


Residual-based multi-filter methodology for all-source fault detection, exclusion, and performance monitoring

Juan Jurado^{1,*}  | John Raquet^{2,†} | Christine M. Schubert Kabban³ | Jonathon Gipson⁴

¹ U.S. Air Force Test Pilot School, Edwards AFB, CA 93524, USA

² IS4S-Dayton, Integrated Solutions for Systems, Inc., Beavercreek, OH 45324, USA

³ Department of Mathematics and Statistics, Air Force Institute of Technology, Beavercreek, OH 45324, USA

⁴ Department of Electrical Engineering, Air Force Institute of Technology, Beavercreek, OH 45324, USA

Correspondence

Juan Jurado, U.S. Air Force Test Pilot School, Edwards AFB, CA, 93524, USA.
Email: Juan.Jurado.1@us.af.mil

*At the time of writing, he was a Ph.D. student at the Air Force Institute of Technology, OH 45324, USA.

†At the time of writing, he was a Professor of Electrical Engineering at the Air Force Institute of Technology, OH 45324, USA.

Funding information

Air Force Materiel Command

Abstract

All-source navigation has become increasingly relevant over the past decade with the development of viable alternative sensor technologies. However, as the number and type of sensors informing a system increases, so does the probability of corrupting the system with sensor modeling errors, signal interference, and undetected faults. Though the latter of these has been extensively researched, the majority of existing approaches have constrained faults to biases and designed algorithms centered around the assumption of simultaneously redundant, synchronous sensors with valid measurement models, none of which are guaranteed for all-source systems. As part of an overall all-source assured or resilient navigation objective, this research contributes a fault- and sensor-agnostic fault detection and exclusion method that can provide the user with performance guarantees without constraining the statistical distribution of the fault. The proposed method is compared against normalized solution separation approaches using Monte-Carlo simulations in a 2D non-GPS navigation problem.

1 | INTRODUCTION

All-source navigation and Assured Position Navigation and Timing (APNT) have become increasingly important research areas over the past two decades, especially as alternative navigation sensor technologies (e.g., vision (Venable, 2016), radio (Curro & Raquet, 2016), magnetic (Canciani & Raquet, 2016), etc.) have been matured and integrated into navigation systems (Grejner-Brzezinska et al., 2016). However, each additional sensor allowed into a navigation system introduces another opportunity for corrupting the navigation solution with errors in sensor modeling, unexpected signal interference, or undetected sen-

sor faults. Of these challenges, the latter has been extensively researched (Bhatti, 2006; Bhatti, Ochieng, & Feng, 2007a, 2007b; Brenner, 1996, 1990; Brumback & Srinath, 1987; Call, Ibis, McDonald, & Vanderwerf, 2006; Joerger, Chan, & Pervan, 2014; Kerr, 1980; Lee et al., 1986; Parkinson & Axelrad, 1988; Sturza, 1988; van Graas & Farrell, 1993; Young & McGraw, 2003) as a multi-sensor fault detection problem where each satellite in the Global Positioning System (GPS) constellation is regarded as a different (albeit identical in nature and synchronous) sensor in the multi-sensor system, and the “fault” is defined as an unmodeled bias that is assumed to only affect one of the sensors (satellites) at any given time. As shown in

Jurado, Raquet, and Schubert Kabban (2019, 2020), our overall research motivation is to create a resilient sensor management system that provides APNT through the online detection and self-correction (i.e., auto-tuning) of sensor models that do not match observed measurements. In support of this overall effort, the specific developments presented in this paper seek to determine when any of the above sources of corruption are present by detecting any general mismatches between a sensor's stated model (i.e., measurement function, function parameters, and error covariance matrix) and its observed measurements, where an unmodeled bias is simply one specific case of a mismatch. Additionally, our research shifts away from identical and synchronous sensors, such as GPS satellites, and focuses on all-source multi-domain (e.g., position, velocity, etc.) and asynchronous sensors. In the following section, we discuss traditional techniques generally used in fault detection and exclusion and highlight the novel developments and adaptations we have made in order to achieve our research objectives.

2 | BACKGROUND

Multi-sensor Fault Detection and Exclusion (FDE) research to date has generally approached the FDE problem through the use of either least-squares (Sturza, 1988; van Graas & Farrell, 1993) or filtered (Joerger et al., 2014; Young & Mcgraw, 2003) approaches. In general, least-squares approaches rely on the availability of redundant measurements and perform the FDE function on a sample-by-sample basis. This approach is useful for systems where simultaneously redundant measurements are available by design, and the sensors in question measure the similar quantities (e.g., the GPS constellation). In contrast, filtered approaches rely on estimation filters, such as the Kalman Filter (KF), Extended Kalman Filter (EKF), and Unscented Kalman Filter (UKF), to integrate measurements from various sensor types into a consolidated state-space estimate that can then be tested for faults using a variety of test statistics. In this case, the fault exclusion function is often provided by employing a series of subfilters, each excluding the measurements from a subset of sensors in order to guarantee the existence of a fault-free solution. One of the most prolific filtered approaches, referred to as Normalized Solution Separation (Young & Mcgraw, 2003), is based on testing the statistical distribution of the difference between the state-space estimate obtained by a filter informed by all available sensors and each of the various exclusion subfilters previously described. Based on our all-source research objective, filtered approaches appear most useful since, by definition, we seek to integrate information from sensors

of various types (i.e., position, velocity, pseudorange, etc.). However, as later shown, Normalized Solution Separation methods were found ineffective for detecting sensor model mismatches (i.e., faults) that do not result in significant differences in state-space estimates between the main filter and the subfilters. One example of such faults is an incorrectly stated sensor measurement error covariance matrix and is later shown in Section 4. Additionally, as part of our research objective, we also seek to provide users with a measure of system performance guarantee beyond the filter-computed estimation error covariance since filter-estimated error statistics are not guaranteed to be consistent in the presence of undetected faults. While the majority of FDE research has solved this problem via the computation of system integrity figures such as Horizontal Protection Level (HPL) and associated alert limits, the lack of an assumed fault-present statistical distribution prevents us from estimating quantities such as a probability of missed detection, which are the basis for the majority of integrity computations.

Given the developments required to enable our all-source research objective, our proposed method, henceforth referred to as Sensor-Agnostic All-source Residual Monitoring (SAARM), provides a significant contribution to the state-of-the-art in that it: enables reliable fault-agnostic and sensor-agnostic FDE across a larger variety of fault types when compared to Normalized Solution Separation and provides a method of establishing system performance guarantees without defining a fault-present condition or distribution, or a probability of missed detection. In this work, we also demonstrate our FDE approach using "all-source" sensors across various domains and with different update rates, which directly addresses the emerging all-source APNT challenge.

The remainder of this paper is divided into three additional sections. Section 3 develops the necessary multi-filter multi-sensor notation, the residual-based test statistic, the fault detection and exclusion process, and the system performance assumptions and guarantees. In Section 4, the detection performance of the proposed method is compared against a normalized solution separation method in a variety of simulated all-source navigation problems. Finally, Section 5 summarizes the research contributions and provides ideas for future work.

3 | METHODOLOGY

3.1 | Multi-sensor multi-filter notation

This section expands the conventional Kalman filter (Kalman, 1960) notation from Maybeck (1982, 1984)

to include estimates from multiple filters as well as measurements from multiple non-identical sensors. The notation and underlying considerations will be crucial in the later development of the residual-space test statistic and the resulting fault exclusion process. Consider a (possibly) nonlinear dynamic system of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] + \mathbf{G}(t)\mathbf{w}(t), \quad (1)$$

where \mathbf{x} is the $N \times 1$ navigation state vector containing the system states, \mathbf{u} is the control input vector, \mathbf{G} is an $N \times W$ linear operator, and \mathbf{w} is a $W \times 1$ white Gaussian noise process with a $W \times W$ continuous process noise strength matrix \mathbf{Q} . Suppose the discretized (Van Loan, 1978) system states are estimated by J separate filters. Then at time $t = t_k$, the system state estimate vector and corresponding state estimation error covariance matrix from filters $j = 1, \dots, J$ are given by $\hat{\mathbf{x}}^{[j]}(t_k)$ and $\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}^{[j]}(t_k)$, respectively. Next, each of the J filters can be informed by any, all, or a subset of I sensors. At time t_k , the $i = 1, \dots, I$ sensor provides (possibly) multidimensional $Z_i \times 1$ measurements of the form

$$\mathbf{z}^{[i]}(t_k) = \mathbf{h}^{[i]}[\mathbf{x}(t_k), \mathbf{u}(t_k), t_k] + \mathbf{v}^{[i]}(t_k), \quad (2)$$

where $\mathbf{h}^{[i]}$ is a (possibly) nonlinear measurement function, and $\mathbf{v}^{[i]}(t_k)$ is a $Z_i \times 1$ discrete white Gaussian noise process with covariance matrix $\mathbf{R}^{[i]}(t_k)$. Immediately prior to a measurement update, the estimated measurement for sensor i from filter j , $\hat{\mathbf{z}}^{[i,j]}$, is generated using

$$\hat{\mathbf{z}}^{[i,j]}(t_k^-) = \mathbf{h}^{[i]}[\hat{\mathbf{x}}^{[j]}(t_k^-), \mathbf{u}(t_k), t_k], \quad (3)$$

while its estimated covariance matrix, $\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{[i,j]}(t_k^-)$, is generated based on the type of filtering algorithm. For example, in a linearized filter (such as an Extended Kalman Filter), it can be computed using

$$\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{[i,j]} = \mathbf{H}^{[i]} \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}^{[j]} \mathbf{H}^{[i]T}, \quad (4)$$

where the time index (t_k^-) is omitted for simplicity, and $\mathbf{H}^{[i]}$ represents the Jacobian of $\mathbf{h}^{[i]}$ about the point $\hat{\mathbf{x}}^{[j]}(t_k^-)$. For information on generating $\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}$ in an UKF, the reader is referred to Wan and Van Der Merwe (2000). Finally, the (so-called) pre-update residual vector computed between sensor i and filter j , $\mathbf{r}^{[i,j]}$, and associated covariance matrix, $\mathbf{P}_{\mathbf{r}\mathbf{r}}^{[i,j]}$, is given by

$$\mathbf{r}^{[i,j]}(t_k) = \mathbf{z}^{[i]}(t_k) - \hat{\mathbf{z}}^{[i,j]}(t_k^-) \quad (5)$$

$$\mathbf{P}_{\mathbf{r}\mathbf{r}}^{[i,j]}(t_k) = \mathbf{R}^{[i]}(t_k) + \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{[i,j]}(t_k^-). \quad (6)$$

3.2 | Fault detection test statistic

Having derived the residual vector, $\mathbf{r}^{[i,j]}(t_k)$, and its associated covariance matrix, $\mathbf{P}_{\mathbf{r}\mathbf{r}}^{[i,j]}(t_k)$, in Equations (5) and (6), we now define a residual-space test statistic to determine if a set of observed residuals between a specific sensor-filter pair are adhering to their expected distribution. Since our goal is to limit the assumptions on the type of fault (i.e., the fault could be a bias, an incorrectly stated noise covariance matrix, or incorrect calibration of measurement function parameters), we did not model two competing distributions as would be needed to employ a Likelihood Ratio Test (LRT) (Kay, 1998). Instead, we focused on the distribution resulting from summing the squared Mahalanobis distance (De Maesschalck, Jouan-Rimbaud, & Massart, 2000) across a sequence of pre-update residuals and selecting a threshold based on a desired probability of false alarm, P_f .

Given a Z_i -dimensional Gaussian distribution with mean $\boldsymbol{\mu}$, and covariance matrix $\boldsymbol{\Sigma}$, the squared Mahalanobis distance, d^2 , between an observation \mathbf{y} , and the centroid of the distribution is then given by

$$d^2 = (\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu}). \quad (7)$$

Additionally, d^2 is known (Casella & Berger, 2002; De Maesschalck et al., 2000) to follow a Chi-Square distribution with Z_i degrees of freedom. Moreover, the sum of M independent d^2 distances is also known to follow a Chi-Square distribution with $M \times Z_i$ degrees of freedom. As proven in Maybeck (1982) and Young and McGraw (2003), Kalman filter pre-update residuals form a zero-mean white sequence. As such, we let $\mathbf{y} = \mathbf{r}^{[i,j]}(t_k)$ from Equation (5), $\boldsymbol{\Sigma} = \mathbf{P}_{\mathbf{r}\mathbf{r}}^{[i,j]}(t_k)$ from Equation (6), and $\boldsymbol{\mu} = \mathbf{0}$. Subsequently, we can compute the fault detection test statistic, $\chi_{[i,j]}^*$, using

$$\chi_{[i,j]}^* = \sum_{s=k-M}^k \mathbf{r}^{[i,j]T}(t_s) [\mathbf{P}_{\mathbf{r}\mathbf{r}}^{[i,j]}(t_s)]^{-1} \mathbf{r}^{[i,j]}(t_s), \quad (8)$$

where M is the number of trailing samples in the residual sequence, a fault is declared if

$$\chi_{[i,j]}^* > \chi^2(1 - \alpha, M \times Z_i), \quad (9)$$

and α is derived from the overall desired system P_f , which is further discussed in Section 3.3. It is important to note the test above can only determine if *any* of the I sensors providing measurement updates to filter j is faulty. In order to determine the actual sensor(s) within filter j that are faulty, additional assumptions and computations must be made, as shown in the next section.

3.3 | Fault identification process

Up to this point, we've defined how a time sequence of residual vectors from a specific sensor-filter combination may be tested for likelihood, without making assumptions on the domain of the sensor measurement, or the type of fault. Here, it is important to emphasize that a fault detection derived from a set of residual vectors from a particular sensor-filter pair (i, j) does not imply that sensor i is faulty. It is only an indication of an inconsistency between the information provided by sensor i and the rest of the sensors informing filter j . In other words, low-likelihood residuals can then either be caused by faulty measurements, $\mathbf{z}^{[i]}$, or faulty estimated measurements, $\hat{\mathbf{z}}^{[i,j]}$, the latter of which is influenced by all sensors informing filter j whose state-space overlaps with sensor i . Therefore, in order to identify and exclude the faulty measurements, we developed a "fault consensus" process that associates the presence of a sensor with the presence of a fault in order to determine the most probable sensor associated with faulty results. Though the proposed method is not limited to just single sensor faults, it is best to begin our discussion with this case before scaling to the generalized multiple simultaneous fault cases. The next two sections develop the single and multiple fault cases, respectively.

3.3.1 | Single serial faults

As described in Section 2, a commonly assumed fault scenario is a single sensor fault per testing epoch (i.e., during a single M -sample test window in our case). Multiple faults are still considered, but restricted to occur serially. In this case, we set up our fault identification process by creating $J = I$ filters, each informed by a unique set of $I - 1$ sensors. In other words, each filter excludes one of the I sensors. Here, it is important to note two points. First, since we expect all-source sensors to be non-identical, some states may become unobservable within a particular filter if the only sensor that has observability over them is excluded from that filter. To prevent potential numerical issues with the covariance of unobservable states, we can remove unobservable states from each subfilter or perform a stochastic observability test (Bageshwar, Gebre-Egziabher, Garrard, & Georgiou, 2009) to detect unbounded growth in a subfilter's position error covariance matrix. Second, as with other parallel-filter methods, a "main filter" that is informed by all sensors is also created, but in our method, we do not use its information for solution separation comparisons, thereby eliminating the need for computing the cross-covariance terms between it and all other filters. Having designed the set of filters using this method guarantees,

under the assumption that, at most, one sensor can fail at a time, at least one of the filters will be completely unaffected by faulty measurements. As shown below, we can then use this axiom in conjunction with the full set of (i, j) residual test results to determine the culprit sensor.

We begin the fault identification process by populating the $I \times J$ (which becomes $I \times I$ in this single-fault case) test results matrix, \mathbf{T} , using

$$\mathbf{T}(i, j) = \begin{cases} 0, & \text{Sensor } i \text{ does not inform filter } j, \\ 0, & \chi^2_{[i,j]}^* \leq \chi^2(1 - \alpha, M \times Z_i) \text{ (no fault detected),} \\ 1, & \chi^2_{[i,j]}^* > \chi^2(1 - \alpha, M \times Z_i) \text{ (fault detected).} \end{cases} \quad (10)$$

Figure 1 illustrates the information from each sensor-filter pair needed to populate \mathbf{T} in the case where the j^{th} filter excludes the i^{th} sensor. In the figure, each of the $i = 1, \dots, I$ rows corresponds to the measurement, $\mathbf{z}^{[i]}$, and its associated error covariance matrix, $\mathbf{R}^{[i]}$, obtained from the i^{th} sensor. These two parameters define the modeled distribution of the sensor measurement and make up the first half of Equations (5) and (6), respectively. Next, each of the $j = 1, \dots, J$ ($J = I$ in this single fault case) columns corresponds to the estimated measurement, $\hat{\mathbf{z}}^{[i,j]}$, and its associated error covariance matrix, $\mathbf{P}_{\hat{\mathbf{z}}}^{[i,j]}$. These two parameters make up the remainder of Equations (5) and (6) and define the modeled distribution of the estimated sensor measurement. As shown in the figure, these last two parameters are influenced by all sensors informing the filter in the j^{th} column, which corresponds to all sensors except the i^{th} sensor.

A fault is declared when \mathbf{T} contains any non-zero entries. Here, it is important to highlight that based on Equation (10), $\mathbf{T}(i, j) = 0$ when Sensor i does not inform filter j , which forces the diagonals of \mathbf{T} to zero in the $J = I$ case. This means a fault can be declared when any of the $I^2 - I$ test statistics from Equation (9) that are contained in \mathbf{T} result in a fault. Given each row in \mathbf{T} is informed by a particular sensor, we expect some interdependence among the "cells" in \mathbf{T} ; however, since this dependence is not predictable *a priori*, we can upper-bound the family-wise error rate on the entire set of tests found in \mathbf{T} using a Bonferroni correction (Bonferroni, 1936), which is not affected by such dependence (Goeman & Solari, 2014). Therefore, in order to guarantee a maximum system (i.e., family-wise) $P_f \leq \alpha_{\max}$, we compute each α from Equation (9) in \mathbf{T} using

$$\alpha = \frac{\alpha_{\max}}{I^2 - I}. \quad (11)$$

Table 4 summarizes individual α values resulting from a series of desired α_{\max} rates in the range $\alpha_{\max} \in \{1 \times 10^{-3}, 1 \times 10^{-1}\}$ along with actual P_f rates achieved in the Monte Carlo simulations detailed in Section 4.

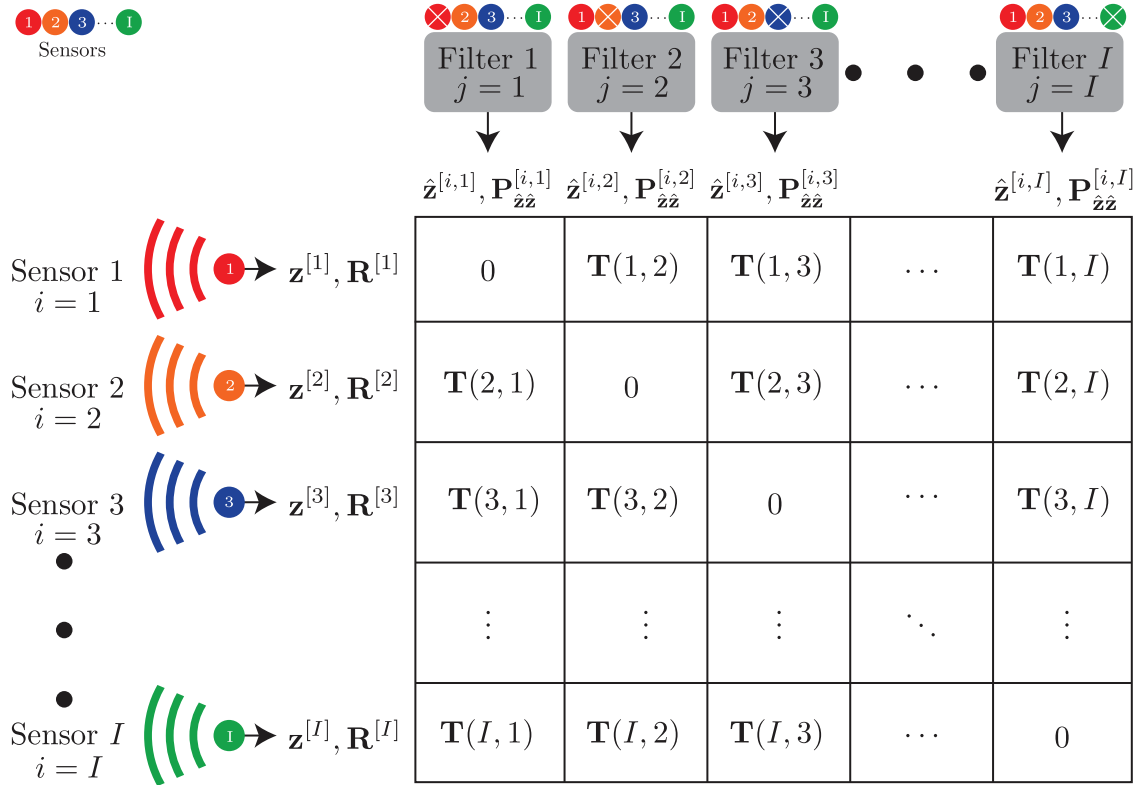


FIGURE 1 Illustration of the multi-sensor multi-filter test statistic matrix, \mathbf{T} [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

Under our proposed method, if a fault is declared, the culprit sensor may only be identified if a consensus is reached. That is, since each sensor is excluded from one filter, we can identify a faulty sensor if only a single filter (namely the filter that excluded it) remains fault free. Mathematically, we first compute the fault scores vector, \mathbf{s} , whose dimension is equal to the number of filters, J , using

$$\mathbf{s}(j) = \sum_{i=1}^I \mathbf{T}(i, j), \tag{12}$$

which produces a sum across the rows (sensors) for each column (filter) in \mathbf{T} . Once computed, we have four possible scenarios:

1. If \mathbf{s} contains all zeros (i.e., $[0 \ 0 \ 0 \ 0]$), then no fault has been detected.
2. If \mathbf{s} contains at least one non-zero and more than one zero (i.e., $[0 \ 0 \ 1 \ 0]$), then a fault is declared, but the culprit is not yet identified. However, the SAARM position estimate is still bounded by the Guaranteed Position Zone (GPZ) defined in Section 3.4.
3. If $\mathbf{s}(j)$ is the only zero remaining in \mathbf{s} (i.e., $[0 \ 1 \ 1 \ 1]$), then a fault is declared and the culprit sensor is the sensor that was excluded from the j^{th} column in \mathbf{T} , or the j^{th} filter, if constructed according to Figure 1.

4. Finally, if \mathbf{s} contains no zeros (i.e., $[1 \ 1 \ 1 \ 1]$), then more than one sensor is faulty, and the assumptions of the test have been violated.

Each of these “states” can be used in conjunction with the performance guarantee computations presented in Section 3.4 in order to continuously inform users of their APNT protection status. Depending on the type and dynamics of the fault, as well as the set and type of sensors in the system, the results in T may continue to change during every epoch and eventually lead to a culprit. If and when a culprit is determined, the corresponding fault-free filter is used as the new main filter, and a new set of $I - 1$ filters is initialized using its state-space estimate and associated error covariance matrix. The process can then be repeated sequentially for multiple serial faults with the assumption that a second fault does not occur during the first M samples after having re-spawned the filter set, which will be addressed in the next section.

3.3.2 | Simultaneous faults

The serial-fault methodology described above can be easily scaled to enable detection of a secondary fault occurring during the first M samples after an initial fault, as

TABLE 1 Sensor-filter configuration for layer $J_1, I = 5$ sensors

Sensor	Included in filter				
	1	2	3 ^a	4	5
1		•	•	•	•
2	•		•	•	•
3	•	•		•	•
4	•	•	•		•
5	•	•	•	•	

^aThis filter is uncorrupted by faulty Sensor 3 measurements

well as multiple simultaneous faults. To do so, we first re-define the number of filters required, J , the structure of the associated test results matrix, \mathbf{T} , and the dimension of the faults score vector, \mathbf{s} , as functions of the assumed maximum number of simultaneous faults, which we define as a “layer.” In general, the number of additional filters required, J , for each layer, N , is given by

$$J_N = \binom{I}{I-N} = \frac{I!}{N!(I-N)!}. \quad (13)$$

As shown in Section 3.3.1, in layer one, we assumed $N = 1$ simultaneous fault was possible and created

$$J_1 = \binom{I}{I-1} = \frac{I!}{1!(I-1)!} = I \quad (14)$$

filters each excluding one sensor, which were then used to populate $\mathbf{T}_1 \in \mathbb{R}^{I \times J_1}$, detect the fault, and identify the single culprit using \mathbf{s}_1 . If we now assume $N = 2$ simultaneous faults are possible, we require an additional

$$J_2 = \binom{I}{I-2} = \frac{I!}{2!(I-2)!} = \frac{I^2 - I}{2} \quad (15)$$

filters each excluding two sensors, which are then used to populate $\mathbf{T}_2 \in \mathbb{R}^{I \times J_2}$. Using this two-layer configuration, the culprit sensor in a single fault scenario can continue to be identified as previously described, using \mathbf{T}_1 and \mathbf{s}_1 . In the case of a simultaneous fault, \mathbf{s}_1 indicates that the single fault assumption has been violated (no zeros remain), which prompts the system to use \mathbf{T}_2 and \mathbf{s}_2 to identify the two culprits. In the case of a secondary fault during the first M samples after an initial fault, the subset of filters in the J_2 layer that excluded the first culprit corresponds exactly to the new J_1 layer of filters needed after re-spawning, which enables uninterrupted fault detection.

For example, consider a system with $I = 5$ sensors where up to two simultaneous sensor faults are assumed. The first layer consists of $J_1 = 5$ filters, and each filter excludes one of the sensors, as shown in Table 1. Using Equation (15), the second layer consists of $J_2 = 10$ filters, and each filter excludes two of the sensors, as shown in Table 2. Sup-

TABLE 2 Sensor-filter configuration for layer $J_2, I = 5$ sensors

Sensor	Included in filter									
	1	2 ^a	3	4	5 ^a	6	7	8 ^a	9 ^b	10
1					•	•	•	•	•	•
2		•	•	•				•	•	•
3	•		•	•		•	•			•
4	•	•		•	•		•		•	
5	•	•	•		•	•		•		

^aThese filters are uncorrupted by faulty Sensor 3 measurements

^bThis filter is uncorrupted by faulty Sensor 3 and Sensor 5 measurements

TABLE 3 Sensor and fault configuration for Monte Carlo simulations

Scenario	Fault	Affected	
		sensor	Other sensors
1	1.0 [m/s] bias	VEL1	POS1, VEL2, POS2
2	2x \mathbf{R} scale	VEL1	POS1, VEL2, POS2

VEL1: 2D velocity sensor, $\mathbf{R} = \text{diag}([1^2, 1^2])$ [m^2/s^2], Update Rate: 0.5 [s]

POS1: 2D position sensor, $\mathbf{R} = \text{diag}([5^2, 5^2])$ [m^2], Update Rate: 1.0 [s]

VEL2: 2D velocity sensor, $\mathbf{R} = \text{diag}([2^2, 2^2])$ [m^2/s^2], Update Rate: 1.5 [s]

POS2: 2D position sensor, $\mathbf{R} = \text{diag}([10^2, 10^2])$ [m^2], Update Rate: 2.0 [s]

pose Sensor 3 experiences a fault. In this case, Filter 3 is uncorrupted by any faulty measurements, and its corresponding column in $\mathbf{T}_1 \in \mathbb{R}^{5 \times 5}$ uniquely contains all zeros. After determining Sensor 3 is the culprit via Equation (12), Sensor 3 is taken offline and a new set of $J_1 = I - 1 = 4$ filters, each excluding one of the remaining four sensors, is spawned. Without a J_2 layer, this would mean the system could not detect a subsequent fault while it repopulates the new $\mathbf{T}_1 \in \mathbb{R}^{4 \times 4}$. However, having the J_2 layer already running, we can see the new J_1 layer of filters is actually equivalent to the subset of J_2 filters that had

TABLE 4 False alarm rate summary for SAARM simulations, Scenario 1 ($I = 4$)

Desired $P_f \leq \alpha_{\max}$	Individual $\alpha = \alpha_{\max}/(I^2 - I)$	Actual P_f
1.0×10^{-3}	8.3×10^{-5}	6.0×10^{-4}
1.7×10^{-3}	1.4×10^{-4}	9.0×10^{-4}
2.8×10^{-3}	2.3×10^{-4}	1.3×10^{-3}
4.6×10^{-3}	3.9×10^{-4}	1.7×10^{-3}
7.7×10^{-3}	6.5×10^{-4}	2.5×10^{-3}
1.3×10^{-2}	1.1×10^{-3}	4.7×10^{-3}
2.2×10^{-2}	1.8×10^{-3}	8.9×10^{-3}
3.6×10^{-2}	3.0×10^{-3}	1.5×10^{-2}
6.0×10^{-2}	5.0×10^{-3}	2.6×10^{-2}
1.0×10^{-1}	8.3×10^{-3}	4.2×10^{-2}

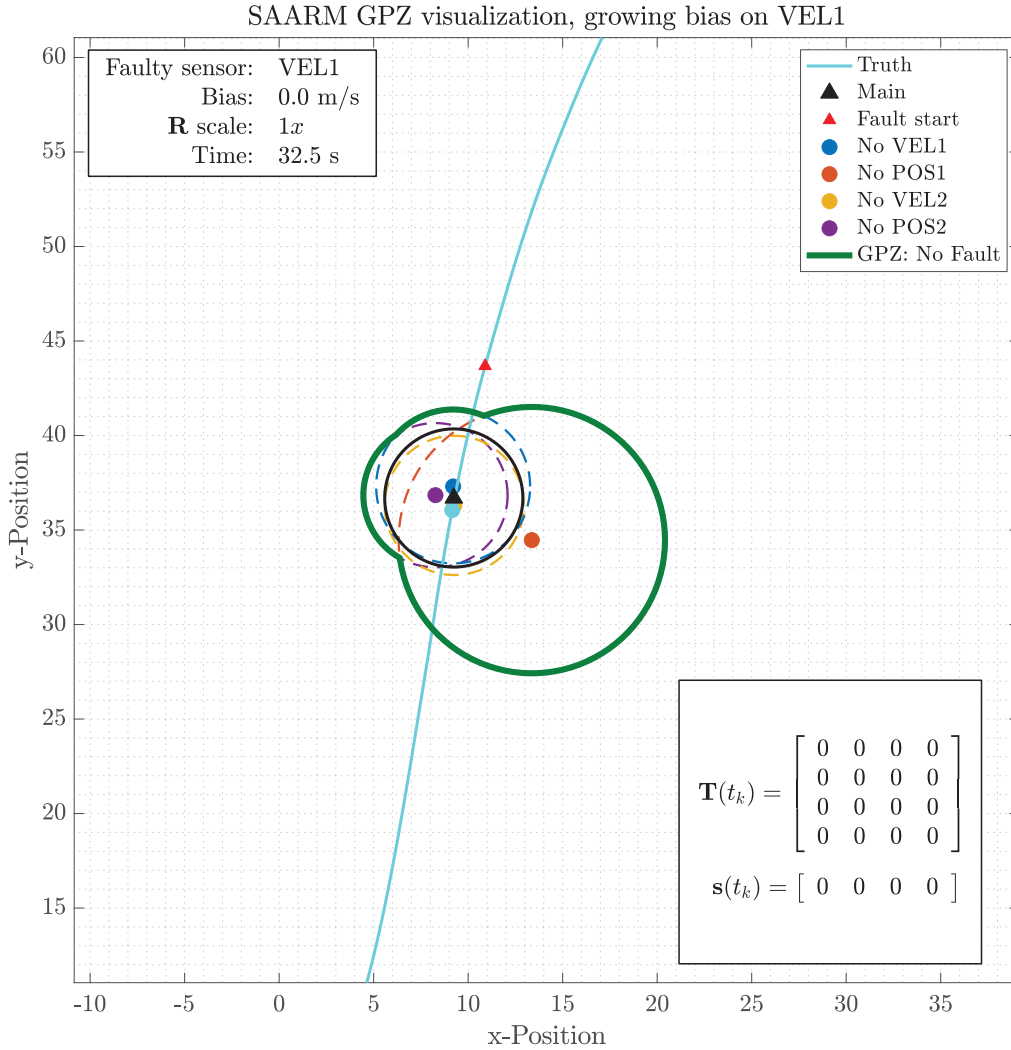


FIGURE 2 Example SAARM GPZ: No fault present. In this example, there is no fault induced into any of the four sensors, and no fault has been detected (all entries in \mathbf{s} are zero), which is shown to the user as a green GPZ. All filters are uncorrupted and the main filter is consistent. The GPZ is comprised of the union of the 95% position error ellipses from all filters and contains the true position at least 95% of the time [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

also excluded Sensor 3, which guarantees uninterrupted fault detection after detecting an initial fault. Finally, suppose both Sensor 3 and Sensor 5 experience a simultaneous fault. In this case, every single filter in the J_1 layer (i.e., every column in Table 1) would be corrupted and no column in \mathbf{T}_1 would contain all zeros. However, the J_2 filter that excluded both Sensor 3 and Sensor 5 would be guaranteed to be uncorrupted by faulty measurements, and its corresponding column in $\mathbf{T}_2 \in \mathbb{R}^{5 \times 10}$ would uniquely contain all zeros. In principle, this process can be scaled up to any number of layers, corresponding to any number of simultaneous faults. It is important, however, to consider the (previously mentioned) stochastic observability of each state as we exclude additional sensors and the trade-off in computational power required to support the

growing number of required filters for each additional layer.

3.4 | Performance assumptions and guarantees

Having defined an all-source fault detection, identification, and exclusion process, we now turn our attention to providing users with a measure of system performance guarantee without constraining the nature of the fault-present condition. Though FDE literature often provides a rigorous definition of system “Integrity,” here we aim to simply provide a guarantee of system performance under a particular set of assumptions. As already mentioned,

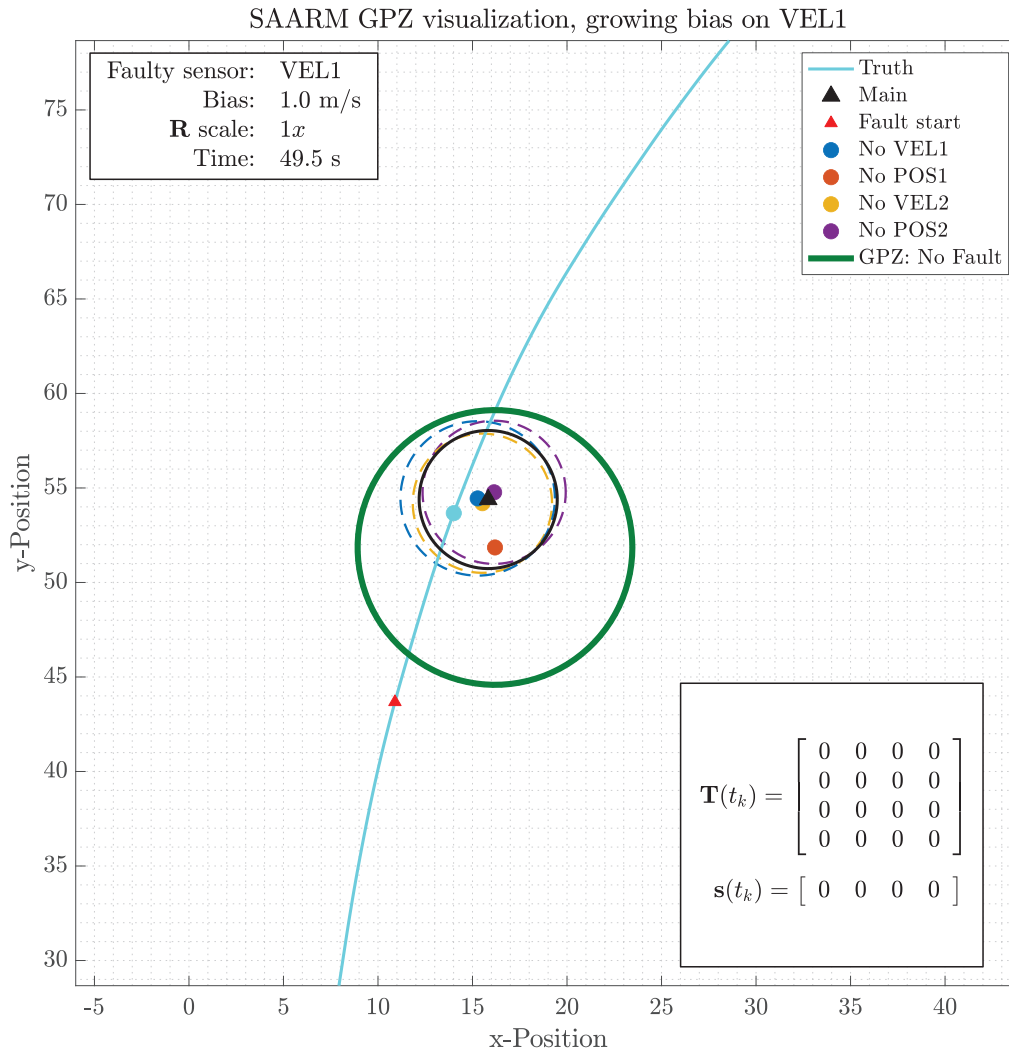


FIGURE 3 Example SAARM GPZ: Undetected fault. In this example, a 1.0 [m/s] bias is affecting the VEL1 sensor, but no fault has been detected yet (all entries in \mathbf{s} are still zero), which is shown to the user as a green GPZ. All filters except Filter 1 are corrupted and potentially inconsistent. The GPZ is comprised of the union of the 95% position error ellipses from all filters, and it is guaranteed to contain the true position at least 95% of the time since one of the filters is guaranteed to be uncorrupted [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

under our all-source fault-agnostic goal, we are unable to define the nature of the fault-present condition or the distribution of the fault-present test statistic, thereby precluding any computations involving the probability of missed detection or missed alert. However, given the multi-filter FDE mechanism already in place, we can still provide users with a GPZ, or a position error ellipsoid based on both the filter-computed position error statistics and the assumption that one of the filters in the system is guaranteed to be fault-free and therefore consistent.

In general, the GPZ is constructed under one condition: *assuming at least one of the filters is informed entirely by properly modeled, uncorrupted sensors, then at least one filter contains consistent state estimation error statistics.* In other words, since it is assumed that one of the filters is fault-free (based on properly designing the set of filters),

then the estimated error statistics from one of the filters truly describe actual errors committed by the filter. Defining α_I as the acceptable error bound, we can then derive an accurate $100(1 - \alpha_I)\%$ error ellipse on the horizontal position using the uncorrupted filter's horizontal position estimate and its associated error covariance matrix. Given the uncorrupted filter is not identifiable prior to determining a culprit, we union the $100(1 - \alpha_I)\%$ horizontal position error ellipses from all of the filters, thereby guaranteeing the true horizontal position is contained within the union with *at least* a $100(1 - \alpha_I)\%$ probability, since the union can only grow the resulting ellipse. This guarantee is valid regardless of the status of the underlying fault detection and identification process.

To clarify the difference between the GPZ and other ellipsoids found in integrity literature, we examine a few

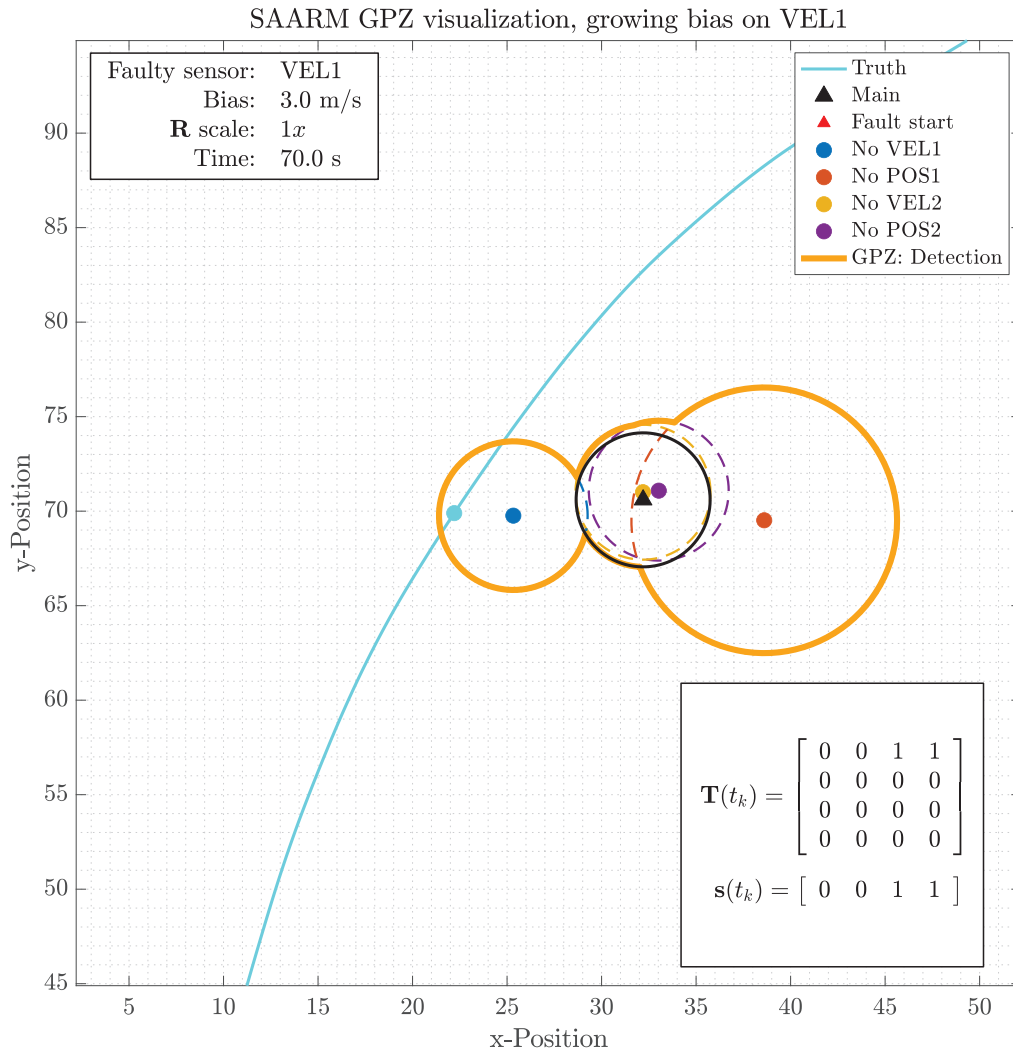


FIGURE 4 Example SAARM GPZ: Unidentified culprit. In this example, a 3.0 [m/s] bias is affecting the VEL1 sensor, and a fault has been detected (at least one entry in \mathbf{s} is non-zero), but no culprit has been identified (there is more than one zero entry in \mathbf{s}), which is shown to the user as an orange GPZ. All filters except Filter 1 (“No VEL1”) are corrupted and the main filter is clearly inconsistent. The GPZ is comprised of the union of the 95% position error ellipses from all filters, and it is guaranteed to contain the true position at least 95% of the time since one of the filters is guaranteed to be uncorrupted [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

important notes. The GPZ derivation begins with each subfilter’s estimate of position. For each subfilter, the acceptable error bound, α_I , represents the probability of the true vehicle position lying outside the error ellipsoid centered on the subfilter’s position estimate and derived from the subfilter’s position error covariance matrix. If an undetected fault is present in the system, only one subfilter is guaranteed to have consistent error statistics, meaning only one $100(1 - \alpha_I)\%$ error-bound ellipsoid is guaranteed to accurately describe the probability of containing the true vehicle position. Since the uncorrupted subfilter is not knowable prior to identifying a culprit, we cannot choose a single correct error-bound ellipsoid to display to the user. Therefore, we union all error-bound ellipsoids from all subfilters, thereby guaranteeing the

“correct one” is contained. Finally, since the correct one is guaranteed to contain the true vehicle position with a $100(1 - \alpha_I)\%$ probability, then unioning additional areas can only increase the probability of capturing the true vehicle position. As later shown, this union does not produce a contiguous shape centered about a mean position estimate, it is simply the union of all individual ellipsoids centered on their respective subfilter position estimate. Note the main filter position estimate and error statistics are not used in the computation of the GPZ.

To illustrate our GPZ, consider a 2D navigation problem using $I = 4$ sensors. The sensor suite is the same as used in the simulations in Section 4 and is summarized in Table 3. As shown, the sensor suite is composed of two 2D position sensors: POS1, POS2, and two 2D velocity sensors: VEL1,

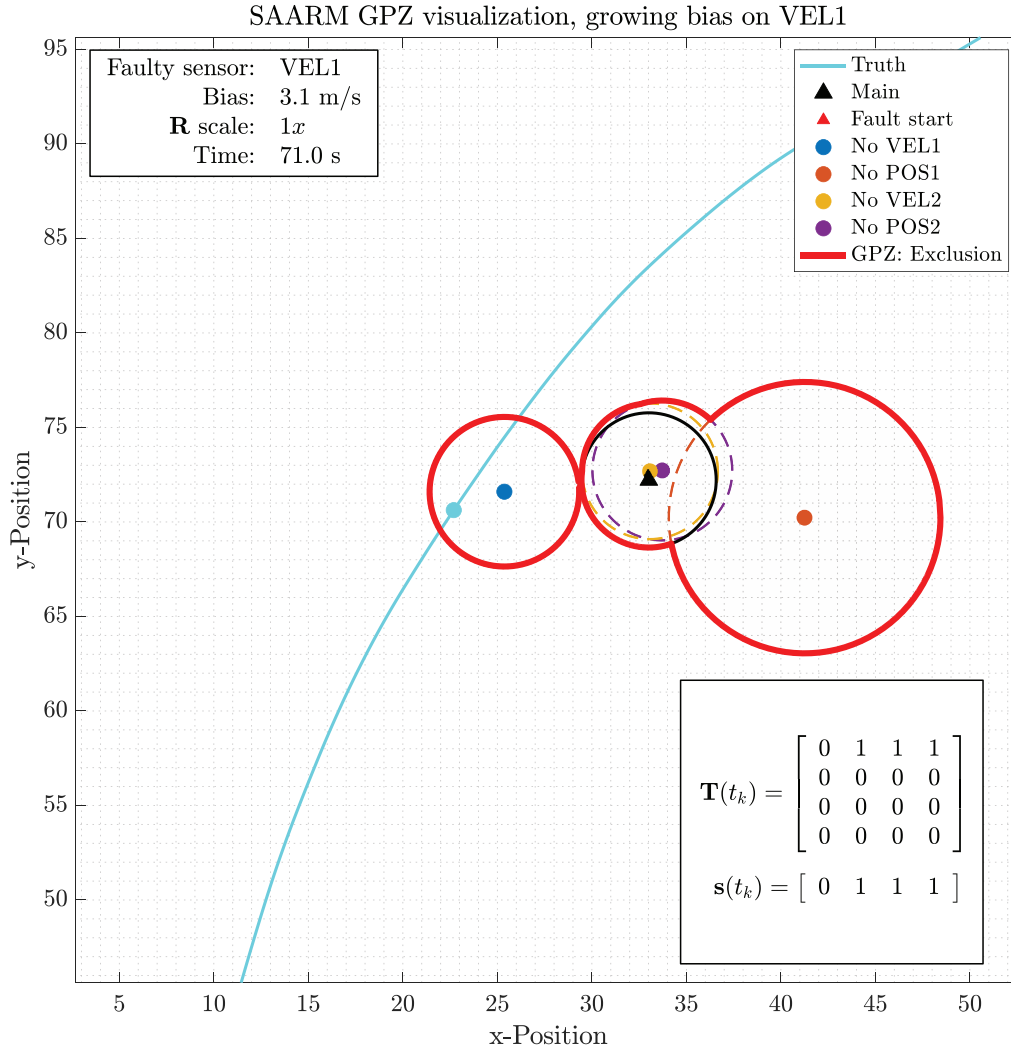


FIGURE 5 Example SAARM GPZ: Culprit identified. In this example, a 3.1 [m/s] bias is affecting the VEL1 sensor, a fault has been detected and the culprit has been identified (there is a single zero-entry in \mathbf{s}), which is shown to the user as a red GPZ. All filters except Filter 1 (“No VEL1”) are corrupted, and the main filter is clearly inconsistent. The GPZ is comprised of the union of the 95% position error ellipses from all filters, and it is guaranteed to contain the true position at least 95% of the time since one of the filters is guaranteed to be uncorrupted. Immediately after this time step, the VEL1 sensor is taken offline, and a new set of filters is re-spawned from Filter 1 [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

VEL2. The system dynamics are propagated using a 2D kinematics model driven by 2D First Order Gauss-Markov (FOGM) acceleration as described in Section 4. In order to best visualize the effects of faults on GPZ, the fault has been defined as a growing velocity bias starting from 0 [m/s] at $t_k = 40$ [s], growing at a rate of 0.1 [m/s/s], and applied to the x -dimension measurements from the VEL1 sensor. Figures 2 through 5 illustrate a time sequence of events along a sample instantiation of the simulation. As shown in the figures, an error bound of $\alpha_f = 0.05$ was used to produce 95% error ellipses. The GPZ derived from the union of the 95% horizontal position error ellipses from all subfilters is guaranteed to contain the true vehicle position at least 95% of the time, regardless of the presence of a fault, ability to detect, or ability to determine a culprit. In all examples, the

main filter is informed by all sensors but its states and their error covariances are not used in the detection of faults or the computation of GPZ. Finally, though these sample illustrations were limited to a single fault and two dimensions, the underlying axiom and assumptions are still valid for multiple faults, and the error ellipses can be scaled to 3D error ellipsoids, if desired.

4 | SIMULATION RESULTS

The proposed method was evaluated via a series of Monte Carlo simulations using two vehicles, each informed by the same set of all-source sensors described below and summarized in Table 3. The FDE function was provided by the

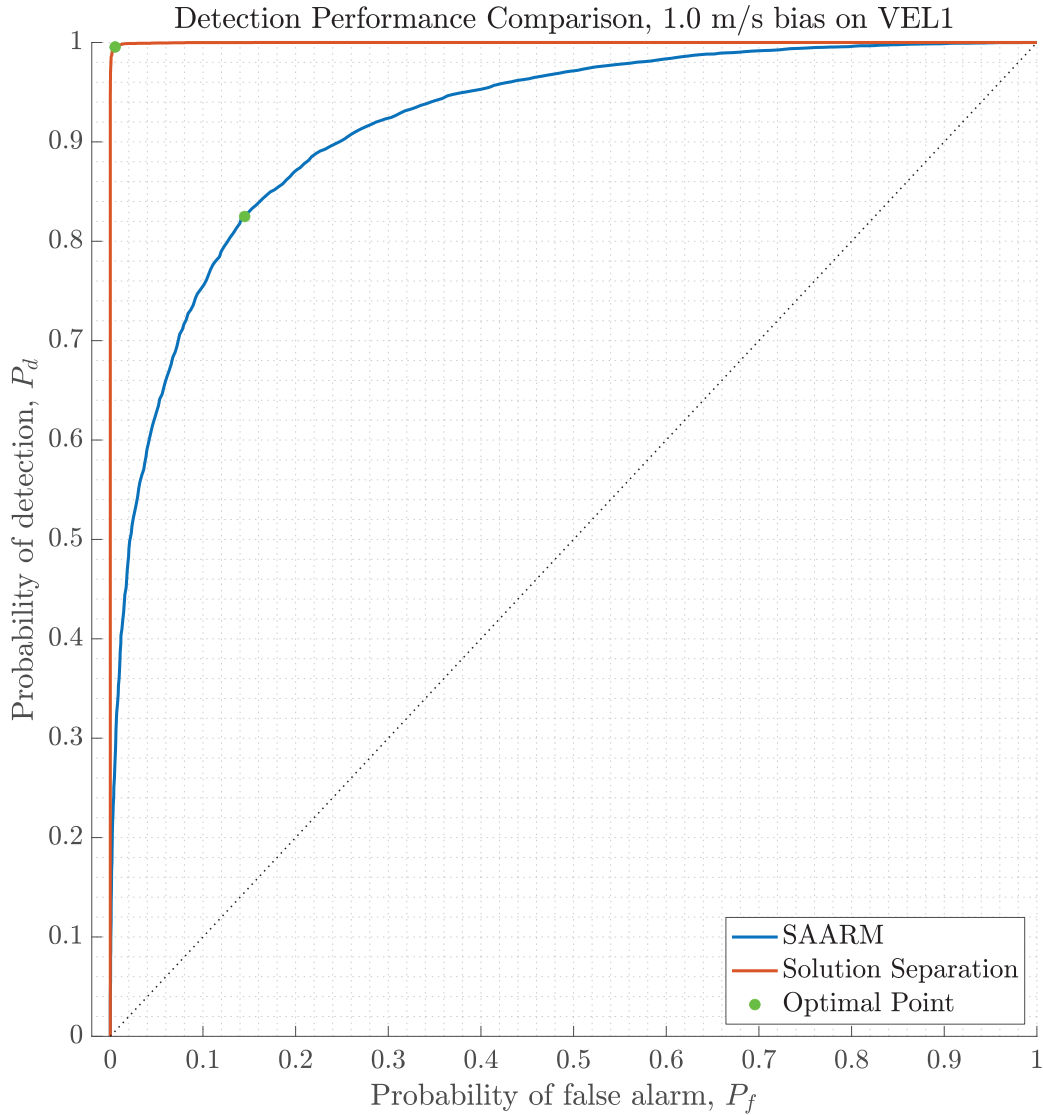


FIGURE 6 Detection performance comparison, SAARM vs. Sol. Sep., 1.0 [m/s] bias. Though both methods appear to provide useful detection performance, the solution separation detector outperforms SAARM by approximately 17% at the optimal Neyman-Pearson detection point [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

proposed SAARM algorithm in the first vehicle while the normalized horizontal position solution separation test statistic as implemented in Young and McGraw (2003) was used for the second vehicle. For all simulations, the true system dynamics were driven by a 2D kinematic model given by

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{x}}_p(t) \\ \dot{\mathbf{x}}_v(t) \\ \dot{\mathbf{x}}_a(t) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_v(t) \\ \mathbf{x}_a(t) \\ -\frac{1}{\tau_a} \mathbf{x}_a(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{w}(t) \end{bmatrix}, \quad (16)$$

where \mathbf{x}_p is the vehicle's 2D position in [m], \mathbf{x}_v is the 2D velocity in [m/s], \mathbf{x}_a is the 2D acceleration in [m/s²] and propagated by a FOGM process with time

constant $\tau_a = 10$ [s], and variance $\sigma_a = 0.01^2$ [m²/s⁴], making $\mathbf{w}(t)$ a 2D white Gaussian noise process with $E[\mathbf{w}(t)\mathbf{w}(t + \tau)^T] = \mathbf{Q}\delta(\tau)$ and

$$\mathbf{Q} = (1.5 \times 10^{-3})^2 \mathbf{I}_{2 \times 2} \text{ [m}^2/\text{s}^5]. \quad (17)$$

Each vehicle in the simulation was informed by two velocity sensors and two position sensors for a total of four ($I = 4$) sensors. Sensor 1 ("VEL1") measurements were modeled as

$$\mathbf{z}^{[1]}(t_k) = \mathbf{x}_v(t_k) + \mathbf{v}_k^{[1]}, \quad (18)$$

$$E[\mathbf{v}_k^{[1]}\mathbf{v}_k^{[1]T}] = \mathbf{R}^{[1]}(t_k) = \begin{bmatrix} 1^2 & 0 \\ 0 & 1^2 \end{bmatrix} \text{ [m}^2/\text{s}^2], \quad (19)$$

Data Distribution: SAARM T Matrix, 1.0 m/s bias on VEL1

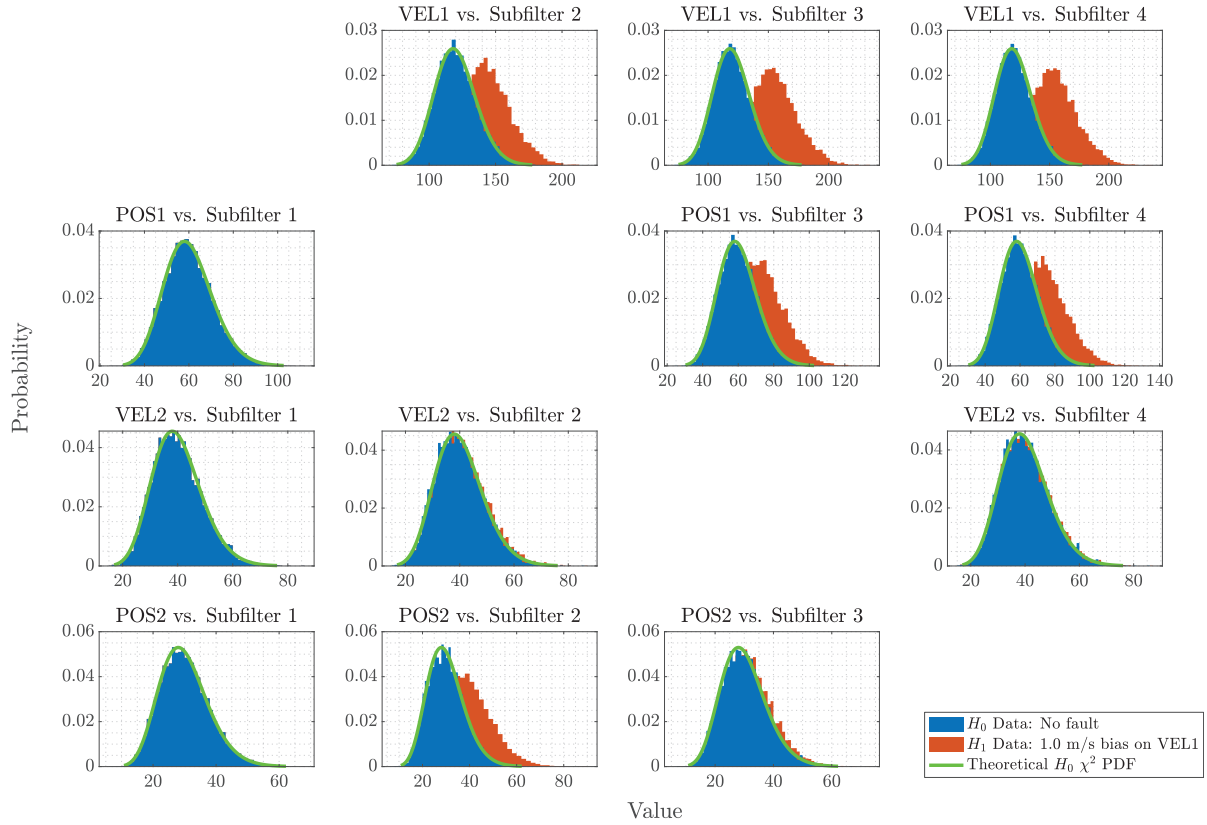


FIGURE 7 Chi-Square test statistic distributions in the SAARM T matrix. The expected H_0 degrees of freedom are derived using Equation (9) and are based on the sensor update rate, measurement dimension, and the SAARM monitoring time window. In this case, the residual test statistic formed between VEL1 and subfilter 3 (“No VEL2”) provided the best detection performance [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

and its update rate was set to 0.5 [s] or 2 [Hz]. Sensor 2 (“POS1”) measurements were modeled as

$$\mathbf{z}^{[2]}(t_k) = \mathbf{x}_p(t_k) + \mathbf{v}_k^{[2]}, \quad (20)$$

$$E \left[\mathbf{v}_k^{[2]} \mathbf{v}_k^{[2]T} \right] = \mathbf{R}^{[2]}(t_k) = \begin{bmatrix} 5^2 & 0 \\ 0 & 5^2 \end{bmatrix} [\text{m}^2], \quad (21)$$

and its update rate was set to 1.0 [s] or 1 [Hz]. Sensor 3 (“VEL2”) measurements were modeled as

$$\mathbf{z}^{[3]}(t_k) = \mathbf{x}_v(t_k) + \mathbf{v}_k^{[3]}, \quad (22)$$

$$E \left[\mathbf{v}_k^{[3]} \mathbf{v}_k^{[3]T} \right] = \mathbf{R}^{[3]}(t_k) = \begin{bmatrix} 2^2 & 0 \\ 0 & 2^2 \end{bmatrix} [\text{m}^2/\text{s}^2], \quad (23)$$

and its update rate was set to 1.5 [s] or 0.67 [Hz], and finally, Sensor 4 (“POS2”) measurements were modeled as

$$\mathbf{z}^{[4]}(t_k) = \mathbf{x}_p(t_k) + \mathbf{v}_k^{[4]}, \quad (24)$$

$$E \left[\mathbf{v}_k^{[4]} \mathbf{v}_k^{[4]T} \right] = \mathbf{R}^{[4]}(t_k) = \begin{bmatrix} 10^2 & 0 \\ 0 & 10^2 \end{bmatrix} [\text{m}^2], \quad (25)$$

and its update rate was set to 2.0 [s] or 0.5 [Hz]. It is important to note here the different update rates among the sensors will lead to a different number of samples (M) and Chi-Square degrees of freedom captured for each sensor within the fixed SAARM monitoring time window discussed below.

The fault-free (H_0) condition was characterized using 10,000 Monte Carlo trials where no faults were introduced into the system and all measurements were drawn from their modeled distributions. Next, two fault-present (H_1) scenarios were simulated, each using additional 10,000 Monte Carlo trials. The two fault-present scenarios are summarized in Table 3. In the first scenario, measurements from Sensor 1 (“VEL1”) were corrupted with an unmodeled constant 1.0 [m/s] x -velocity bias, starting at $t_k = 40$ [s]. In the second scenario, the measurements from Sensor 1 (“VEL1”) sensor were corrupted by scaling $\mathbf{R}^{[1]}$ by a factor of 2 \times , again starting at $t_k = 40$ [s], and

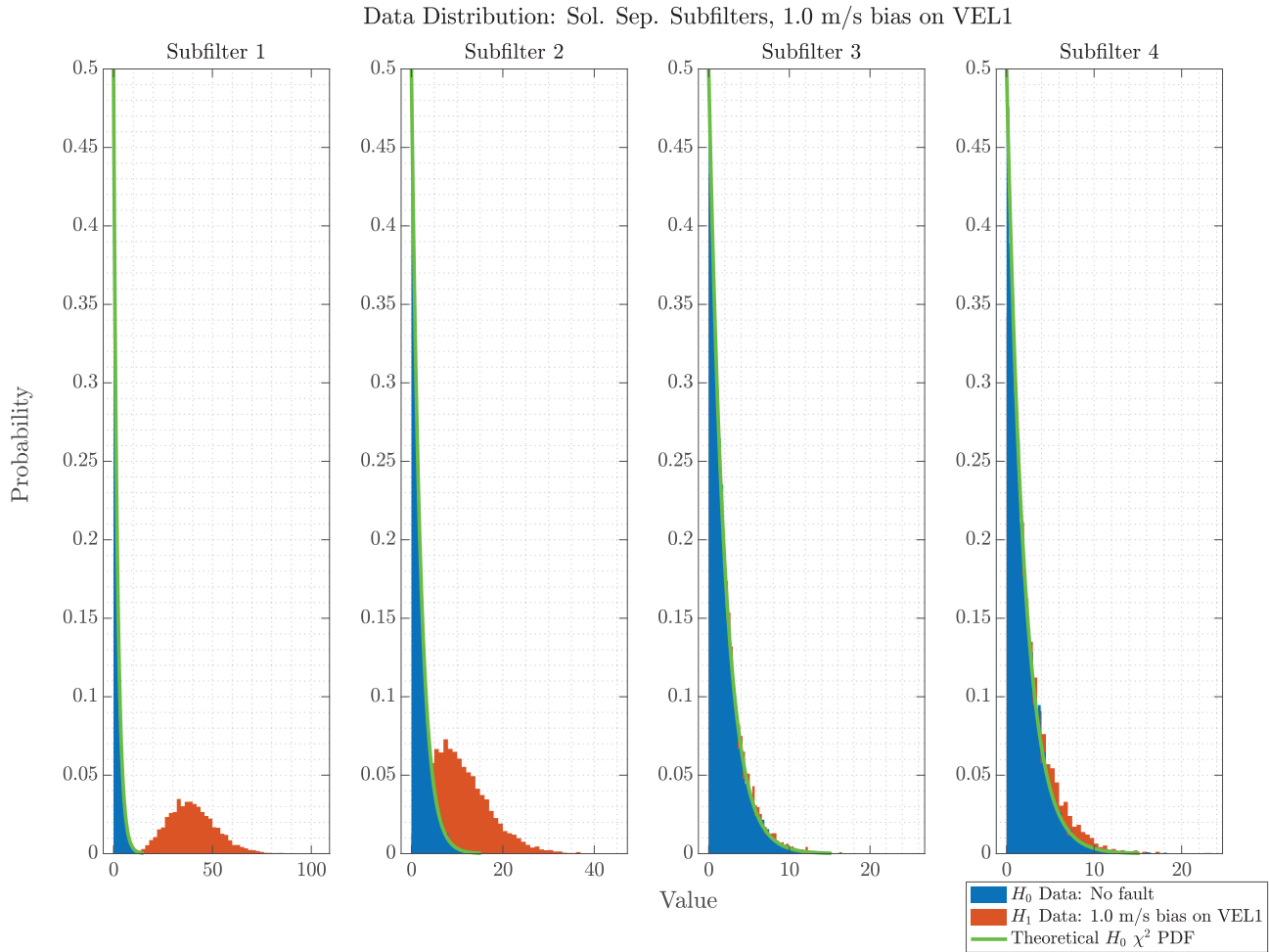


FIGURE 8 Chi-Square solution separation test statistic distributions for each of the required subfilters. The expected H_0 degrees of freedom (i.e., 2) are based on testing the 2D horizontal position solution separation vector. In this case, subfilter 1 (“No VEL1”) provided the best detection performance [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

without specifying the change in the filter measurement model.

For each trial, the initial state estimation error covariance matrix was set to

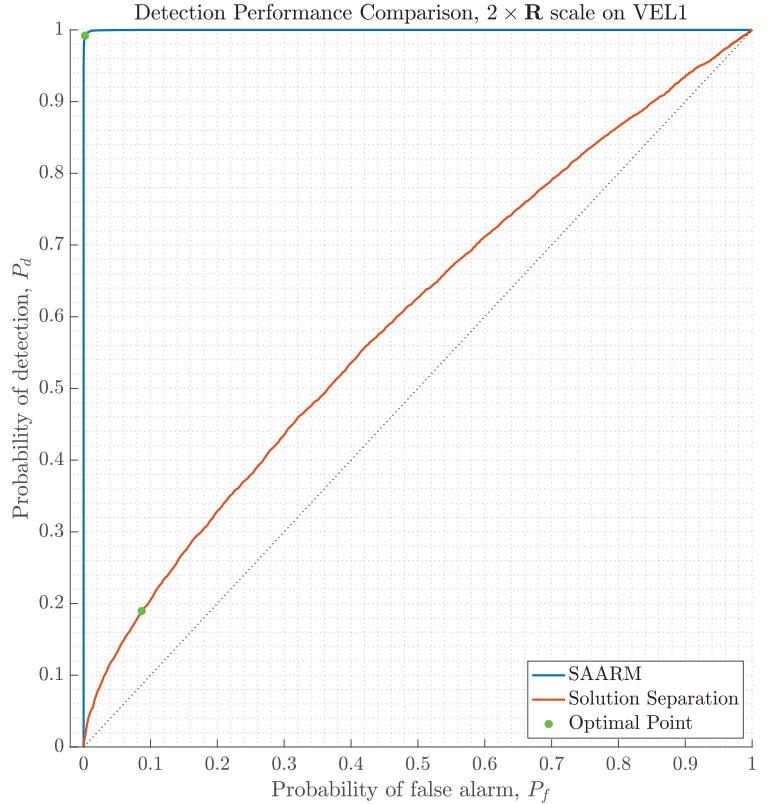
$$\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}(t_0) = \text{diag}([1 \ 1 \ 1 \ 1 \ 0.01^2 \ 0.01^2]), \quad (26)$$

while the initial state estimate, $\hat{\mathbf{x}}(t_0)$, was set to zeros, and the true initial state was drawn from a $\mathcal{N}(\mathbf{0}, \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}(t_0))$ distribution. Each trial was propagated using $\Delta t_k = 0.5$ [s], starting at $t_k = 0$ [s]. The SAARM monitoring period was set to 30 [s], yielding $M = 60$ for Sensor 1, $M = 30$ for Sensor 2, $M = 20$ for Sensor 3, and $M = 15$ for Sensor 4 in Equation (9). All trials were terminated at $t_k = 70$ [s], at which point both the SAARM and solution separation tests statistics were recorded. Here, it is important to note that at the time each trial was terminated, the associated SAARM test matrix, \mathbf{T} , was based on the sum of residual-based Chi-Square samples over the trailing 30 [s] (i.e., the monitoring period), while the normalized solution separation test

statistic was based on the single solution-separation vector associated with the final measurement update prior to termination.

Figure 6 compares the detection performance between the SAARM and normalized solution separation test statistics for Scenario 1 (1.0 [m/s] bias). Meanwhile, Figures 7 and 8 illustrate the test statistic distributions for the H_0 and H_1 conditions in the SAARM \mathbf{T} matrix and each solution separation subfilter, respectively. As shown, in the case of a bias-type fault, both SAARM and normalized solution separation provide useful detection performance, with solution separation outperforming SAARM by approximately 17% at the optimal Neyman-Pearson (Kay, 1993) detection point. The results from Figure 7 indicate SAARM detection performance was mostly derived from the VEL1 vs. subfilter 3 (“No VEL2”) residual test statistic. Meanwhile, the results from Figure 8 indicate solution separation detection performance was mostly derived from the subfilter 1 (“No VEL1”) solution separation test statistic.

FIGURE 9 Detection performance comparison, SAARM vs. Sol. Sep., $2 \times \mathbf{R}$ scale. The solution separation detector performs marginally and is outperformed by SAARM by approximately 81% at the optimal Neyman-Pearson detection point [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]



Data Distribution: SAARM T Matrix, $2 \times \mathbf{R}$ scale on VEL1

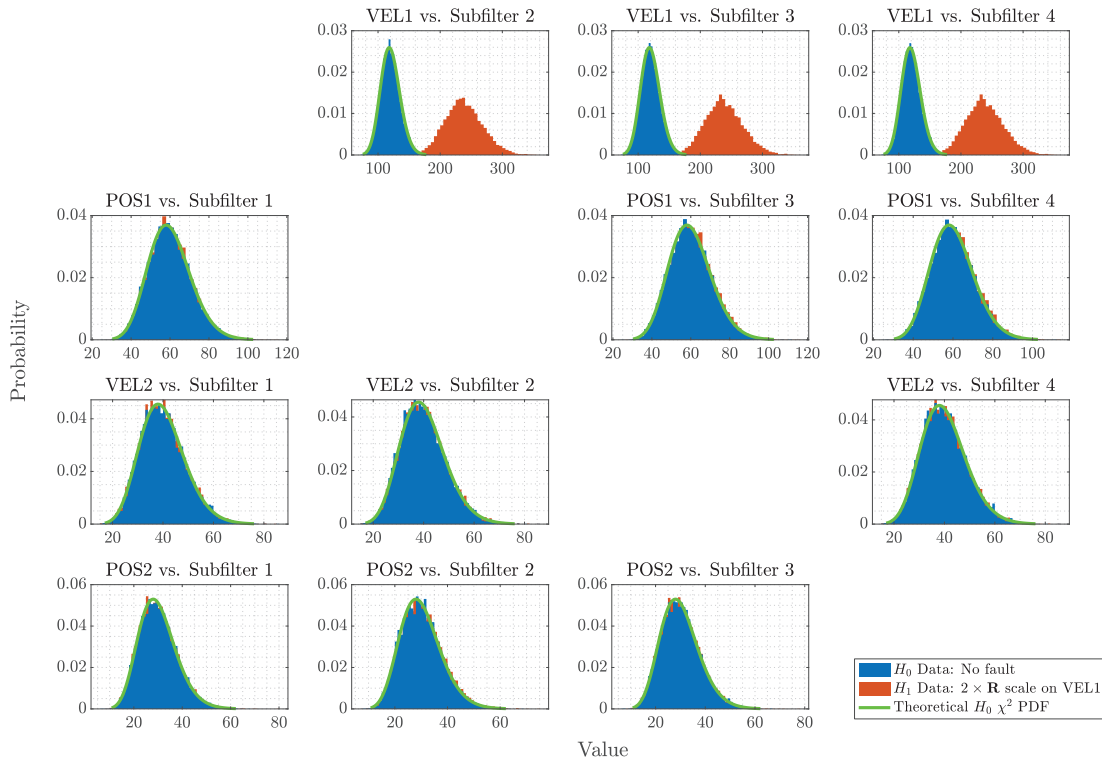


FIGURE 10 Chi-Square test statistic distributions in the SAARM T matrix. The expected H_0 degrees of freedom are derived using Equation (9) and are based on the sensor update rate, measurement dimension, and the SAARM monitoring time window. In this case, the residual test statistic formed between VEL1 and subfilter 3 (“No VEL2”) provided the only detection performance, which was marginal [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

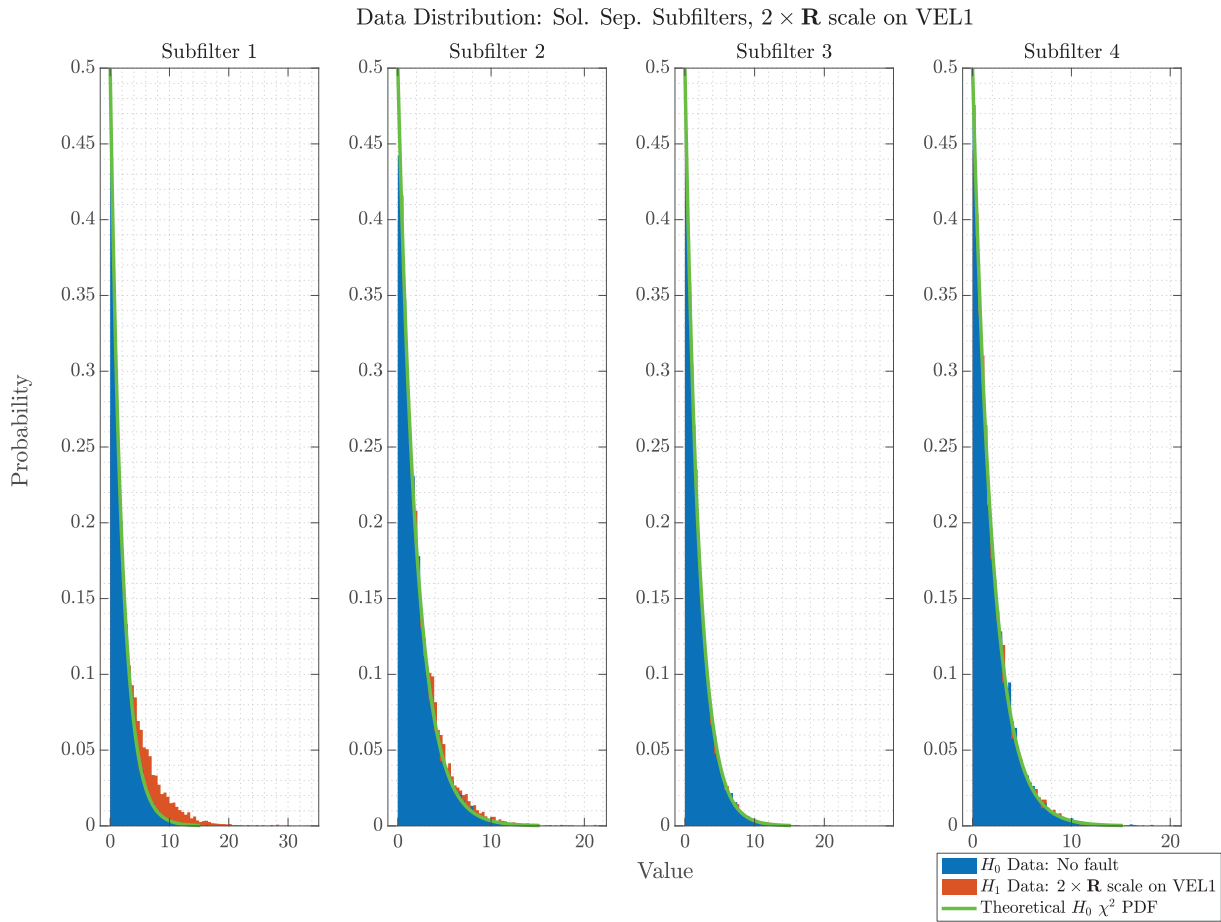


FIGURE 11 Chi-Square solution separation test statistic distributions for each of the required subfilters. The expected H_0 degrees of freedom (i.e., 2) are based on testing the 2D horizontal position solution separation vector. In this case, subfilter 1 (“No VEL1”) provided the best detection performance [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

Figure 9 compares the detection performance between the SAARM and normalized solution separation test statistics for Scenario 2 ($2 \times \mathbf{R}$ scale). Similarly, Figures 10 and 11 illustrate the test statistic distributions for the H_0 and H_1 conditions in the SAARM \mathbf{T} matrix and each solution separation subfilter, respectively. In the case of a wrongly stated covariance matrix fault type, both SAARM and normalized solution separation provide some detection performance; however, in this case, the solution separation detection performance was marginal, with SAARM outperforming it by approximately 81% at the optimal detection point. Again, the results from Figure 10 indicate SAARM detection performance was mostly derived from the VEL1 vs. subfilter 3 (“No VEL2”) residual test statistic, while the results from Figure 8 indicate solution separation detection performance was mostly derived from the subfilter 1 (“No VEL1”) solution separation test statistic.

Table 5 summarizes the performance of the proposed GPZ error bound. Each row of Table 5 is based on 10,000 Monte Carlo trials and displays the percentage of time the true horizontal position was contained within the 95%

TABLE 5 Horizontal position error consistency, 95% error bounds

Scenario	Percentage of samples within error bound	
	Main filter covariance	SAARM GPZ
Fault-free scenario	95.19%	99.72%
Scenario 1: 1 [m/s] bias	78.63%	98.97%
Scenario 2: 2x \mathbf{R} scale	94.04%	99.60%

error bound derived from either the main filter position error covariance matrix or the SAARM GPZ. As shown, the 95% error bound derived from the main filter position error covariance matrix underestimated the true error in both fault-present scenarios. Meanwhile, the 95% error bound derived from the SAARM GPZ successfully bounded the true error *at least* 95% of the time, as expected.

The results from the above scenarios indicate fault types that systematically corrupt the state-space estimate (namely the position estimate) of the subfilters that are

more easily detected via solution separation methods than via sensor-specific (velocity in this case) residual test statistics. In contrast, fault types that do not systematically affect the subfilter solutions are not easily detected via solution separation methods, while sensor-specific residual test statistics provide excellent detection performance. It is also important to note that based on these two cases, the SAARM approach generally provides the best detection performance if the fault type is unknown or not constrained to a bias, which exactly meets the original research objective laid out in Section 1. Additionally, though not shown here for brevity, a third set of Monte-Carlo trials with a larger velocity bias (2.0 [m/s]) led to identical detection performance between SAARM and solution separation (i.e., when the bias is large enough, both detectors perform equally). The motivation for using the proposed method is also highlighted by the ability to provide performance guarantees without a fault-present model via the proposed GPZ.

5 | CONCLUSIONS

This research has proposed a novel method for fault detection and exclusion in all-source navigation systems. The proposed method, referred to as Sensor-Agnostic All-source Residual Monitoring, was shown to provide reliable FDE in cases where the fault type is unknown or unconstrained, such as when using emerging all-source sensors with measurement models that are not well understood. When compared to traditional filtered approaches such as normalized solution separation, the proposed method was shown to be approximately 17% less sensitive to detecting bias-type faults, but outperformed such methods by approximately 81% when the fault was defined by changes in covariance. Finally, the mechanization of the proposed method lended itself to a practical and robust guarantee of performance for the user in the form of a so-called Guaranteed Position Zone. The proposed GPZ ellipsoid error bound was shown to contain the true vehicle position within the specified error rate during all phases of the FDE process and without the need to constrain the fault-present distribution or probability of missed detection. This research directly enables self-correcting plug-and-play open architecture navigation systems as well as APNT in the challenging application of all-source multi-domain navigation. Future work in this area will focus on experimental results using real-world data, further modeling and simulation in the case of multiple simultaneous faults, and research into how to compare the GPZ ellipsoid to a predetermined alert limit in order to warn users of degraded performance.

DISCLAIMER

The views expressed in this paper are those of the authors, and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

ORCID

Juan Jurado  <https://orcid.org/0000-0003-3600-6239>

REFERENCES

- Bageshwar, V. L., Gebre-Egziabher, D., Garrard, W. L., & Georgiou, T. T. (2009). Stochastic observability test for discrete-time kalman filters. *Journal of Guidance, Control, and Dynamics*, 32(4), 1356–1370. <https://doi.org/10.2514/1.38128>
- Bhatti, U. I. (2006). An improved sensor level integrity algorithm for GPS/INS integrated system. In *Proceeding of ION GNSS 19th International Technical Meeting*, pp. 3012–3023. Retrieved from <https://www.ion.org/publications/abstract.cfm?articleID=7063>
- Bhatti, U. I., Ochieng, W. Y., & Feng, S. (2007a). Integrity of an integrated GPS/INS system in the presence of slowly growing errors. Part I: A critical review. *GPS Solutions*, 11(3), 173–181. <https://doi.org/10.1007/s10291-006-0048-2>
- Bhatti, U. I., Ochieng, W. Y., & Feng, S. (2007b). Integrity of an integrated GPS/INS system in the presence of slowly growing errors. Part II: Analysis. *GPS Solutions*, 11(3), 183–192. <https://doi.org/10.1007/s10291-006-0049-1>
- Bonferroni, C. E. (1936). *Teoria statistica delle classi e calcolo delle probabilita*. Libreria internazionale Seeber.
- Brenner, M. (1990). Implementation of a RAIM monitor in a GPS receiver and an integrated GPS/IRS. In *Proceedings of the 3rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1990)*, pp. 397–406. Retrieved from <https://www.ion.org/publications/abstract.cfm?articleID=5044>
- Brenner, M. (1996). Integrated GPS/inertial fault detection availability. *NAVIGATION*, 43(2), 111–130. <https://doi.org/10.1002/j.2161-4296.1996.tb01920.x>
- Brumback, B., & Srinath, M. (1987). A chi-square test for fault-detection in kalman filters. *IEEE Transactions on Automatic Control*, 32(6), 552–554. <https://doi.org/10.1109/TAC.1987.1104658>
- Call, C., Ibis, M., McDonald, J., & Vanderwerf, K. (2006). Performance of Honeywell's Inertial/GPS Hybrid (HIGH) for RNP operations. In *2006 IEEE/ION Position, Location, And Navigation Symposium*, p. 244. <https://doi.org/10.1109/PLANS.2006.1650610>
- Canciani, A., & Raquet, J. (2016). Absolute positioning using the earth's magnetic anomaly field. *NAVIGATION*, 63(2), 111–126. <https://doi.org/10.1002/navi.138>
- Casella, G., & Berger, R. L. (2002). *Statistical inference* (2nd ed.). Pacific Grove, CA: Duxbury.
- Curro, J., & Raquet, J. (2016). Navigation using VLF environmental features. In *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 373–379. <https://doi.org/10.1109/PLANS.2016.7479723>
- De Maesschalck, R., Jouan-Rimbaud, D., & Massart, D. L. (2000). The mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1), 1–18. [https://doi.org/10.1016/S0169-7439\(99\)00047-7](https://doi.org/10.1016/S0169-7439(99)00047-7)

- Goeman, J. J., & Solari, A. (2014). Multiple hypothesis testing in genomics. *Statistics in Medicine*, 33(11), 1946–1978. <https://doi.org/10.1002/sim.6082>
- Grejner-Brzezinska, D. A., Toth, C. K., Moore, T., Raquet, J. F., Miller, M. M., & Kealy, A. (2016). Multisensor navigation systems: A remedy for GNSS vulnerabilities? *Proceedings of the IEEE*, 104(6), 1339–1353. <https://doi.org/10.1109/JPROC.2016.2528538>
- Joerger, M., Chan, F.-C., & Pervan, B. (2014). Solution separation versus residual-based raim. *NAVIGATION*, 61(4), 273–291. <https://doi.org/10.1002/navi.71>
- Jurado, J., Raquet, J., & Schubert Kabban, C. M. (2019). Autonomous and resilient management of all-source sensors for navigation. In *Proceedings of the ION 2019 Pacific PNT Meeting*, pp. 142–159. <https://doi.org/10.33012/2019.16800>
- Jurado, J. D., Raquet, J. F., & Schubert Kabban, C. M. (2020). Single-filter finite fault detection and exclusion methodology for real-time validation of plug-and-play sensors. *IEEE Transactions on Aerospace and Electronic Systems*. Advance online publication. <https://doi.org/10.1109/TAES.2020.3010394>
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45. Retrieved from <https://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf>
- Kay, S. M. (1993). *Fundamentals of statistical signal processing, volume I: Estimation theory*. Upper Saddle River, NJ: Prentice Hall.
- Kay, S. M. (1998). *Fundamentals of statistical signal processing, volume II: Detection theory*. Upper Saddle River, NJ: Prentice Hall.
- Kerr, T. (1980). Statistical analysis of a two-ellipsoid overlap test for real-time failure detection. *IEEE Transactions on Automatic Control*, 25(4), 762–773. <https://doi.org/10.1109/TAC.1980.1102423>
- Lee, Y. C., et al. (1986). Analysis of range and position comparison methods as a means to provide GPS integrity in the user receiver. In *Proceedings of the 42nd Annual Meeting of the Institute of Navigation*, pp. 1–4. Retrieved from <https://www.ion.org/publications/abstract.cfm?articleID=12197>
- Maybeck, P. S. (1982). *Stochastic models, estimation, and control: Volume 1*. New York, NY: Academic Press.
- Maybeck, P. S. (1984). *Stochastic models, estimation, and control: Volume 2*. New York, NY: Academic Press.
- Parkinson, B. W., & Axelrad, P. (1988). Autonomous GPS integrity monitoring using the pseudorange residual. *NAVIGATION*, 35(2), 255–274. <https://doi.org/10.1002/j.2161-4296.1988.tb00955.x>
- 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>
- van Graas, F., & Farrell, J. L. (1993). Baseline fault detection and exclusion algorithm. In *Proceedings of the 49th Annual Meeting of The Institute of Navigation*, pp. 413–420. Retrieved from <https://www.ion.org/publications/abstract.cfm?articleID=4504>
- Van Loan, C. F. (1978). Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3), 395–404. <https://doi.org/10.1109/TAC.1978.1101743>
- Venable, D. T. (2016). *Improving real world performance of vision aided navigation in a flight environment* (PhD dissertation). Air Force Institute of Technology WPAFB.
- Wan, E. A., & Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pp. 153–158. <https://doi.org/10.1109/ASSPCC.2000.882463>
- Young, R. S., & McGraw, G. A. (2003). Fault detection and exclusion using normalized solution separation and residual monitoring methods. *NAVIGATION*, 50(3), 151–169. <https://doi.org/10.1002/j.2161-4296.2003.tb00326.x>

How to cite this article: Jurado J, Raquet J, Schubert Kabban CM, Gipson J. Residual-based multi-filter methodology for all-source fault detection, exclusion, and performance monitoring. *NAVIGATION*. 2020;67: 493–509. <https://doi.org/10.1002/navi.384>