

# Detecting Slowly Accumulating Faults Using a Bank of Cumulative Innovations Monitors in Kalman Filters

John D. Quartararo\* | Steven E. Langel

Electronic Systems Innovation Center,  
The MITRE Corporation, Bedford,  
MA, USA

## Correspondence

\*J. D. Quartararo, Electronic Systems  
Innovation Center, The MITRE  
Corporation, Bedford, MA, USA  
Email: [johndq@gmail.com](mailto:johndq@gmail.com)

## Summary

Extended Kalman filters (EKFs) that monitor innovations over time have been demonstrated to be effective at detecting slowly accumulating measurement faults (Quartararo & Langel, 2020; Tanil et al., 2018). This paper first demonstrates that a single cumulative monitor becomes increasingly sensitive to measurement error model uncertainty as the accumulation interval increases, leading to false alarm and detection rates that can differ significantly from predefined design parameters. In response, a bank of finite-length cumulative innovations monitors is explored for fault detection in multisensor navigation systems. A novel extension to traditional covariance analysis (*Covariance Analysis Including Expected Values* or CAIEV) is developed to accommodate measurement faults and is used in addition to Monte Carlo simulations to present detection results for a variety of GPS fault profiles and inertial measurement unit (IMU) grades. Data for time-to-detect is presented alongside the position-domain bias induced by the fault at the time of detection. We show that the monitor bank can reliably detect the presence of faulty measurements after the position-domain bias has reached only tens of meters using tactical and aviation-grade IMUs for the cases considered, an improvement over other innovations-based techniques.

## Keywords

anti-spoofing, fault detection, GNSS, GPS, inertial, innovations monitoring, Kalman filtering, sensor fusion

## 1 | INTRODUCTION

Modern navigation systems are increasingly utilizing multiple sensors for positioning, navigation, and timing (PNT), most often beginning with an inertial measurement unit (IMU) and a global navigation satellite system (GNSS) receiver. The extended Kalman filter (EKF) remains the most widely used filter for multisensor fusion, and much effort has been devoted to designing EKFs for operation under nominal conditions. However, additional sensors provide more than improved accuracy and availability of navigation. They can also be cross-checked against each other to detect the presence of, and exclude, faulty measurements. This paper focuses on quantifying the ability of an IMU to detect slowly accumulating Global Positioning System (GPS) measurement faults. We use the GPS/inertial sensor

suite for demonstrative purposes, but the techniques developed in the paper extend naturally to EKF that utilize any number and type of sensors.

Numerous methods have been developed for fault detection and exclusion in navigation systems. Autonomous integrity monitoring methods are developed in Lee (1988) and Sturza (1988) that can detect a single satellite fault by leveraging the redundancy of measurements from multiple GNSS satellites. Innovations and residuals-based fault detection algorithms are commonly used in navigation for detecting multiple satellite faults and have been studied extensively (see Mehra and Peschon [1971], Joerger and Pervan [2013], and Crespillo et al., [2017]). However, the previous references only considered *snapshot* implementations, in which detection test statistics were defined solely based on current measurements. While snapshot approaches are effective at detecting abrupt faults, they have difficulty detecting persistent faults that are small at any one epoch but eventually accumulate to a large magnitude. For robust protection against slowly accumulating faults that could be present on multiple satellite signals, cumulative fault detection methods offer a promising solution.

Some examples of slowly accumulating measurement faults may include GPS satellite clock anomalies and incorrect orbit ephemeris parameters (Bhatti & Ochieng, 2007). Because it can be difficult to model the behavior of these faults over time, detection algorithms that make no assumptions regarding the temporal behavior of measurement faults are often employed. In Tanil et al. (2018) and Diesel and Luu (1995), monitors were developed that accumulated innovations over time to provide the capability of detecting slowly growing GPS measurement faults. Innovations monitors are attractive because they can naturally leverage sensor and input information that is not necessarily a complete position estimate (barometric altimeter measurements, a partial [two-axis] IMU, fewer than four GNSS pseudorange measurements from a different constellation, etc.).

Solution separation is a different but related technique that can also be used for fault detection. Advanced Receiver Autonomous Integrity Monitoring (ARAIM; Blanch et al., 2007; WG-C ARAIM Subgroup, 2016) is one example where faults are detected by comparing solutions from different subsets of available GNSS signals. Recently, Kujur et al. (2020) used the discrepancy between a GPS/IMU fused position estimate and an IMU-only position as a test statistic to detect faults. However, dead-reckoning solutions (e.g., inertial-only) become less accurate with time and therefore there is a certain time window beyond which absolute measurements must be used to correct the otherwise unbounded error growth. Solution separation implementations that address faults in non-GNSS sensor measurement and in Kalman filter dynamic models are sparse.

A practical but often overlooked aspect of innovation-based fault detection is the impact of real-world model uncertainty. A main contribution of this work is the conclusion that the cumulative innovations monitor becomes increasingly sensitive to model uncertainty as the accumulation interval increases, a topic which was recently explored in Kujur et al. (2019). In response, this work proposes a novel implementation of innovations monitoring that uses a bank of finite-length cumulative monitors to mitigate the adverse effects of real-world model uncertainty. These effects become important for longer-duration runtimes which may last for tens of minutes or hours and the monitor bank provides a practical solution to this problem.

A complicating factor in the analysis of EKF fault detection performance is that undetected faults can induce biases in the estimate error that produce a significant discrepancy between the true and estimated state vectors. It is not obvious how to account for this effect in EKF covariance analysis. Traditional methods are

generally not sufficient because they linearize the measurement and state dynamic models about the true state. A large discrepancy between the true and estimated states means that several terms in the Taylor series could be required to form an accurate approximation. In response, we derived a novel extension of traditional covariance analysis called *Covariance Analysis Including Expected Values* (CAIEV) that makes it possible to analyze EKFs in the presence of measurement faults. We used this approach together with a high-fidelity Monte Carlo simulation testbed to demonstrate the fault detection performance of the monitor bank.

## 2 | MOTIVATION FOR MULTI-MONITOR APPROACH

This section motivates the use of multiple monitors for fault detection by first exploring the practical issues associated with a single cumulative monitor. The cumulative monitor falls in the class of chi-square monitors, meaning that its test statistic under fault-free conditions is a chi-square random variable. It will be shown later that the test statistic is formed by accumulating inner products of Gaussian random vectors weighted by the inverse of their covariance matrix. In an EKF, the covariance matrix is determined based on models of the underlying system, which are never precisely accurate. As a result, the test statistic is not truly chi-square distributed under fault-free conditions. Rife (2013) provides a general treatment of how chi-square monitors are impacted by covariance matrix uncertainty. This section uses a simple model where uncertainty is defined by a single parameter  $\alpha$  to show that the cumulative monitor becomes more sensitive to uncertainty as the accumulation interval increases.

Let  $\mathbf{x}_k \in \mathbb{R}^m$  be a vector of independent zero-mean Gaussian random variables at time index  $k$  with covariance matrix  $\mathbf{P} = \sigma^2 \mathbf{I}_m$ , where  $\mathbf{I}_m$  is the  $m$ -dimensional identity matrix. Assume that  $\mathbf{P}$  is the same for all  $k$ . Then the cumulative monitor test statistic at time index  $k$  can be written as:

$$s_k = \mathbf{x}_1^T \mathbf{P}^{-1} \mathbf{x}_1 + \dots + \mathbf{x}_k^T \mathbf{P}^{-1} \mathbf{x}_k = \frac{1}{\sigma^2} (\mathbf{x}_1^T \mathbf{x}_1 + \dots + \mathbf{x}_k^T \mathbf{x}_k) \quad (1)$$

When  $\sigma^2$  is known precisely,  $s_k$  is nominally chi-square distributed with  $mk$  degrees of freedom. Now suppose that the covariance matrix  $\mathbf{P} = \alpha \sigma^2 \mathbf{I}_m$  is used instead of  $\mathbf{P}$ , where  $\alpha$  is a scalar. Then the resulting test statistic can be written as  $\bar{s}_k = \alpha^{-1} s_k$ . For  $\alpha \neq 1$ ,  $\bar{s}_k$  is a scalar multiple of a central chi-square random variable, which is known to follow a gamma distribution (Coelho, 2020). Of course, the fact that  $\alpha$  is likely not equal to one cannot be exploited because  $\alpha$  is unknown. Therefore, a monitor threshold  $T_k$  can only be determined for a desired false alarm probability under the assumption that  $\bar{s}_k$  is chi-square distributed. Nevertheless, it is possible to use the gamma distribution to assess the impact that  $\alpha \neq 1$  has on the monitor's true probability of false alarm  $P_{\text{fa},k}$ . From the gamma *cumulative distribution function* (CDF; Abramowitz et al., 1988):

$$P_{\text{fa},k} = P(\bar{s}_k > T_k) = \Gamma\left(\frac{T_k \alpha}{2}, \frac{mk}{2}\right) \quad (2)$$

where  $\Gamma(x, a) = \frac{1}{\Gamma(a)} \int_x^\infty t^{a-1} e^{-t} dt$  is the upper incomplete gamma function. Suppose that  $m = 8$  and that  $T_k$  is determined based on a desired false alarm probability of  $10^{-4}$ . Figure 1 shows  $P_{\text{fa},k}$  as a function of the time index  $k$  for different values of  $\alpha$ .

There is no uncertainty when  $\alpha = 1$ , and the probability of false alarm is equal to its desired value of  $10^{-4}$ . Notice how the sensitivity to uncertainty increases with

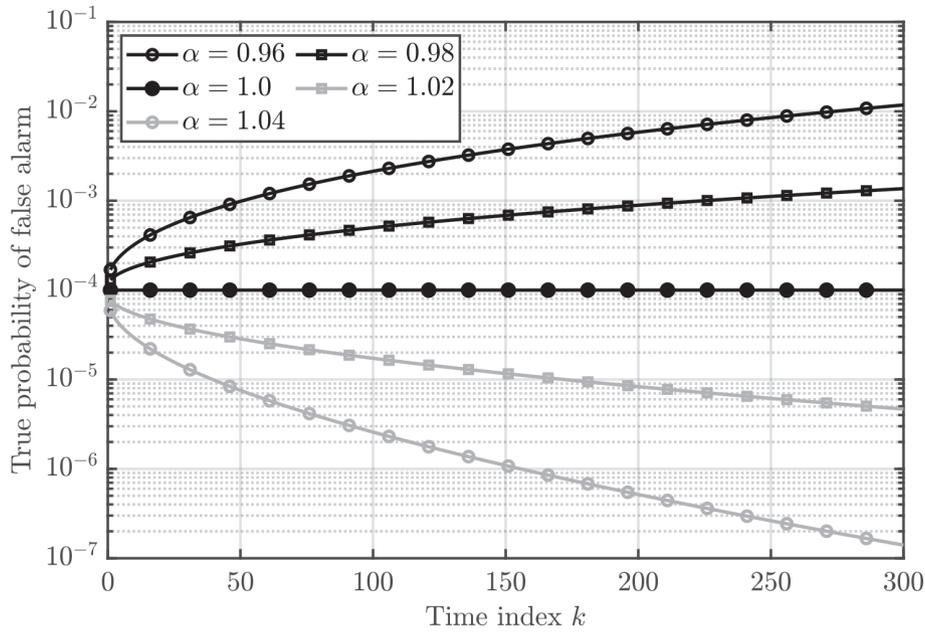


FIGURE 1 Effect of model uncertainty on the true probability of false alarm for a cumulative chi-square monitor

the number of accumulations. Even for modest values like  $\alpha = 1.04$  and  $\alpha = 0.96$ , compared to its desired value, the probability of false alarm is three orders of magnitude less in the former case and two orders of magnitude greater in the latter case. The discrepancy will continue to grow with the number of accumulations. Given this behavior for small levels of uncertainty likely to be encountered in practice, one should expect a monitor that accumulates indefinitely to either far exceed its desired performance or fall drastically short. Therefore, from a practical perspective, it is desirable to limit the number of accumulations in the monitor.

It is also worth pointing out the detection behavior of the cumulative monitor. Faulty measurements induce a bias  $\boldsymbol{\mu}_k$  in  $\boldsymbol{x}_k$ , resulting in a test statistic that is non-centrally chi-square distributed with noncentrality parameter  $\lambda_k = \boldsymbol{\mu}_k^T \mathbf{P}^{-1} \boldsymbol{\mu}_k$ . To assess the effect that accumulation can have on the detectability of faults, we focus on the specific case where a Kalman filter, applied to a time invariant linear system, has reached steady state. Under these conditions, a sequence of faulty measurements produces the same sequence of noncentrality parameters, regardless of when faulty measurements first appear. While this is not true in general for a transient filter or one operating on a time-varying system, considering this special case will highlight the additional adverse effect accumulation can have on fault detection.

In detection analysis, it is common to quantify the instantaneous *minimum detectable error* (MDE), which for us is the smallest value of  $\lambda$  at a given epoch that can be detected with a specified probability. Consider the same monitor described earlier where  $m = 8$  and the threshold is set based on a desired false alarm probability of  $10^{-4}$ . Figure 2 depicts MDE as a function of the time index  $k$  for a detection probability of 0.999. The MDE increases with the accumulation interval (i.e., the monitor becomes less sensitive to faults over time). Depending on when a sequence of faulty measurements begins, the cumulative monitor may or may not be able to detect its presence. For example, suppose that faulty measurements begin appearing at time index  $k = 50$  that result in a noncentrality parameter

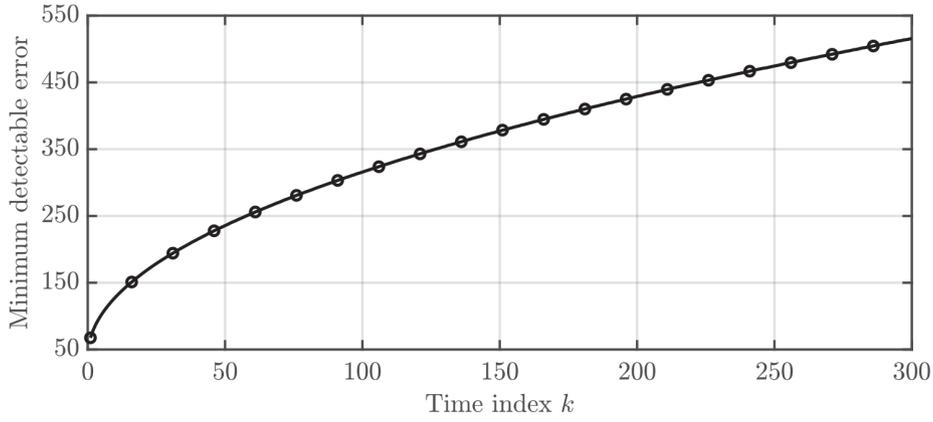


FIGURE 2 Minimum detectable error for a cumulative chi-square monitor

$\lambda = 250$ . Figure 2 shows that the cumulative monitor would be able to detect the presence of faulty measurements with a probability greater than 0.999. However, if the faulty sequence instead began appearing at time index  $k = 200$ , the monitor would not be able to detect the faults with high probability. It would therefore be beneficial to allow the start time of the monitor to vary.

Figures 1 and 2 demonstrate that: a) it is desirable to limit the accumulation interval of the monitor, and b) it is advantageous to consider monitors with varying start times to increase the probability of detection. One way to achieve these two goals is to utilize a bank of chi-square monitors, each with its own unique accumulation interval and start time. This paper explores the design, implementation, and performance of such an approach for fault detection in EKF.

### 3 | ALGORITHM DESCRIPTIONS

#### 3.1 | Problem Setup

This paper is concerned with the nonlinear estimation problem:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) \end{aligned} \quad (3)$$

where  $\mathbf{x}_k$  is the state vector,  $\mathbf{u}_k$  is a known input vector,  $\mathbf{z}_k \in \mathbb{R}^m$  is the measurement vector and  $\mathbf{w}_k, \mathbf{v}_k$  are zero-mean process and measurement noise vectors, respectively. An estimate of the state vector is obtained using an EKF, which is defined by the recursion shown in Table 1 (Brown & Hwang, 2012).

TABLE 1  
Extended Kalman filter algorithm

Time Update	Measurement Update
$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \mathbf{0})$ $\mathbf{P}_{k+1}^- = \mathbf{F}_k \mathbf{P}_k^+ \mathbf{F}_k^T + \mathbf{G}_k \mathbf{W}_k \mathbf{G}_k^T$	$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{J}_k \mathbf{V}_k \mathbf{J}_k^T)^{-1}$ $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, \mathbf{0})]$ $\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$

$\mathbf{F}_k$  and  $\mathbf{G}_k$  are the Jacobians of  $\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)$  with respect to  $\mathbf{x}_k$  and  $\mathbf{w}_k$ , and  $\mathbf{H}_k, \mathbf{J}_k$  are the Jacobians of  $\mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)$  with respect to  $\mathbf{x}_k$  and  $\mathbf{v}_k$ . The matrices  $\mathbf{W}_k$  and  $\mathbf{V}_k$  are the process and measurement noise covariance matrices, respectively.

A reality in most estimation applications is that, at times, the measurement vector  $\mathbf{z}_k$  does not conform to its predefined model. That is, the measurement presents as  $\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) + \mathbf{d}_k$  with  $\mathbf{d}_k$  being an unknown fault vector. Assimilating a faulty measurement in the EKF induces a bias in the state vector estimate that is generally not captured by the covariance matrix. Therefore, it is critical to detect when a measurement is faulty so that the EKF does not incorporate the measurement.

### 3.2 | Snapshot Innovations Monitor

The *innovations vector*  $\mathbf{y}_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, \mathbf{0})$  in Table 1 can be used for fault detection. It is shown in Todling (2000) and Grewal and Andrews (2014) that the covariance matrix of  $\mathbf{y}_k$  is  $\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{J}_k \mathbf{V}_k \mathbf{J}_k^T$ . The *snapshot innovations monitor* forms the test statistic  $s_k = \mathbf{y}_k^T \mathbf{S}_k^{-1} \mathbf{y}_k$  at each measurement epoch, which is chi-square distributed with  $m$  degrees of freedom (DOF) under fault-free conditions. For a given  $P_{\text{fa}}$ , the detection threshold  $T_k$  is set using the inverse chi-square CDF where the number of DOFs  $m$  can vary with the number of measurements at each time step (e.g., number of GNSS satellites in view). That is:

$$T_k = F_{\chi_m^2}^{-1}(1 - P_{\text{fa}}) \quad (4)$$

Normally, the EKF is an unbiased estimator to first order, meaning that the estimate error vector  $\boldsymbol{\varepsilon}_k^-$  is zero-mean. When measurement faults are incorporated into the filter,  $\boldsymbol{\varepsilon}_k^-$  is biased. Therefore, the mean of  $\mathbf{y}_k$  under faulted conditions can be expressed as  $\boldsymbol{\mu}_k = \mathbf{H}_k E[\boldsymbol{\varepsilon}_k^-] + \mathbf{d}_k$  and the test statistic becomes noncentrally chi-square distributed with  $m$  degrees of freedom and noncentrality parameter  $\lambda_k = \boldsymbol{\mu}_k^T \mathbf{S}_k^{-1} \boldsymbol{\mu}_k$ . In this case, the probability of missed detection is  $P_{\text{md}} = F_{\chi_m^2; \lambda}^{-1}(T_k)$ , such that  $F_{\chi_m^2; \lambda}^{-1}$  is the noncentral chi-square inverse CDF.

### 3.3 | Infinite Horizon (IH) Innovations Monitor

To protect a navigation system from slowly growing faults, it is necessary to observe the behavior of the measurements over time. A natural way to do this in the context of innovations monitoring is to accumulate normalized innovations. We consider the approach in Tanil et al. (2018) in which the cumulative test statistic is defined as the sum of normalized innovations from time index one to time index  $k$ . That is, a summation of normalized test statistics over the last  $k$  measurement update epochs:

$$s_k = \sum_{i=1}^k \mathbf{y}_i^T \mathbf{S}_i^{-1} \mathbf{y}_i \quad (5)$$

The innovations vector is uncorrelated through time (Todling, 2000; Stengel, 2012), meaning that  $s_k$  is also a chi-square random variable. Therefore, design and analysis of the *infinite horizon* (IH) monitor (i.e., setting thresholds and predicting  $P_{\text{md}}$ ) is analogous to the design and analysis of a snapshot monitor. The number of degrees of freedom that define the threshold is now  $mk$  and the noncentrality parameter that governs detection performance is  $\lambda_k = \sum_{i=1}^k \boldsymbol{\mu}_i^T \mathbf{S}_i^{-1} \boldsymbol{\mu}_i$ . Given that the

degrees of freedom increase ad infinitum, we expect the IH monitor’s performance to become increasingly sensitive to model uncertainty with time (see Figure 1).

### 3.4 | FIND Algorithm

In Section 2, we suggested that a bank of finite length cumulative innovations monitors could circumvent the practical deficiencies of the IH monitor. Another practical benefit of a bank of differently sized windows would be the ability to detect accumulating fault profiles that occur over different time scales. From this point on the paper refers to the bank of finite cumulative monitors as the *Faulty Innovations Detector* or the FIND algorithm. Let the integer  $N$  denote the number of cumulative monitors, in addition to the snapshot monitor which is always included to catch abrupt faults. Figure 3 shows one example of a monitor bank, which consists of four cumulative monitors and one snapshot monitor. The first step of the FIND algorithm is to store the weighted norm of the innovations vectors over  $NB$  measurement epochs, where  $B$  is an integer configuration parameter. Next, we break up this time interval into  $N$  intervals, each spanning an integer multiple of  $B$  measurement epochs. Lastly, we form the triangular  $N$ -monitor structure shown in Figure 3.

Referring to the example in Figure 3, the normalized innovations at the current time ( $m_0$ ) are subjected to a standard snapshot innovations-based fault detector. The current ( $m_0$ ) and most recent previous normalized innovations ( $m_{-1}$ ) are summed into the first cumulative monitor (the most recent two since  $B = 2$ ). The second cumulative monitor in the bank sums the four (i.e.,  $2B$ ) most recent measurements ( $m_{-3:0}$ ), the next the six (i.e.,  $3B$ ) most recent measurements ( $m_{-5:0}$ ), and so on. Each constituent monitor’s threshold is derived from its allocated  $P_{fa}$  (which we will discuss shortly) and the number of degrees of freedom in its test statistic, which can vary if the number of measurements varies at each time step (e.g., due to a varying number of GNSS satellites over time).

A larger product  $NB$  leads to a monitor bank that covers a longer span of measurement time history, and smaller values of the step size  $B$  produce a set of monitors with finer granularity from the short to long durations. Selection of  $N$  and  $B$  are application dependent. Longer time histories allow for the detection of slower, more pernicious fault accumulations and, while smaller step sizes can increase the computation load (i.e., more monitors), they may provide better coverage across a wider variety of fault time scales. These are trade-offs that would have to be analyzed in the context of a specific use case, such as the detection performance

Time-series of measurement updates:

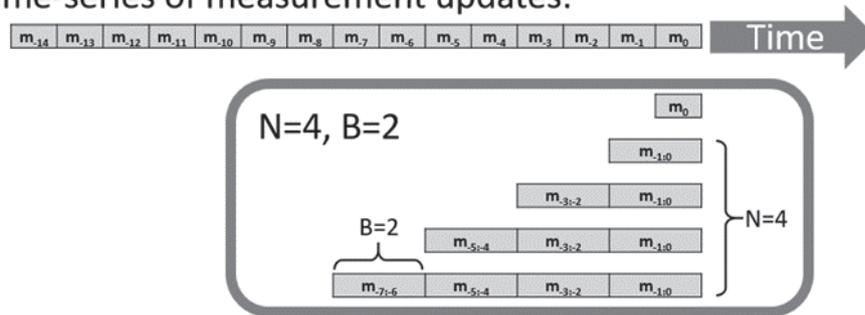


FIGURE 3 One example of a FIND configuration

against a specific set of faults given the choice of design parameters  $N$ ,  $B$ , and  $P_{fa}$ . A *use case* would include any relevant quantities such as the sensor suite, Kalman filter design, fault detection algorithm design parameters, fault models, environment conditions such as the radiofrequency (RF) environment, GNSS signals, trajectory, mission duration, etc.

In addition to reducing sensitivity to error model uncertainty, the FIND algorithm addresses the fact that the onset of faulty measurements is unknown and ensures that a test statistic with minimal fault-free noise accumulation will capture a faulty measurement sequence. Figures 5 and 6 in Section 6 show that both aspects increase the likelihood that the FIND algorithm can detect slowly accumulating persistent faults compared to the IH monitor considered in Tanil et al. (2018).

The detection scheme for the FIND algorithm is simple. If the test statistic for any monitor exceeds its threshold, the FIND algorithm declares that faulty measurements are present. Given a minimum acceptable probability of false alarm  $P_{fa,req}$  that applies to the entire monitor bank, we allocate a portion of  $P_{fa,req}$  to each monitor that is used to set a detection threshold according to Equation (4). It is clear from Figure 3 that the monitors are correlated, and ideally the FIND algorithm would account for this when allocating  $P_{fa,req}$ . Developing a sound approach for exploiting monitor cross-correlation is difficult and is left as a topic for future work.

Here, we take a conservative approach and use the fact that for events  $A_1, \dots, A_n$  with  $P(A_1) + \dots + P(A_n) \leq 1$ , (Hailperin, 1965):

$$P(A_1 \cup \dots \cup A_n) \leq P(A_1) + \dots + P(A_n) \quad (6)$$

Associating  $A_i$  with the event that the  $i$ -th monitor false alarms, the left-hand side of Equation (6) is the probability of false alarm for the FIND algorithm. Thus, Equation (6) shows that if we allocate the false alarm budget equally across all monitors, the actual probability of false alarm for the bank would be less than the required value. In other words, equal allocation is conservative.

Using this result, the  $i$ -th FIND algorithm monitor is given the allocation:

$$P_{fa,i} = P_{fa,req} / (N + 1) \text{ for } i = 1, \dots, N + 1 \quad (7)$$

in which we remind the reader that there are  $N + 1$  monitors in the bank including the snapshot monitor (see Figure 3). Note that the FIND algorithm can be represented by a single test statistic. Denoting the test statistic and threshold of the  $i$ -th monitor as  $s_i$  and  $T_i$ , respectively, the FIND test statistic is given by:

$$s_{\text{FIND}} = \max \left( \frac{s_{\text{snapshot}}}{T_{\text{snapshot}}}, \frac{s_1}{T_1}, \frac{s_2}{T_2}, \dots, \frac{s_N}{T_N} \right) \quad (8)$$

and has a threshold of one. This follows by observing that if  $s_{\text{FIND}}$  exceeds one, then it must be that at least one of the  $s_i$  values is larger than its threshold.

For certain instantiations of the monitor bank, equal allocation of  $P_{fa,req}$  does not appear to be overly conservative (see Appendix A for Monte Carlo simulation results). However, it is important to point out that a conservative false alarm rate comes at the expense of a higher missed detection rate, a trade-off that should be considered in practice. As is usually the case, whether an attempt is made to take advantage of monitor cross-correlation will depend on the application.

## 4 | IMPLEMENTATION DETAILS

### 4.1 | Tightly-Coupled GPS/Inertial EKF

The GPS/inertial *tightly-coupled filter* (TCF) implementation used in this study is very similar to the TCF algorithm described in Groves (2015), which estimates the receiver kinematic state as an Earth-centered, Earth-fixed (ECEF) position ( $\mathbf{r}_{eb}^e$ , in m), ECEF velocity ( $\mathbf{v}_{eb}^e$ , in m/s), and nav-to-body rotation matrix (attitude)  $\mathbf{C}_n^b$ . The *nav* frame is the local North-East-Down (NED) coordinate frame (Groves, 2015). Attitude is implemented as an error-state formulation where the state vector contains a three-element attitude error tilt vector  $\Psi_n^b$  that is periodically used to correct the full attitude estimate, and is maintained separately from the state vector (a standard approach to attitude estimation, due to fundamental challenges with the estimation of 3D attitude [Markley, 2003]). The filter also estimates clock bias and clock frequency error states which are modeled as a two-state clock process (Brown & Hwang, 2012).

Several augmented states are also included: IMU biases are estimated with both a constant (*turn-on*) and first-order Gauss-Markov process (*in-run*) component for each of the three accelerometers and three gyroscopes. The IMU bias estimates are used to compensate the raw IMU measurements during the time update (inertial navigation equations). During the EKF's time update, the position, velocity, and attitude are propagated forward by solving a system of differential equations (given by equations 5.37, 5.35, and 5.24 of Groves [2015]) with a Runge-Kutta 4(5)-based *ordinary differential equation* (ODE) solver. The turn-on and in-run bias states are propagated according to their differential equations (either a constant or first-order Gauss-Markov process).

Pseudorange and pseudorange-rate measurements were incorporated from the GNSS tracking loops and modeled as only having white noise. Typically, satellite clock/ephemeris and pseudorange bias states are estimated as two separate first-order Gauss-Markov processes for each satellite, although those processes were disabled for this testing since those errors were not present in the synthetic measurements for processing.

### 4.2 | Scenario Configuration

All testing used a single simulated airborne trajectory, in which an aircraft moves at a constant speed of 100 m/s about a racetrack shape with a 15-km semi-major axis, resulting in a lap period of about 13 minutes. The aircraft banks turn up to a maximum roll angle of 10°. The altitude is nearly constant at about 10 km although there are some slight deviations on the order of 10 meters as the racetrack shape does not wrap with the WGS-84 Earth ellipsoid. The true trajectory is used to generate ideal inertial measurements before random noise and biases are added to form synthetic measurements for processing. Each fault profile is defined as a position and velocity offset relative to this truth trajectory and is used to derive the synthetic GPS measurements.

Simulated pseudorange and Doppler measurements from GPS coarse acquisition (C/A) signals (transmitted at  $L1 = 1,575.42$  MHz) were used as the GNSS observations. A two-state clock (Brown and Hwang, 2012) tuned as an oven-controlled crystal oscillator (OCXO) was used both by the simulation to drive the generation of synthetic measurements and by the filter for its dynamics model and process noise tuning. From Brown and Hwang (2012), the clock parameters for an OCXO

TABLE 2  
IMU model parameters

	Accelerometers (per-axis)			Gyroscopes (per-axis)		
	Bias (m/s <sup>2</sup> )		Noise Root PSD	Bias (rad/s)		Noise Root PSD
	Turn-On	In-Run	( $\frac{\text{m/s}^2}{\sqrt{\text{Hz}}}$ )	Turn-On	In-Run	( $\frac{\text{rad/s}}{\sqrt{\text{Hz}}}$ )
Tactical	$0.9 \times 1e-2$	$0.1 \times 1e-2$	$\sqrt{1e-6}$	$0.9 \times 5e-5$	$0.1 \times 5e-5$	$\sqrt{1e-9}$
Aviation	$0.9 \times 1e-3$	$0.1 \times 1e-3$	$\sqrt{1e-7}$	$0.9 \times 5e-8$	$0.1 \times 5e-8$	$\sqrt{1e-12}$

were  $h_0 = 2e-25 \text{ sec}^{-1}$  and  $h_{-2} = 5e-25 \text{ sec}$ . This study explored two grades of IMU: an *aviation grade* and a lower-quality *tactical grade*. The various characteristics and tunings for a variety of IMU grades were obtained from Groves (2015) and are summarized in Table 2. The time constants for the IMU in-run biases were set to 3,600 seconds.

This study examined a single class of GNSS fault profiles that implemented an accumulated position drift in the vertical channel. These were not necessarily worst-case fault profiles, and more complete coverage is a topic for future work. We chose a quadratic pulloff of vertical position as it is believed to be a particularly challenging case to detect, specifically when utilizing IMUs. For the first 20 minutes (1,200 seconds), no faults were present. The profiles are referenced/named in later sections by the *fault magnitude* according to the total position pulloff at the 1,800 second mark (10 minutes since fault onset; 250, 1,000, 4,000 meters, etc.). All GPS C/A signals were consistent with the true trajectory up to the 20-minute mark. After the 20-minute mark, signals were nominally consistent with the vertical drift fault. The inertial measurements were always consistent with the true trajectory.

## 5 | ANALYSIS METHODS

This section discusses the analysis methods used to predict and quantify the performance of the FIND algorithm relative to the snapshot and IH innovations monitors. Covariance analysis and Monte Carlo simulations are both used for this purpose. To the best of our knowledge, a clear justification for using covariance analysis to analyze EKF behavior in the presence of faults has not been given. This section fills that gap by deriving a new, generalized approach to EKF covariance analysis capable of accommodating measurement faults.

### 5.1 | Covariance-Based Fault Analysis

*Covariance analysis* is a common technique for assessing the performance of a Kalman filter. The theory is well established for linear KFs and can also be applied relatively easily to EKFs under fault-free conditions. However, it is not obvious how one would (or could) use covariance analysis to predict EKF behavior under faulted measurement conditions. This subsection shows how such an analysis can be conducted, and its role in assessing the performance of the FIND algorithm.

To begin, let's take a slightly different approach to EKF analysis and view the propagation equations in Table 1 as the noise-free propagation of the nonlinear system:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}'_k, \mathbf{u}_k, \mathbf{w}_k) \\ \mathbf{x}'_k &= \mathbf{x}_k + \mathbf{K}_k [\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)] \end{aligned} \quad (9)$$

In other words,  $\hat{\mathbf{x}}_{k+1}^- = \mathbf{x}_{k+1}$  when  $\mathbf{w}_k = \mathbf{0}$  and  $\hat{\mathbf{x}}_k^+ = \mathbf{x}'_k$  when  $\mathbf{v}_k = \mathbf{0}$ . The estimate error vectors  $\mathbf{e}_{k+1}^- = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-$  and  $\mathbf{e}_k^+ = \mathbf{x}'_k - \hat{\mathbf{x}}_k^+$  evolve according to:

$$\begin{aligned}\mathbf{e}_{k+1}^- &= \mathbf{f}(\mathbf{x}'_k, \mathbf{u}_k, \mathbf{w}_k) - \mathbf{f}(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \mathbf{0}) \\ \mathbf{e}_k^+ &= \mathbf{e}_k^- + \mathbf{K}_k [-\mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) + \mathbf{h}(\hat{\mathbf{x}}_k^-, \mathbf{0})]\end{aligned}\quad (10)$$

Assuming that  $\hat{\mathbf{x}}_k^+$  is close to  $\mathbf{x}'_k$  and that  $\hat{\mathbf{x}}_k^-$  is close to  $\mathbf{x}_k$ , the first-order approximations:

$$\begin{aligned}\mathbf{f}(\mathbf{x}'_k, \mathbf{u}_k, \mathbf{w}_k) &\approx \mathbf{f}(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \mathbf{0}) + \mathbf{F}_k \mathbf{e}_k^+ + \mathbf{G}_k \mathbf{w}_k \\ \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) &\approx \mathbf{h}(\hat{\mathbf{x}}_k^-, \mathbf{0}) + \mathbf{H}_k \mathbf{e}_k^- + \mathbf{J}_k \mathbf{v}_k\end{aligned}\quad (11)$$

are used to obtain the linear propagation equations:

$$\begin{aligned}\mathbf{e}_{k+1}^- &\approx \mathbf{F}_k \mathbf{e}_k^+ + \mathbf{G}_k \mathbf{w}_k \\ \mathbf{e}_k^+ &\approx \mathbf{H}_k \mathbf{e}_k^- - \mathbf{J}_k \mathbf{v}_k\end{aligned}\quad (12)$$

Equation (12) is the familiar estimate error propagation for the Kalman filter, with the one caveat that the matrices require the specification of the estimate vector  $\hat{\mathbf{x}}_k$ . For EKF covariance analysis,  $\hat{\mathbf{x}}_k$  is never actually constructed. That would require simulating the random processes  $\mathbf{w}_k$  and  $\mathbf{v}_k$  and running the EKF, which is more in line with Monte Carlo simulation than covariance analysis. Instead, we eliminate the need to simulate random sequences by linearizing about the expected value of the trajectory in Equation (9).

Using the first order approximation that for any nonlinear random function  $f(x)$ ,  $E[f(x)] \approx f(E[x])$ :

$$\begin{aligned}E[\mathbf{x}_{k+1}] &\approx \mathbf{f}(E[\mathbf{x}'_k], \mathbf{u}_k, \mathbf{0}) \\ E[\mathbf{x}'_k] &\approx E[\mathbf{x}_k] + \mathbf{K}_k \{E[\mathbf{z}_k] - \mathbf{h}(E[\mathbf{x}_k], \mathbf{0})\}\end{aligned}\quad (13)$$

Thus, EKF covariance analysis is a combination of propagating the trajectory in Equation (13) and using it to form the matrices in Equation (12) that ultimately enable propagation of the estimate error covariance matrix.

No assumptions were made in the earlier analysis regarding whether  $\mathbf{z}_k$  was fault-free or contained measurement faults. The only assumption was that  $\hat{\mathbf{x}}_k^+$  was close to  $\mathbf{x}'_k$  and that  $\hat{\mathbf{x}}_k^-$  is close to  $\mathbf{x}_k$ , which we assume to be valid under both fault-free and faulted measurement conditions. Therefore, simultaneously propagating the expected state trajectory and the covariance matrix is justified to analyze the performance of EKFs under faulted measurement conditions. We will subsequently refer to this analysis method as Covariance Analysis Including Expected Values, or CAIEV.

The last item to clarify is how to compute the expected value  $E[\mathbf{z}_k]$  in Equation (13). Under the assumed faulty measurement model  $\mathbf{z}_k = \mathbf{h}(\bar{\mathbf{x}}_k, \mathbf{v}_k) + \mathbf{d}_k$ , where we now use  $\bar{\mathbf{x}}_k$  to denote the true state trajectory,  $E[\mathbf{z}_k]$  can be approximated to first order as:

$$E[\mathbf{z}_k] = E[\mathbf{h}(\bar{\mathbf{x}}_k, \mathbf{v}_k) + \mathbf{d}_k] \approx \mathbf{h}(E[\bar{\mathbf{x}}_k], \mathbf{0}) + \mathbf{d}_k \quad (14)$$

The true trajectory evolves according to  $\bar{\mathbf{x}}_{k+1} = \mathbf{f}(\bar{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{w}_k)$ . Thus,  $E[\bar{\mathbf{x}}_k]$  propagates as  $E[\bar{\mathbf{x}}_{k+1}] \approx \mathbf{f}(E[\bar{\mathbf{x}}_k], \mathbf{u}_k, \mathbf{0})$ , to the first order.

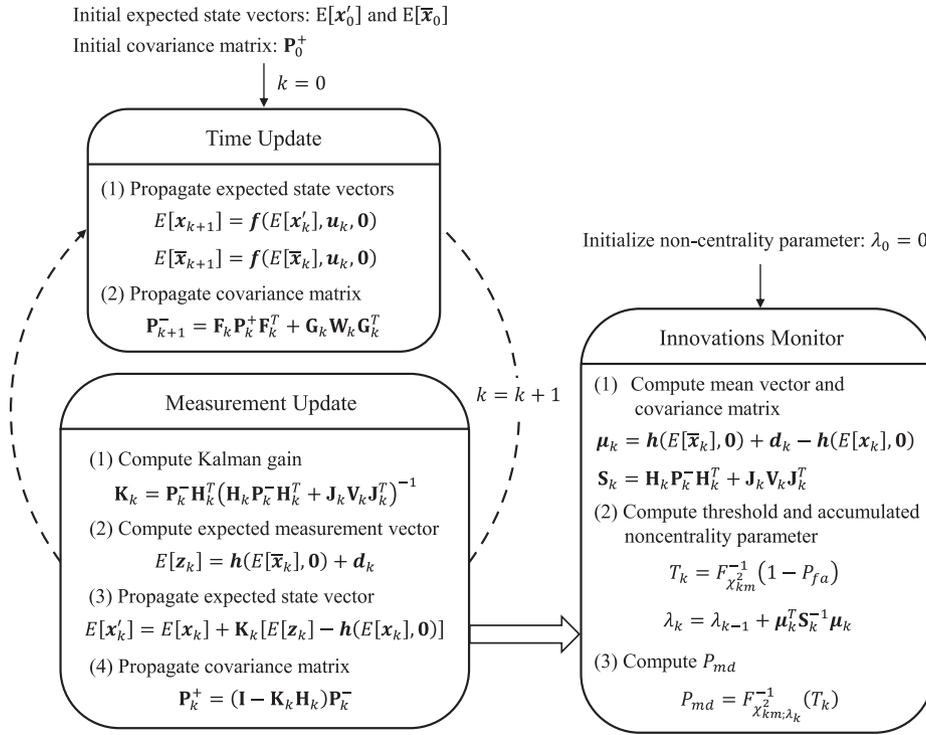


FIGURE 4 Performance analysis flowchart for cumulative innovations monitor

From the standpoint of fault detection, the expected values propagated using CAIEV enable prediction of the mean of the EKF innovations vector, denoted as  $\boldsymbol{\mu}_k$ . Specifically:

$$\boldsymbol{\mu}_k = \mathbf{h}(E[\bar{\mathbf{x}}_k], \mathbf{0}) + \mathbf{d}_k - \mathbf{h}(E[\mathbf{x}_k], \mathbf{0}) \quad (15)$$

Figure 4 summarizes the CAIEV approach and its use in predicting the performance of a cumulative innovations monitor. For the FIND algorithm, the steps in the box labeled *Innovations Monitor* are conducted for each monitor in the bank.

## 5.2 | Monte Carlo Simulation

We used a high-fidelity simulation testbed for the Monte Carlo analysis, the Global Navigation Satellite System Test Architecture (GNSSTA; Quartararo et al., 2016). High-fidelity semi-analytic models for tracking channels generate synthetic complex correlations which include the effects of code phase mismatch between the replica being driven by the tracking loops and true signal(s) in the simulated environment. They account for effects of the autocorrelation function specific to that signal type (Doppler, power, noise, carrier phase, data symbols, etc.). These are described as *semi-analytic* because the tracking model does not actually process baseband data as in a software receiver mode, but instead uses descriptions of the receiver clock and the Doppler, data symbols, power, and noise profiles of the signals in the environment as a function of time to create a synthetic correlation result to be passed downstream to the rest of the receiver processing chain (Borio et al., 2010; Spilker et al., 1996). From there, pseudoranges, Doppler estimates, decoded navigation messages, position solutions, etc., are derived and fed into the EKF.

As with the synthetic GNSS measurements, synthetic IMU measurements are generated to be consistent with the true receiver trajectories. Noise is added to the nominal measurements according to the emulator configuration (i.e., simulated IMU quality/grade); this includes turn-on biases, in-run biases, white noise, etc. (Quartararo et al., 2016).

## 6 | RESULTS

Performance of the FIND algorithm is first compared to the snapshot and IH algorithms using CAIEV. For FIND,  $N = 60$  monitors are used with  $B = 10$  measurements per block. We chose these values in conjunction with the 1-Hz GPS measurement update to cover a time span of 10 minutes with monitors increasing in duration by steps of 10 seconds. A topic for future work is to study how to more rigorously select the FIND design parameters for a specific application and set of faults and fault profiles to consider. The  $P_{fa}$  for the snapshot, IH, and FIND bank was set to  $10^{-5}$  and, given the 1-Hz measurement rate, about 0.86 false alarms would therefore be expected per day of continuous operation.

Figure 5 shows the missed detection performance over time for the tightly coupled GPS/inertial EKF with a tactical-grade IMU. It is worth discussing how the probability of missed detection ( $P_{md}$ ) was determined for FIND given that multiple, correlated monitors were involved. A missed detection occurs when the test statistic  $s$  from every monitor fails to exceed its threshold  $T$ . Therefore,  $P_{md}$  can be written as:

$$P_{md} = P(s_1 < T_1 \cap s_2 < T_2 \cap \dots) \quad (16)$$

It is difficult to compute  $P_{md}$  in Equation (16) because the test statistics are mutually correlated. In response, we use the upper bound (Hailperin, 1965):

$$P(s_1 < T_1 \cap s_2 < T_2 \cap \dots) \leq \min(P(s_1 < T_1), P(s_2 < T_2), \dots) \quad (17)$$

to predict  $P_{md}$  for FIND using CAIEV. That is, at each time instance, we extract the minimum  $P_{md}$  across all monitors in the bank. It is this minimum value that is plotted over time in the right-most panel of Figure 5.

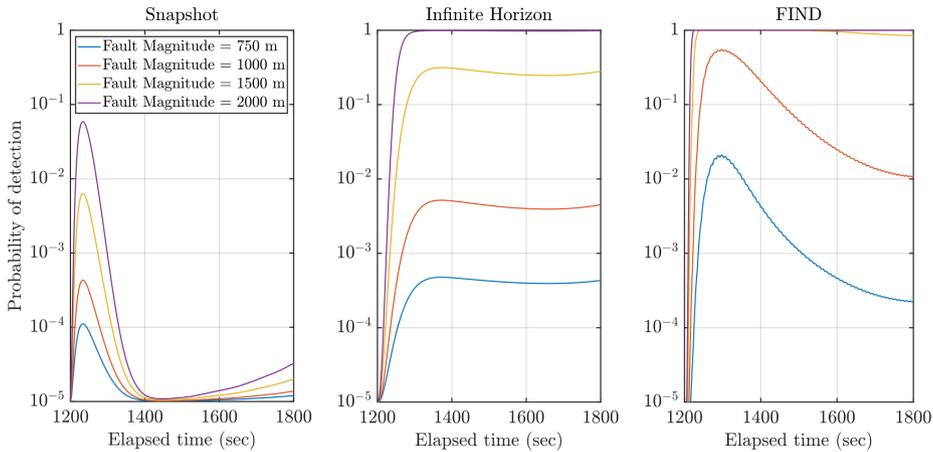


FIGURE 5 Probability of detection as a function of time for a tightly coupled GPS/inertial EKF equipped with a tactical grade IMU (CAIEV)

For the quadratic fault profile, the snapshot monitor is unable to reliably detect the presence of faulty measurements for fault magnitudes up to 2,000 m. The IH monitor clearly has better detection performance compared to the snapshot monitor, but also struggles with smaller fault magnitudes. Only for the largest fault magnitude of 2,000 m does the IH method provide a high probability of detection ( $P_d$ ). An observation in Figure 5 is that for all fault magnitudes, FIND's highest  $P_d$  over the entire fault duration always exceeds that of the IH monitor. For example, with a fault magnitude of 750 m, the highest  $P_d$  for FIND is approximately 0.025 whereas it is only  $4.75 \times 10^{-4}$  for IH. In other words, covariance analysis predicts that the  $N = 60$ ,  $B = 10$  instantiation of the FIND algorithm outperforms the IH monitor. However, it is worth pointing out that other instantiations may result in different relative assessments between FIND, IH, and the snapshot monitor.

One way to understand the theoretical basis for why the FIND algorithm may perform better than IH is by considering the *degrees of freedom* (DOF) in the test statistic. Qualitatively, DOFs capture the allowable variation in the test statistic, analogous to how DOFs capture the allowable motion of a dynamic body. For example, a body with a single DOF can only translate along a line, whereas a body with six DOFs can translate and rotate in three dimensions. The IH test statistic can have many DOFs. Thus, if the IH monitor encounters a faulty measurement sequence after it has been accumulating for a while, it could be that the faults are "allowable" within the DOF, resulting in poor detection performance. With the FIND algorithm's finite-length accumulations, the DOFs of each monitor could be substantially less compared to IH, making it harder to justify that the faulty behavior is due to natural causes. As a result, the detection probability for the FIND algorithm would be higher than IH.

It is also worth discussing the behavior of the FIND  $P_d$  curves for fault magnitudes of 1,000 m and 1,500 m. The curves initially rise when faults begin to appear at 1,200 seconds, indicating that the likelihood of fault detection increases with exposure time. However, this trend does not continue for the entire fault duration. After approximately 1,300 seconds, the  $P_d$  curves begin to fall, and it becomes less likely that the measurement faults will be detected. This is explained as follows.

The analysis conducted here does not include exclusion (i.e., faulty measurements are always incorporated by the EKF) regardless of alarm status. Therefore, the state estimate, which serves as the EKF linearization point, begins to follow the faulty state trajectory with prolonged exposure to the fault, resulting in less sensitivity to the presence of faulty measurements and lower probabilities of detection. It is important to note that an actual implementation of the FIND algorithm would execute an exclusion algorithm upon fault detection.

A concise performance summary is obtained by extracting the best-case (highest)  $P_d$  over time for each fault magnitude. This metric is shown in Figure 6 for many IMU grades, fault profiles, and fault detection algorithm configurations, and the results reiterate the fact that the best detection performance is achieved using the FIND algorithm given a particular IMU grade.

The CAIEV analysis performed in this section has two important limitations. First, it is based on first-order approximations and thus does not fully capture second-order or nonlinear effects. Second, it assumes that the EKF has perfect knowledge of measurement and process noise statistics and therefore does not capture the effects of real-world model uncertainty. Nevertheless, the CAIEV analysis is useful for performing initial studies of fault detection performance. In parallel, we also perform high fidelity Monte Carlo simulations to better understand the effect of processes that cannot easily be replicated in covariance analysis.

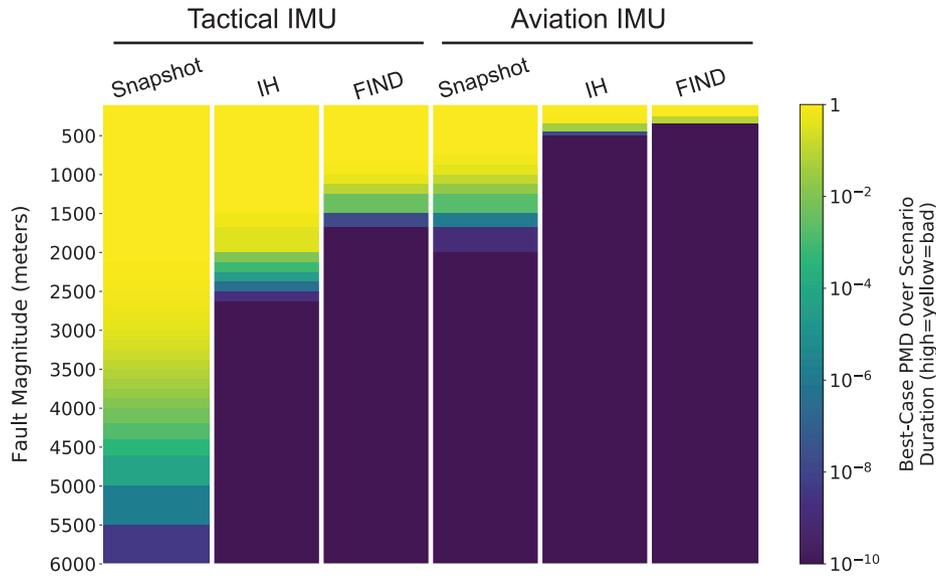


FIGURE 6 Best case  $P_{md}$  for snapshot, IH, and FIND monitors (CAIEV analysis)

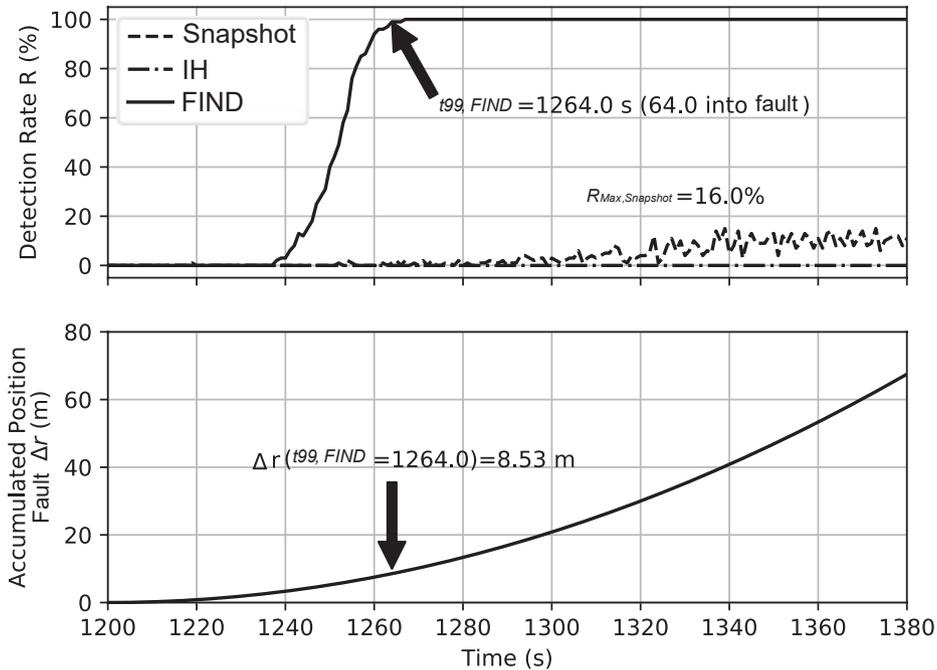


FIGURE 7 Monte Carlo detection rate example for three fault detection algorithms with an aviation-grade IMU, 750-meter pulloff fault, 100 runs

For several combinations of IMU grade (aviation or tactical) and measurement fault profile, we ran 100 Monte Carlo simulations, each simulating 30 minutes of navigation with all fault detection algorithms active. Just as with the covariance analysis, fault exclusion was not conducted in the Monte Carlo analysis (i.e., measurements were always ingested by the filters regardless of the fault detection alarms). An example plot is shown in Figure 7 to illustrate the metrics that are presented later. The top plot shows the percentage of runs,  $R$ , in which each detector triggered, and the bottom plot shows the corresponding position pulloff,  $\Delta r$ , of the fault. By  $t = 1,264$  seconds (64 seconds after the onset of the faults), 99% of FIND

**TABLE 3**

Maximum (best) detection rate over time ( $R_{\max}$ , as percentage of 100 runs) for various configurations and algorithms (across 0–30 minute span where faults start at 20-minute mark)

	Tactical			Aviation			
	Snapshot	IH	FIND	Snapshot	IH	FIND	
250				2	8	8	
500		No Data			2	8	100
750				19	48	100	
1,000	0	0	1	84	100	100	
1,250	1	1	50	99	100	100	
1,500	2	7	100				
2,000	8	6	100				
2,500	40	0	100				
3,000	90	3	100		No Data		
3,500	100	2	100				
4,000	100	100	100				

runs had detected faults. This may initially seem slow, but it is important to note that the quadratic fault profile had only reached approximately nine meters of total position pulloff compared to the 6.4 km traveled by the aircraft in that same period.  $R_{\max}$  indicates the maximum (best) detection rate achieved across all time (out of 30 minutes) for a particular configuration and algorithm.

Table 3 summarizes the values of  $R_{\max}$  (see Figure 7) for different scenarios and sensor configurations. Performance was strongly tied to the grade of IMU; coupled with the limitations of processing resources and time, data was not collected for every permutation of fault and IMU grade. Data was collected for the aviation-grade IMU up to where 99% detection rates were observed, however in the future it would be useful to fill out those cases, particularly for the second two metrics. Utilizing an aviation-grade IMU, the FIND algorithm achieved excellent detection for the 500- and 750-meter fault profiles, and the IH did well for the 750-meter fault.

The Monte Carlo results show reasonable agreement with the CAIEV predictions but there were some differences. Specifically, within the tactical-grade IMU results, the snapshot monitor generally outperformed IH whereas the CAIEV results predicted that IH would provide better fault detection than the snapshot monitor. This difference may be explained by the fact that CAIEV does not account for the model mismatch effect which more severely impacts long accumulating monitors such as IH, leading to a potential loss of detection sensitivity as evident in these Monte Carlo simulations. This discrepancy was not seen, however, in the aviation-grade results, where the Monte Carlo results agreed with the CAIEV predictions that the FIND algorithm would outperform IH, and IH would outperform snapshot. Another complicating factor here may be that the length of accumulation during the fault activity varied between the two cases, as evidenced by the time-to-detect metrics that will be described next; shorter detection indicates less time for the IH monitor to recover from the long fault-free accumulation before the faults began, which was driven by model mismatch.

Tables 4 and 5 summarize the time-to-detect results for the Monte Carlo simulations (the time to achieve 99% detection rate). Since the fault scenarios were

**TABLE 4**

Time to 99% detection rate (since fault onset in sec) from Monte Carlo runs for various configurations and algorithms; dashes indicate where 99% detection rates were not achieved.

	Tactical			Aviation		
	Snapshot	IH	FIND	Snapshot	IH	FIND
250				-	-	-
500		No Data		-	-	130
750				-	-	62
1,000	-	-	-	-	197	41
1,250	-	-	-	123	144	27
1,500	-	-	44			
2,000	-	-	21			
2,500	-	-	16			
3,000	-	-	13		No Data	
3,500	21	-	11			
4,000	14	111	10			

**TABLE 5**

Accumulated position fault (in m) at the 99% detection time from Monte Carlo runs for various configurations and algorithms; dashes indicate where 99% detection rates were not achieved.

	Tactical			Aviation		
	Snapshot	IH	FIND	Snapshot	IH	FIND
250				-	-	-
500		No Data		-	-	23
750				-	-	8
1,000	-	-	-	-	108	5
1,250	-	-	-	53	72	3
1,500	-	-	8			
2,000	-	-	2			
2,500	-	-	2			
3,000	-	-	1		No Data	
3,500	4	-	1			
4,000	2	137	1			

selected to be subtle and to explore the boundaries of performance of the FIND algorithm, not all of the algorithm/scenario/IMU configurations ever achieved a 99% detection rate (note that only some of the  $R_{max}$  entries in Table 3 meet or exceed 99%).

As a result, the tables are somewhat sparse, but we can see that the algorithms did eventually achieve strong detection performance and typically did so within a few seconds to minutes of the fault onset, as indicated by the  $t_{99}$  values in Table 4. While detection rates on the order of tens of seconds to minutes may seem slow, it

is important to remember that the fault profiles are very slow accumulations and as a result the total position error achieved after a minute may only have amounted to a few meters. The total position error (in meters) at the corresponding  $t_{99}$  times are shown Table 5. For the cumulative monitors that had valid data, they typically triggered only when tens of meters of pulloff had occurred, which is very small compared to the kilometers the aircraft may have truly traveled over that time (recall that the aircraft speed is 100 m/s). Stronger pulloff faults tend to be detected more quickly and the FIND algorithm typically triggers when only a few meters of actual position pulloff have been achieved at that point.

## 7 | SUMMARY

This paper derived and analyzed a novel approach to utilize additional sensors for GPS fault detection. The proposed algorithm, FIND, uses a bank of cumulative innovations monitors in an EKF for sensitivity to slow and fast accumulating faults and to minimize the effect of real-world model uncertainty on detection performance. Analysis of the new method was conducted using an adaptation of first-order covariance analysis (the CAIEV approach introduced in this paper) and high-fidelity Monte Carlo simulations. Performance was demonstrated for a tightly coupled GPS/inertial integration architecture, and results showed that the instantiation of the FIND algorithm studied in this paper provides superior detection performance for the cases considered over existing approaches to innovations monitoring.

## 8 | FUTURE WORK

While improved detection performance for the FIND algorithm versus other algorithms has been demonstrated for specific cases in this paper through covariance analysis and high-fidelity Monte Carlo simulations, significant future work could be performed. For a more realistic assessment, hardware-in-the-loop testing would be the next step. This would include error sources that these simulations did not study; for example, the impacts of multipath, atmospheric (ionosphere/troposphere) delays, IMU misalignments and scale factor errors, second-order effects on sensors such as vibratory conditions (which can affect clocks and IMUs), gravity, and gravity-squared error effects, etc.

The dimensionality of the test space is large, and the number of test cases/vectors is intractable, but at least some additional permutations of trajectory, scenario duration, IMU grades, fault profiles, GNSS constellations (e.g., multi-GNSS such as Galileo), and RF environments (e.g., jamming) would provide additional rigor to support the performance conclusions. Additional study on optimal selection of FIND configuration parameters (number of monitors  $N$  and block size  $B$ ) may yield additional performance improvement.

This study focused exclusively on fault detection but of course mitigation is a critical component to the hardening of PNT equipment. Mitigation is typically straightforward for a snapshot monitor because it can simply exclude the single offending measurement set under evaluation without tainting the recursive filter state. For cumulative monitors, detection indicates a problem with the accumulated history of measurements in the recursive filter. A straightforward approach would be to save the filter state history and non-satnav sensor measurements, rewind in time equal to the size of the window of the alarming monitor and re-play

without the satnav measurements. Depending on the length of the accumulation, this may be infeasible, but there may be clever solutions that allow an efficient hybrid between a purely recursive vs. batch filter approach (i.e., it may be unnecessary to save every Kalman filter state/covariance, save every IMU measurement, etc.). Another approach would be to have a bank of parallel filters running (which process different measurements) that the user equipment could swap over to or copy from. A recent approach using fading filters is described in Tanil et al. (2019).

As described in Section 2, any mismatch between the real errors and the models of those errors can lead to discrepancies between a predicted probability of false alarm and the actual false alarm rate (and the same concern applies to detection). Beyond simply getting a parameter value wrong (e.g., the variance for a Gaussian noise process), real system errors may not even be Gaussian-distributed (e.g., multipath errors) although engineers typically use this model. It is critical to study and verify actual false alarm rates, and future work should consider efforts in that area. An in-depth sensitivity analysis of the impacts on performance due to model mismatch would help improve understanding in this area.

The approaches outlined in this paper have only examined validating measurement sets as a whole; they do not determine which of the measurements are faulty and, as a result, discard all of them if the test statistic exceeds the detection threshold. It would be advantageous to improve upon this so that subsets of good measurements can still be incorporated. For example, ARAIM (Blanch et al., 2007; WG-C ARAIM Subgroup, 2016) can sift out subsets of self-consistent measurements (noting *self-consistent* does not necessarily mean *correct* without independent information/sensors such as clocks and inertial sensors).

The probability of detection analysis did not address the correlated nature of the accumulated test statistic. The IH source material (Tanil et al., 2018) contains a comment on this as well, and the authors posit that the analysis is probably still generally valid, although it would be valuable to address this flaw. Since we saw detection rates and performance consistent with predictions, the effect is probably not massive, but it would be useful analytical work to mathematically characterize this effect and its impact on performance predictions.

We were able to show that the sensitivity loss is strongly affected by model mismatch which accumulates over time. However, we have noticed that even in the covariance and a lower-fidelity Monte Carlo analysis where the models perfectly match the errors being added to the synthetic measurements, very long accumulations (IH algorithm) still suffer from some loss of sensitivity over time. This could be due to the correlation in the accumulated test statistics that the predictive analysis does not account for, but further work should seek to understand the source of this residual effect.

Lastly, while the monitor bank concept is general to any Kalman filter, we have only explored it in the specific cases of one filter that processes inertial and GNSS measurements, relying on the inertial and clocks to help detect faulty GNSS measurements. Since the core algorithm translates to any filter, other sensor configurations could naturally and automatically benefit. For example, a filter that also incorporates altitude information from a barometric altimeter may be able to significantly increase detection of faults (particularly in the vertical direction as we examined in this paper).

## REFERENCES

- Abramowitz, M., Stegun, I. A., & Romer, R. H. (1988). Handbook of mathematical functions with formulas, graphs, and mathematical tables. *American Journal of Physics*, 56. <https://doi.org/10.1119/1.15378>
- Bhatti, U., & Ochieng, W. (2007). Failure modes and models for integrated GPS/INS systems. *The Journal of Navigation*, 60(2), 327–348. <https://doi.org/10.1017/S0373463307004237>
- Blanch, J., Ene, A., Walter, T., & Enge, P. (2007). An optimized multiple hypothesis RAIM algorithm for vertical guidance. *Proc. of the 20th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2007)*, Fort Worth, TX, 2924–2933. <https://www.ion.org/publications/abstract.cfm?articleID=7644>
- Borio, D., Anantharamu, P. B., & Lachapelle, G. (2010). Semi-analytic simulations: An extension to unambiguous BOC tracking. *Proc. of the 2010 International Technical Meeting of the Institute of Navigation*, San Diego, CA, 1023–1036. <https://www.ion.org/publications/abstract.cfm?articleID=8882>
- Brown, R. G., & Hwang, P. Y. C. (2012). *Introduction to random signals and applied Kalman filtering with MATLAB exercises* (4th ed.). John Wiley & Sons, Inc.
- Coelho, C. (2020). On the distribution of linear combinations of chi-square random variables. In A. Bekker, D.-G. Chen, & J. T. Ferreira (Eds.), *Computational and methodological statistics and biostatistics* (pp. 211–252). [https://doi.org/10.1007/978-3-030-42196-0\\_9](https://doi.org/10.1007/978-3-030-42196-0_9)
- Crespillo, O. G., Grosch, A., Skaloud, J., & Meurer, M. (2017). Innovation vs residual KF based GNSS/INS autonomous integrity monitoring in single fault scenario. *Proc. of the 30th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2017)*, Portland, OR, 2126–2136. <https://doi.org/10.33012/2017.15136>
- Diesel, J., & Luu, S. (1995). GPS/IRS AIME: Calculation of thresholds and protection radius using chi-square methods. *Proc. of the 8th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 1995)*, Palm Springs, CA, 1959–1964. <https://www.ion.org/publications/abstract.cfm?articleID=2520>
- Grewal, M. S., & Andrews, A. P. (2014). *Kalman filtering: Theory and practice with MATLAB* (4th ed.). Wiley-IEEE Press.
- Groves, P. D. (2015). Principles of GNSS, inertial, and multisensor integrated navigation systems 2nd ed. [book review]. *IEEE Aerospace and Electronic Systems Magazine*, 30(2), 26–27. <https://doi.org/10.1109/MAES.2014.14110>
- Hailperin, T. (1965). Best possible inequalities for the probability of a logical function of events. *The American Mathematical Monthly*, 72(4), 343–359. <https://doi.org/10.2307/2313491>
- Joerger, M., & Pervan, B. (2013). Kalman filter-based integrity monitoring against sensor faults. *Journal of Guidance, Control, and Dynamics*, 36(2), 349–361. <https://doi.org/10.2514/1.59480>
- Kujur, B., Khanafseh, S., & Pervan, B. (2020). A solution separation monitor using INS for detecting GNSS spoofing. *Proc. of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020)*, 3210–3226. <https://www.ion.org/publications/abstract.cfm?articleID=17535>
- Kujur, B., Tanil, C., Khanafseh, S., & Pervan, B. (2019). Sensitivity of innovation monitors to uncertainty in error modeling. *Proc. of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019)*, Miami, FL, 3266–3274. <https://www.ion.org/publications/abstract.cfm?articleID=17066>
- Lee, Y. C. (1988). New concept for independent GPS integrity monitoring. *NAVIGATION*, 35(2), 239–254. <https://doi.org/10.1002/j.2161-4296.1988.tb00954.x>
- Markley, F. L. (2003). Attitude error representations for Kalman filtering. *Journal of Guidance, Control, and Dynamics*, 26(2). <https://doi.org/10.2514/2.5048>
- Mehra, R. K., & Peschon, J. (1971). An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7(5), 637–640. [https://doi.org/10.1016/0005-1098\(71\)90028-8](https://doi.org/10.1016/0005-1098(71)90028-8)
- Quartararo, J. D., Cerruti, A., Miller, S., Shapiro, A., & Singh, R. (2016). Providing flexible integrated navigation tools and implementations to the PNT community. In *Proceedings of the 2016 ION Joint Navigation Conference*, Dayton, OH, 284–301.
- Quartararo, J. D., & Langel, S. E. (2020). Detecting slowly accumulating faults using a bank of cumulative innovations monitors in Kalman filters. *Proc. of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020)*, 2367–2381. <https://www.ion.org/publications/abstract.cfm?articleID=17655>
- Rife, J. (2013). The effect of uncertain covariance on a chi-square integrity monitor. *NAVIGATION*, 60(4), 291–303. <https://doi.org/10.1002/navi.45>
- Spilker Jr., J. J., Axelrad, P., Parkinson, B. W., & Enge, P. (1996). *Global positioning system: Theory and applications* (vol. 1). American Institute of Aeronautics and Astronautics. <https://doi.org/10.2514/4.866388>
- Stengel, R. F. (2012). *Optimal control and estimation*. Dover Publications.
- Sturza, M. A. (1988). Navigation system integrity monitoring using redundant measurements. *NAVIGATION*, 35(4), 483–501. <https://doi.org/10.1002/j.2161-4296.1988.tb00975.x>

- Tanil, C., Khanafseh, S., Joerger, M., Kujur, B., Kruger, B., de Groot, L., & Pervan, B. (2019). Optimal INS/GNSS coupling for autonomous car positioning integrity. *Proc. of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019)*, Miami, FL, 3123–3140. <https://doi.org/10.33012/2019.17013>
- Tanil, C., Khanafseh, S., Joerger, M., & Pervan, B. (2018). An INS monitor to detect GNSS spoofers capable of tracking vehicle position. *IEEE Transactions on Aerospace and Electronic Systems*, 54(1), 131–143. <https://doi.org/10.1109/TAES.2017.2739924>
- Todling, R. (2000). Estimation theory and atmospheric data assimilation. In P. Kasibhatla, M. Heimann, P. Rayner, N. Mahowald, R. G. Prinn, & D. E. Hartley (Eds.). *Inverse methods in global biogeochemical cycles* (vol. 114, pp. 49–65). <https://doi.org/10.1029/GM114p0049>
- WG-C ARAIM Subgroup. (2016). *WG-C ARAIM subgroup, milestone 3 report* (Tech. Rep.). EU-US Cooperation on Satellite Navigation, Working Group C - ARAIM Technical Subgroup.

**How to cite this article:** Quartararo, J. D., & Langel, S. (2022) Detecting slowly accumulating faults using a bank of cumulative innovations monitors in Kalman filters. *NAVIGATION*, 69(1). <https://doi.org/10.33012/navi.507>

## APPENDIX A: EFFECT OF MONITOR CROSS-CORRELATION ON FALSE ALARM RATE

This appendix investigates the effect that monitor cross-correlation has on the actual false alarm rate of the FIND algorithm. In addition to showing that equally allocating  $P_{fa}$  to each monitor is conservative, we will also provide insight into how conservative this allocation is. The analysis is carried out through Monte Carlo simulations with a total number of monitors in the FIND bank varying from five to 60. Using Figure 3 as a guide, we use  $B = 1$  and assume for simplicity that each of the  $m_i$  is a chi-square random variable with 10 degrees of freedom.

Suppose that the overall false alarm requirement is  $10^{-4}$  and that we allocate this budget equally to each monitor. Recall from Section 3.4 that the FIND test statistic can be written as:

$$s_{\text{FIND}} = \max \left( \frac{s_{\text{snapshot}}}{T_{\text{snapshot}}}, \frac{s_1}{T_1}, \frac{s_2}{T_2}, \dots, \frac{s_N}{T_N} \right) \quad (\text{A1})$$

where  $N$  is the number of cumulative monitors in the bank (varying between four and 59 for our case). The threshold for  $s_{\text{FIND}}$  is  $T_{\text{FIND}} = 1.0$ .

To assess the behavior of  $s_{\text{FIND}}$  under fault-free conditions, we randomly simulate one billion samples. A single sample is obtained by first generating independent  $m_i$ , each of which is a chi-square random variable with 10 degrees of freedom. Then the normalized test statistics  $s_i/T_i$  are formed by summing the appropriate subsets of the  $m_i$  and normalizing by the threshold.

Lastly, the  $s_i/T_i$  are substituted into Equation (A1) to obtain one sample of  $s_{\text{FIND}}$ . The true false alarm rate for a given threshold is determined empirically by dividing the number of samples exceeding the threshold by the total number of samples. The results of this analysis are shown in Figure A1. Notice that for the FIND threshold of one, the true false alarm rate is less than the desired value of  $10^{-4}$ , with the discrepancy increasing as the number of monitors increases. Hence, the Monte Carlo analysis supports the claim that equal allocation of the false alarm requirement to each monitor is a conservative approach.

Figure A1 also provides an indication of how conservative an equal false alarm allocation is as a function of the number of monitors. With five monitors, the true false alarm rate is  $7.58 \times 10^{-5}$ , 24% below the required level. If 60 monitors are used, the true false alarm rate is  $2.78 \times 10^{-4}$ , or 72% below the desired value. It is difficult to make general statements about whether it is worth trying to exploit monitor cross-correlation when setting thresholds for the FIND monitor bank.

Not only will the answer depend on the application, but it will also depend on the form of the bank. This appendix only considers instantiations where the number of degrees of freedom was equal for each of the  $m_i$ , and the results could vary significantly with degrees of freedom. Nevertheless, if the bank structure is known a priori, the Monte Carlo approach outlined in this section could serve as a data-driven method for setting monitor thresholds that take advantage of cross-correlation.

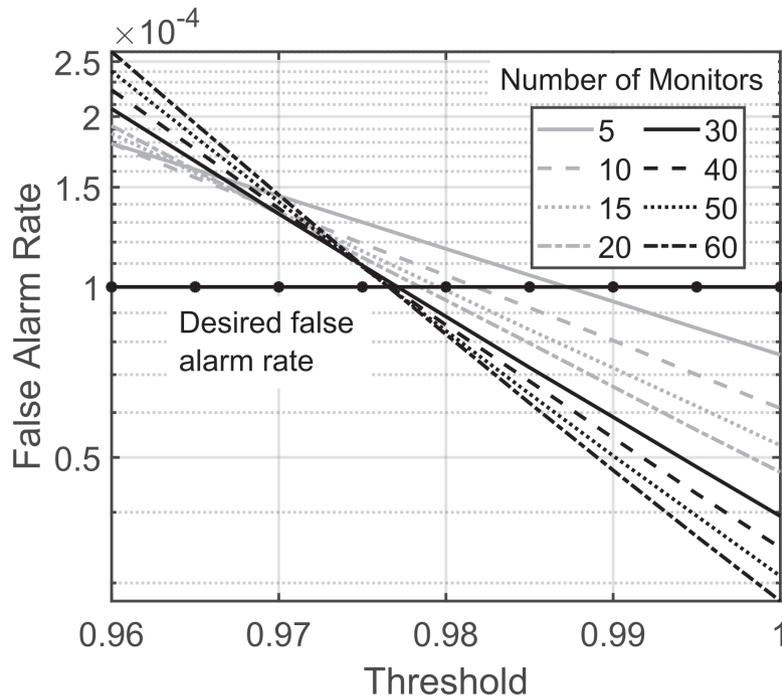


FIGURE A1 FIND false alarm rate as a function of the detection threshold