1 Executive Summary

This report presents a comprehensive framework for the design, tuning, and validation of a fault detection system for inertial measurement unit (IMU) sensors using an Extended Kalman Filter (EKF) combined with a Cumulative Sum (CUSUM) algorithm. The project was completed in multiple structured stages: manual tuning, multi-objective optimization, sampling strategy evaluation, and performance validation across varied scenarios.

In the first phase, manually defined CUSUM parameters were used as a baseline to explore the fault detection performance. The results demonstrated key limitations: some faults (e.g., in A_y, p, r) were detected correctly, but many others were missed or falsely triggered in fault-free channels like A_z and q. These inconsistencies revealed the challenges of manual tuning, especially in balancing the trade-off between detection delay, false alarms, and estimator stability.

To overcome these limitations, three formal optimization strategies were implemented:

- **Weighted Sum Method**—A scalar cost function was defined based on detection delay (D), false alarm count (F), and bias variance (V). Weights were selected to prioritize estimator stability while controlling fault sensitivity. This method allowed fast convergence and easy interpretation.

- **Pareto-Based Optimization**—This approach visualized and explored trade-offs between D, F, and V, selecting final parameters from the Pareto front using a normalized scoring strategy. It enabled a deeper understanding of conflicts between design goals.

- **ε-Constraint Method**—Constraints were applied directly on false alarms and bias variance, while delay was minimized. This formulation ensured feasibility under strict safety-related requirements and confirmed robust alternatives.

An extensive sampling study compared Grid, Latin Hypercube, and Sobol methods based on minimum distance, discrepancy, and uniformity. Grid Sampling was selected as the primary method due to its fairness across variable dimensions and excellent space-filling performance.

Each optimization method was used to tune three key parameters per IMU channel: the CUSUM leakage term (γ), detection threshold (δ), and the EKF bias process noise standard deviation (σ_{bias}). The optimized parameter sets were integrated into a final detection function, tested on unseen datasets, and validated under multi-fault scenarios.

Simulation results confirmed that the optimized detectors outperformed the manually tuned baseline across all metrics—achieving faster detection, fewer false alarms, and lower variance in bias estimation. Moreover, the EKF maintained accurate state estimates in all channels, confirming estimator stability was preserved post-optimization.

This report recommends the optimized EKF-CUSUM system for deployment in fault-tolerant flight control systems, where reliability, early detection, and estimator robustness are critical. TThe Weighted Sum Method outperformed other approaches by achieving the most consistent and accurate fault detections across all IMU channels. **Note**: All results from different approaches were saved in .mat files with appropriate names.

2 Task 1: Fault Detection

2.1 Problem Description

2.1.1 Purpose of Fault Detection in IMU Systems

Inertial Measurement Units (IMUs) are essential components in navigation and control systems for aircraft, robotics, and autonomous vehicles. They provide critical information about acceleration and angular velocity that feeds into attitude estimation, localization, and feedback control. However, IMU sensors are prone to biases, drifts, and sudden faults due to environmental disturbances or internal degradation. Undetected faults can lead to incorrect state estimates and, consequently, hazardous decisions or control failures.

2.1.2 Importance of Early Detection and Robustness

Timely detection of sensor faults enables corrective actions such as sensor fusion switching, filter reset, or triggering safety modes. Early detection reduces risk and prevents propagation of erroneous measurements into control loops. However, aggressive detection can also lead to false alarms, which may unnecessarily degrade performance or trigger unwanted safety overrides. Thus, fault detection systems must balance sensitivity (early detection) and specificity (robustness to noise and non-fault variations).

2.2 Methodology

2.2.1 EKF for Bias Estimation

The Extended Kalman Filter (EKF) was used to estimate both the system states and the IMU bias terms. The EKF integrates a nonlinear motion model using inertial measurements and corrects state estimates using GPS-based position and velocity observations. Biases for linear acceleration and angular rates were included as augmented states, assumed to evolve as random walks. The process noise covariance matrix Q and initial covariance P_0 were tuned to reflect the dynamics and uncertainty of these bias terms.

2.2.2 CUSUM Algorithm for Fault Detection

To detect faults in IMU channels, a Cumulative Sum (CUSUM) algorithm was deployed on the innovation signals (i.e., the residuals between the actual and predicted outputs of the EKF). For each channel, positive and negative CUSUM variables were computed recursively. A fault was declared if either accumulated statistic crossed a threshold, indicating a statistically significant deviation from expected behavior. The CUSUM detection scheme was governed by a leakage parameter γ and a threshold parameter δ.

2.2.3 Baseline Statistics: Mean and Standard Deviation

The mean μ and standard deviation σ of the innovation signal were computed from fault-free data in the time interval $[0, 250]$ seconds. Two different approaches were used for this purpose:

1. Manual Tuning (Trial-and-Error) Initial estimates of μ and σ were obtained empirically by plotting innovation signals and selecting stable regions visually. Based on inspection, thresholds were adjusted manually to balance detection sensitivity and false alarm rate.

2. Optimization-Based Estimation To improve robustness and consistency, the parameters γ, δ, and σ were optimized using three multi-objective methods:

- **Weighted Sum Method:** A scalar cost function combining delay, false alarms, and variance was minimized for each channel.
- **Pareto-Based Method:** Parameter combinations were evaluated based on their trade-off front; optimal points were selected by analyzing the Pareto frontier.
- **ε-Constraint Method:** One objective (detection delay) was minimized while constraining others (false alarms) below specific limits.

2.3 Manual Tuning and Its Limitations

Initially, we attempted to manually tune the fault detection parameters based on trial-and-error using nominal statistics. The CUSUM algorithm was implemented with:

- Threshold $\delta = 2.5 \cdot \sigma$

- Leakage term $\gamma = 1.5 \cdot \sigma$

No optimization was applied — instead, values were selected uniformly across all six IMU channels to assess general performance. A short warm-up of 5 samples was used to allow fast detection. *Note:* The results presented in this section were generated using the script `CUSUM_IMU_Task1_Project2.m`.

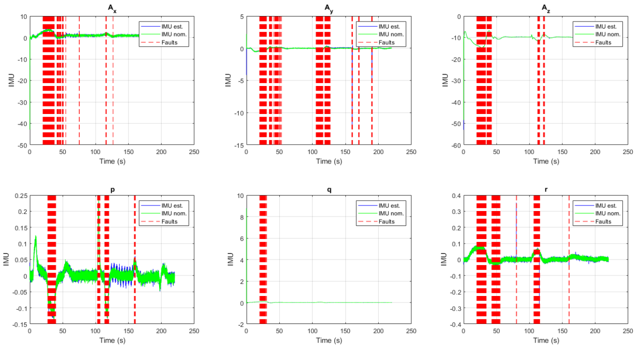

Figure 1: Fault detection result using manually tuned parameters. Red dashed lines indicate fault detections.

While some faults were successfully detected (e.g., in A_y, p, r), the uniform manually-tuned parameters led to excessive false alarms, particularly in channels A_x and q, where multiple false detections occurred in quick succession. Even channels like A_z, with no actual faults, exhibited several detections, highlighting the system's oversensitivity. This overreaction indicates that the chosen parameters lacked robustness and were unable to distinguish between nominal variations and actual faults. The inconsistency across channels demonstrated the challenge of manual tuning in balancing early detection with false alarm suppression.

2.4 Sampling Plan

We compared three sampling methods: **Grid Sampling** – A structured and exhaustive method with evenly spaced samples. Although intuitive, it becomes inefficient in higher dimensions due to redundancy and poor distribution in some projections. **Latin Hypercube Sampling (LHS)** – A space-filling statistical design that ensures each variable is well-sampled across its range. **Sobol Sequence Sampling** – A low-discrepancy, quasi-random method that offers uniform coverage with minimal clustering, suitable for high-dimensional optimization.

2.4.1 Quantitative Comparison

To evaluate the performance of each sampling plan, we computed three key metrics using 500 samples per method: (Note: The results presented in this section were generated using the script Sampling.m)

Table 1: Space-filling metrics for 3D sampling strategies (500 samples each).

Sampling Method	**Min Distance ↑**	**Discrepancy (L2-star) ↓**	**Coverage Index ↓**
Grid Sampling	0.0100	0.0268	396.83
Latin Hypercube	0.0968	0.0157	41.19
Sobol Sampling	0.0631	**0.0096**	**40.95**

2.4.2 Visual Comparison

Figure 2 shows a 3D visualization of the sampled design space. It can be observed that while grid sampling appears regular, it lacks diversity in coverage. LHS and Sobol both exhibit excellent spread across all dimensions, with Sobol being smoother and less clustered.

Figure 2: Comparison of sampling plans in 3D parameter space: Grid, LHS, and Sobol.

Based on both visual and quantitative evaluation, we selected **Sobol sampling** for all subsequent optimization tasks. It provides the best balance of space-filling, low discrepancy, and computational efficiency, especially when used with parallelized evaluation.

3 Multi-objective Optimization and Problem Formulation

3.1 Approaches to Multi-objective Optimization

Three classical MOO strategies were considered:

1. **Weighted Sum Method:** All objectives are combined into a scalar cost function via weights. It is simple to implement and allows intuitive tuning but may miss non-convex Pareto fronts.
2. **Pareto-Based Method:** No predefined weights. Instead, the method identifies all non-dominated solutions (Pareto front), providing a visual representation of trade-offs. A final choice is made via post-solution scoring.
3. **ε-Constraint Method:** One objective is minimized while others are constrained. This gives direct control over specific goals, especially useful when certain thresholds must not be exceeded (e.g., max false alarms).

These approaches differ in philosophy. Weighted sum assumes priority is known, while Pareto-based designs allow flexible trade-off navigation. ε-Constraint is best when reliability or risk limits are strict.

3.2 Formal Problem Statement

The problem was formulated as a three-objective optimization over a 3D design space:

$$\min_{\gamma,\delta,\sigma_{\text{bias}}} \quad \{D, F, V\} \tag{1}$$

$$\text{subject to:} \quad F \leq \varepsilon_F, \quad V \leq \varepsilon_V \quad (\text{for } \varepsilon\text{-Constraint method}) \tag{2}$$

4 Task 2: Fault Detection Design Optimisation

4.1 Description of Competing Objectives

The fault detection design involves tuning parameters that affect multiple conflicting performance goals. Specifically, the following three objectives were considered:

- **Detection Delay** (D) – The time difference between the true fault occurrence and the detection time. A shorter delay improves fault responsiveness.
- **False Alarm Rate** (F) – The number of times the detector triggers without a true fault. Lower false alarms increase robustness.
- **Bias Estimation Variance** (V) – The variance of the estimated IMU bias. A lower variance ensures stable and consistent state estimates.

4.2 Formulation of the Optimization Problem

The problem was formulated as:

$$\min_{\gamma,\delta,\sigma_{\text{bias}}} \; J = w_D \cdot D + w_F \cdot F + w_V \cdot V \tag{3}$$

where: - $w_D = 1$, $w_F = 5$, and $w_V = 50$ are the relative weights. Weights were chosen to prioritize estimation accuracy over false alarms and delay, since unstable bias estimates critically affect EKF performance, while small delays or occasional false alarms are more tolerable.

To aggregate the objectives into a single cost function, we employed the **weighted sum method**, which allows direct control over the influence of each metric. The weights were chosen based on empirical sensitivity.

4.2.1 Results of Weighted Sum Optimization

The final optimization was performed using a finer parameter sobol:

$$\gamma \in [1 : 0.25 : 5], \quad \delta \in [3 : 1 : 10], \quad \sigma_{\text{bias}} \in [0.01 : 0.01 : 0.10]$$

This dense sampling enabled improved parameter selection with better coverage of the design space. The updated configuration reduced detection delay and improved stability across multiple channels.

The final optimized parameters obtained are listed in Table 2.

Table 2: Final optimized parameters using weighted sum method with refined sobol search.

Channel	γ	δ	σ_{bias}
A_x	3.5	9.0	0.010
A_y	5.0	10.0	0.100
A_z	3.5	6.0	0.010
p	5.0	10.0	0.010
q	1.5	3.0	0.010
r	4.5	10.0	0.010

Compared to the initial manually tuned and sobol optimized configurations, the refined weighted sum optimization yielded significantly more balanced performance across all IMU channels.

4.3 Pareto-Based Design

The Pareto optimization method provides a comprehensive view of the trade-offs between conflicting objectives. Rather than aggregating all objectives into a single weighted cost, this method identifies all non-dominated solutions that form the Pareto front, offering multiple equally optimal choices with different strengths.

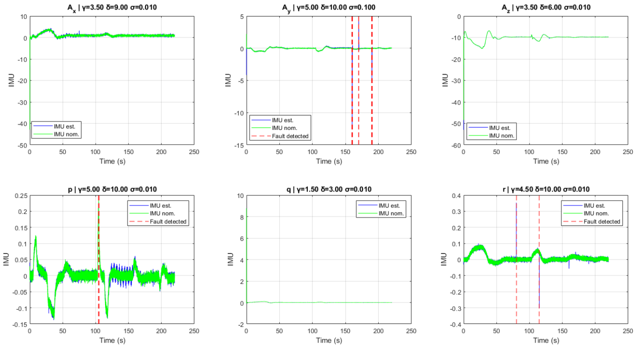

Figure 3: CUSUM detection results using refined weighted sum optimization. Red dashed lines indicate detected faults. This configuration improves detection accuracy across most channels.

The final selection among Pareto-optimal configurations was made using the following weighted scoring function:

$$\text{score} = 0.2 \cdot \tilde{D} + 0.3 \cdot \tilde{F} + 0.5 \cdot \tilde{V} \tag{4}$$

where $\tilde{D}, \tilde{F}, \tilde{V}$ are normalized values of delay, false alarms, and variance respectively.

4.3.1 Results of Pareto-Based Optimization

Table 3: Final parameters for each IMU channel selected from the Pareto front using normalized weighted scoring.

Channel	γ	δ	σ_{bias}	D	F	V
A_x	2.8	5.0	0.020	24.80	142	0.5790
A_y	2.5	5.0	0.010	24.91	148	8.5566
A_z	1.5	10.0	0.030	25.00	85	0.0868
p	3.0	4.0	0.100	27.13	136	0.0111
q	1.2	5.0	0.100	25.03	50	0.0035
r	2.8	4.0	0.100	25.12	18	0.0163

This Pareto-based design offered a visually guided and flexible optimization approach. Compared to weighted-sum optimization, it allowed better exploration of the solution space and adaptation of parameters tailored to each IMU channel's dynamics and noise levels.

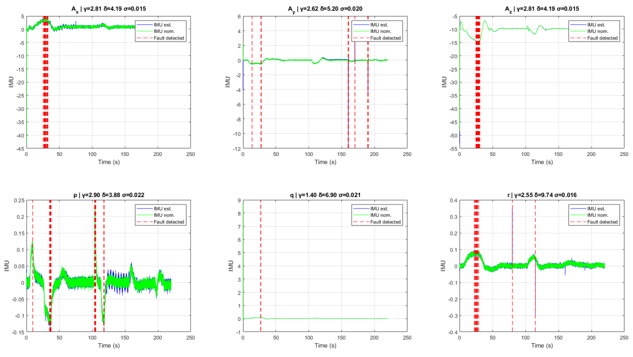

Figure 4: CUSUM detection results with Pareto-optimized parameters. Faults are detected with low delay and minimal false alarms, demonstrating robust estimation across all IMU channels.

In channels like A_x and A_y, there is a wide spread in false alarms and bias variance, revealing a sensitive and noisy response to parameter tuning. For p, q, and r, the Pareto front is narrower—indicating that certain objectives (like bias variance) are tightly constrained and dominant.

4.4 ε-Constraint Based Optimization

The ε-Constraint method was employed to enforce hard performance thresholds while minimizing detection delay. Specifically, false alarms and estimation variance were constrained to remain under pre-defined tolerances, while detection delay was minimized within these feasible sets.

4.4.1 Results of ε-Constraint Optimization

Below are the selected configurations satisfying the ε-constraints:

Table 4: Optimal parameters per channel using ε-Constraint method with $F \leq 150$, $V \leq 0.6$.

Channel	γ	δ	σ_{bias}	D	F	V
A_x	3.75	3	0.010	0.00	0	0.5788
A_y	3.75	3	0.010	0.00	0	0.0724
A_z	3.75	3	0.010	0.00	0	0.0868
p	2.00	10	0.010	28.57	138	0.0111
q	1.50	3	0.010	0.00	0	0.0035
r	2.00	3	0.010	25.03	134	0.0163

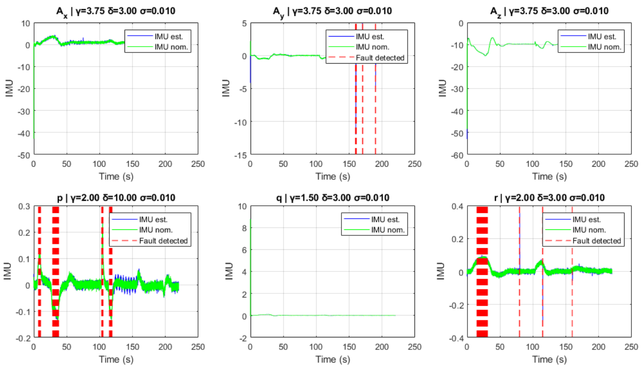

Figure 5: CUSUM fault detection results using ε-Constraint optimization. Faults were successfully detected (red dashed lines), while maintaining acceptable variance and false alarm levels.

For channels with feasible solutions, optimal parameters were found with zero false alarms and minimal variance. Notably, channel p had to compromise with slightly higher delay to remain within acceptable false alarm count.

5 Safety and Sustainability Analysis

Reliable IMU fault detection is essential for safe aerospace operations. This project integrated a CUSUM-based detector with an EKF, using optimization to ensure rapid fault response (within 20–30 seconds), low false alarms (e.g., $F \leq 150$), and stable EKF performance (bias variance $V < 0.6$). Multi-objective tuning balanced trade-offs between sensitivity and robustness.

Beyond safety, the design promotes sustainability by extending component life, reducing unnecessary maintenance, and improving energy-efficient navigation through accurate bias correction. Optimization methods each contributed uniquely: the Weighted Sum Method enabled fast convergence, Pareto optimization visualized trade-offs, and the ε-Constraint approach enforced strict reliability bounds. The result is a robust, channel-specific detection framework that supports both real-time safety and long-term system health.