Winter 2021

Resilience for Multi-filter All-source Navigation Framework with Integrity

Jonathon S. Gipson
Air Force Institute of Technology

Robert C. Leishman
Air Force Institute of Technology

Recommended Citation

Gipson, Jonathon S. and Leishman, Robert C., "Resilience for Multi-filter All-source Navigation Framework with Integrity" (2021). *Faculty Publications*. 771.
https://scholar.afit.edu/facpub/771

Resilience for Multi-filter All-source Navigation Framework with Integrity

Jonathon S. Gipson*[1] and Robert C. Leishman†[2]

[1] *Autonomy and Navigation Technology Center* Wright-Patterson AFB, OH
[2] *Director, Autonomy and Navigation Technology Center U.S. Air Force Institute of Technology*, Wright-Patterson AFB, OH

May 11, 2021

Abstract

The Autonomous and Resilient Management of All-source Sensors (ARMAS) framework monitors residual-space test statistics across unique sensor-exclusion banks of filters, (known as *subfilters*) to provide a resilient, fault-resistant all-source navigation architecture with assurance. A critical assumption of this architecture, demonstrated in this paper, is fully overlapping state observability across all subfilters. All-source sensors, particularly those that only provide partial state information (altimeters, TDoA, AOB, etc.) do not intrinsically meet this requirement.

This paper presents a novel method to monitor real-time overlapping position state observability and introduces an "observability bank" within the ARMAS framework, known as Stable Observability Monitoring (SOM). SOM uses real-time stability analysis to provide an intrinsic awareness to ARMAS of the capabilities of the fault detection and exclusion (FDE) functionality. We define the ability to maintain consistent all-source FDE to recover failed sensors as *navigation resilience.* A resilient FDE capability then is one that is "aware" of when it requires more sensor information to protect the consistency of the FDE and integrity functions from corruption. SOM is the first demonstration of such a system, for all-source sensors, that the authors are aware.

A multi-agent 3D environment simulating both GNSS and position and velocity alternative navigation sensors was created and individual GNSS pseudorange sensor anomalies are utilized to demonstrate the capabilities of the novel algorithm. This paper demonstrates that SOM seamlessly integrates within the ARMAS framework, provides timely prompts to augment with new sensor information from other agents, and indicates when framework stability and preservation of all-source navigation integrity are achieved.

*jonathon.gipson@afit.edu
†robert.leishman@afit.edu

1 Introduction

Introduced in 2018, ARMAS provided a generalized framework for real-time management of heterogeneous, asynchronous all-source sensors [9]. This framework was resilient to corruption from mismodeled, uncalibrated, and faulty sensors and was accomplished by combining sensor validation, FDE, recalibration, and remodeling modes in a single architecture. Sensor-Agnostic All-source Residual Monitoring (SAARM) was designed to provide all-source Fault Detection and Exclusion (FDE) and navigation integrity functions within the ARMAS framework [11]. In this context, all-source navigation resiliency is the ability to maintain consistent all-source FDE operations with the ability to recover failed sensors. The pluggable Bayesian filters provided by the SCORPION estimation architecture afforded needed flexibility to spawn, propagate, and remove estimation filters on the fly [12]. SAARM required the designer to maintain a set of unique navigation subfilters (each unique subfilter excludes measurements from a different sensor) to maintain resilience to a single simultaneous sensor failure. A primary assumption of the FDE and integrity functions was the ability to maintain overlapping position state observability. This paper presents a novel method to monitor real-time overlapping position state observability and introduces an "observability bank" within the ARMAS framework. These additions to ARMAS use real-time observability analysis at the layer 2 subfilter level (each unique layer 2 subfilter excludes measurements from two different sensors) to provide a timely indication to augment the framework with new sensor data, thus protecting the consistency of the ARMAS FDE and integrity functions from corruption.

2 Background

2.1 Autonomous Resilient Management of All-source Sensors (ARMAS)

The ARMAS framework was designed to gracefully recover from multiple types of failure modes (bias, model mismatched, and/or sensor miscalibration) while attempting to maintain a consistent, uncorrupted navigation estimate. ARMAS employs a set of SCORPION pluggable EKF estimators to address the following nonlinear navigation problem:

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

where $\mathbf{x}$ is a $N \times 1$ state vector of a vehicle's position, velocity, and attitude. The measurement error states vector α is of dimension $M \times 1$, $\mathbf{u}$ is the control input vector, $\mathbf{G}$ is an $(N + M) \times W$ linear operator, and $\mathbf{w}$ is a $W \times 1$ white noise process defined by a $W \times W$ continuous process noise strength matrix, $\mathbf{Q}$.

The state estimates are propagated through optimally combining the state process model, sensor-specific calibration parameters, and measurement updates from j = $1...J$ available all-source sensors. The measurement model for the j^{th} sensor is described by:

$$\mathbf{z}_k^{[j]} = \mathbf{h}^{[j]}\left[\mathbf{x}(t), \alpha^{[j]}(t), \mathbf{u}(t), t, \mathbf{p}^{[j]}\right] + \boldsymbol{v}_k^{[j]}, \tag{2}$$

where $\mathbf{h}^{[j]}$ is the nonlinear measurement function for the j^{th} sensor, $\alpha^{\mathbf{[j]}}$ is an $L \times 1$ subset of α which contains additional error states needed to process sensor measurements, $\mathbf{p^{[j]}}$ is a $P \times 1$ user-selectable model parameter vector for $\mathbf{h}^{[j]}$, and $\mathbf{v_k}$ is a $Z \times 1$ discrete white noise process with covariance defined by matrix $\mathbf{R_k^{[j]}}$.

The $Z \times 1$ measurement residual for sensor j, $\mathbf{r_k^{[j]}}$ is defined by

$$\mathbf{r}_k^{[j]} = \mathbf{z}_k^{[j]} - \mathbf{h}^{[j]}\left[\hat{\mathbf{x}}_k^-, \hat{\alpha}_k^{[j]-}, \mathbf{u}_k, t_k, \hat{\mathbf{p}}_k^{[j]}\right], \tag{3}$$

where $\hat{\mathbf{x}}_k^-$, $\hat{\alpha}_k^{[j]-}$, and $\hat{\mathbf{p}}_k^{[j]}$ are estimated quantities. Assuming white Gaussian noise, the measurement residual from (3) is expected to follow the distribution

$$\mathbf{r}_k^{[j]} \hookrightarrow \mathcal{N}\left(\mathbf{0}_{N\times 1}, \mathbf{S}_k^{[j]}\right), \tag{4}$$

$$\mathbf{S}_k^{[j]} = \mathbf{H}_k^{[j]}\mathbf{P}_k^-\mathbf{H}_k^{[j]^T} + \mathbf{R}_k^{[j]}, \tag{5}$$

where $\mathbf{P}_k^-$ is the $(N+M)\times(N+M)$ state estimate covariance matrix at time t_k and $\mathbf{H}_k^{[j]^T}$ is the $Z\times(N+M)$ Jacobian of $\mathbf{h}^{[j]}$.

Sensors are initialized in one of two modes: trusted or untrusted. Untrusted sensors are required to enter a sensor validation mode prior to being brought into monitoring mode [10]. In validation mode, ARMAS employs a likelihood function to monitor the statistical distribution of a user-defined monitoring period composed of recent Kalman pre-update residuals. A Chi-square, χ^*, test statistic is used to detect excursions outside a user-defined threshold across the sampling period. Sensors in validation mode are excluded from impacting the main state estimates using a Schmidt partial update [2]. Trusted sensors are directly brought online into monitoring mode. In monitoring mode, sensor measurements are allowed to update the main state estimates. ARMAS employs the same pre-update residual likelihood function used in the validation mode to monitor sensor performance. A detailed explanation of monitoring mode, including FDE and integrity functions is given in section 2.2.

Once a fault is detected, the sensor is no longer "trusted" and is quarantined from affecting the core navigation state estimate, $\hat{\mathbf{x}}^{[j]}$. ARMAS attempts to reinitialize the sensor via validation mode. If this fails, ARMAS attempts to repair and recover the faulty sensor via two separate modes: sensor calibration and remodeling. In calibration mode, user-selectable sensor parameters, $\mathbf{p}^{[j]}$ and/or $\alpha^{[j]}$ are estimated using residual monitoring from trusted sensors that have observability of $\mathbf{x}$. If there is a single calibration parameter, ARMAS attempts to correct the calibration using residual monitoring and sends the sensor back to validation mode. If linked extrinsic calibration parameters exist, (e.g. camera lever arm and camera orientation within $\mathbf{p}^{[j]}$ or $\alpha^{[j]}$), these are estimated individually and sequenced based on convergence of the state covariance matrix to maintain state observability.

If the recalibrated sensor fails to pass sensor validation, the sensor enters remodeling mode where ARMAS attempts to modify the measurement model, $\mathbf{h}^{[j]}$, based on $1...S$ user-defined measurement models. S concurrent filters are spawned (each with a unique measurement model), and an epoch of measurement residuals is gathered against the core navigation estimate $\mathbf{x}$. The 'winning' sensor measurement model is selected based on which filter best matches the prescribed distribution (4) during the residual epoch. The sensor then enters validation mode. If the remodeling mode does not result in a new model selection, and Resilient Sensor Recovery (RSR) is activated, the sensor periodically re-enters validation mode after a user-selectable time period in an attempt to overcome a temporal anomaly [9]. Figure 1 is a state transition diagram depiction of these modes. The result is a framework compatible with heterogeneous, asynchronous all-source sensors with the benefits of resilience against various sensor calibration, modeling, and temporal faults.

One assumption that is not explicitly discussed in [9] is that ARMAS requires overlapping state observability [6] to detect anomalous sensor behavior. As discussed above, the system monitors Kalman pre-update residuals between sensor measurements and subfilter estimates to continuously judge whether sensor measurements adhered to the distribution prescribed by the sensor model. Anomalous sensor behaviors (e.g. bias, gain, model mismatch, high noise, etc.) are only observable if there are other sensors with comparable observability into the state estimate. If anomalous behavior is detected, the ARMAS framework attempts to recover the sensor through recalibration, remodeling, and re-validation. Without overlapping state observability, it is impossible to determine if a sensor is misbehaving or if it can be re-validated. The criticality of this assumption for ARMAS is highlighted in the analysis below.

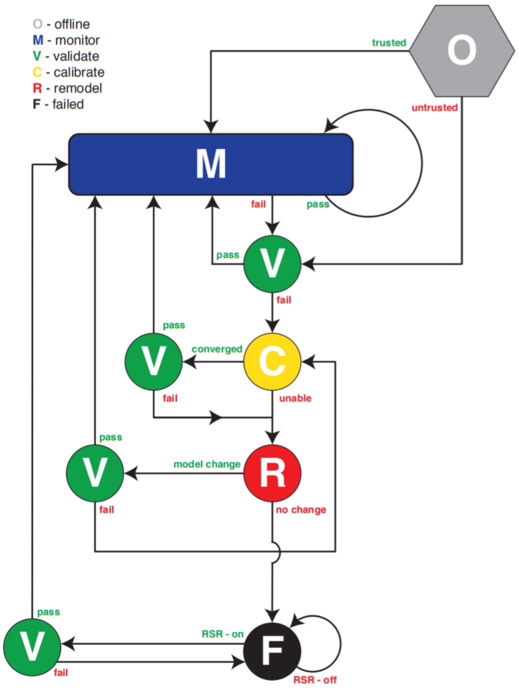


Figure 1: ARMAS Framework State Diagram [9]. A sensor begins at point **O** (origin) and is "trusted" or "untrusted". SAARM and the new contribution SOM reside within the **M** (Monitoring) mode.

2.2 Sensor-Agnostic All-source Residual Monitoring (SAARM)

SAARM assumes a system form of

$$\dot{\mathbf{x}}(t) = \boldsymbol{f}[\mathbf{x}(t), \boldsymbol{\alpha}(t), \boldsymbol{u}(t), t] + \boldsymbol{G}(t)\boldsymbol{w}(t) \tag{6}$$

where $\mathbf{x}$ is a $N \times 1$ state vector of a vehicle's position, velocity, and attitude. The measurement error states vector α is of dimension $M \times 1$, $\mathbf{u}$ is the control input vector, $\mathbf{G}$ is an $(N + M) \times W$ linear operator, and $\mathbf{w}$ is a $W \times 1$ white noise process defined by a $W \times W$ continuous process noise strength matrix, $\mathbf{Q}$. SAARM estimates system states with J separate subfilters. At time $t = t_k$, the system state vector and state estimation covariance matrix are defined by

$$\hat{\mathbf{x}}^{[j]}(t_k) \text{ and } \mathbf{P}^{[j]}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}(t_k),$$

for$j = 1 \ldots J$ separate subfilters. Each of these subfilters is informed by a subset of $I - 1$ sensors. At $t = t_k$, the i^{th} sensor provides measurements given by

$$\mathbf{z}^{[i]}(t_k) = \mathbf{h}^{[i]}\left[\hat{\mathbf{x}}^{[j]}(t_k^-), \mathbf{u}(t_k), t_k\right] \tag{7}$$

where $\mathbf{h}^{[i]}$ is the nonlinear measurement function, $\mathbf{u}(t_k)$ is the control input function, and $\mathbf{v}^{[i]}(t_k)$ is a discrete white noise process of dimension $\mathbf{Z}_i \times 1$ defined by covariance matrix $\mathbf{R}^{[i]}(t_k)$. The pre-update measurement estimate for sensor i from filter j is defined by

$$\hat{\mathbf{z}}^{[i,j]}(t_k) = \mathbf{h}^{[i]}\left[\hat{\mathbf{x}}^{[j]}(t_k^-), \mathbf{u}(t_k), t_k\right], \tag{8}$$

where the estimated covariance matrix is defined by

$$\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}{}^{[i,j]}(t_k^-) = \mathbf{H}^{[i]}(t_k^-)\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}{}^{[j]}(t_k^-)\mathbf{H}^{[i]}(t_k^-)^T. \tag{9}$$

Using (8) and (9), the “pre-update residual” vector between sensor i and filter j, $\mathbf{r}^{[i,j]}$ and its covariance matrix, $\mathbf{P}_{rr}^{[i,j]}$ are defined as

$$\mathbf{r}^{[i,j]}(t_k) = \mathbf{z}^{[i]}(t_k) - \hat{\mathbf{z}}^{[i,j]}(t_k^-), \tag{10}$$

$$\mathbf{P}_{rr}^{[i,j]}(t_k) = \mathbf{R}^{[i]}(t_k) + \mathbf{P}_{\hat{z}\hat{z}}^{[i,j]}(t_k^-). \tag{11}$$

Fault detection relies on computing a moving average of recent residual-space test statistics formed by pre-update residual vectors from (10) and (11). ARMAS is designed to detect any sensor behavior which is inconsistent with the stated measurement model within the limitations of the stated significance level, *alpha*. Although not examined in this paper specifically, the ARMAS framework provides additional options for the user to provide candidate models and/or calibration schemes which could be used to validate and recover a failed sensor. The likelihood function focuses on a single residual-space statistic derived from the Mahalanobis distance, d, given by

$$d^2 = (\mathbf{y} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{y} - \mu), \tag{12}$$

where μ and $\mathbf{\Sigma}$ are the mean and covariance of a Z_i-dimensional Gaussian distribution. It is known that a sum of M independent d^2 distances follows a χ^* distribution with Z degrees of freedom [4] given by

$$\chi^* = \sum_{s=k}^{k+M} d^2(t_s), \tag{13}$$

$$d^2(t_k) = \mathbf{r}^T(t_k)[\mathbf{P}_{rr}(t_k)]^{-1}\mathbf{r}(t_k). \tag{14}$$

The set of pre-update residuals is known to be a zero-mean, white sequence [16]. The fault detection test for M pre-residuals is composed of the following hypotheses:

$$H_0 : \chi^*_{[i,j]} < \chi^2(1 - \alpha/2, M \times Z_i) \tag{15}$$

$$H_1 : \chi^*_{[i,j]} > \chi^2(1 - \alpha/2, M \times Z_i) \tag{16}$$

where α is the probablity of false alarm and M is the number of averaged pre-residual samples. H_0 refers to the hypothesis where the fault is not present in filter j. H_1 refers to the hypothesis where a fault is present in filter j. The resulting hypothesis test forms the basis of the fault detection algorithm.

Once a fault is detected, a agreement of all other subfilters is utilized to exclude the faulty sensor. With $J = I$ subfilters, SAARM can only exclude single faults within each residual monitoring epoch (i.e. M-sample moving average). In this scenario, each subfilter is informed by a different subset of $I - 1$ sensors (i.e. each subfilter is missing a single sensor). SAARM also assumes that all states are observable by all subfilters. In addition to $J = I$ subfilters, a main filter is maintained to generate a full navigation state estimate for user output. Accordingly, cross-covariance terms between the main filter and any other filters are not used for any computation. For this scenario, SAARM provides an axiom for fault exclusion: *under the assumption that, at most, one sensor can fail simultaneously, at least one of the J subfilters will be completely unaffected by faulty measurements* [9].

The fault consensus is tallied in a $\mathbf{T}$-matrix of dimension $I \times J$ and uses the following convention:

$$\mathbf{T}(i,j) = \begin{cases} 0, & \text{Sensor } i \text{ not associated with filter } j \\ 0, & \chi^*_{[i,j]} < \chi^2(1-\alpha/2, M \times Z_i), \text{ No Fault, } H_0 \\ 1, & \chi^*_{[i,j]} > \chi^2(1-\alpha/2, M \times Z_i), \text{ Fault, } H_1 \end{cases} \tag{17}$$

Figure 2 shows the relationship between I sensors and J subfilters required for “fault agreement“ sensor exclusion. The rows correspond to the $i = 1...I$ sensors and the columns correspond to

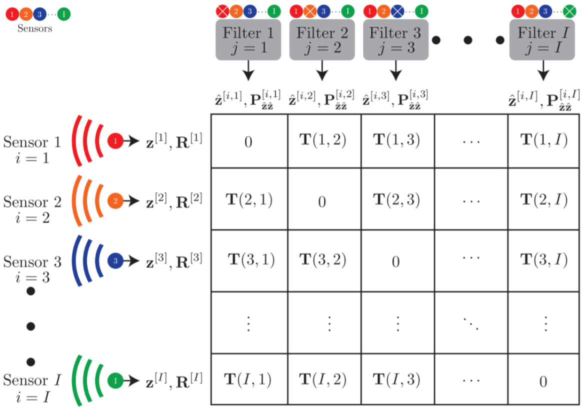

Sensors
Filter 1 $j = 1$
Filter 2 $j = 2$
Filter 3 $j = 3$
Filter I $j = I$
$\hat{\mathbf{z}}^{[i,1]}, \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{[i,1]}$
$\hat{\mathbf{z}}^{[i,2]}, \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{[i,2]}$
$\hat{\mathbf{z}}^{[i,3]}, \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{[i,3]}$
$\hat{\mathbf{z}}^{[i,I]}, \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}^{[i,I]}$
Sensor 1 $i = 1$
Sensor 2 $i = 2$
Sensor 3 $i = 3$
Sensor I $i = I$
$\mathbf{z}^{[1]}, \mathbf{R}^{[1]}$
$\mathbf{z}^{[2]}, \mathbf{R}^{[2]}$
$\mathbf{z}^{[3]}, \mathbf{R}^{[3]}$
$\mathbf{z}^{[I]}, \mathbf{R}^{[I]}$
0 | $\mathbf{T}(1,2)$ | $\mathbf{T}(1,3)$ | $\cdots$ | $\mathbf{T}(1,I)$
$\mathbf{T}(2,1)$ | 0 | $\mathbf{T}(2,3)$ | $\cdots$ | $\mathbf{T}(2,I)$
$\mathbf{T}(3,1)$ | $\mathbf{T}(3,2)$ | 0 | $\cdots$ | $\mathbf{T}(3,I)$
$\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$
$\mathbf{T}(I,1)$ | $\mathbf{T}(I,2)$ | $\mathbf{T}(I,3)$ | $\cdots$ | 0

Figure 2: SAARM T-Matrix for $i = 1...I$ All-Source Sensors [11]. If Sensor 3 were faulty, for example, each subfilter that includes that sensor would report a fault (all across row three) and only subfilter 3 would remain consistent. Filter 3 would then be promoted to the main filter level and a new bank of FDE subfilters must be populated.

the $j = 1...J$ subfilters. Each row contains measurements, $\mathbf{Z}^{[i]}$, and the measurement covariance matrix, $\mathbf{R}^{[i]}$, from the i^{th} sensor. Each column contains the estimated measurement, $\hat{\mathbf{z}}^{[i,j]}$, and its covariance matrix, $\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}{}^{[i,j]}$.

Based on the stated convention, a fault is declared when $\mathbf{T}$ contains a single nonzero entry (i.e. at least one subfilter detected H_1). After a fault is declared, SAARM waits for an agreement from the remaining subfilters until a single fault-free subfilter remains. It is assumed that the last remaining fault-free subfilter is the one which does not contain the faulty sensor. After fault exclusion, the fault-free subfilter is elevated to "main filter" status and the pre-update residual monitoring epoch is restarted with $I - 1$ total sensors and $J - 1$ subfilters. Each newly spawned subfilter now contains $I - 2$ sensors. The faulty sensor is removed from monitoring mode and follows the state diagram shown in Figure 1. Of note, SAARM is able to detect the occurrence of multiple simultaneous faults but is only able to provide multiple fault exclusion when initialized with additional subfilter layers.

In summary, SAARM provides all-source sensor FDE and integrity for various sensor fault types. To provide resiliency to a single fault, ARMAS is required to instantiate and maintain a quantity of subfilters equal to the quantity of all-source sensors. A separate main filter is maintained strictly for user output. Fault identification is based on a sequence of χ^* statistical tests of pre-update measurement residuals. Fault exclusion is based on a subfilter agreement approach which is tallied in a novel T matrix. If every subfilter has position state observability, then SAARM provides a method for all-source position integrity via the union of all subfilter position covariance estimates. This integrity concept is based on the assumption that the framework is able to maintain at least one uncorrupted subfilter. The next section describes the motivation for a novel layer in ARMAS used for real-time observability analysis.

2.3 Motivation for Stable Observability Monitoring

The ARMAS framework with SAARM was originally conceived and simulated with basic linear 2D position and velocity sensors and assumed fully overlapping state observability within the FDE layer. All-source sensors, particularly those that only provide partial state information do not intrinsically exhibit this characteristic. In collaborative navigation scenarios, retention of autonomy is desirable as long as a stable, resilient solution can be maintained. Another key motivation for SOM is the ability to determine when to augment with collaborative information and a method to measure the sufficiency of the collaborative information. In early 2020, the ARMAS framework was applied to a flight test dataset [1] (Figure 3). The flight

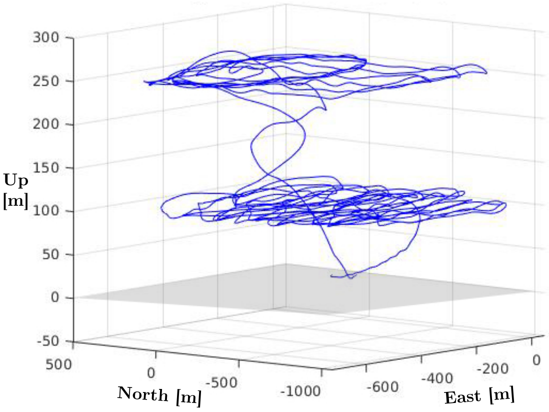


Figure 3: sUAS Flight Test Data, 12 Oct 2018, Camp Atterbury, IN

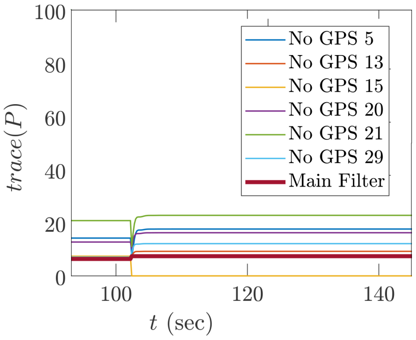


Figure 4: Trace of *aposteriori* Layer 1 Subfilter Position Covariance, Simulated Single GPS 15 Pseudorange Sensor Failure

was conducted by the ANT Center on 12 October 2018 at Camp Atterbury, IN where a small Unmanned Aerial System (sUAS) took off and landed at Himsel Army Airfield (HAA). During the 27-minute dataset, the aircraft flew patterns at approximately 250 and 100 meters above the local surface. The aircraft used the ANT Center's Scorpion framework [12] to provide a GNSS/INS-coupled 'truth' navigation solution. This analysis consists of individual pseudorange measurements extracted from six Global Positioning System (GPS) Satellite Vehicles (SV) for nonlinear processing in ARMAS as individual sensors.

For analysis, we configured ARMAS with a sensor package consisting of 6 individual pseudorange sensors, one for each visible SV. This means each filter in the FDE layer is equipped with a unique combination of 5 sensors. The latter half of the flight test dataset contains numerous pseudorange sensor dropouts. During analysis, it was observed that sensor dropouts tended to cause spurious behavior in ARMAS. This behavior occurs when less than 5 SVs are visible, meaning each layer 1 subfilter is unable to perform a stable position state estimate with less than 4 SVs. Further analysis reveals that SAARM is unable to provide an agreement to identify a failed sensor when the FDE subfilter layer loses overlapping position state observability. In other words, SAARM can detect a sensor fault but can not exclude the faulty sensor if even a single FDE layer subfilter loses position state observability due to dropout, poor geometry, etc. Since the initial simulation of ARMAS was performed with fully overlapping position state observability in the FDE layer, this deficiency was overlooked.

Consider a scenario where SV 'GPS 15' in a 6-SV constellation is excluded by ARMAS due to a simulated pseudorange bias at $t = 100$ sec. Figure 4 shows a local observability analysis for this scenario with the remaining observability subfilters, FDE layer subfilters, and Main Filter. Note there are 10 layer 2 subfilters remaining (from the original 15) in the observability

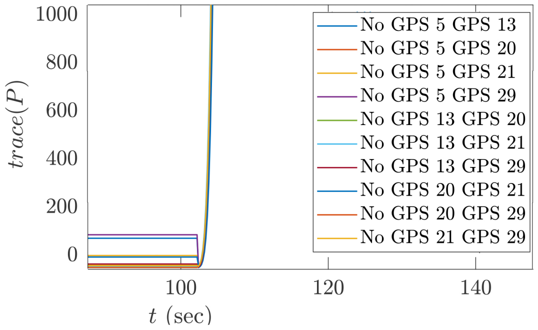


Figure 5: Trace of *aposteriori* Layer 2 Subfilter Position Covariance, Simulated Single GPS 15 Pseudorange Sensor Failure

layer, corrresponding to 5 remaining SVs (after GPS 15 was excluded). Note that each layer 2 subfilter in the observability layer is informed by exactly 4 unique pseudorange sensors until GPS 15 is removed. Since 4 unique pseudoranges are required to constrain a stable 3D position solution with clock bias estimation, the unbounded position state covariance matrix indicates a complete loss of observability after $t = 100$ sec. As mentioned, a single pseudorange sensor dropout in this scenario will result in the loss of position state observability and is evidenced by an increase in the position state covariance estimate. If an additional sensor failure occurs, the decision-making FDE layer of ARMAS will struggle to provide a subfilter agreement to determine the culprit sensor due to a reduction in position state observability. This analysis forms the genesis of local observability monitoring at the layer 2 "sub-subfilter" level to preserve the consistency of the layer 1 subfilter FDE and integrity functions of ARMAS.

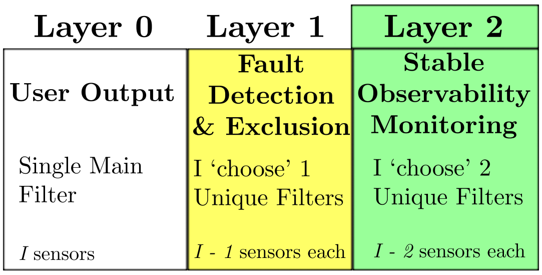
3 Novel Observability Layer 2 "Sub-subfilter" Bank

In the previous section, we began the motivation for monitoring observability at a layer deeper than the FDE and integrity operations to preserve estimation consistency and framework resiliency. The uncorrupted subfilter guarantee provided by the ARMAS framework for a single simultaneous fault enables SAARM to extend a similar guarantee for all-source position integrity [11]. The previously stated fault exclusion axiom is extended:

Assuming at least one of the subfilters is informed entirely by properly modeled, uncorrupted sensors, then at least one subfilter contains consistent state estimation error statistics [11]. **If the states of interest in each layer 2 subfilter are observable and stabilizable, then each layer 1 subfilter inherits these properties.**

This means that the physical region encompassed by the position covariance estimates of all layer 1 subfilters contains the true navigation state within the statistical significance of the fault detection tests. That said, SAARM requires overlapping position observability across all layer 1 subfilters to perform consistent FDE operations and guarantee the preservation of at least one uncorrupted subfilter for position integrity. For SAARM to guarantee position state observability at the layer 1 subfilter level, an additional layer of subfilters is required (Figure 6). Each unique subfilter in the observability layer excludes measurements from two sensors. The purpose of this layer is to provide a means for observability analysis one layer deeper than the decision-making FDE layer to maintain resiliency to a single simultaneous sensor fault.

For example, if a failed sensor is excluded, the FDE layer will be repopulated with new subfilters, each missing a single unique sensor and the failed sensor. Prior to the sensor exclusion, a subset of the observability layer contains the set of filters needed to spawn the new FDE subfilter layer after the exclusion. The purpose of monitoring the observability one layer deeper



Layer 0	Layer 1	Layer 2
User Output	**Fault Detection & Exclusion**	**Stable Observability Monitoring**
Single Main Filter	I ‘choose’ 1 Unique Filters	I ‘choose’ 2 Unique Filters
I sensors	*I - 1* sensors each	*I - 2* sensors each

Figure 6: ARMAS Framework with Novel Observability Layer for Resiliency to One Simultaneous Sensor Failure.

than the decision-making FDE layer is to provide a mechanism to warn the user in the event that a single sensor failure could jeopardize the consistency of the FDE and integrity operations. This warning comes in the form of a user “observability warning” to add an additional sensor to the framework. If only a single subfilter in the FDE layer loses overlapping position observability, SAARM is unable to provide a subfilter agreement required to identify and exclude the failed sensor. Once overlapping position observability is lost in the FDE layer, ARMAS can no longer guarantee at least one consistent uncorrupted subfilter which is required to preserve solution integrity. Due to potentially variable Fisher information available from all-source navigation sensors, it is critical that a resilient all-source navigation framework contains a method to monitor observability prior to corruption of the decision-making FDE layer.

To maintain resilience to F simultaneous sensor failures, the number of concurrent layer 2 subfilters in the observability filter bank, N, required to monitor observability for I sensors is:

$$N = \binom{I}{F+1} = \frac{I!}{(F+1)(I-(F+1))!} \tag{18}$$

As one might expect, the processing requirements to monitor each layer 2 subfilter position state covariance are non-trivial. For example, an 8 sensor system requires 28 concurrent SAARM layer 2 subfilters for resiliency to a single simultaneous sensor failure. When summed with the main filter and 8 traditional subfilters, a total of 37 concurrent estimation filters must be maintained. This method monitors overlapping position observability at the processing expense of factorial growth in the required quantity of concurrent estimation filters. A major benefit of this approach is evident in the event of a sensor failure. Since a subset of the observability filter bank will form the new FDE layer, maintenance of these filters in the observability layer eliminates FDE and integrity operation downtime normally required during FDE layer re-initialization. Additionally, if we monitor the magnitude of the state estimate variance in each layer 2 subfilter we can determine which sensors provide critical Fisher information about our state(s) of interest prior to fault detection and exclusion events.

4 Stable Observability Monitoring (SOM)

The ability to maintain stable *aposteriori* estimates of system states is a primary indicator of overall estimator stability [6]. A primary goal of observability analysis is to measure the influence measurements have on system states [16]. Observability analysis has been applied to fused estimation with a variety of approaches including information matrix [8], error covariance analysis [21], and others [15] [7] [13]. It is well understood that a discrete linear time varying system is globally observable if the rank of its observability matrix $\mathbf{M}$ is of rank n for all time indices k. The degree of local observability can be defined as the measure of the singularity of $\mathbf{M}$ over a finite set of k [3]. For linear systems, the observability Gramian can be obtained as a solution of the the Lyapunov equation [5].

Directly related to the observability Gramian is the Fisher Information matrix which is a measure of the certainty of the state estimate due to measurement data alone [14] [17] [20]. The discrete recursive definition of the Fisher information matrix $\boldsymbol{F}$ is

$$\boldsymbol{F}(t_k^+) = \mathbf{\Phi}^T(t_k)\boldsymbol{F}(t_k^-)\mathbf{\Phi}(t_k) + \mathbf{H}^T(t_k)\mathbf{R}(t_k)^{-1}\mathbf{H}(t_k) \tag{19}$$

where the Fisher information contained in a single update at t_k is $\mathbf{H}^T(t_k)\mathbf{R}(t_k)^{-1}\mathbf{H}(t_k)$ which is the same term used to generate

$$\mathbf{P}(t_k^+)^{-1} = \mathbf{P}(t_k^-)^{-1} + \mathbf{H}^T(t_k)\mathbf{R}(t_k)^{-1}\mathbf{H}(t_k). \tag{20}$$

This relationship can be leveraged for observability analysis in a nonlinear estimator. If the system model is stochastically controllable and observable, then $\mathbf{P}(\mathbf{t_k^+})$ is uniformly bounded from above [16]. Stabilizable states have a unique positive-definite $\mathbf{P}(\mathbf{t_k^+})$ [16]. By monitoring post-update covariance matrices over time, we can ascertain if the signal to noise ratio in the system enables stabilized estimation. For the purposes of this paper, we focus on the stability of the position states because we are particularly interested in preserving the consistency of the navigation integrity solution provided by the FDE sublayer in ARMAS.

The user-defined monitoring epoch for the sum of Mahalanobis distances in (13) adjusts the sensitivity of the SAARM monitoring test within the ARMAS framework [18]. This is a moving average of recent residual-space test statistics formed by pre-update residual vectors from (10) and (11). The length of 'monitoringTime' ($M\Delta t$) is an ARMAS tuning parameter which is used to adjust for detection sensitivity for temporal anomalies. The ARMAS 'monitoringTime' is designed to contain a sufficient quantity of samples to meet central limit theorem [19] criteria for the desired α. The pluggable estimation architecture provided by SCORPION enables propagation of multiple layers of subfilters. We record and monitor the post-update position covariances in each $n = [1...N]$ layer 2 subfilter in the observability bank. An observability flag O_k is set for layer 2 subfilter n for t_k according to

$$O_{k,i}(n) = \begin{cases} 1, & \text{if } \mathbf{P}_j^{\mathbf{[n]}}(t_k^+) > \mathbf{P}_j^{\mathbf{[n]}}(t_{k-M}^+)\beta \\ 1, & \mathbf{P}_j^{\mathbf{[n]}}(t_k^+) > \mathbf{P}_{j,max} \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

where $\mathbf{P}_i^{\mathbf{[n]}}(t_k^+)$ is the jth diagonal of the post-update state covariance matrix for layer 2 subfilter n, $\mathbf{P}_i(t_{k-M}^+)$ is the post-update variance for state estimate element j for layer 2 subfilter n exactly M samples prior to t_k, $\mathbf{P}_{j,max}$ a user-defined limit for maximum steady-state state estimate covariance, and $\beta \in [1, \infty)$ is the state estimate covariance transient growth threshold. If $k < M$ (e.g. a newly initialized filter), an observability flag is not set. A more compact form used to monitor position states is

$$O_{k,pos}(n) = \begin{cases} 1, & \text{if } tr\big(\mathbf{P}_{\mathbf{pos}}^{\mathbf{[n]}}(t_k^+)\big) > tr\big(\mathbf{P}_{\mathbf{pos}}^{\mathbf{[n]}}(t_{k-M}^+)\big)\beta \\ 1, & \text{if } tr\big(\mathbf{P}_{pos}^{\mathbf{[n]}}(t_k^+)\big) > tr\big(\mathbf{P}_{pos,max}\big) \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

where $\mathbf{P}_{\mathbf{pos}}^{\mathbf{[n]}}(t_k^+)$ is the most recent post-update position covariance matrix for layer 2 subfilter n, $\mathbf{P}_{\mathbf{pos}}^{\mathbf{[n]}}(t_{k-M}^+)$ is the post-update position covariance matrix for layer 2 subfilter n exactly M samples prior to t_k $\mathbf{P}_{pos,max}$ is a user-defined limit for maximum steady-state position state estimate covariance, and $\beta \in [1, \infty)$ is the state estimate covariance transient growth threshold. The trace is the sum of the diagonal elements of the matrix $\mathbf{P_k}$, which represent variances of the system state estimates. If the $tr\big(\mathbf{P}_{\mathbf{pos}}^{\mathbf{[n]}}(t_k^+)\big)$ converges, then the individual position estimate variances also converge. When applying (22), it is important to ensure units are identical across the grouped states.

By measuring the difference between the trace of post-update position covariance matrices, we can determine if the position state information contained in the $\mathbf{H}^T(t_k)\mathbf{R}(t_k)^{-1}\mathbf{H}(t_k)$ terms in the previous M measurements have resulted in a stable mean estimate of all position elements in that subfilter. The user sets β as a tuning parameter for acceptable transient variance growth per monitoring epoch. To maintain resilience to a sensor failure, a user prompt to augment ARMAS with an additional sensor is triggered if at least a single layer 2 subfilter observability test sets to 1 (23). The newly added sensor will directly enter monitoring mode if it is considered 'trusted' or must pass through sensor validation if 'untrusted'.

$$SOM_{AddNewSensor} = \begin{cases} true, & \text{if } \sum_{n=1}^{N} O_k(n) > 0 \\ false, & \text{otherwise} \end{cases} \tag{23}$$

where N is the quantity of layer 2 subfilters.

Once a new sensor is successfully added into ARMAS monitoring mode, each layer 2 subfilter gains another sensor. The results of (21) or (22) are ignored until the newly requested sensor completes an ARMAS monitoring period after entering monitoring mode. If post update variance stability is regained after the requisite ARMAS monitoring period, then (21) or (22) will set to 0 for each stable layer 2 subfilter and the observability warning is rescinded. This method flags the presence of an information deficiency with respect to the estimated states of interest in real-time across multiple layer 2 subfilters in a novel observability bank which is intended to preserve the consistency of the FDE and integrity operations in ARMAS.

5 State Estimate Covariance Transient Growth Threshold (Beta) and Maximum State Estimate Covariance Limits

The state estimate covariance transient growth threshold, $\beta \in [1, \infty)$, sets SOM framework sensitivity to transient loss of Fisher information for the state(s) of interest (21). The maximum state estimate covariance limit is simply an upper bound for state estimate covariance and is designed to set a minimum steady-state information threshold for the framework. The validation gate employed by SAARM is a moving window of residual-space test statistics in the form of Mahalanobis distances. The power of SAARM's Chi-squared distributed hypothesis test is dependent on the ARMAS framework's ability to produce a stable and consistent pre-update residual covariance matrix, Σ (14). By operating SOM's state estimate covariance monitoring scheme in the layer 2 sublayer (See Figure 6), we are able to peek forward at framework stability in the event of a single unknown sensor failure.

Since the pre-update residual covariance matrix is a function of the state estimate covariance and the observation model in (9), the post-update state estimate covariance $\mathbf{P}^+$ is a direct indicator for estimator observability and stabilizability [16]. The least stabilizable layer 2 subfilter is informed by exactly 1 fewer sensor than the least stabilizable layer 1 subfilter. By monitoring the layer 2 subfilter level, this allows for a sensor augmentation request before a potential loss of stability in the FDE operations. Since T-matrix exclusion operations employed by SAARM require a unanimous fault agreement to exclude a failed sensor (17), it is particularly important that the stability of the layer 1 subfilter estimates is ensured. Furthermore, since ARMAS navigation integrity is provided by the union of the layer 1 subfilter position covariance ellipses, position integrity can be stabilized if the position states are selected as the SOM states of interest.

A selection of $\beta = 1.0$ is the most sensitive case which results in a request for an additional sensor when any layer 2 subfilter state estimate covariance grows by any amount in a single ARMAS monitoring epoch according to (23). This extreme case will maximize use of available offboard information. In the opposite case, large values of β will produce a less sensitive framework without a tendency to request additional sensor augmentation, where $\beta \to \infty$ performs identically to legacy ARMAS. For initial implementation, we recommend beginning with a small value for $\beta \approx 2.0$. This tuning parameter should be adjusted by the user to set the desired balance between autonomy and framework estimation stability for the state(s) of interest. The next section describes a set of reconfigurable GNSS scenarios used to assess the effectiveness of SOM.

6 Simulation

This section describes a set of four example 3D scenarios designed to assess the impacts of SOM on ARMAS FDE operations and untrusted sensor validation. Individual SVs are arranged in random stationary GNSS constellations in a local tangential frame (LTF). In scenarios 1-4, we initialize three aircraft operating a standard EKF, ARMAS, and ARMAS-SOM (22) with 4, 5, 6, and 7 trusted pseudorange sensors, respectively. We assume the remaining untrusted SVs are available for augmentation via SOM. The availability of additional untrusted sensor information can be analagous to offboard augmentation in a collaborative navigation scenario. Each constellation contains 1 SV directly overhead and 9 SVs evenly distributed in azimuth with discrete random uniform elevations between 45 and 63.4 degrees (See Fig. 7). Each scenario consists of two sensor anomalies introduced at fixed times: (1) a growing pseudorange bias (linear ranging ramp) from $t = 240$ to $t = 330$ sec on a trusted sensor and (2) validation of an untrusted pseudorange sensor with a constant 40 meter bias at $t = 360$ sec. The growing pseudorange bias on the trusted sensor is used to assess FDE in monitoring mode (i.e. how large is the bias before it is detected and excluded). The constant bias is used to measure probability of detection in validation mode after recovery from a sensor exclusion.

Consider a three-dimensional example with two vehicles obtaining their navigation solutions from an EKF within the ARMAS framework

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{x}}_p(t) \\ \dot{\mathbf{x}}_v(t) \\ \dot{\mathbf{x}}_a(t) \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{x}}_v(t) \\ \dot{\mathbf{x}}_a(t) \\ -\frac{1}{\tau_a}\dot{\mathbf{x}}_v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathbf{w(t)} \end{bmatrix} \tag{24}$$

where $\mathbf{x_p}$ is the vehicle's position (m), $\mathbf{x_v}$ is the vehicle's velocity (m/s), $\mathbf{x_a}$ is the vehicle's acceleration (m/s^2), and $\tau_a = 90$ seconds is a time-constant associated with a First-order Gauss-Markov (FOGM) process. A 3D white noise process is given by $w(t)$ where E[$\mathbf{w}$(t) $\mathbf{w}$(t+τ)^T] = $\mathbf{Q}\delta(\tau)$ and

$$\mathbf{Q} = (\mathbf{1.0 \times 10^{-2}})^\mathbf{2}\mathbf{I_{3\times3}}\ (\mathbf{m^2/s^4}) \tag{25}$$

The model from (24) is used to generate randomly initialized vehicle trajectories for each trial. Initial velocities and accelerations are normally distributed with $\sigma_{accel} = 1 \times 10^{-2}$ (m/s^2) and $\sigma_{vel} = 5$ (m/s). Figure 8 shows 300 sample truth trajectories for this scenario.

Each vehicle is initialized identically with initial state and covariance estimates:

$$\hat{\mathbf{x}}(0) = \begin{bmatrix} 0 & 0 & 200 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{26}$$

$$\mathbf{P(0)} = \mathbf{diag}\Bigg(\begin{bmatrix} 30 & 30 & 30 & 10 & 10 & 10 & 0.01 & 0.01 & 0.01 \end{bmatrix}^\mathbf{2} \Bigg). \tag{27}$$

Each aircraft receives discrete measurements from a constellation of stationary satellite vehicles (SV). Although there are 10 SVs (labelled GPS1-10), the aircraft are initialized with a

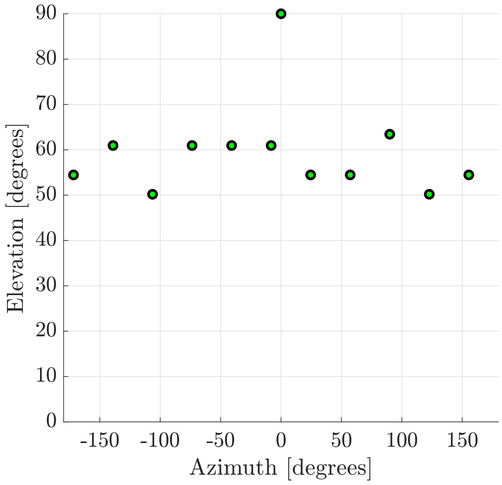


Figure 7: Sample of 10-SV Stationary Constellation Skyplot with Discrete Random Uniform Elevations and 1 Satellite Directly Overhead

trusted subset of this constellation varying from 4 to 7 SVs. An additional fixed satellite with a random uniform elevation between 45 and 63.4 degrees is introduced at $t = 360$ sec and is corrupted with a constant 40 meter pseudorange range bias. This constellation is designed to provide coverage at high elevation angles between approximately 45 degrees and 63.4 degrees with a single satellite directly overhead (Figure 7).

Individual pseudorange measurements are performed according to (28)

$$\rho_i = \sqrt{(X_{SV,i} - X_u) + (Y_{SV,i} - Y_u) + (Z_{SV,i} - Z_u)} + b_u, \tag{28}$$

where ρ_i is the pseudorange to SV i, with fixed coordinates (X_{SV}, Y_{SV}, Z_{SV}), estimated user coordinates are (X_u, Y_u, Z_u), and an estimated GNSS receiver clock bias is b_u. The pseudorange measurement covariance $\mathbf{R_{SV}} = 10^2$ m^2. A receiver clock bias b_u is independently estimated as an additional state in each EKF. All coordinates are expressed in the LTF.

Pseudorange measurements, $\hat{\mathbf{z}} = h(\hat{\mathbf{x}})$, are performed according to (30) for the estimated position states (X_u, Y_u, Z_u, b_u) within $\hat{\mathbf{x}}$. The measurement Jacobian $\boldsymbol{H}$ is:

$$\boldsymbol{H} = \begin{bmatrix} \frac{-(X_{i,SV} - X_u)}{r_i} & \frac{-(Y_{i,SV} - Y_u)}{r_i} & \frac{-(Z_{i,SV} - Z_u)}{r_i} & 1 \end{bmatrix} \tag{29}$$

where

$$r_i = \sqrt{(X_{SV,i} - X_u) + (Y_{SV,i} - Y_u) + (Z_{SV,i} - Z_u)} \tag{30}$$

is the distance from the platform to an SV.

In addition to initial pseudorange sensors, additional unbiased pseudorange sensors may be requested automatically by SOM at any time during the scenario. The other two approaches (legacy ARMAS, EKF) are limited to the sensors provided during initialization. In this respect, SOM has a clear advantage over the two legacy approaches. The point of this analysis is to show that ARMAS-SOM can detect a threat to navigation resilience, provide timely sensor augmentation, and successfully preserve the consistency of the FDE and validation operations in ARMAS. In the middle of the scenario, an insidious growing position bias is injected into a single trusted pseudorange sensor to test the ARMAS FDE process. The size of the position bias at exclusion is recorded. Near the end of the scenario, an untrusted pseudorange sensor with a fixed bias is added into validation at $t = 360$ sec to test ARMAS sensor validation.

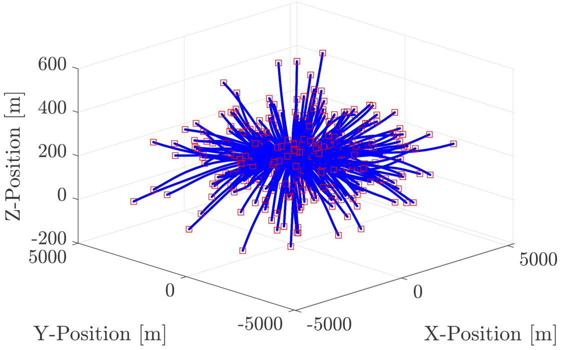


Figure 8: Sample Truth Trajectories, 300 Runs

7 Numerical Results

The following section presents the results for three identical aircraft (EKF, ARMAS, and ARMAS-SOM equipped) in four scenarios with 4, 5, 6, and 7 initial trusted pseudorange sensors. We assume the remaining untrusted SVs are available for augmentation via SOM. The acceptable steady state variance growth term β from (22) is set to 3.0 for all runs. This was achieved by initially setting β to 1.0 and incrementally increasing until desired sensitivity is achieved. The maximum position state estimate covariance is $\mathbf{P}_{pos,max} = diag_{3\times3}(20\text{m}^2)$ (22).

In each scenario, a single trusted pseudorange sensor experiences a sensor anomaly (linear ramp) from $t = 240$ to 330 sec and an untrusted biased sensor (40 meters) is added at $t = 360$ sec. The EKF simply trusts all information provided, has no ability to detect faults nor perform sensor validation, and is included only as a performance baseline. ARMAS is equipped with FDE capabilities and validation of untrusted sensors. The ARMAS recalibration and remodeling modes are not active. ARMAS-SOM includes all of the aforementioned ARMAS capabilities and adds the ability to request additional untrusted pseudorange sensors at any time. SOM monitors the stability of the state covariance estimates in the observability subfilters (layer 2) (22). Since each observability layer subfilter exclude 2 sensors, the minimum number of pseudorange sensors at the user output level (layer 0) is 6. If one of the trusted sensors is excluded, the quantity of sensors at the top level must remain at least 6 to ensure stability at the layer 2 subfilter level.

7.1 Scenario 1

In scenario 1, each aircraft is initialized with 4 trusted pseudorange sensors receiving information from a random stationary constellation of SVs in the LTF. A summary of the results is shown in Table 1. Clearly, ARMAS-SOM outperforms both ARMAS and the EKF, especially in terms of RSS error and detection rate for a biased sensor. This is expected because ARMAS-SOM augments itself with 2 additional untrusted pseudorange sensors using the ARMAS validation process prior to the sensor anomaly event at $t = 240$ sec. Once the spoofed sensor is excluded, ARMAS-SOM requests one additional untrusted sensor to stabilize the observability layer prior to validation of the untrusted biased sensor. This results in a total of 3 added pseudorange sensors for a total of 7. With ARMAS, the exclusion of the spoofed sensor results in only 3 pseudorange sensors which is insufficient to provide a stable 3D position estimate with clock bias estimation. This is evidenced by the large growth in estimated standard error at approx. $t = 245$ sec in Figure 9. Figure 10 shows the mean RSS error and standard deviation for 1000 monte carlo trials. It is clear that ARMAS does not recover well from the exclusion of the spoofed sensor with a bias detection rate of 0.01 is clearly outperformed by ARMAS-SOM with a detection rate of 0.99. The EKF simply trusts all information provided and the navigation solution is carried off by the linear ramp sensor anomaly.

4 Trusted SVs, 1000 Trials					
Aircraft	**Grand Mean (Median) [m]**	**Std. Dev. [m]**	**Mean GPS4 Anomaly Exclusion Mag. [m]**	**Detection Rate 40 m Bias**	**Mean Total Added Sensors**
EKF	916.0 (76.53)	1,720.8	–	–	–
ARMAS	694.3 (76.52)	684.3	44.53	0.01	–
SOM	16.78 (10.15)	18.25	41.51	0.99	3.00

Table 1: Scenario 1 Results

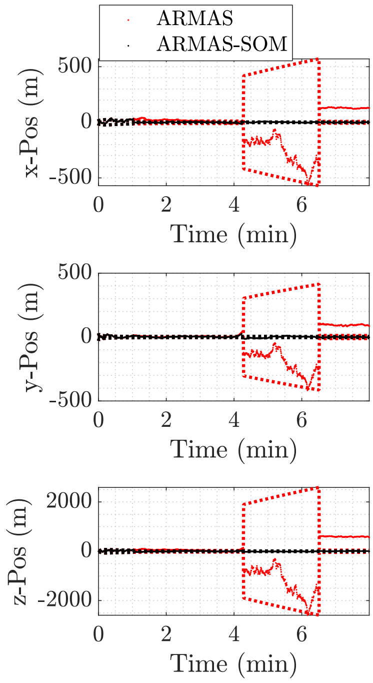


Figure 9: Scenario 1 State Estimation Error for 1 Run with Sensor Anomaly from $t = 240$ to 330 sec and Biased Sensor added at $t = 360$ sec

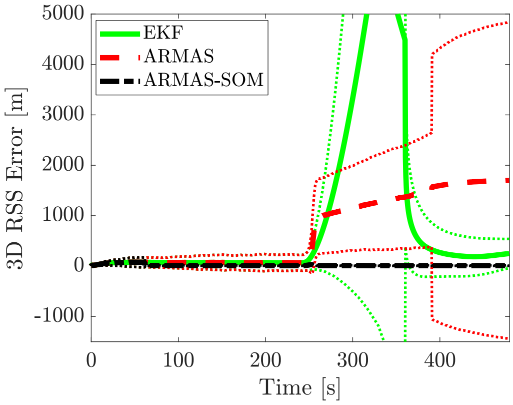


Figure 10: Scenario 1 Mean 3D RSS Error Comparison for 1000 Runs with Sensor Anomaly from $t = 240$ to 330 sec and Biased Sensor added at $t = 360$ sec

7.2 Scenario 2

In scenario 2, each aircraft is initialized with 5 trusted pseudorange sensors. A summary of the results is shown in Table 2. ARMAS-SOM outperforms both ARMAS and the EKF. Figure 12 shows the improvement in estimation performance achieved by ARMAS-SOM sensor augmentation at approximately $t = 70$ sec. This is also visible at approximately $t = 70$ sec

5 Trusted SVs, 1000 Trials					
Aircraft	**Grand Mean [m] (Median)**	**Std. Dev. [m]**	**Mean GPS4 Anomaly Magnitude at Exclusion [m]**	**Detection Rate for 40 meter Bias**	**Mean Total Added Sensors**
EKF	283.5 (36.42)	411.3	–	–	–
ARMAS	39.07 (22.44)	34.22	40.36	0.58	–
SOM	12.14 (9.48)	7.25	37.93	1.00	2.00

Table 2: Scenario 2 Results

with the estimated standard error estimates in Figure 11. Once the spoofed sensor is excluded, ARMAS-SOM requests one additional untrusted sensor to stabilize the observability layer prior to validation of the unstrusted biased sensor. This results in a total of 2 added pseudorange sensors for a total of 7. With ARMAS, the exclusion of the spoofed sensor results in 4 pseudorange sensors which is sufficient to provide a stable 3D position estimate with clock bias estimation at the main filter level. Figure 10 shows the mean RSS error and standard deviation for 1000 monte carlo trials. It is clear that ARMAS maintains a stable solution but has difficulty with proper validation of the untrusted biased sensor with a detection rate of 0.58 and is outperformed by ARMAS-SOM with a detection rate of 0.99. The EKF simply trusts all information provided and the navigation solution is exploited by the biased information.

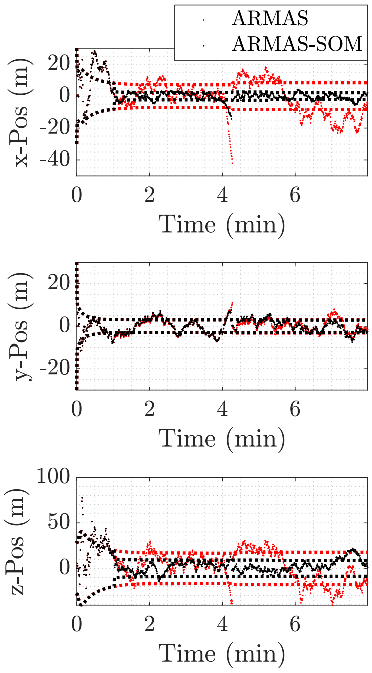


Figure 11: Scenario 2 State Estimation Error for 1 Run with Linear Pseudorange Ramp Bias from $t = 240$ to 330 sec and Biased Sensor added at $t = 360$ sec

6 Trusted SVs, 1000 Trials					
Aircraft	**Grand Mean [m] (Median)**	**Std. Dev. [m]**	**Mean GPS4 Anomaly Magnitude at Exclusion [m]**	**Detection Rate for 40 meter Bias**	**Mean Total Added Sensors**
EKF	153.7 (25.11)	213.6	–	–	–
ARMAS	17.73 (12.94)	9.00	38.44	0.85	–
SOM	12.34 (11.76)	3.97	38.44	1.00	1.00

Table 3: Scenario 3 Results

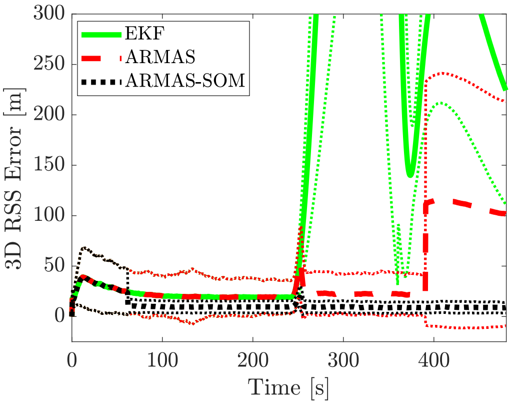


Figure 12: Scenario 2 Mean 3D RSS Error Comparison for 1000 Runs with Sensor Linear Pseudorange Ramp Bias from $t = 240$ to 330 sec and Biased Sensor added at $t = 360$ sec

7.3 Scenario 3

In scenario 3, each aircraft is initialized with 6 trusted pseudorange sensors. A summary of the results is shown in Table 3. ARMAS-SOM outperforms both ARMAS and the EKF. Figures 13 and 14 show that ARMAS and ARMAS-SOM performance is nearly identical until ARMAS-SOM augments with an additional untrusted sensor at approximately $t = 310$ sec after spoofed sensor exclusion. This results in a total of 1 added pseudorange sensor for a total of 7. With ARMAS, the exclusion of the spoofed sensor results in 5 pseudorange sensors at the main filter level which is sufficient to provide a stable 3D position estimate with clock bias estimation in each layer 1 subfilter (each containing 4 pseudorange sensors). It is clear that ARMAS maintains a stable solution but has difficulty with proper validation of the untrusted biased sensor with a detection rate of 0.85 and is outperformed by ARMAS-SOM with a detection rate of 1.00. The EKF simply trusts all information provided and the navigation solution is exploited by the biased information.

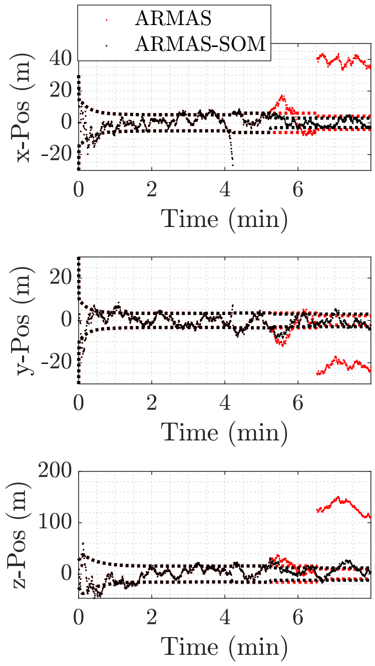


Figure 13: Scenario 3 State Estimation Error for 1 Run with Sensor Linear Pseudorange Ramp Bias from $t = 240$ to 330 sec and Biased Sensor added at $t = 360$ sec

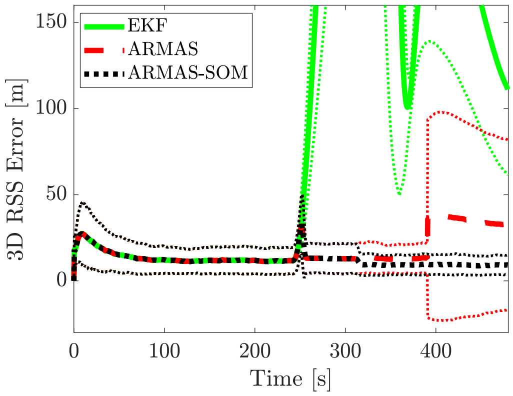


Figure 14: Scenario 3 Mean 3D RSS Error Comparison for 1000 Runs with Sensor Linear Pseudorange Ramp Bias from $t = 240$ to 330 sec and Biased Sensor added at $t = 360$ sec

7.4 Scenario 4

In scenario 4, each aircraft is initialized with 7 trusted pseudorange sensors. A summary of the results is shown in Table 4. ARMAS-SOM performs identically to ARMAS and both outperform the EKF. Figures 15 and 16 show that ARMAS and ARMAS-SOM performance is nearly identical. ARMAS-SOM does not request any sensor augmentation for a total of 7 pseudorange sensors. The EKF simply trusts all information provided and the navigation solution is exploited by the biased information.

7 Trusted SVs, 1000 Trials					
Aircraft	**Grand Mean [m] (Median)**	**Std. Dev. [m]**	**Mean GPS4 Anomaly Magnitude at Exclusion [m]**	**Detection Rate for 40 meter Bias**	**Mean Total Added Sensors**
EKF	85.52 (20.5)	113.4	–	–	–
ARMAS	11.04 (10.08)	2.55	36.75	0.99	–
SOM	11.04 (10.08)	2.55	36.75	0.99	0.00

Table 4: Scenario 4 Results

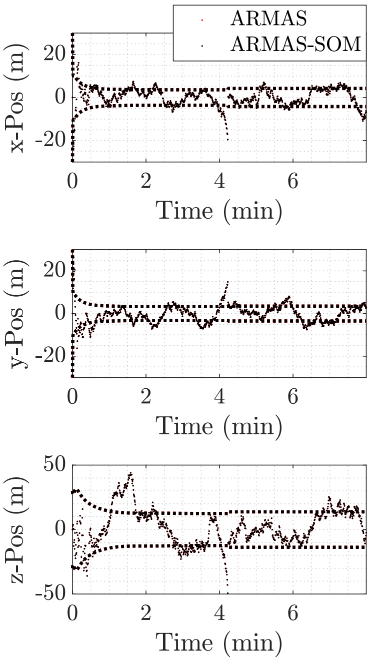


Figure 15: Scenario 4 State Estimation Error for 1 Run with Sensor Linear Pseudorange Ramp Bias from $t = 240$ to 330 sec and Biased Sensor added at $t = 360$ sec

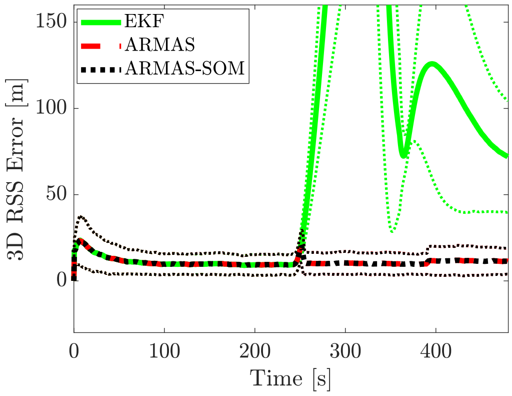


Figure 16: Scenario 4 Mean 3D RSS Error Comparison for 1000 Runs with Sensor Linear Pseudorange Ramp Bias from $t = 240$ to 330 sec and Biased Sensor added at $t = 360$ sec

7.5 Across Scenarios

Figure 17 shows a summary of detection rates for each of the four 1000-run scenarios. Clearly, ARMAS validation consistency is a function of stable observability in the FDE (layer 1) subfilters. In order guarantee resilience to a single simultaneous sensor failure, stable observability must be established at the Observability (layer 2) subfilters.

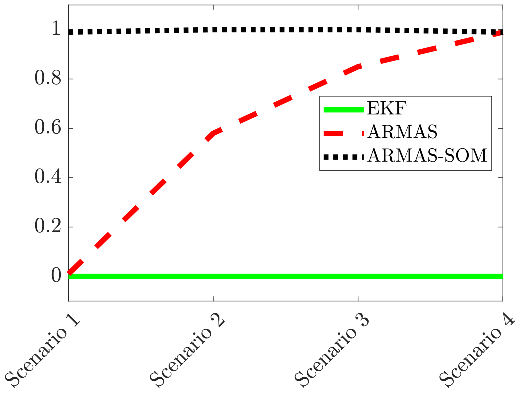


Figure 17: Summary of Detection Rates for 40 meter Biased Pseudorange Sensor

8 Conclusion

This paper addresses a critical vulnerability of ARMAS and provides a convenient method to monitor real-time navigation resilience and eliminate subfilter respawn downtime in the event of a sensor failure. This method presents a novel "observability bank" operating in a layer of I choose 2 subfilters. These additions to ARMAS provide real-time stable observability analysis via monitoring of the layer 2 subfilter aposteriori covariance matrices.

The ARMAS framework was originally developed with linear 2D position and velocity sensors which provided fully overlapping position observability. Initial analysis of ARMAS with GNSS pseudorange data from a sUAS flight test at Camp Atterbury, IN showed that ARMAS operations can become inconsistent if the FDE layer subfilters lose overlapping position estimation observability. SOM monitors each layer 2 subfilter for both observability and stabilizability.

To maintain resilience to a single simultaneous sensor failure, we must assume that a single sensor may fail at any time. Since the "observability" bank contains a subset of subfilters which will form the new FDE layer after a sensor exclusion, the observable and stabilizable properties guaranteed by SOM are inherited by the newly formed FDE layer. Furthermore, SOM provides the user with a timely warning to augment with additional sensor data and provides a notification when the augmented sensor information is sufficient for resilience to a single simultaneous sensor failure. A Monte Carlo analysis of four example scenarios proves that a loss of overlapping position observability in the FDE layer can result in an inability to exclude a failed sensor and inadvertent validation of a corrupted sensor, resulting in undetected corruption of the main navigation solution. SOM is shown to guarantee ARMAS framework resilience to a single simultaneous sensor failure and is proven by the preservation of the ARMAS FDE and validation processes.

References

[1] Andy Appleget, Robert C. Leishman, and Jonathon S. Gipson. "Evaluation of Sensor-Agnostic All-Source Residual Monitoring for Navigation". In: *IEEE/ION PLANS* (2021).

[2] Kevin M. Brink. "Partial-update Schmidt-Kalman filter". In: *Journal of Guidance, Control, and Dynamics* 40.9 (2017), pp. 2214–2228. ISSN: 15333884. DOI: 10.2514/1.G002808. URL: https://doi.org/10.2514/1.G002808.

[3] Zhe Chen. "Local Observability and Its Application to Multiple Measurement Estimation". In: *IEEE Transactions on Industrial Electronics* 38.6 (1991), pp. 491–496. ISSN: 15579948. DOI: 10.1109/41.107106.

[4] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. "The mahalanobis distance". In: *Chemometrics and intelligent laboratory systems* 50.1 (2000), pp. 1–18.

[5] Khalid Abd El Mageed Hag Elamin and Mirghani Fath Elrahman Taha. "On the steady-state error covariance matrix of Kalman filtering with intermittent observations in the presence of correlated noises at the same time". In: *Proceedings - 2013 International Conference on Computer, Electrical and Electronics Engineering: 'Research Makes a Difference', ICCEEE 2013* (2013), pp. 15–22. DOI: 10.1109/ICCEEE.2013.6633901.

[6] Fredric M Ham. "Observability, Eigenvalues, and Kalman Filtering". In: *Ieee Transactions On Aerospace And Electronic Systems* 2 (1983), pp. 269–273.

[7] Robert Hermann and Arthur J. Krener. "Nonlinear Controllability and Observability". In: *IEEE Transactions on Automatic Control* 22.5 (1977), pp. 728–740. ISSN: 15582523. DOI: 10.1109/TAC.1977.1101601.

[8] Sinpyo Hong et al. "Observability measures and their application to GPS/INS". In: *IEEE Transactions on Vehicular Technology* 57.1 (2008), pp. 97–106. ISSN: 00189545. DOI: 10.1109/TVT.2007.905610.

[9] Juan Jurado, John F. Raquet, and Christine M. Schubert-Kabban. "Autonomous and Resilient Management of All-source sensors for Navigation". In: *ION* (2018).

[10] Juan Jurado, John Raquet, and Christine M Schubert Kabban. "Single-Filter Finite Fault Detection and Exclusion Methodology for Real-time Validation of Plug-and-play Sensors". In: *IEEE Transactions on Aerospace and Electronic Systems* (2020).

[11] Juan Jurado et al. "Residual-Based Multi-Filter Methodology for All-Source Fault Detection, Exclusion, and Performance Monitoring". In: *ION* (2019).

[12] Kyle Kauffman et al. "Scorpion: A Modular Sensor Fusion Approach for Complementary Navigation Sensors". In: *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. Portland, OR, Apr. 2020.

[13] J. P. LeCadre. "Properties of Estimability Criteria for Target Motion Analysis". In: 145.2 (1998).

[14] Maodeng Li, Dayi Wang, and Xiangyu Huang. "Study on the observability analysis based on the trace of error covariance matrix for spacecraft autonomous navigation". In: *IEEE International Conference on Control and Automation, ICCA* (2013), pp. 95–98. ISSN: 19483449. DOI: 10.1109/ICCA.2013.6565112.

[15] Wei Li, Zhen Yang, and Haifeng Hu. "A new variant of sum-product algorithm for sensor self-localization in wireless networks". In: *2013 IEEE/CIC International Conference on Communications in China, ICCC 2013* Iccc (2013), pp. 628–632. DOI: 10.1109/ICCChina.2013.6671189.

[16] Peter S. Maybeck. *Stochastic Models, Estimation, and Control Volume 1*. Virginia: Navtech, 1982.

[17] Nathan D. Powel and Kristi A. Morgansen. "Empirical Observability Gramian for Stochastic Observability of Nonlinear Systems". In: *arXiv* Cdc (2020), pp. 6342–6348. ISSN: 23318422. arXiv: 2006.07451.

[18] John D. Quartararo and Steven E. Langel. "Detecting slowly accumulating faults using a bank of cumulative innovations monitors in kalman filters". In: *Proceedings of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2020* 20 (2020), pp. 2367–2381. DOI: 10.33012/2020.17655.

[19] Mathieu Rouaud. "Probability, statistics and estimation". In: *Propagation of uncertainties* (2013), p. 191. URL: http://www.incertitudes.fr/book.pdf.

[20] Prateep Roy, Arben Çela, and Yskandar Hamam. "On the relation of FIM and controllability gramian". In: *Proceedings - 2009 IEEE International Symposium on Industrial Embedded Systems, SIES 2009* (2009), pp. 37–41. DOI: `10.1109/SIES.2009.5196189`.

[21] Yonggang Tang et al. "INS/GPS integration: Global observability analysis". In: *IEEE Transactions on Vehicular Technology* 58.3 (2009), pp. 1129–1142. ISSN: 00189545. DOI: `10.1109/TVT.2008.926213`.