accumulation of raw IMU errors leads to rapid, unconstrained growth in position uncertainty [4]. While an Extended Kalman Filter (EKF) is designed to dampen these errors by estimating sensor biases during nominal flight, its reliance on Gaussian assumptions makes it vulnerable to structured interference. In scenarios involving GPS jamming or spoofing, the EKF may mistakenly incorporate "fundamentally flawed" data into its state estimate, potentially leading to faster divergence than simpler, non-probabilistic estimators [3].

3) Safety Metrics and Robustness: Traditional metrics for judging estimators, such as Root Mean Square Error (RMSE), quantify historical accuracy but fail to describe the future "survivability" of the airframe [2], [3]. While existing research into Fault Detection and Isolation (FDI) focuses on identifying when a sensor has failed, there is limited literature on the specific time-window available for human or automated intervention before a flight envelope violation occurs [3]. This study fills that gap by moving beyond traditional accuracy metrics to the "Time-to-Violation" (TTV) framework. TTV maps the divergence rates of different algorithms against a predefined safe flight corridor, providing a more relevant safety buffer for mission go/no-go decisions in high-risk environments.

C. Paper Roadmap

This paper is organized as follows: Section II describes the estimation methods and the 6-DOF simulation used. Section III details the failure injection methodology. Section IV describes the Time-to-Violation mapping, followed by the results in section V and conclusions in Section VI.

II. Estimation Methods and Simulation

This study leverages the ASEN 5128 6-DOF simulation environment to evaluate estimator performance, comparing different estimator complexities and their respective error profiles. Three distinct estimation strategies are analyzed to determine their robustness under sensor degradation.

A. Basic Dead Reckoning (DR)

Dead reckoning (DR) serves as the primary baseline for navigation by integrating inertial measurement unit sensors (IMU) without the benefit of external corrections, such as GPS. In this method, the aircraft's state is propagated using raw inertial data.

The kinematic equations used to propagate position ($\mathbf{p}$) and orientation ($\mathbf{o}$) over time are defined as:

$$\dot{\mathbf{p}}^E = \mathbf{R}_B^E \cdot \mathbf{v}_B^E \tag{1}$$

$$\dot{\mathbf{o}}_{B/E} = T \cdot \boldsymbol{\omega}_B \tag{2}$$

While DR is a fundamental navigation technique, it inevitably drifts due to the integration of stochastic sensor noise and deterministic biases inherent in "cheap" MEMS hardware [3]. Consequently, this method is expected to diverge rapidly from the true flight path even under nominal conditions [4].

B. Simple Estimator (Low-Pass Filtered Inversion)

Serving as an intermediate baseline, the Simple Estimator utilizes standard autopilot estimation logic. This method applies sensor inversion, deriving state variables directly from sensor measurements, coupled with low-pass filtering to remove high-frequency noise. This approach provides a comparison point between raw integration and complex statistical filtering.

All the sensor inversion equations are described by Beard and McLain in detail, but the general approach is [2]:

$$\hat{\mathbf{x}} = f(LPF(\mathbf{y})) \tag{3}$$

where $\mathbf{y}$ represents the measurement and $f()$ is any function that is applied to it, in order to derive the estimate $\hat{\mathbf{x}}$, such as for altitude or height estimation, this is:

$$\hat{\mathbf{h}} = \sqrt{\frac{2}{\rho} LPF(\mathbf{y}_{pres_dyn})} \tag{4}$$

C. Extended Kalman Filter (EKF)

The EKF implemented in this study represents a cascaded architecture that fuses multi-sensor data to provide optimal estimates under nominal Gaussian noise assumptions [1]. Unlike a monolithic filter, this design utilizes decoupled estimation stages to reduce computational overhead while maintaining statistical rigor for critical states.

State Vectors: The estimator is partitioned into two distinct filters. The attitude filter tracks orientation, while the GPS smoothing filter tracks position, velocity, and wind states:

$$\mathbf{x}_{att} = [\phi, \theta]^T \tag{5}$$

$$\mathbf{x}_{gps} = [p_n, p_e, V_g, \chi, w_n, w_e, \psi]^T \tag{6}$$

Models and Linearization: Each filter utilizes a non-linear process model to describe state evolution and a measurement model to relate sensor readings — such as GPS, accelerometers, and pressure sensors — to the states. Because sUAS dynamics are inherently non-linear, the implementation

employs linear approximations through the computation of Jacobian matrices ($\mathbf{A}$ and $\mathbf{H}$) during the prediction and update steps [2].

Prediction and Update: For each time step T_s, the state and covariance ($\mathbf{P}$) are propagated:

$$\mathbf{P}_{k+1} = \mathbf{P}_k + T_s(\mathbf{A}\mathbf{P}_k + \mathbf{P}_k\mathbf{A}^T + \mathbf{Q}) \tag{7}$$

The measurement update then applies the Kalman gain to correct the state based on the residual between actual sensor data and the measurement model. Despite its accuracy, this EKF structure remains susceptible to "filter smearing" or divergence when sensors provide non-nominal data, such as GPS spoofing or IMU bias drift [3].

D. 6-DOF Simulation Environment

To validate these estimators, the ASEN 5128 non-linear simulation environment is utilized. This environment models the complex flight dynamics of an sUAS, incorporating specific aircraft parameters and environmental constants. The simulation replicates realistic hardware characteristics, including specific sensor noise profiles, to ensure the injected failures accurately reflect real-world flight risks [2].

1) The Ttwistor UAS: The aircraft utilized in the 6-DOF nonlinear simulation environment is the University of Colorado's Ttwistor Unmanned Aircraft. The Ttwistor is a specialized, twin-engine variant of the Tempest UAS, originally designed for high-consequence sensing missions such as supercell thunderstorm penetrations [5].

Airframe and Geometry

The Ttwistor is a high-aspect-ratio fixed-wing platform designed for aerodynamic efficiency and payload stability. The physical and inertial parameters used in this study were directly used from a datasheet provided by Prof. Eric Frew in ASEN 5128 class. The key physical specifications are summarized in Table I.

TABLE I: Ttwistor Physical and Inertial Specifications

Parameter	Symbol	Value	Units
Mass	m	5.74	kg
Wingspan	b	3.067	m
Wing Area	S	0.6282	m^2
Mean Chord	c	0.208	m
Aspect Ratio	AR	14.97	-
Moment of Inertia (X)	I_x	1.201	kg·m^2
Moment of Inertia (Y)	I_y	0.932	kg·m^2
Moment of Inertia (Z)	I_z	2.074	kg·m^2
Gravity	g	9.81	m/s^2

Propulsion and Control Surfaces

The Ttwistor utilizes a twin-engine configuration that provides redundant propulsion and enhanced climb performance. The propulsion system is modeled using the framework provided by Beard and McLain, which assumes zero thrust at zero throttle. The simulated propulsion parameters include a propeller area (S_{prop}) of 0.0707 m^2 and a motor constant (k_{motor}) of 30.

Flight control is achieved through three primary aerodynamic surfaces, each characterized by specific stability derivatives that dictate the vehicle's response to control inputs:

- **Elevator:** Primary longitudinal control with an aerodynamic derivative of $Cm_{\delta e} = -0.06$.
- **Ailerons:** Lateral and roll control with a derivative of $Cl_{\delta a} = -0.02$.
- **Rudder:** Directional and yaw control with a derivative of $Cn_{\delta r} = -0.000856$.

Aerodynamic Stability Derivatives

The 6-DOF simulation environment utilizes nondimensional stability derivatives to characterize the aerodynamic forces and moments acting on the Ttwistor UAS. These values, summarized in Table II, define the "plant model" that the Extended Kalman Filter uses for state prediction. A mismatch between these parameters and the actual flight dynamics (e.g., due to airframe damage) would represent an additional source of estimation error not explored in this specific study.

TABLE II: Nondimensional Stability Derivatives for Ttwistor UAS

Longitudinal		Lateral-Directional	
$C_{L\alpha}$	6.1967	$C_{Y\beta}$	-0.3672
$C_{m\alpha}$	-1.6340	$C_{l\beta}$	-0.0807
C_{Lq}	10.1376	$C_{n\beta}$	0.0806
C_{mq}	-24.3761	C_{Yp}	-0.6499
$C_{L\delta e}$	0.0068	C_{lp}	-0.6866
$C_{m\delta e}$	-0.0600	C_{np}	-0.0394
		C_{Yr}	0.2134
		C_{lr}	0.1197
		C_{nr}	-0.0523
		$C_{Y\delta a}$	-0.0008
		$C_{l\delta a}$	-0.0200
		$C_{n\delta a}$	-0.0001
		$C_{Y\delta r}$	0.0031
		$C_{l\delta r}$	0.0002
		$C_{n\delta r}$	-0.0009

Sensor Suite and Noise Characterization

The robustness of the state estimation algorithms is evaluated based on their response to raw sensor data corrupted by stochastic noise and systematic errors. The simulated sensor suite is based on the physical hardware integrated into the Ttwistor UAS. The noise characteristics for each sensor, summarized in Table III, are derived from the manufacturers' datasheets for the specific MEMS components.

TABLE III: Ttwistor Hardware Sensor Specifications

Sensor Type	Model/Component	Noise (σ)	Units
Accelerometer	ADXL325	$0.0025 \cdot g$	m/s^2
Rate Gyro	ADXRS540	0.13	deg/s
Abs. Pressure	MP3H6115A	0.01	kPa
Dyn. Pressure	MPXV5004G	0.002	kPa
GPS Position (N/E, D)	-	0.21, 0.40	m
GPS Velocity	-	0.01	m/s

E. Failure Injection Methodology

To evaluate the "Time-to-Violation" (TTV) metric, two primary failure modes were injected during the cruise phase of the flight: 1) **GNSS Outage:** A total loss of position and ground-speed updates, forcing the estimators to rely solely on dead reckoning or inertial integration. 2) **IMU Bias Drift:** A linear bias ramp injected into the pitch-rate gyro to simulate thermal instability or sensor degradation.

III. Failure Injection

The primary objective of this research is to move beyond nominal performance and characterize estimator behavior under different failure conditions. To achieve this, specific sensor failures are programmatically injected into the 6-DOF nonlinear simulation environment. These failures are designed to mimic real-world hardware and environmental challenges that sUAS face in complex missions.

A. Failure Modeling Philosophy

The injection framework is designed to evaluate how each estimator — Dead Reckoning, the Simple Estimator, and the EKA — manages to transition from a "nominal" state to a "degraded" state. By systematically introducing these faults, we can profile the divergence rates and calculate the resulting **Time-to-Violation (TTV)** of the safe flight corridor.

B. Nominal Sensor Model

To define failure, we fist establish the nominal measurement model for the onboard sensors (IMU and GPS).

1) Inertial and Pressure Sensors: The IMU and pressure suite measurements are modeled with additive white Gaussian noise and deterministic biases.

Accelerometers & Gyros:

$$\mathbf{y}_{accel} = \frac{f_B}{m} + \eta_{accel} \tag{8}$$

$$\mathbf{y}_{gyro} = \omega_B + \eta_{gyro} \tag{9}$$

where $\eta \sim N(0, \sigma^2)$.

Pressure Sensors:

$$\mathbf{y}_{pres} = \overline{\mathbf{y}} + b_{pres} + \eta_{pres} \tag{10}$$

where $\overline{\mathbf{y}}$ is the pressure derived from altitude (absolute pressure) or airspeed (dynamic pressure), b represents a constant deterministic bias and $\eta \sim N(0, \sigma^2)$.

Full Measurement Vector: The measurement vector provided by the IMU suite is:

$$\mathbf{y}_{inertial} = [y_{accel}, y_{gyro}, y_{pres_abs}, y_{pres_dyn}]^T \tag{11}$$

2) GNSS Model: Unlike the IMU, the GPS position noise v_{gps} is modeled as a persistent Gauss-Markov noise:

$$\mathbf{v}_{gps,k} = \mathbf{k}_{exp}\mathbf{v}_{gps,k-1} + \eta_{gps,k} \tag{12}$$

The resulting measurement for position (p_n, p_e, p_h) is:

$$\mathbf{y}_{pos} = \mathbf{p}_{true} + \mathbf{v}_{gps} \tag{13}$$

Additionally, the GNSS sensor suite also provides ground speed (V_g) and course angle (χ) using the following equations:

$$\mathbf{y}_{V_g} = \mathbf{V}_{\mathbf{g}_{true}} + \eta_{V_g} \tag{14}$$

$$\mathbf{y}_{\chi} = \boldsymbol{\chi}_{true} + \frac{\eta_{\chi}}{V_g} \tag{15}$$

where $\eta \sim N(0, \sigma^2)$ and notably the course angle noise is scaled by $1/V_g$, so heading uncertainty increases as the aircraft slows down.

Full Measurement Vector: The measurement vector provided by the GPS sensor is:

$$\mathbf{y}_{gps} = [p_n, p_e, p_h, V_g, \chi]^T \tag{16}$$

C. Injected Failure Scenarios

Three primary failure types are utilized to represent some of the most common and high-consequence mission risks to sUAS state estimation:

1) GNSS Signal Loss (Blackout): This scenario simulates total unavailability of GNSS signal due to environmental conditions, such as signal masking in urban canyons, deep valleys, or during occultation.

Mechanism: The GPS measurement update is completely severed at a predetermined timestamp t_{fail}:

$$\mathbf{y}_{gps}(t) = \text{NaN}, \forall t \geq t_{fail} \tag{17}$$

Expected Impact: Without the corrective feedback loop of a GNSS signal, estimators must rely solely on inertial propagation. In Dead Reckoning, this leads to rapid, unconstrained growth in position uncertainty due to the lack of stabilizing measurement.

2) GNSS Spoofing (Malicious Interference): Spoofing represents a "hard failure" where the sensor provides fundamentally flawed "truth" data due to intentional jamming or malicious interference.

Mechanism: Instead of losing signal, the GPS measurement is replaced with a false trajectory after a predetermined timestamp t_{fail}. It is modeled as an injected bias function which is added to the persistent noise process:

$$\mathbf{y}_{pos}(t) = \mathbf{p}_{true} + \mathbf{v}_{gps} + \delta\mathbf{p}(t), \forall t \geq t_{fail} \tag{18}$$

where $\delta\mathbf{p}(t)$ represents a constant velocity spoofing attack:

$$\delta\mathbf{p}(t) = \mathbf{v}_{spoof} \cdot (t - t_{fail}) \tag{19}$$

Expected Impact: This tests the "filter smearing" susceptibility of the EKF. Because the EKF treats incoming data with statistical weight, it may mistakenly incorporate flawed data into its state estimate, potentially leading to faster divergence than simpler, non-probabilistic methods.

3) IMU Bias Drift (Long-Duration Fatique): This scenario simulates "soft failures" caused by hardware fatique or thermal noise in low-cost MEMS sensors.

Mechanism: The nominal code assumes zero bias for the IMU, therefore the failure is modeled by introducing a time-varying bias or drift $b(t)$ which is added to the raw accelerometer or gyroscope readings.

$$\mathbf{y}_{accel} = \frac{f_B}{m} + b(t) + \eta_{accel} \tag{20}$$

$$\mathbf{y}_{gyro} = \omega_B + b(t) + \eta_{gyro} \tag{21}$$

The bias drift $b(t)$ is defined as:

$$b(t) = \beta_{IMU} \cdot (t - t_{fail}) \tag{22}$$

Expected Impact: While an EKF is designed to estimate and dampen sensor biases during nominal flight, an unmodeled or accelerating drift can lead to slow, deceitful degradation of the navigation solution. This failure mode is critical for identifying the window available for human or automated intervention before safety limits are exceeded.

D. Simulation Methodology

The methodology follows a three-phase approach within the simulation environment:

- **Baseline Phase:** The aircraft operates under standard atmospheric and nominal sensor noise conditions to establish a performance benchmark for each estimator.
- **Failure Phase:** The failure is injected with varying levels of intensity and duration to simulate different risk levels.
- **Divergence Profiling:** The simulation quantitatively measures the rate at which the estimated state deviates from the ground truth state, providing the data necessary to map the TTV.

IV. Time to Violation - TTV

The core contribution of this research is the formalization of the **Time-to-Violation (TTV)** maetric. Its main goal is quantitatively describing the operational window available to a human or automated pilot before an aircraft exits a predefined safe flight corridor.

A. Formal Definition

TTV is defined as the elapsed time between the onset of a sensor failure (t_{fail}) and the moment the estimated state ($\hat{\boldsymbol{x}}$) or the true state ($\boldsymbol{x}_{true}$) deviates beyond a safety threshold (ϵ). Mathematically, for a given state i (such as lateral position or altitude), TTV is expressed as:

$$\mathbf{TTV}_i = \min\{t : |\boldsymbol{x}_i(t) - \boldsymbol{x}_{ref}(t)| > \epsilon_i\} - t_{fail} \tag{23}$$

where $\boldsymbol{x}_{ref}$ is the commanded trajectory and ϵ_i represents the maximum allowable deviation for that specific state. This metric identifies the critical timestamps at which emergency software interrupts or manual interventions must be triggered to prevent aircraft loss.

B. Operational Significance and Intervention Windows

The utility of TTV lies in its ability to define the time left before a critical violation might occur for different actors in the system:

- **Human-in-the-Loop:** For a pilot, TTV represents the maximum reaction time to recognize a "sensor-mismatch" alarm and take manual control of the aircraft.
- **Autonomous Safeguards:** For flight control software, the TTV dictates the time period before the system must trigger a "Return to Home" (RTH) or an emergency landing sequence.
- **Mission Control Board:** By comparing TTV against the expected mission duration in a contested area, operators can make quantitative mission go/no-go decisions.

C. Safe Flight Corridors

The "corridor" is a multidimensional boundary defined by mission-specific constraints:

- **Spatial Constraints:** Lateral and vertical bounds to prevent collisions with terrain or obstacles.
- **Dynamic Constraints:** Maximum allowable deviations in velocity or attitude to prevent aerodynamic stalls or structural failure.
- **Operational Constraints:** Geofencing limits in urban or contested environments to avoid unauthorized airspace.

D. Simulation Safe Flight Corridor

For the simulation in this research, the safe flight corridor is defined as a multidimensional tube surrounding the commanded trajectory. There are three error boundaries defined to achieve this safe flight corridor.

1) Cross-Track Error ϵ_{ct}: The maximum allowable lateral deviation from the planned path, is set at **10 meters**, which reflects a standard "narrow" corridor for autonomous navigation between obstacles or within geofenced zones.

2) Vertical Error ϵ_h: The altitude ceiling and floor boundaries to prevent Controlled Flight Into Terrain (CFIT) or unauthorized entry into higher-altitude regulated airspace are more strict and therefore set to **5 meters**.

3) Attitude Error ϵ_{att}: If estimates diverge due to sensor failure, the flight controller might induce a "death spiral" or it might stall. Therefore a threshold of **30**° is set for the deviation from the nominal attitude, which would indicate a loss of stable flight control.

During the simulation phase, the TTV is computed by comparing the 6-DOF ground truth against the onboard estimate in real-time. The simulation monitors the state vector at every time step T_s and flags the violation event the moment the boundary condition is met:

$$\text{Violation} = \begin{cases} \sqrt{\Delta p_n^2 + \Delta p_e^2} > \epsilon_{ct} \\ |\Delta p_h| > \epsilon_h \\ |\Delta \phi| \text{ or } |\Delta \theta| > \epsilon_{att} \end{cases} \tag{24}$$

This allows for the profiling of different failure intensities against the resulting safety window.

V. Results and Discussion

A. Baseline error comparison

The baseline error comparison between the three estimators is done on a single 60s simulation run to visualize how each estimator performs without any sensor failures. Figure 1 shows the position error, Figure 2 the velocity error, and Figure 3 the attitude error over time.

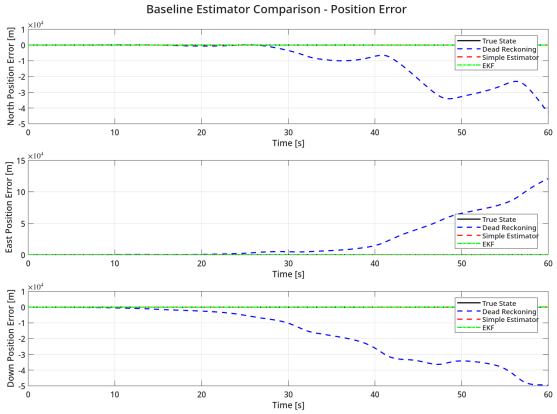


Fig. 1: Baseline position error comparison between Dead Reckoning, Simple Estimator and EKF (True State in black).

Both Simple Estimator and EKF stay very close to the true state across the whole simulation, but Dead Reckoning shows significant errors starting at 20s for North and East position and 10s for Down Position.

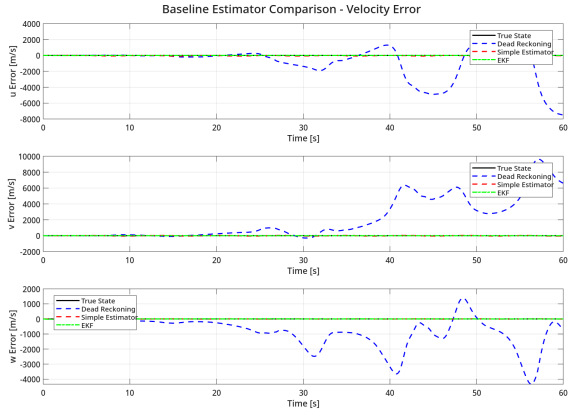


Fig. 2: Baseline velocity error comparison between Dead Reckoning, Simple Estimator and EKF (True State in black).

Much like in Figure 1, Figure 2 shows a similar picture with Simple Estimator and EKF performing well but Dead Reckoning starting to struggle after 10-20s. This suggests — as expected — that Dead Reckoning is an estimator that might not even be reliable enough to use even without sensor failures.

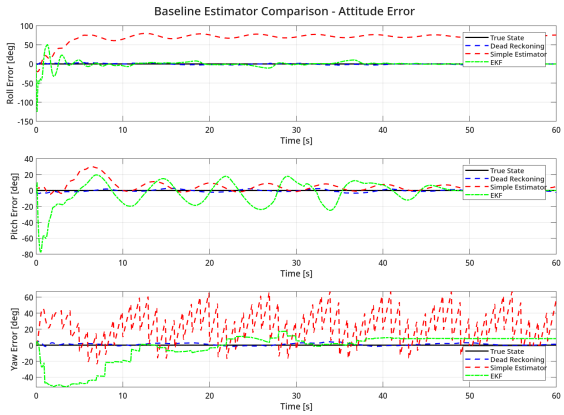


Fig. 3: Baseline attitude error comparison between Dead Reckoning, Simple Estimator and EKF (True State in black).

The Attitude errors are generally much higher, though Dead Reckoning actually performed better than the two other estimators. But what's important for both Simple Estimator and EKF, is that the error did not grow and/or blow up as the simulation continued.

B. TTV Results by fault scenario

Since there are stochastic processes involved in the estimators, the simulation has been run for $N = 100$ times and averaged across all runs to gather significant TTV values. Generally, if an estimator did not violate a boundary within the simulation time, the TTV used is $TTV_{max} = t_{end} - t_{fail} = 60s - 10s = 50s$. This is a reasonable value for a single iteration, but must be considered when averaging all the runs. However, since this only limits the upper boundary for the TTV-metric, the results below can be considered accurate and rather show a worse picture for the estimators, meaning a TTV of $48s$ might actually be higher, since in some cases it might have reached e.g. $75s$. Furthermore, the minimum increase in TTV is given by the simulation timestep as $\Delta TTV_{min} = \Delta t = 0.1s$.

No fault (baseline)

The first scenario to analyze is the baseline, with no failure injection but just letting the simulation run as is. Even in this scenario t_{fail} was subtracted from the actual time of violation, to give a more comparable result (where TTV's are always based on the $t = 10s$ start-time).

As expected, the EKF performed much better than Dead Reckoning and Simple Estimator, not violating either vertical nor attitude boundaries and only violating Cross-Tracking

TABLE IV: Baseline TTV values

	Dead Reckoning	Simple Estimator	EKF
$\mathrm{TTV_{ct}}$	0.10s	0.33s	15.80s
$\mathrm{TTV_{h}}$	0.10s	11.40s	50.00s*
$\mathrm{TTV_{att}}$	43.78s	0.10s	50.00s*

*=Did not violate before end of simulation

boundary after $15.8s$. The simple estimator does a better job at keeping the altitude, only violating it after $11.4s$, whereas the Dead Reckoning simulation immediately violates that boundary. However, Dead Reckoning performs very well in tracking the attitude reasonably for $43.78s$, something the Simple Estimator immediately fails (after $0.1s$). The boundaries seem to be too tight for DR and Simple Estimator to work, even without any injected sensor failures.

Therefore, the Cross-Tracking error boundary was adjusted to $50m$ to simulate a wide or open airspace with the intent to gather more meaningful data in this analysis. This resulted in the following baseline:

TABLE V: Baseline TTV values after safe flight corridor revision

	Dead Reckoning	Simple Estimator	EKF
$\mathrm{TTV_{ct}}$	0.10s	48.31s	50.00s*
$\mathrm{TTV_{h}}$	0.10s	11.30s	50.00s*
$\mathrm{TTV_{att}}$	42.66s	0.10s	50.00s*

*=Did not violate before end of simulation

While EKF was now able to stay within the boundaries for the entire duration of the simulation, the Simple Estimator was now also able to stay within the Cross-Tracking boundary for almost $50s$, all other metrics stayed more or less the same. Still, Dead Reckoning failed immediately in Cross-Tracking as well as Altitude estimation, which goes to show that even with those boundaries, Dead Reckoning is too unreliable to be used.

GNSS Signal Loss (Blackout)

The first simulated sensor failure is total loss of GNSS signal.

TABLE VI: TTV values under GNSS Signal Loss

	Dead Reckoning	Simple Estimator	EKF
$\mathrm{TTV_{ct}}$	0.10s	1.31s	4.66s
$\mathrm{TTV_{h}}$	0.10s	11.21s	50.00s*
$\mathrm{TTV_{att}}$	13.74s	0.10s	49.55s

*=Did not violate before end of simulation

The performance in altitude estimation did not suffer, but that is because altitude is calculated from IMU sensors, not GPS signals. Both the Simple Estimator and the EKF suffered immensely in Cross-Tracking performance from the GNSS signal loss, violating the boundaries after just $1.31s$ and $4.66s$ respectively.

Dead Reckoning's ability to estimate attitude was significantly harmed, resulting in a TTV_{att} lowered by $\sim 75\%$, whereas EKF performed reasonably well in attitude estimation.

GNSS Spoofing (Malicious Interference)

This was tested with various intensities, first $v_{spoof} = 2$:

TABLE VII: TTV values under GNSS Spoofing ($v_{spoof} = 2$)

	Dead Reckoning	Simple Estimator	EKF
$\mathrm{TTV_{ct}}$	0.10s	4.24s	4.84s
$\mathrm{TTV_{h}}$	0.10s	11.30s	50.00s*
$\mathrm{TTV_{att}}$	10.79s	0.10s	50.00s*

*=Did not violate before end of simulation

Next with increased intensity $v_{spoof} = 3$:

TABLE VIII: TTV values under GNSS Spoofing ($v_{spoof} = 3$)

	Dead Reckoning	Simple Estimator	EKF
$\mathrm{TTV_{ct}}$	0.10s	3.62s	4.00s
$\mathrm{TTV_{h}}$	0.10s	11.30s	50.00s*
$\mathrm{TTV_{att}}$	9.80s	0.10s	50.00s*

*=Did not violate before end of simulation

And finally with $v_{spoof} = 5$:

TABLE IX: TTV values under GNSS Spoofing ($v_{spoof} = 5$)

	Dead Reckoning	Simple Estimator	EKF
$\mathrm{TTV_{ct}}$	0.1 s	3.00s	3.00s
$\mathrm{TTV_{h}}$	0.1 s	11.30s	50.00s*
$\mathrm{TTV_{att}}$	10.36s	0.10s	46.46s

*=Did not violate before end of simulation

Again, just as with GNSS Signal Loss, Dead Reckoning's ability to estimate attitude was harmed significantly, though the spoof intensity did not seem to play a major role. While both the Simple Estimator and EKF showed a significant drop in Cross-Tracking performance, the drop for EKF was more significant, such that they essentially performed at a similar or even the same level. EKF's attitude estimation also suffered slightly.

IMU Bias Drift (Long-Duration Fatique)

IMU Bias drift was also tested with various intensities, first $\beta_{IMU} = 0.01$:

TABLE X: TTV values under IMU Bias Drift ($\beta_{IMU} = 0.01$)

	Dead Reckoning	Simple Estimator	EKF
$\mathrm{TTV_{ct}}$	0.10s	48.41s	50.00s*
$\mathrm{TTV_{h}}$	0.10s	11.30s	50.00s*
$\mathrm{TTV_{att}}$	8.75s	0.10s	50.00s*

*=Did not violate before end of simulation

And then with $\beta_{IMU} = 0.05$:

Overall, IMU Bias Drift only seemes to harm attitude estimation, not altitude estimation. Since Simple Estimator's attitude estimation TTV was already low, it was not affected,

TABLE XI: TTV values under IMU Bias Drift ($\beta_{IMU} = 0.05$)

	Dead Reckoning	Simple Estimator	EKF
$\mathrm{TTV_{ct}}$	0.10s	48.81s	50.00s*
$\mathrm{TTV_{h}}$	0.10s	11.20s	50.00s*
$\mathrm{TTV_{att}}$	1.95s	0.10s	30.28s

*=Did not violate before end of simulation

but both Dead Reckoning's and EKF's attitude estimation performance suffered. Dead Reckoning was already impacted by the lower intensity, where the boundary was violated after just $8.75s$ compared to the baseline value of $43.78s$. EKF was not affected at lower intensity, but at higher intensity its TTV dropped by 40%.

Averaged across scenarios

While the results above indicate trends across 100 runs, looking at the TTV performances averaged across all the scenario shows something interesting:

TABLE XII: TTV values averaged across scenarios

	Dead Reckoning	Simple Estimator	EKF
$\mathrm{TTV_{ct}}$	0.100s	22.527s	23.785s
$\mathrm{TTV_{h}}$	0.100s	11.273s	50.000s*
$\mathrm{TTV_{att}}$	14.005s	0.100s	46.613s

*=Did not violate before end of simulation

And if we exclude scenario 1 (baseline) and only average across the fault injection scenarios:

TABLE XIII: TTV values averaged across failure-scenarios

	Dead Reckoning	Simple Estimator	EKF
$\mathrm{TTV_{ct}}$	0.100s	18.229s	19.416s
$\mathrm{TTV_{h}}$	0.100s	11.268s	50.000s*
$\mathrm{TTV_{att}}$	9.230s	0.100s	46.048s

*=Did not violate before end of simulation

While the EKF generally still outperforms not just Dead Reckoning but also the Simple Estimator, the margins are pretty small, indicating that depending on the exact "Safe Flight Corridor" defined and the mission requirements as well as available hardware, one might not benefit all too much from EKF's more complex state estimation. However, one must consider that the current TTV is capped at the total simulation duration, meaning that for longer simulations, EKF might potentially outperform Simple Estimator by higher margins. Furthermore, attitude estimation of Simple Estimator is really bad and would need to be improved.

Single result over time

To show how the error evolves and triggers TTV, a single run with GNSS Spoofing (intensity $v_{spoof} = 5$) is visualized in Figure 4.

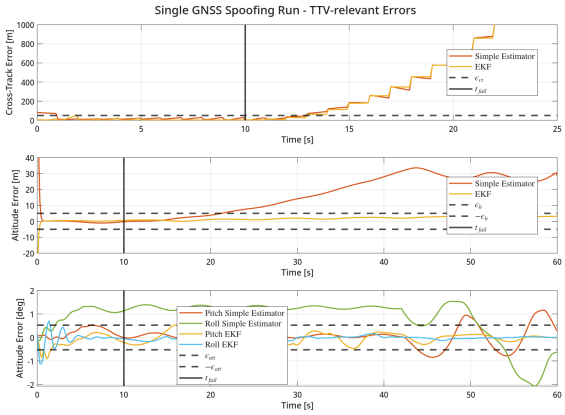
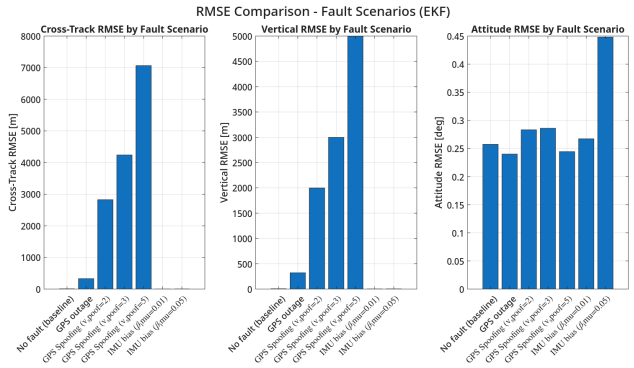


Fig. 4: TTV-relevant errors (Cross-Track, Vertical, Attitude) evolution of a single GNSS Spoofing Run over time ($v_{spoof} = 5$) with the vertical black line showing the start of failure injection (t_{fail}) and the dashed black lines showing the "Safe Flight Corridor" defined.

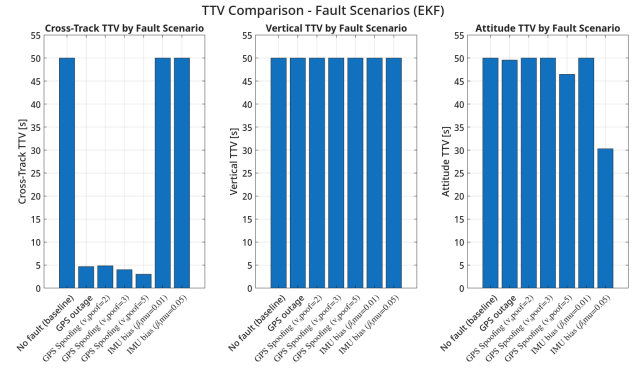
While Cross-Track errors blow up for Simple Estimator and EKF similarly a few seconds after t_{fail}, vertical error or altitde error stays within the boundaries for EKF but not for Simple Estimator. The third plot in Figure 4 most importantly shows that while the attitude errors (except for Roll in Simple Estimator) mostly stay within the bounds, TTV actually measures the first time the boundaries are violated, no matter if the error doesn't blow up afterwords. This allows TTV to capture some information that would be invisible in other metrics like RMSE.

C. Comparison of TTV with RMSE

To compare Root Mean Squared Error (RMSE) metric and the newly developed Time-to-Violation (TTV) metric, the different metric values across the failure scenarios (mentioned in section V-B) are shown in Figure 5a (RMSE) and Figure 5b (TTV) respectively.



(a) RMSE Comparison



(b) TTV Comparison

Fig. 5: Complete Fault Scenario Analysis for the EKF.

These results demonstrate that while RMSE and TTV serve distinct purposes, both are essential for evaluating estimator performance. For instance, Figure 5a shows that the EKF is highly sensitive to spoofing intensity; the RMSE for $v_{spoof} = 5$ is more than double that of $v_{spoof} = 2$. In contrast, the TTV plot in Figure 5b reveals that the time taken to violate safety boundaries remains relatively consistent across these scenarios.

This divergence is even more pronounced in vertical errors (altitude). While the RMSE suggests that GPS spoofing — unlike a simple GNSS signal loss or blackout — leads to severe altitude estimation errors, the TTV plot shows that vertical safety boundaries were actually never breached during the simulation period ($t_{sim} - t_{fail}$).

Figures 6 and 7 further compare the relative safety margins based on RMSE (Figure 6) or TTV (Figure 7). For RMSE, its inverse is used such that a higher value indicates better performance of an estimator.

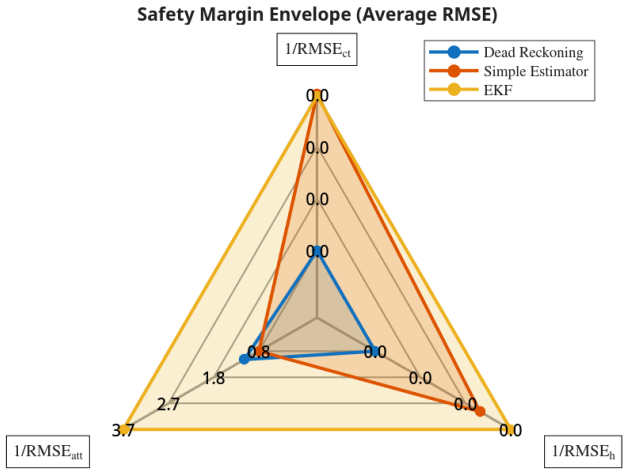


Fig. 6: RMSE-based safety margins comparison between Dead Reckoning, Simple Estimator and EKF using 1/RMSE to show better performance at higher values.

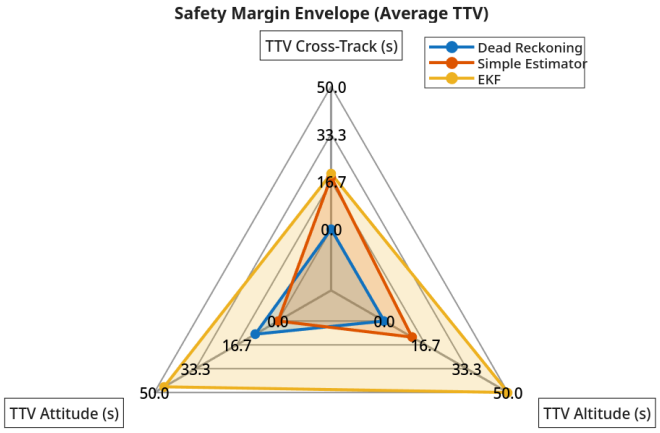


Fig. 7: TTV-based safety margins comparison between Dead Reckoning, Simple Estimator and EKF.

While they both generally show that EKF performs best, followed by Simple Estimator and then Dead Reckoning, the TTV-based plot allows for a more intuitive comparison as the scales and units of all three TTV-metrics is the same, whereas RMSE (or actually the inverse used in these plots) show some relative comparison without allowing the viewer to understand what the meaning behind the values shown actually is. Furthermore, the TTV-based plot clearly shows that while EKF performs very well in attitude and altitude estimation, its performance in Cross-Track estimation is not even close to the maximum. This is not visible in Figure 6's plot.

VI. Conclusion

The research conducted in this study has established a quantitative framework for evaluating the robustness of state estimation methods for small Uncrewed Aircraft Systems (sUAS) through the introduction and application of the Time-to-Violation (TTV) metric. By moving the focus away from traditional, reflection-based accuracy metrics like Root Mean Square Error (RMSE), TTV provides a mission-critical, forward-looking assessment of flight safety. This study successfully mapped the divergence characteristics of three distinct estimation strategies — Dead Reckoning (DR), a Simple Estimator based on Sensor-Inversion, and an Extended Kalman Filter (EKF) — under conditions of sensor degradation, providing a clear picture of the safety buffers available in autonomous flight.

A. Synthesis of Estimator Performance

The comparative analysis revealed that the relationship between estimator complexity and operational safety is non-linear. Under nominal flight conditions, the EKF performed as theoretically expected, providing the highest degree of state stability and the lowest variance across all states. This performance translated to a maximum TTV of 50.00 seconds (the simulation limit). However, the introduction of structured failures, such as GNSS Signal Loss, GNSS Spoofing and IMU bias drift, highlighted the "brittleness" of probabilistic filters such as EKF.

One of the most significant findings was the EKF's response to GNSS spoofing. While the EKF is designed to weight sensors based on their characterized noise covariance, the injection of a persistent, structured bias (spoofing) led to a rapid "pulling" of the state estimate. In these scenarios, the TTV for the EKF dropped to levels nearly identical to the ones of the Simple Estimator. This suggests that without an external integrity monitoring system, the EKF's mathematical complexity does not inherently provide a safety advantage against systematic or even hostile sensor errors. In contrast, the Simple Estimator's lack of historical state integration meant it reacted more "honestly" to the instantaneous sensor data, though it lacked the smoothing required for precise control.

Pure Dead Reckoning was — as expected — mostly unusable even for very short mission durations and without any injected failures.

B. The TTV Metric as a Design Paradigm

A core objective of this research was to validate TTV as a superior or complementary metric for safety-critical autonomous systems. Traditional RMSE-based evaluations often penalize estimators for high-frequency noise that may not actually threaten the vehicle's safety envelope. Throughout the 100-run Monte Carlo simulations, it was observed that an estimator could maintain a high RMSE—specifically in altitude during IMU drift, while still maintaining a safe TTV status.

This leads to an important conclusion for sUAS architecture: safety should be defined by the vehicle's proximity to or violation of its operational boundaries (the "Safe Flight Corridor") rather than its (average) deviation from a theoretical ground truth. By adopting a TTV-centric design philosophy, engineers can build "safety-aware" flight controllers that prioritize different states based on the remaining time-to-violation. For instance, if the cross-track TTV is 2 seconds while the altitude TTV is 30 seconds, the flight controller can prioritize lateral corrections or trigger an emergency hover-in-place command specifically for the lateral channel.

C. Hardware and Resource Considerations

The results also carry weight for resource-constrained platforms. The computational overhead of the EKF, which involves the continuous propagation of the covariance matrix and the calculation of the Kalman Gain, is justified when high-precision navigation is required. However, for low-cost, disposable sUAS platforms where the mission goal is simply to stay within a 50m corridor, the Simple Estimator provided a remarkably similar TTV profile during GNSS outages. This suggests that in specific "low-risk" flight phases, systems could potentially throttle down their estimation complexity to conserve battery life and CPU cycles without resulting in an equivalent loss in the safety buffer.

D. Limitations of this Study

Despite the insights gained, several limitations must be acknowledged. The 6-DOF simulation environment, while robust, assumed a relatively static atmospheric model. Real-world turbulence and wind gusts would likely degrade the performance of Dead Reckoning much faster than seen in these results, as DR lacks the feedback mechanisms to compensate for unmodeled external forces. Furthermore, the "Safe Flight Corridor" was defined as a static boundary; in dynamic environments, such as urban air mobility (UAM), these boundaries would be constantly shifting based on nearby obstacles and other aircraft, requiring a dynamic TTV calculation that was beyond the scope of this work.

This study also could not provide meaningful comparative data when the "Safe Flight Corridor" was too narrow or small. While this overall suggests that in those cases, the usage of more complex estimators like the EKF might be even more important, the effect of sensor failures in such narrow corridors could not be evaluated.

E. Future Work

The findings of this paper open several promising paths for future exploration. The most immediate extension involves the development of a *Hybrid Switching Estimator*. Based on the TTV profiles generated in this study, a system could be designed to monitor the residuals of the EKF. If the residuals indicate a high likelihood of spoofing or catastrophic failure, the system could "hot-swap" to a simplified, high-pass filtered estimator to prevent the EKF from pulling the vehicle out of its flight corridor and to simultaneously conserve battery life.

Secondly, integrating Machine Learning (ML) for anomaly detection presents an interesting opportunity to improve TTV usage. While the EKF struggled to distinguish between high-noise and spoofed data, a Recurrent Neural Network (RNN) trained on nominal flight patterns could potentially identify the ”signature” of a failing sensor before a corridor violation occurs, effectively extending the TTV by providing an earlier warning where TTV could then be used to estimate how long the flight controller or pilot has left to land the sUAS.

F. Final Remarks

In summary, this research has demonstrated that robustness in sUAS navigation is a function of both algorithmic accuracy (captured by RMSE) and the temporal buffer provided during failure states (captured by TTV). While the Extended Kalman Filter remains the superior choice for high-precision tasks, its advantage narrows significantly in the presence of sensor failures. The Time-to-Violation metric serves as a vital bridge between control theory and operational safety, providing a clear, actionable measurement for the next generation of autonomous aerial vehicles. By designing with TTV in mind, we can create systems that are not only more accurate but more resilient to the unpredictable challenges of real-world flight. It furthermore enables Flight Controllers and Mission Designers to evaluate their mission goals against potential risks.

REFERENCES

[1] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton, NJ: Princeton University Press, 2012.

[3] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2nd ed., 2013.

[4] A. Kissai and M. Smith, “Uav dead reckoning with and without using ins/gps integrated system in gps denied polar regions,” *International Journal of Aeronautics and Aerospace Engineering*, vol. 1, no. 2, pp. 58–67, 2019.

[5] J. Roadman, J. Elston, B. Argrow, and E. W. Frew, “Mission performance of the tempest uas in supercell storms,” *AIAA Journal of Aircraft*, 2012.