Influence of GNSS Integrity Monitoring on Single and Multiple Fault Event Probabilities

Jason Rife, Tufts University

ABSTRACT

Cases of multiple simultaneous faults are generally considered low probability in integrity analysis of GNSS augmentation systems. This assumption is not necessarily valid given the sensitivity limitations of practical integrity monitors. This paper introduces a method to account for monitor sensitivity in quantifying the probability that undetected faults occur. Analysis of the two-fault case suggests that the conventional method of designating a small integrity risk allocation to cover all multiple fault events does not necessarily ensure integrity for difficult-to-detect, small and moderate-sized faults.

INTRODUCTION

GNSS augmentation systems support navigation for safety-critical applications such as aviation. These systems monitor for anomalous measurement errors and enable users to compute rigorous error bounds. Precision augmentation system architectures exploit stationary reference stations, which communicate differential corrections and safety data to users. These architectures include Ground-Based Augmentation Systems (GBAS), Space-Based Augmentation Systems (SBAS), and Ground-based Regional Augmentation Systems (GRAS) [1]. For certain applications which do not require differential corrections, augmentation functions can be accomplished without a stationary reference-station infrastructure through Aircraft-Based Augmentation Systems (ABAS). An example of this last is Receiver Autonomous Integrity Monitoring (RAIM) [2].

The primary role of augmentation systems is to validate navigation sensor integrity risk (the probability that an undetected error exceeds a specified error bound). Integrity is quantified in terms of a probability allocation per unit time. In aviation, maximum integrity risk is specified by flight regime; examples include en route flight ($10^{-7}$ per hour), precision approach ($2 \cdot 10^{-7}$ per 150 s), and automated landing ($10^{-9}$ per 30 s) [1].

Error bounds, also called Protection Levels (PLs), are constructed to protect integrity under nominal and off-nominal conditions. Typically, a portion of the total integrity budget is allocated for the purpose of bounding nominal measurement errors with a PL. This allocation is illustrated for a hypothetical augmentation system in Figure 1. The magnitude of the PL is computed to ensure that only a very small fraction of errors (less than or equal to the integrity allocation) lies outside the bound. If the PL is large, the augmentation system may not be available for certain demanding navigation applications. Precision aircraft landing, for instance, requires the PL not exceed an alert limit of 10 m in the vertical direction. To support high availability, it is desirable that the PL be “tight” (as small in magnitude as possible).

Achieving tight error bounds requires integrity monitoring algorithms, which detect and exclude anomalous GNSS measurements. Most often, integrity monitors rapidly alert users to exclude any measurement corrupted by a large system fault. Because monitors exclude large faults, the magnitude of fault-condition PLs can be reduced, as these PLs need only bound errors caused by small and moderate faults which remain undetected.

Typically, fault-condition PLs are only evaluated for events involving a single system fault. By contrast, PLs are not typically constructed for events involving multiple simultaneous faults. Rather, multiple fault events of all sizes are considered low probability, and this small...
probability is typically covered by an allocation from the overall integrity budget (as shown in Figure 1). If the probability of multiple-fault events does not exceed the integrity allocation, no additional error bounds or monitors are needed to cover the multiple-fault case. This simplification is an enormous benefit for certifying augmentation systems and also justifies certain monitoring schemes which assume no more than one measurement fault at a time (such as conventional RAIM strategies).

Assumptions about the low probability of multiple fault events are only valid, however, if faults can be quickly detected and resolved. By convention, it is assumed that fault events are independent and last only for a brief period before being detected. Hence, the probability of a two-event fault over a given window of time is commonly estimated by squaring the probability of a single fault occurring over that window. Though this approach makes sense for large faults, which can be quickly detected, it makes less sense for small and moderate faults, which may persist for extended periods of time. The longer a small or moderate fault persists, the more likely a second fault will occur, resulting in a multiple-fault event.

Multiple-fault events are an important issue in the modernization of the GNSS and its augmentations. The risk of multiple satellite fault events will increase as upgraded constellations (GPS and GLONASS) and new constellations (Galileo and Compass) place a larger number of GNSS satellites into orbit [1]. Multiple-fault events associated with persistent faults are also an issue for blended SBAS/ABAS architectures proposed by the GNSS Evolutionary Architecture Study (GEAS) [3]. This effort, sponsored by the FAA, is directed at enabling a worldwide LPV-200 precision-approach capability, similar to what is already possible through the Wide Area Augmentation System (WAAS) in North America. The GEAS architectures exploit aircraft-based monitoring to ensure rapid detection of large faults and ground-based monitoring to ensure detection of moderate and small faults over extended periods of time.

The purpose of this paper is to define a method for quantifying the risks associated with persistent faults. The new approach incorporates monitor missed-detection probabilities, which depend both on the duration faults persist and on their observability over that period. As an analog to the basic fault probability used in conventional integrity analysis, the new approach computes an undetected fault probability.

The paper is organized as follows. The paper’s next section develops a method for relating fault probabilities to monitor performance and quantifies missed-detection probabilities in terms of the time correlation of monitor input signals. A final section discusses the implications of the new model for aviation applications.

PROBABILITY OF UNDETECTED FAULTS

This section models the probability of fault events which persist for extended times. Typically, satellite faults last for hours or days, until the Operational Control Segment (OCS) for the faulted satellite deactivates it or flags it unhealthy in the broadcast navigation message. Integrity monitors are needed for safety-critical applications to enable more rapid fault detection.

Previous integrity analyses have accounted for the probability of fault events without explicitly considering the role of integrity monitoring. For the most part, fault analyses have focused on the probability that a new fault will occur over a given time period. In this context, integrity budgets are allocated using a fault rate \( R_f \) that is derived from historical data.

\[
R_f \approx \frac{\text{Number of Occurrences}}{\text{Unit Time}} \quad (1)
\]

This approach does not explicitly account for fault duration. Rather, faults are implicitly assumed to be brief, and fault probabilities are estimated by integrating the fault rate over an application-specific exposure interval \( t_{\text{exp}} \).

\[
P_f = \int_0^{t_{\text{exp}}} R_f \, dt \quad (2)
\]

As an example, precision landing applications specify an exposure window of 150 s. Satellite faults are believed to occur at a rate of no more than \( 10^{-4} \) per satellite per hour [4]. Conservatively assuming no more than 24 satellites are visible at a time, the overall satellite fault rate \( R_f \) can thus be estimated as less than or equal to \( 24 \times 10^{-4} \) per hour. Integrating the fault rate over the exposure window gives:

\[
P_f \leq 10^{-4} \text{ per 150 s} . \quad (3)
\]

The exposure window approach does not account for persistent faults. Longer duration faults present a bigger risk, however, and should conceptually result in a higher fault probability \( P_f \). To account for this limitation, it is necessary to generalize the notion of a fault probability.

This paper develops an approach for generalizing the conventional fault probability \( P_f \) as an undetected fault probability \( P_{\text{ud}} \). This risk of an undetected fault depends on monitor sensitivity, which can be characterized as the probability that the monitor has failed to detect the fault by a time \( t \) following the initial occurrence of the fault. This missed-detection probability, \( P_{\text{md}}(t) \), is a conditional probability that assumes a fault has occurred.
\[ P_{uf}(t) = \left\{ \begin{array}{ll} 0 & \text{t < } t_{\text{exposure}} \\ 1 & \text{otherwise} \end{array} \right. \]  

(7)

More general forms of \( P_{\text{mad}} \), as the next section discusses, may be an arbitrary monotonically-decreasing functions.

Missed-detection probabilities also impact the risk posed by multiple simultaneous faults. In the past, the risk of a multiple-fault event has been computed through multiplication, under the assumption that fault events are independent and occur with equal probability. Thus, the multiple-fault risk \( P_{\text{mf}} \) is conventionally assessed as:

\[ P_{\text{mf}} = P_f^N. \]  

(8)

As in the case of (2), this multiple-fault probability can be related to the differential fault rate \( R_f \) through integration. In the case of two independent faults, for instance, the differential fault rate can be integrated over a pair of overlapping exposure windows to obtain the two-fault probability.

\[ R_{uf} = R_f^2 \]  

(9)

\[ P_{2uf} = \int_{t_1}^{t_2} \int_{t_1}^{t_2} R_{uf} \, dt_1 \, dt_2 \]  

Here the "2" subscript in the notation for the fault probability (\( P_{2uf} \)) and fault rate (\( R_{2uf} \)) designates the two-fault case.

The integral (9) is easily generalized to incorporate missed-detection probabilities, just as (2) was generalized to obtain (6). For this purpose, the rate of undetected faults \( R_{2uf} \) is substituted for the baseline fault rate \( R_{2f} \). If the monitors for the two faults are independent, then the undetected fault rates can be defined separately with \( t_1 \) designating the time since one event and \( t_2 \), the time since the other (with \( t_1 \leq t_2 \)).

\[ R_{2uf}(t_1, t_2) = R_{uf,1}(t_1) R_{uf,2}(t_2) \]  

(10)

Each of these independent expressions can be expanded in terms of the fault rates and the missed-detection probabilities, as described by (5).

\[ R_{2uf}(t_1, t_2) = R_f^2 P_{\text{md,1}}(t_1) P_{\text{md,2}}(t_2) \]  

(11)

Substituting the undetected fault probability \( R_{2uf} \) for the baseline fault probability \( R_{2f} \) in equation (9) and extending the limits of integration to cover all possible past fault times gives the following equation for the probability of two simultaneous undetected faults \( P_{2uf} \).

\[ P_{2uf} = \int_{t_1}^{t_2} \int_{t_1}^{t_2} R_{uf} \, dt_1 \, dt_2 \]  

(12)

In this case, the undetected fault probability is equivalent to half the conventional fault probability when monitor detection occurs within the exposure time. That is (12) equals half of (8) when \( N = 2 \) and \( P_{\text{mad}} \) is given by (7).

This manuscript will focus on one- and two-fault events for monitors implemented digitally. To analyze these systems, it is necessary to convert equations (6) and (12) to a discrete form. In the discrete form, integrals are replaced with summations over evenly distributed samples with sample period \( \Delta t \).

\[ P_{uf} = \sum_{k=0}^{N} R_f(k) \Delta t = \sum_{k=0}^{N} R_f P_{\text{mad}}(k) \Delta t \]  

(13)

\[ P_{2uf} = \sum_{k=0}^{N} \sum_{l=k}^{N} R_{2uf}(k,l) \Delta t^2 \]  

(14)
MONITOR PERFORMANCE MODEL

GNSS Faults and Time-to-Alert Requirements

GNSS augmentation systems rely on integrity monitors to detect rare measurement anomalies, which might otherwise introduce hazardously large biases into user position solutions. When integrity monitors detect a signal anomaly, the augmentation system can then warn users to exclude affected measurements. Based on this detection and exclusion process, augmentation system users can establish rigorous error bounds to evaluate whether or not they may conduct a safety-critical application, such as aircraft precision approach and landing [5].

Faults that affect GNSS navigation are generally divided into three categories: satellite, signal-in-space and receiver faults. Satellite faults (such as a clock failure) corrupt the GNSS signals directly. Signal-in-space faults (such as anomalous delays caused by severe ionosphere storms) degrade GNSS signals that pass through certain regions of the atmosphere. Receiver faults may affect an individual user or, in the case of a faulty reference station, an entire group of users leveraging differential corrections from that station [6].

For precision applications, it is critical to detect and exclude anomalous measurements quickly, before positioning errors grow overly large. An overly large error is one for which user-evaluated error bounds no longer apply [7]. In general, error bounds continue to apply for some time after the onset of a fault, since measurement filtering slows the rate at which faulty measurements are incorporated in the user navigation solution. When user error bounds do eventually become invalid, the augmentation system must provide a warning to users within a specified time, called the time-to-alert. For safety critical applications, like aviation, the time-to-alert is measured in seconds (10 s for nonprecision approach, 6 s for precision approach, and as little as 1-2 seconds for automated landing [1]).

Fault Monitor

Monitors detect faults by processing raw GNSS measurements. From these, monitor statistics are generated and compared to a threshold. If a monitor statistic exceeds its threshold, the associated measurement is flagged and users are warned to exclude that measurement for the purposes of positioning.

As an example, a distorted code signal can be detected using a Signal Deformation (SD) monitor. The SD monitor looks for so-called “evil waveforms” by computing a set of early, late and prompt correlator values. Differences between these measurements and nominal values (that describe a typical, approximately triangular code-correlation peak) can be linearly combined to generate a monitor statistic which responds to distortions in the shape of the correlator peak. By comparing this statistic to a threshold, the monitor can detect and exclude SD anomalies which might otherwise result in large errors of tens of meters or more [8].

Monitor statistics are typically smoothed by a linear filter to reduce noise. To model the influence of a linear filter on monitor performance, the monitor statistic \( y(k) \) can be written as a function of the unsmoothed input signal \( x(k) \) for each sample time \( k \). For the analysis of this paper, a first-order, linear time-invariant filter will be assumed, as is used in GNSS augmentation systems like WAAS [9] and LAAS [10]. The amount of smoothing applied by the digital filter is specified by a constant \( \alpha \in (0,1] \).

\[
y(k) = (1-\alpha)y(k-1) + \alpha x(k)
\]

Augmentation system integrity depends on rapid detection of faults by comparing \( y(k) \) to a threshold \( T \). This relationship can be used to define the missed-detection probability more precisely.

\[
P_{md}(k) = P\left[y(k) < T \; \forall \; k \in [0,k]\right]
\]

Missed detections occur if the monitor statistic has failed to exceed the threshold at any time since the onset of a fault. Because \( P_{md} \) depends on the behavior of the monitor statistic over a period of time, statistical correlation must be considered in evaluating (16).

Time Correlation of the Monitor Statistic

This subsection models the time correlation in the monitor statistic \( y \), which results both from the smoothing process of (15) and from existing correlation in the raw data \( x \). Though the raw data may exhibit nonstationary statistics, this paper develops an illustrative example, by modeling the raw data as the result of a first-order, time-invariant Markov process. This first-order process generates a model of the raw data \( x \) by applying the following process to a white noise signal \( w(k) \). The result is designated \( x_{nom} \), where the “nom” subscript indicates a random noise component present even under nominal conditions.

\[
x_{nom}(k) = (1-\beta)x_{nom}(k-1) + \beta w(k)
\]

Here \( \beta \) is a parameter that defines the level of correlation in the random noise \( x_{nom} \), with \( \beta \in (0,1] \).

Under faulted conditions, the nominal random noise \( x_{nom} \) is offset by an additional fault-induced bias term \( x_{fl} \). The following equation describes the combined input to the smoothing process (15) when a fault is present.

\[
x(k) = x_{nom}(k) + x_{fl}(k)
\]
To explore a range of correlation behaviors, three values of $\beta$ are considered in this paper. First, a fully-correlated case models $\beta$ approaching 0. In this case, $x_{\text{nom}}$ is a random variable, but one with a constant value through time. A second model is the fully-independent case, in which $\beta$ equals one. In this case, each value of $x_{\text{nom}}$ has no dependence on the previous value. A third model is the partially-correlated case, in which $\beta$ takes an intermediate value. In this paper, the partially-correlated case uses a value of $\beta$ equal to 0.05 (that represents a 10-second time constant for a 2 Hz sampled system, $\beta = 1/10 \text{ s/2 s}^{-1} = 0.05$).

The overall correlation of the smoothed monitor statistic $y$ can be computed by combining the noise process model of (17) with the filter smoothing model of (15), as illustrated by the block diagram of Figure 2. For subsequent analysis, the smoothing filter will be assumed to have the form of (15) evaluated with a 100-second time constant ($\alpha = 0.005$). Accounting for the cascade of the noise and smoothing processes, the monitor statistic $y$ can be related directly to the white noise signal $w$. For the fully-correlated case, this relationship has the following form where $w_0$ is a random variable, constant for all time.

$$y(k) = (1-\alpha)y(k-1) + \alpha x_{\text{nom}}(k) + \alpha w_0$$  \hspace{1cm} (19)

For the fully-independent case, a first-order difference equation relates the monitor statistic to the white noise signal.

$$y(k) = (1-\alpha)y(k-1) + \alpha x_{\text{nom}}(k) + \alpha w(k)$$  \hspace{1cm} (20)

For the partially-correlated case, the relationship is second order.

$$y(k) = \gamma_1 y(k-1) + \gamma_2 y(k-2) + \gamma_3 w(k) + \gamma_4 x_{\text{nom}}(k) + \gamma_5 x_{\text{nom}}(k-1)$$  \hspace{1cm} (21)

The $\gamma$ parameters in this equation have the following values, governed by the time constants of the filter and noise processes.

$$\gamma_1 = (1-\alpha) + (1-\beta)$$
$$\gamma_2 = (1-\alpha)(\beta-1)$$
$$\gamma_3 = \alpha\beta$$
$$\gamma_4 = \alpha$$
$$\gamma_5 = \alpha(\beta-1)$$  \hspace{1cm} (22)

It is intuitive that the missed-detection probability will be worst (highest) in the fully-correlated case, since there is only one opportunity to obtain a noise value $w_0$ that puts the monitor statistic above the detection threshold. The missed-detection probability will be most favorable (lowest) in the fully-independent case, since there are many opportunities to obtain a noise value $w(k)$ that puts the monitor statistic above the detection threshold. The partially-correlated case lies somewhere between the two extremes.

**Transient Analysis**

In order to evaluate the risk posed by an undetected fault, it is necessary to assess $P_{\text{md}}(k)$ in light of the time correlation of the monitor statistic $y$. In the absence of a threshold, the properties of the correlation could be computed from the linear relationships (19)-(21) as the solution of a discrete Lyapunov equation [11]. The thresholding process is nonlinear, however, because the threshold clips the tails of the monitor statistic distribution $p(y(k))$ at each time step. Hence, the monitor statistic distribution is non-Gaussian even if the white noise input is Gaussian.

Histogram-based numerical methods were used to compute the evolution of the monitor statistic probability distribution $p(y(k))$ through time. Monte Carlo simulations are often useful to model nonlinear random processes [12]. In the current application, however, the monitor detection probability per time step may be extremely small (probabilities of $10^{-10}$ or less). Modeling these small levels of probability with a Monte Carlo method is not practical, as exceedingly large particle populations would be required ($10^{12}$ or more particles). The histogram-based simulation approach can achieve arbitrary levels of precision in the far tails of a Probability Density Function (PDF) using a relatively small number of discretized points for low-dimensional systems ($10^{6}$ histogram bins used to simulate a two-dimensional joint PDF).

The appendix describes full details of the simulations used to evaluate $P_{\text{md}}(k)$ for the fully-independent and the partially-correlated cases, governed by the dynamic processes (20) and (21) respectively.

No simulations are needed for the fully-correlated case, since the random input $w_0$ is not a function of time. For this case, $P_{\text{md}}(k)$ can be computed simply by inverting the Gaussian cumulative distribution function $Q$ to determine

![Figure 2. Modeling the Monitor Statistic.](image-url)
for the deterministic monitor statistic bias \( \bar{y}(k) \).

\[
P_{md}(k) = Q^{-1}(-T + \bar{y}(k)) + \left(1 - Q^{-1}(T + \bar{y}(k))\right)
\]

\[
\bar{y}(k) = (1 - \alpha)y(k-1) + \alpha x_{md}(k)
\]

(23)

For all simulations, the nominal standard deviation \( \sigma_y \) for the monitor statistic (in the absence of a threshold) was normalized to one. The threshold \( T \) was set to a value of \( 6\sigma_y \), which is representative for typical integrity monitors used in augmentation systems. The fault-mode input to the monitor smoothing filter was assumed to be a step function, rising instantaneously from zero to \( B \).

For this paper, a range of bias values were simulated, with \( B \in [\sigma_y, 14\sigma_y] \). Representative curves for \( P_{md}(k) \) are illustrated in Figure 3 for the fully-correlated, fully-independent, and partially-correlated cases for two levels of bias. Initially, the monitor has little opportunity to detect the fault, and so the missed-detection probability \( P_{md}(k) \) starts close to one. For longer times (i.e., for higher values of \( k \)), the bias grows, more probability shifts across the monitor threshold, and the missed-detection probability \( P_{md}(k) \) decreases accordingly. For the fully-independent and partially-correlated cases, \( P_{md}(k) \) converges toward zero. In the fully-correlated case, \( P_{md}(k) \) converges toward a constant value, greater than zero.

For the fully-correlated case, the nonzero value of \( P_{md}(k) \) in steady-state reflects an eventual lack of new information available for detection. Only one independent measurement is available in this case, no matter how much time has elapsed.

To contrast the fully-correlated case with the other cases, it is useful to define a per-epoch missed detection probability \( \phi \).

\[
\phi(k) = \frac{P_{md}(k)}{P_{md}(k-1)}
\]

(25)

The time evolution of this quantity is illustrated for the three cases in Figure 4. After dipping initially, \( \phi \) returns to unity for the fully-correlated case. By comparison, for the cases of fully-independent and partially-correlated input noise, \( \phi \) decreases continuously toward a steady-state value. This steady-state value can be interpreted as the largest eigenvalue of the PDF evolution equations (see Appendix).

Of the three cases, the partially-correlated case best represents monitors typically implemented in GNSS augmentation systems. The fully-correlated and fully-independent cases can be considered as limiting results, which bound monitor performance for correlation times longer or shorter than those (100 s smoothing and 10 s noise processes) assumed for the partially-correlated simulation.

**Computing Undetected Fault Probabilities Using \( P_{md} \)**

Using the techniques described in the previous section, the transient for the missed-detection probability \( P_{md}(k) \) can be simulated for an arbitrary number of time steps. For faults introducing only a small bias \( B \), the missed-detection probability may converge very slowly toward zero. Because simulating the missed-detection transient for extended time periods is computationally expensive (particularly for second-order process of the partially-correlated case), an alternative means is desired for modeling the evolution of the \( P_{md} \) transient. In fact, computing the undetected fault probability using (13) or (14) requires that the \( P_{md} \) transient be modeled to infinity!

Fortunately, the three correlation cases discussed in the previous sections can all be extended to infinity by analytical bounding methods. To accomplish this, the \( P_{md} \) transient is first simulated to a transition time step, designated as \( k^* \). For this paper, \( k^* \) was set to 500 s (i.e., 1000 time steps for 2 Hz sampling). The duration of the simulation is arbitrary, but longer durations lead to tighter bounds, as discussed below. The 500 s transition time was selected to allow \( \phi \), as illustrated in Figure 4, to converge near its steady-state value.

Beyond the transition time \( k^* \), \( P_{md} \) may be conservatively bounded by introducing a modeled per-epoch missed-detection probability \( \phi_m \), which is a constant greater than or equal to the actual value \( \phi \). To show that this approach provides a tight and conservative upper bound for \( P_{md} \), it is useful to express the relationship between \( P_{md} \) and \( \phi \) in the following way.

\[
P_{md}(k) = \phi_m(k^*) \prod_{k=k^*}^{k} \phi(k), \quad k \geq k^*
\]

(26)

As long as \( \phi_m \geq \phi(k) \), the following equation provides an upper bound for missed-detection probability after the transition step at \( k^* \).

\[
P_{md}(k) \leq P_{md}(k^*) \phi_m^{k-k^*}, \quad k \geq k^*
\]

(27)
For the fully-correlated case, the steady-state value of $\phi$ is unity, and $\phi_m$ must be set accordingly ($\phi_m = 1$). For the fully-independent and partially-correlated input noise cases, $\phi_m$ approaches its steady state value from above, as illustrated in Figure 4. Hence for these cases, $\phi_m$ can be set conservatively based on the value of $\phi$ at the transition time, $\phi_m = \phi(k^*)$.

Extending $P_{md}(k)$ to large times is necessary for computation of the undetected fault probabilities, defined by (13) and (14). These equations, which involve summations to infinity, can both be evaluated analytically (as convergent geometric series) using the bound (27).

For the one-satellite fault case, combining (13) and (27) gives the following expression.

$$P_{uf} = R_f \Delta t \left( \sum_{k=0}^{k^*} P_{md}(k) + P_{md}(k^*) \sum_{k^*}^{\infty} \phi_m^{-k^*} \right)$$  \hspace{1cm} (28)

The sum of the geometric series can be evaluated analytically [13], resulting in the following equation.

$$P_{uf} = R_f \Delta t \left( \sum_{k=0}^{k^*} P_{md}(k) + \frac{P_{md}(k^*)}{1 - \phi_m} \right)$$  \hspace{1cm} (29)
The two-satellite fault case can be decomposed into three terms, as follows, and the infinite geometric series evaluated in a similar manner.

\[
P_{2uf} = R_f^2 \Delta t^2 \left( \sum_{k=0}^{k^*} \sum_{l=0}^{k^*-1} P_{md,1}^{}(k) P_{md,2}^{}(l) \right)
+ \sum_{k=k^*}^{\infty} \sum_{l=0}^{k^*-1} P_{md,1}^{}(k) P_{md,2}^{}(l)
\]

Evaluating the geometric series for values of \(P_{md}^{}\) beyond \(k^*\), the two-satellite undetected fault probability can be rewritten with finite bounds.

\[
P_{2uf} = R_f^2 \Delta t^2 \left( \sum_{k=0}^{k^*} \sum_{l=0}^{k^*-1} P_{md,1}^{}(k) P_{md,2}^{}(l) \right)
\]

\[
+ \frac{P_{md,2}^{}(k^*)}{1 - \phi_{md,2}^{}(k^*)} \sum_{k=0}^{\infty} \phi_{md,1}^{}(k)
\]

\[
+ \frac{P_{md,1}^{}(k^*)}{1 - \phi_{md,1}^{}(k)} \sum_{k=0}^{\infty} \phi_{md,1}^{}(k)
\]

AUGMENTATION SYSTEM APPLICATIONS

The major objective of this paper is to assess the impact of small, persistent faults on augmentation system integrity. The central issue is to identify which faults can be bounded, and which cannot be bounded, by a multiple-fault integrity allocation of the form illustrated in Figure 1. By evaluating undetected fault risk over a range of threat magnitudes, it is possible to identify the tolerable threats given a particular integrity allocation.

To identify tolerable faults, equations (29) and (31) can be evaluated for a range of faults, modeled by (24) as step inputs of varying magnitude \(B\). Results are illustrated in Figure 5, which plots undetected fault risk for the one-satellite case \((P_{uf}^{})\) as a function of threat magnitude, and in Figure 6, which plots contours of undetected fault risk for the two-satellite case \((P_{2uf}^{})\). Only the partially-correlated case is represented in the figures, since results for the fully-independent case are similar, and results for the fully-correlated case diverge (for \(\phi_{md} = 1\)).

Computed results confirm that large faults, which monitors quickly detect, satisfy integrity requirements but that small faults, which persist for extended periods, do not. To satisfy integrity requirements, the undetected fault probability should not exceed the integrity allocations for the single-fault or multiple-fault events (approximately 10^{-4} and 10^{-8} for Category I LAAS, for instance). As the figures indicate, the undetected fault risk is very large for small faults, particularly those with bias values \((B/\sigma^{})\) smaller than the monitor threshold \((T/\sigma^{})\) of 6. Though these smaller faults are rare, monitors cannot detect and resolve them quickly, so the associated risk exceeds the integrity allocation. In fact, very small threats \((B/\sigma^{} < 2)\) persist for a duration longer than the assumed interval between fault events \((1/R_f^{})\). These extreme cases violate the fundamental assumption that fault events are rare, and hence the proposed analysis does not apply to the smallest of faults (i.e. undetected fault “probabilities” greater than one would result).

Because a dedicated integrity allocation cannot cover small, persistent threats, alternative strategies are needed.

Figure 5. Undetected Fault Probability for a Single Satellite Fault \((P_{uf}^{})\) as a Function of Bias Size Assuming a Fault Rate \((R_f^{})\) of 10^{-4} per hour.

Figure 6. Contours of Undetected Fault Probability for a Two Satellite Fault \((P_{2uf}^{})\) as a Function of Bias Size on Each Satellite, for a Fault Rate \((R_f^{})\) of 10^{-4} per hour.
One possibility is to consider tighter bounds on the threat spaces used to characterize fault modes. If appropriate, a minimum bound on fault magnitude could preclude small, persistent threats and their associated integrity risk. Another possibility is to introduce error bounds, also known as protection levels, to cover two-satellite fault cases. A third, complementary option is to introduce specialized process-control monitors, such as sigma or cusum monitors [14], to reduce the likelihood of persistent faults.

As a final note, it is also significant to note that the results plotted in Figure 5 and Figure 6 do not account for communication delays. In practice, monitor alert messages do not immediately reach users. A short time-to-alert is allocated for monitor messages to be transmitted to users [7]. For some proposed systems, such as the RRAIM and ARAIM architectures proposed for GEAS, broadcast message latency may be significant [3]. To account for undetected fault risks for these systems, it may be necessary to delay the start of the \( P_{md} \) transient by the difference \( \Delta \) between the broadcast latency and the allowed time-to-alert. In this case, a delayed \( P_{md} \) could be defined as follows.

\[
P_{md,\text{delayed}}(k) = \begin{cases} 
1 & k < \Delta \\
\left[ P_{md}(k - \Delta) \right] & k \geq \Delta 
\end{cases}
\]  

(32)

SUMMARY AND CONCLUSIONS

This paper developed a method to account for the risks associated with small, persistent faults that impact the integrity of GNSS augmentation systems. Specifically, the paper focused on cases of one or two simultaneous satellite faults. The resulting methods compute an undetected fault probability \( P_{uf} \), which generalizes the conventional fault probabilities used in the analysis of augmentation system integrity. The undetected fault probability explicitly accounts for monitor performance, through a dependence on the monitor missed-detection probability as a function of time.

Analysis based on these methods indicates that a simple allocation from an overall integrity risk budget may not be sufficient to cover all fault events. In particular, small and persistent faults present a sufficient integrity risk to merit additional mitigation through the introduction of a tightened threat space, new error bounds, or specialized integrity monitoring to detect persistent faults.

REFERENCES


APPENDIX

This appendix describes the histogram-based numerical simulations used to obtain $P_m(k)$. Histogram simulations approximate operations on continuous PDFs by modeling those distributions with a discrete set of bins. For each bin $i$ and for time step $k$, the discrete probability bin has a value labeled $\rho_i(k)$ which is approximately equal to the value of the actual probability distribution for the monitor statistic $p(y(k))$, evaluated at the center location of the $i$-th bin, labeled $y_i$.

$$\rho_i(k) \approx p(y_i(k)) \quad (33)$$

Together, the set of histogram bin probabilities $\rho(k)$ can be represented as a vector $\rho(k)$. The same method can also be used to represent multivariate probability distributions. In the multivariate case, each element of the vector $\rho(k)$ refers to a bin location described by a set of center locations.

$$\rho_i(k) \approx p\left( (x(k), y(k))_i \right) \quad (34)$$

PDF Updates

For both the univariate and multivariate cases, the probability distribution model must be updated at each time step. The processes described in this paper consist of linear and nonlinear operations on the monitor statistic (scalar multiplication, addition, and thresholding) which correspond to linear operations on its PDF (interpolation, convolution, and clipping). Time-step updates may thus be represented as matrix operations on $\rho(k)$.

For example, consider the affine process which scales a single random variable and shifts it by a constant.

$$y(k+1) = \xi_1 y(k) + \xi_2 \quad (35)$$

For this operation, the PDF $\rho(k+1)$ can be computed by mapping the values of $\rho(k)$ to new locations $y(k+1)$ that are scaled (by $\xi_1$) and shifted (by $\xi_2$) relative to the original $y(k)$. Probability values must also be weighted by a constant to ensure the integral of the new PDF is one. These operations can be implemented by an interpolation algorithm, such as linear or cubic interpolation [12]. If the interpolation coefficients are stored in a matrix $A$, then the process described by equation (35) modifies the probability distribution as follows.

$$\rho(k+1) = A \rho(k) \quad (36)$$

In practice, the interpolation matrix $A$ is sparse and may be quicker to implement as an interpolation (\texttt{interpl} in Matlab) rather than as a matrix multiplication.

The addition of two random variables may also be computed as a matrix operation on $\rho(k)$. Consider the addition of the random variable $y(k)$ to a white noise variable $w(k)$.

$$y(k+1) = y(k) + w(k) \quad (37)$$

For this operation, the output distribution is the convolution of two input distributions. The convolution operator can be implemented using a convolution matrix $C$, where each row of the convolution matrix is a discrete representation of the shifted PDF for $w(k)$.

$$\rho(k+1) = C \rho(k) \quad (38)$$

Again, the matrix format of this equation is convenient for analysis, but inefficient for practical implementation. Since $C$ is analytically sparse, in the multivariate case, or numerically sparse, in the univariate case where the tails of the PDF for $w(k)$ are vanishingly small, it is more computationally efficient to implement convolution directly (i.e., using \texttt{conv} in Matlab) rather than as a matrix multiplication.

Even the thresholding process can be implemented as a matrix operation on the distribution $\rho(k)$. Specifically, the thresholding operation sets all values of the PDF to zero for values of $y_i$ greater than $T$ or less than $-T$. This operation can be implemented by multiplying $\rho(k)$ by a diagonal threshold matrix $\Gamma$ with diagonal values of one (between the thresholds) or zero (beyond the thresholds). A subsequent scaling step can restore the integral of the PDF model to one.

Fully-Independent Input Signal

The procedure for updating the PDF models in the fully-independent case and the partially-correlated case are similar but not identical. The fully-independent case is considered first, since it involves only a one-dimensional PDF, which is easier to manipulate than the two-dimensional joint PDF for the partially-correlated case.

To simulate the monitoring process for a fully-independent input signal, as described by the (20), it is useful to break the problem into several parts. First, it is convenient to remove the deterministic fault term $x_f$. For this purpose, $y$ is decomposed into an equivalent sum of a deterministic component $\overline{y}$ and a random component $\hat{y}$. This decomposition significantly reduces the grid resolution needed to simulate the evolution of $\rho(k)$.

$$y(k) = \hat{y}(k) + \overline{y}(k)$$

and

$$\hat{y}(k) = (1 - \alpha) \hat{y}(k-1) + \alpha w(k) \quad (39)$$

$$\overline{y}(k) = (1 - \alpha) \overline{y}(k-1) + \alpha x_{f,k}$$

The addition of two random variables may also be computed as a matrix operation on $\rho(k)$. Consider the addition of the random variable $y(k)$ to a white noise variable $w(k)$. The processes described in this paper consist of linear and nonlinear operations on the monitor statistic (scalar multiplication, addition, and thresholding) which correspond to linear operations on its PDF (interpolation, convolution, and clipping). Time-step updates may thus be represented as matrix operations on $\rho(k)$.

For example, consider the affine process which scales a single random variable and shifts it by a constant. The processes described in this paper consist of linear and nonlinear operations on the monitor statistic (scalar multiplication, addition, and thresholding) which correspond to linear operations on its PDF (interpolation, convolution, and clipping). Time-step updates may thus be represented as matrix operations on $\rho(k)$.
Based on this two-component decomposition, it is possible to evaluate the $P_{md}$ condition of (16) with the following equivalent expression.

$$P_{md}(k) = P \left\{ -T - \bar{y}(k) < \hat{y}(k) < T - \bar{y}(k) \right\} \quad \forall \: k \in [0,k]$$

(40)

This formulation has the advantage that the probability distribution for the random component $\hat{y}$ changes relatively little between subsequent time steps, allowing for a more effective use of the histogram grid.

The update for the PDF model for $\hat{y}$ consists of an interpolation operation (based on a scaling by 1-$\alpha$), a convolution operation (based on the addition of the variable $\alpha w$), and a thresholding operation. Together, these three steps may be represented by matrices for interpolation ($A$), convolution ($C$), and thresholding ($\Gamma$) which operate on the PDF model $\rho(k)$ to generate an intermediate likelihood vector $L(k+1)$.

$$L(k+1) = \Gamma(k)CAp(k)$$

(41)

The probability distribution at the new time step can be computed from the likelihood by normalizing the total probability to one. The normalizing value $\phi$ can be obtained by “integrating” the likelihood over the histogram bins, each spaced at a regular interval of $\Delta y$.

$$\phi = \sum_i L_i(k+1)\Delta y$$

(42)

To preserve an overall probability of one, the resulting PDF model at the new time step is normalized by this quantity, which is equivalent the missed-detection probability per epoch defined by (25).

$$\rho(k+1) = \frac{1}{\phi(k)} \Gamma(k)CAp(k).$$

(43)

The quantity $\phi$ can be interpreted as a per epoch missed-detection probability because the thresholding step, associated with the $\Gamma$ matrix, is the only matrix multiplication which changes the total probability of $\rho(k)$. By comparison, the interpolation and convolution steps both preserve total probability. That is, $C$ and $A$ are constructed as unitary matrices. The total probability of a missed detection over an extended time can thus be computed as the product of the $\phi$ values at all time step subsequent to the onset of the fault.

$$P_{md}(k) = \Pi_{k=0}^k \phi(\hat{k})$$

(44)

**Partially-Correlated Input Signal**

For the partially-correlated input signal, the update process is second order, as described by (21). Computing the probability distribution for the second-order process requires modeling a joint PDF for a pair of monitor statistic values at successive time steps: $p(y(k), \hat{y}(k-1))$.

An important feature of this multivariate PDF is that the joint distribution for $y(k)$ and $\hat{y}(k-1)$ is highly correlated. Accordingly, most of the probability in the joint PDF is concentrated in a relatively narrow band around the line $y(k) = \hat{y}(k-1)$.

Higher accuracy can be obtained (for the same total number of histogram bins) if bins are concentrated within this narrow band. To maximize simulation accuracy, it is thus useful to transform the probability distribution from one involving successive time steps, $y(k)$ and $\hat{y}(k-1)$, to one involving the current time step and a difference, $y(k)$ and $\delta(k)$, where $\delta$ is defined as follows.

$$\delta(k) = y(k) - y(k-1)$$

(45)

Substituting (45) into (21) gives the following pair of equations to describe the second-order process.

$$\begin{align*}
\delta(k) &= (\gamma_1 + \gamma_2 - 1) y(k-1) - \gamma_2 \delta(k) \\
&+ \gamma_3 w(k) + \gamma_4 x_{flt}(k) + \gamma_5 x_{flt}(k-1) \\
y(k) &= y(k-1) + \delta(k)
\end{align*}$$

(46)

These equations can be further transformed, in the manner of (39) to transfer the deterministic component to the threshold, leaving only random components $\hat{\delta}$ and $\hat{\hat{y}}$.

$$\begin{align*}
\hat{\delta}(k) &= (\gamma_1 + \gamma_2 - 1) \hat{y}(k-1) - \gamma_2 \hat{\delta}(k) + \gamma_3 w(k) \\
\hat{\hat{y}}(k) &= \hat{\hat{y}}(k-1) + \hat{\delta}(k)
\end{align*}$$

(47)

With this formulation of the dynamic process, the PDF model can be first updated in $\hat{\delta}$ and subsequently updated in $\hat{\hat{y}}$. The first update involves a scaling (by $-\gamma_2$) and a shift (by the $\hat{\hat{y}}$ term) that can be implemented as an interpolation matrix $A_{\delta}$. The first update also involves the addition of a white noise term, which can be implemented by the convolution matrix $C$. The second update step of (48) introduces another shift operation that can be represented by a second interpolation matrix $A_{\hat{\hat{y}}}$. Finally, the threshold matrix $\Gamma$ is applied. Accounting for distribution normalization by $\phi$, the complete PDF update for this second-order process largely resembles that for the first-order process, as described by (43). The only significant difference is an additional $A$-matrix, which designates a second interpolation step.
\[ p(k+1) = \frac{1}{\phi(k)} \Gamma(k)A_{\phi}CA_{\phi}p(k) \] (48)

The transient \( P_{md}(k) \) for the second-order process is computed using (44), in the same manner as for the first-order process.

**Steady-State Solution**

A significant result of the previous sections is that the likelihood update is linear, even though the underlying process, which involves thresholding a random variable, is nonlinear. The likelihood may thus be analyzed as a linear process, in terms of its eigenvectors and eigenvalues. For this purpose, it is useful to rewrite the first-order process of (43) as a recursive equation for likelihood.

\[ L(k+1) = \Gamma(k)CA \cdot L(k) \] (49)

In this form, the likelihood update is expressed as a time-varying linear system. The time-varying term \( \Gamma \) is associated with the thresholding step, where the threshold is shifted by the deterministic component of the monitor statistic \( \overline{y} \). This deterministic component \( \overline{y} \) converges towards a constant value in steady-state. Thus, in steady-state, the likelihood update converges toward the following linear time-invariant form.

\[ L(k+1) = \Gamma_{ss}CA \cdot L(k) \] (50)

Also, given that the system is stable (with all eigenvalues less than one), the solution converges toward the dominant eigenmode in steady state. Thus, for large values of \( k \), the PDF model \( p(k) \) converges toward the dominant eigenvector of the matrix product \( \Gamma_{ss}CA \). Once all other eigenmodes have decayed away, the matrix’s largest eigenvalue \( \lambda_{max} \) may be interpreted as a ratio of the magnitudes of the solution vector between successive time steps. This ratio is equivalent to the per-epoch missed detection probability. Hence, \( \phi(k) \) converges to \( \lambda_{max} \) in steady-state. This property supports the tightness of the bound specified by (27). As long as the system has converged close to steady-state at the transition time \( k^* \), then \( \phi(k^*) \) is approximately equal to \( \lambda_{max} \), and hence \( \phi(k^*) \approx \phi(k) \).