Resilient Collaborative All-source Navigation

1st Jonathon S. Gipson
Autonomy and Navigation Technology Center
U.S. Air Force Institute of Technology
Wright-Patterson AFB, OH
jonathon.gipson@afit.edu

2nd Robert C. Leishman
Director, Autonomy and Navigation Technology Center
U.S. Air Force Institute of Technology
Wright-Patterson AFB, OH
robert.leishman@afit.edu

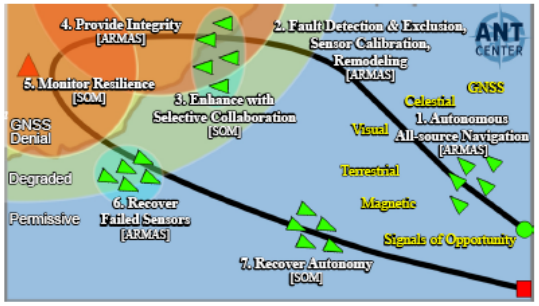

Fig. 1. The ARMAS-SOM framework provides intrinsic all-source resilience "awareness" via novel selective offboard collaboration which improves navigation integrity and facilitates graceful recovery to autonomy.

***Abstract*—The Autonomous and Resilient Management of All-source Sensors with Stable Observability Monitoring (ARMAS-SOM) framework fuses collaborative all-source sensor information in a resilient manner with fault detection, exclusion, and integrity solutions recognizable to a Global Navigation Satellite System (GNSS) user. This framework uses a multi-filter residual monitoring approach for fault detection and exclusion which is augmented with an additional "observability" Extended Kalman Filter (EKF) sub-layer for resilience. We monitor the *a posteriori* state covariances in this sub-layer to provide intrinsic awareness when navigation state observability assumptions required for integrity are in danger. The framework leverages this to selectively augment with offboard information and preserve resilience. By maintaining split parallel *collaborative* and *proprioceptive* frameworks and employing a novel "stingy collaboration" technique, we are able maximize efficient use of network resources, limit the propagation of unknown corruption to a single donor, prioritize high fidelity donors, and maintain consistent collaborative navigation without fear of double-counting in a scalable processing footprint. Lastly, we preserve the ability to return to autonomy and are able to use the same intrinsic awareness to notify the user when it is safe to do so.**

***Index Terms*—collaborative, distributed, all-source, navigation, resilient, recovery**

I. Introduction

Due to widespread awareness of Global Navigation Satellite System (GNSS) vulnerabilities, there is increased interest in alternative navigation technologies such as Visual [1], Signals of Opportunity [2], Magnetic [3], and others [4][5][6]. The fusion of these information sources for localization is known as *all-source navigation* [7]. Since all-source sensors typically reside on individual vehicles connected in a distributed wireless network, consistent collaborative fusion of this decentralized information is highly desired.

A primary navigation user requirement is a timely and accurate estimate of navigation error to include impending degradation known as *navigation integrity* [8] which is currently provided through GNSS techniques such as Receiver Autonomous Integrity Monitoring (RAIM) [9]. With GNSS navigation, simultaneously redundant, synchronous measurements with validated measurement models are readily available. With all-source navigation, none of these are guaranteed. In 2018, Autonomous and Resilient Management of All-source Sensors (ARMAS) [10] was introduced to provide a generalized framework to fuse all-source sensor information which leveraged SCORPION pluggable Bayesian estimators to gracefully manage multiple concurrent state estimation subfilters [11]. In 2019, the monitoring mode of the ARMAS framework was redesigned to provide a means for all-source integrity, known as Sensor-Agnostic All-source Residual Monitoring (SAARM) [8].

The following section provides a brief history of all-source navigation to provide context for our research. We continue with brief overviews of classical model estimation, residual monitoring and applicability to multi-sensor Fault Detection and Exclusion (FDE). The background section ends with a brief explanation of estimator credibility. These concepts are fundamental to the ARMAS framework [10] which is augmented by Stable Observability Monitoring (SOM) [12]. The next section describes methods for managing cross-correlation and introduces split *collaborative* and *proprioceptive* ARMAS-SOM modules which enable consistent collaborative augmentation with the option to gracefully recover autonomy. These improvements are demonstrated in a 3D all-source air vehicle navigation environment which employs collaborative estimator credibility analysis, proportion of resilient time, and navigation error improvement.

II. Background

A. Recursive Bayesian Estimation

Small cumulative measurement errors form an integration bias or "drift", the primary error source for an Inertial Navigation System (INS). Some form of recursive model estimation, like an EKF, is often used to maintain an INS error model. This

strategy incorporates trusted external GNSS and/or geodetic measurements to periodically correct integration bias. Navigation is based on accurate estimation of a vehicle's system states. This is accomplished via a task called model estimation.

Modern navigation, based on recursive model estimation, can trace its roots to the Kalman Filter (KF) algorithm [13], presented in 1960. In a Kalman filter, the system states, $\mathbf{x}$, are estimated recursively in real-time by propagating a state estimate, $\hat{\mathbf{x}}$, and associated state covariance, $\mathbf{P}$. The process (dynamics) model is given by

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{G}\mathbf{w}(t), \tag{1}$$

with

$$E[\mathbf{w}(t)] = 0, \tag{2}$$

$$E[\mathbf{w}(t)\mathbf{w}(t)^T(t+\tau] = \mathbf{Q}\delta(\tau), \tag{3}$$

where $\mathbf{x}$ is the system state vector, $\mathbf{u}$ is the system input control vector, and $\mathbf{w}$ is a vector of white noise components. $\mathbf{F}$, $\mathbf{B}$, and $\mathbf{G}$ are linear operator matrices for the state vector, control input vector, and noise vector, respectively.

In 1978, the Van Loan method was introduced [14] to provide a method to discretize 1 and 3 for use in digital, time-sampled computer systems. The resulting discrete process noise strength matrix, $\mathbf{Q}_d$, discrete control input matrix, $\mathbf{B}_d$, and discrete state transition matrix, $\mathbf{\Phi}$, are calculated by linearizing the system about a single time sample, Δt.

Assuming state estimate observability, measurements, $\mathbf{z}$, and associated measurement covariances, $\mathbf{R}$, are used to perform a state estimate update. The measurements are mapped to the states by the observation model, $\mathbf{H}$. The discrete measurement model is given by

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{4}$$

where

$$E[\mathbf{v}_k] = 0, \tag{5}$$

$$E[\mathbf{v}_k\mathbf{v}_{k+\Delta t}^T] = \mathbf{R}\delta_{k,k+\Delta t}, \tag{6}$$

and $\mathbf{z}$ is the sensor measurement vector. The discrete Kalman filter is initialized with state estimates, $\mathbf{x}_0$, and associated initial state covariance $\mathbf{P}_0$. The initial state estimate of $\hat{\mathbf{x}}$ is propagated in the discrete Kalman filter using

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{\Phi}\hat{\mathbf{x}}_k^+ + \mathbf{B}_d\mathbf{u}_k, \tag{7}$$

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}\mathbf{P}_k^+\mathbf{\Phi}^T + \mathbf{Q}_d. \tag{8}$$

This update is performed by optimally combining the stochastic components of the state estimate and of the measurements via the Kalman gain, K, using

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^T[\mathbf{H}\mathbf{P}_k\mathbf{H}^T + \mathbf{R}]^{-1}, \tag{9}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-], \tag{10}$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-. \tag{11}$$

With properly modeled state-transition and measurement models, the result is optimal stochastic estimation of the vehicle states. The state-transition model makes use of mathematical relationships between system states to provide additional observability to states which cannot be directly measured. Recursive estimation is the primary means of model estimation used by this research to approach the all-source navigation problem.

B. Residual Monitoring

This brief overview of residual monitoring based on a likelihood function for a Kalman filter or EKF is based on a more thorough introduction [15]. Residual monitoring is a useful technique for sensor fault detection. The goal of this section is to set the stage for a later explanation of the SAARM algorithm implemented by the ARMAS framework. After a sensor measurement $\mathbf{z}$ is obtained, it is possible to extract a set of pre-update KF residuals, $\mathbf{r}$, at time k, resulting in

$$\mathbf{r}_k = \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-. \tag{12}$$

If we apply Gaussian white-noise assumptions, the expected statistical distribution of these residuals is

$$\mathbf{r}_k \hookrightarrow \mathcal{N}\left(\mathbf{0}, \mathbf{H}\mathbf{P}_k^-\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right), \tag{13}$$

which results in a Gaussian-Normal probability density function given by

$$p(\mathbf{r}_k|\mathbf{\Sigma}_k) = \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}_k|}} e^{-\frac{1}{2}(\mathbf{r}_k)^{\mathrm{T}}\mathbf{\Sigma}_k^{-1}(\mathbf{r}_k)} \text{ and} \tag{14}$$

$$\mathbf{\Sigma}_k = \mathbf{H}\mathbf{P}_k^-\mathbf{H}^{\mathrm{T}} + \mathbf{R}. \tag{15}$$

If we assume a set of N residuals have been collected preceding t_k, the resulting likelihood function is given by

$$L_{N_k} = \prod_{j=k-N+1}^{k} \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}_j|}} e^{-\frac{1}{2}(\mathbf{r}_j)^{\mathrm{T}}\mathbf{\Sigma}_j^{-1}(\mathbf{r}_j)}. \tag{16}$$

In lieu of differentiation, this expression can be simplified by taking advantage of the monotonically increasing natural logarithm function, resulting in the log-likelihood function given by

$$\mathcal{L}_{N_k} = \sum_{i=k-N+1}^{k} -\log(|2\pi\mathbf{\Sigma}_i|) - \frac{1}{2}\mathbf{r}_i^{\mathrm{T}}\mathbf{\Sigma}_i^{-1}\mathbf{r}_i. \tag{17}$$

If we examine a residual vector $\mathbf{r}_k$ containing the N most recent residuals, the log-likelihood function acts as a moving window to compile the cumulative statistical likelihood for this set. A predefined threshold can be used to determine if the set of residuals falls outside the expected distribution which would trigger a failure detection. This forms the basis for the fault detection and exclusion algorithm espoused by ARMAS-SOM.

C. Autonomous Resilient Management of All-source Sensors (ARMAS)

Introduced in 2018, ARMAS provides a generalized framework for real-time management of heterogeneous, asynchronous all-source sensors [10]. This framework is resilient to faulty sensors and combines sensor validation, FDE, recalibration, and remodeling modes in a single architecture. ARMAS

employs a set of pluggable EKF estimators to address the following nonlinear navigation problem:

$$\dot{\mathbf{x}}(t) = \boldsymbol{f}[\mathbf{x}(t), \boldsymbol{\epsilon}(t), \boldsymbol{u}(t), t] + \boldsymbol{G}(t)\boldsymbol{w}(t) \tag{18}$$

where $\mathbf{x}$ is a $N \times 1$ state vector of a vehicle's position, velocity, and attitude. The measurement error states vector $\boldsymbol{\epsilon}$ 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, $\mathbf{Q}$.

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. In validation mode, ARMAS uses a likelihood function described by (17) to monitor the statistical distribution of a user-defined monitoring period made up of recent Kalman pre-update residuals. A Chi-square, χ^*, test statistic is used to detect excursions outside a user-defined threshold. Sensors in validation mode are prevented from impacting the main state estimates. Trusted sensors proceed directly to monitoring mode where 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.

Once a fault is detected and culprit sensor is identified, the sensor is no longer "trusted" and is quarantined from affecting the core navigation state estimate, $\hat{\mathbf{x}}^{[j]}$. ARMAS attempts to revalidate the sensor, but if this fails the sensor enters repair and recovery modes. Figure 2 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. SAARM resides in the 'M' monitoring mode block in Figure 2 and provides a special form of residual monitoring for FDE and a position integrity region called the Guaranteed Position Zone (GPZ). For resilience to a single faulty sensor, this approach requires the designer to maintain one differently configured navigation subfilter for each sensor. 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. 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.

D. Stable Observability Monitoring (SOM)

The previous section summarized the ARMAS framework and the SAARM algorithm used to perform fault detection, exclusion, and integrity functions. The framework requires overlapping state observability to perform each of these functions. As previously stated, the framework's basic goal is to maintain at least one subfilter which is never corrupted by a faulty sensor. This is required to maintain navigation estimation consistency and provide a method for all-source integrity. The following section describes a method to accomplish this within the monitoring mode of ARMAS. Clearly, understanding the process noise, measurement noise, and measurement model are key to assessing the observability of a system state variable. Introduced in 2021, SOM provides a measure of state estimate observability in the form of a scalar and is used to directly quantify the observability of a state-variable. SOM augments SAARM and resides in the 'M' monitoring mode block in Figure 2. Since the ARMAS framework assumes state observability for both FDE and integrity, it is critical that a method for monitoring real-time position state observability is implemented.

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

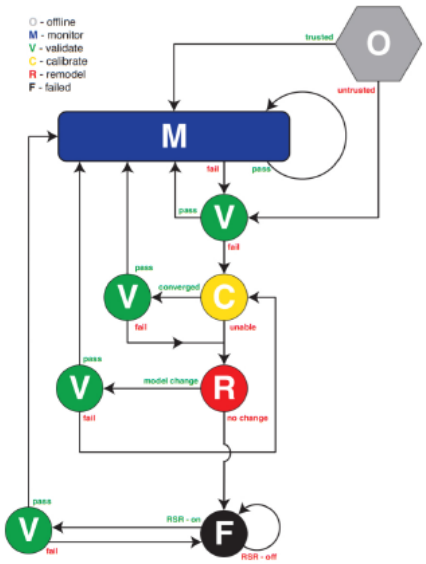

Fig. 2. ARMAS Framework State Diagram [10]

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

Fig. 3. ARMAS Monitoring Mode with User Output Layer 0, Fault Detection and Exclusion Sub-Layer 1, and Observability Sub-Layer 2 for Stable Observability Monitoring (SOM)

observability at the layer 1 subfilter level, an additional layer of subfilters is required (Figure 3). 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. 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 [8]. SOM allows the following extension of the previously stated fault exclusion axiom:

> *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* [8]. ***If the states of interest in each layer 2 subfilter are observable and stabilizable, then each layer 1 subfilter inherits these properties.***

Assuming no more than one simultaneous sensor failure, 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.

SOM records and monitors 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

$$O_{k,pos}(n) = \begin{cases} 1, & \text{if } tr\left(\mathbf{P}^{[n]}_{pos}(t_k^+)\right) > tr\left(\mathbf{P}^{[n]}_{pos}(t_{k-M}^+)\right)\beta \\ 1, & \text{if } tr\left(\mathbf{P}^{[n]}_{pos}(t_k^+)\right) > tr\left(\mathbf{P}_{pos,max}\right) \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

where $\mathbf{P}^{[n]}_{pos}(t_k^+)$ is the most recent post-update position covariance matrix for layer 2 subfilter n, $\mathbf{P}^{[n]}_{pos}(t_{k-M}^+)$ 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 $tr\left(\mathbf{P}^{[n]}_{pos}(t_k^+)\right)$ converges, then the individual position estimate variances also converge. When applying (19), it is important to ensure units are identical across the grouped states.

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 (20). The newly added sensor will directly enter monitoring mode if it is considered 'trusted' or must pass through sensor validation if 'untrusted'.

$$SOM_{Collaborate} = \begin{cases} true, & \text{if } \sum_{n=1}^{N} O_k(n) > 0 \\ false, & \text{otherwise} \end{cases} \tag{20}$$

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 (20) are ignored until the newly requested sensor completes an ARMAS monitoring epoch. If post update variance stability is regained, then (20) 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. Due to potentially variable navigation state information available from all-source navigation sensors, it is critical that ARMAS is always "aware" of when it lacks enough information to maintain consistency.

E. Estimator Credibility Analysis

Estimator performance is always dependent on available measurements [15]. Accordingly, estimators should be compared using the same data set which is sufficiently large to provide statistically significant results. At first glance, position mean-squared error (MSE) appears to be a simple, practical performance metric to compare different estimators. For n total measurements, the bias $\tilde{\boldsymbol{x}}$ of state estimate $\hat{\boldsymbol{x}}$ at time i is shown by (21). The MSE of $\tilde{\boldsymbol{x}}$ is shown by (22).

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \hat{\boldsymbol{x}}_i \tag{21}$$

$$MSE = \frac{1}{n}\sum_{i}^{n} \tilde{\boldsymbol{x}}_i{}^2 \tag{22}$$

Although easily understandable, this definition of MSE only provides information about the first moment of estimation error. Estimators also maintain a self-assessment of the second moment of estimation error, known as the error covariance $\boldsymbol{P}$. The Normalized Estimation Error Squared (NEES) metric, $\boldsymbol{\epsilon}_i$, is Chi-Square distributed, assumes Gaussian errors, and provides a quantitative assessment of the estimator's credibility from pessimistic to optimistic [16]. NEES is calculated according to Eq. 23.

$$\boldsymbol{\epsilon}_i = (\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i)^T \boldsymbol{P}_i^{-1} (\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i) \tag{23}$$

For an m-dimensional state estimate with n total measurements, the Average NEES is shown by Eq. 24.

$$\bar{\epsilon} = \frac{1}{nm}\sum_{i=1}^{n} \boldsymbol{\epsilon}_i \tag{24}$$

NEES represents the square of the distance between the state estimate $\hat{\boldsymbol{x}}$ and the truth $\boldsymbol{x}$, normalized by the error covariance $\boldsymbol{P}$. Average NEES can be interpreted as a uni-dimensional average of this squared distance. Average NEES quantifies estimator credibility by assessing the accuracy of employed assumptions [16]. If the Average NEES is near 1, then the estimator is likely credible. If the Average NEES is much less (greater) than 1, it is pessimistic (optimistic).

III. Methods for Managing Cross-Correlated Information

We must be careful because fusing collaborative sensor data can result in dependency between state estimates and measurements if a relative measurement is later re-used without taking historical dependencies into account. This problem is called double-counting [17]. Distributed networks are scalable and resilient to changes in topology but are particularly vulnerable to this problem. If dependencies are not properly managed, the estimator can become inconsistent by overestimating available sensor information and eventually diverge. For the purposes of this discussion, we define a collaborative exchange as a transaction between a requesting recipient and a cooperative donor.

A rudimentary fix for this problem involves increasing Bayesian estimator process noise to account for unmodelled dependencies between measurements and the state estimate. This strategy leads to increased state estimate covariance and degradation in overall estimator performance. The following section analyzes several methods to properly account for dependencies when fusing measurements: Covariance Intersection, Interleaved Update Algorithm, and novel Split Approach.

1) Covariance Intersection Method: The limitations of managing unknown cross-correlation between state estimates and measurements in decentralized estimation with arbitrary network topologies are widely known. This motivated the CI method [17][18] which fuses the probability densities of two measurements with unknown cross-correlation into a consistent covariance estimate. The CI technique is recommended as a good estimator for highly cross-correlated measurements. The CI method utilizes a conservative estimate of P_{ab} to guarantee estimator consistency. Although the CI method is highly useful and rather ubiquitous, a primary downside for collaborative all-source navigation is the inability to recover previously fused information in the event of a sensor corruption discovered *a posteriori*.

2) Interleaved Update (IU) Algorithm: The IU algorithm [19] provides a multi-filter approach to the inconsistency problem. This algorithm retains consistency by maintaining separable state estimates $\mathbf{X}_i$ and covariance estimates $\mathbf{P}_i$ for collaborative offboard measurements. For example, the covariance matrices associated with vehicles i, j are assembled in a block diagonal matrix $\mathbf{P}_{i,j}(t)$ as shown in (25).

$$\mathbf{P}_{i,j}(t) = \left[\begin{array}{c|c} \mathbf{P}_i(t) & 0 \\ \hline 0 & \mathbf{P}_j(t) \end{array}\right] \tag{25}$$

The timestamp of the most recent measurement update from each vehicle is maintained in a $\mathbf{T}_i$ matrix where each row q represents a filter and each column i represents an offboard vehicle number. Row 1 in $\mathbf{T}_i$ corresponds to the set of states which has never been updated by an offboard range measurement. All filters are always updated by onboard measurements.

Each time a collaborative measurement is broadcast from vehicle i to another vehicle, it contains current copies of $\mathbf{X}_i$, $\mathbf{P}_i$, and $\mathbf{T}_i$. With these parameters, the receiving vehicle j always has access to a state estimate and covariance which have only been updated by sensors on the transmitting vehicle i. This uncorrelated information can be used by vehicle j without inconsistency.

A vehicle i receives a broadcast from vehicle j containing information $\mathbf{X}_j$, $\mathbf{P}_j$, and $\mathbf{T}_j$ to update filter q onboard vehicle i. If information from vehicles other than j is present, the optimal combination of filter information is selected for fusion by examining the trace of the fused covariance estimate $\mathbf{P}_i^q$. The combination of filter information that results in the $argmin\left[trace(\mathbf{P}_i^q)\right]$ is used to update filter q onboard vehicle i.

The IU algorithm requires each vehicle to maintain 2^n simultaneous Kalman filters. The author speculated that 30 unique vehicle IDs are attainable with this framework [19]. Exponential scalability is highly undesirable, especially for multi-filter residual monitoring algorithms like ARMAS [20]. Even so, the idea of retaining a proprioceptive solution uncorrupted by offboard collaboration is a great insurance policy which enables eventual recovery to autonomy from collaboration.

3) Split Approach (aka Stingy Collaboration): If each vehicle maintains two parallel instances of ARMAS-SOM, one collaborative and the other proprioceptive, then we can completely eliminate double-counting and simultaneously mitigate the possibility of propagating corrupt donor data across the network.

For example, a single wingman in a 5-vehicle collaborative network unknowingly broadcasts corrupted proprioceptive donor information to all vehicles (Figure 4). Since each wingman's collaborative estimation framework is propagating a subfilter "No Wingman 3" which excludes any measurements from Wingman 3, a single FDE event is required to remove all historical contributions from Wingman 3. Throughout this process, Wingman 3 is able to receive consistent proprioceptive donor information from the other networked vehicles which facilitates FDE actions onboard Wingman 3 to recover from the fault.

We claim that the ARMAS-SOM framework will detect the faulty donor and exclude them from the donor's navigation solution. In other words, a single fault detection and exclusion action is required by the recipient to eliminate all inconsistencies resulting from the donor. Although this approach does not glean the maximum quantity of information from the network, it significantly maximizes efficent use of network resources, lowers the risk of repeatedly propagating inconsistent information, and provides a mechanism to eliminate the faulty information completely, which is desirable for integrity. Lastly, this approach can be used to prioritize high-fidelty donors over less desirable ones.

If more than one donor vehicle is corrupted (i.e. we violate the single simultaneous failure assumption), the recipient vehicle can revert to the proprioceptive instance of ARMAS-SOM which completely eliminates any possibility of offboard corruption. This is particularly important because ARMAS-SOM must always maintain at least one uncorrupted subfilter to provide integrity. If the donors shared collaborative solutions which contained contributions from other corrupted

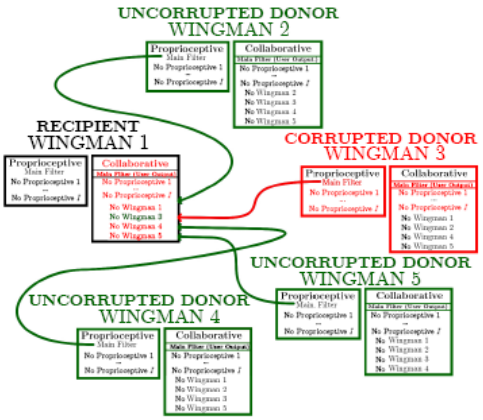

Fig. 4. Stingy Collaboration Example: Wingman 1 receives corrupted collaborative donor information from Wingman 3. A single fault detection and exclusion action is required to remove all historical contributions from Wingman 3.

donor vehicles, an exponentially scaled approach like the IU Algorithm above would be required to back-track and sort out how to fuse the information without loss of consistency. This split approach is useful as an insurance policy to regain autonomous navigation by eliminating all contributions from other vehicles.

We offer the following guidelines for collaborative all-source navigation:

- The key to integrity is the ability to maintain at least one consistent estimator.
- Inconsistency can result from on-board corruption, off-board corruption, or by double-counting off-board information.
- The distributed architecture used to fuse collaborative information should be scalable and allow *a posteriori* consistency recovery.
- A proprioceptive filter is a great insurance policy.

IV. Simulation

Consider a three-dimensional example in a local tangential (navigation) frame with five independent distributed "wingmen" air vehicles at different formation location, each operating the ARMAS-SOM framework. Each vehicle maintains split, parallel autonomous (proprioceptive) and collaborative instances of ARMAS-SOM propagating 10-state EKFs for position, velocity, acceleration, and GNSS clock bias states. The kinematics block is defined by

$$\dot{\mathrm{x}}(t) = \begin{bmatrix} \dot{\mathrm{x}}_p(t) \\ \dot{\mathrm{x}}_v(t) \\ \dot{\mathrm{x}}_a(t) \end{bmatrix} = \begin{bmatrix} \dot{\mathrm{x}}_v(t) \\ \dot{\mathrm{x}}_a(t) \\ -\frac{1}{\tau_a}\dot{\mathrm{x}}_v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathrm{w}(t) \end{bmatrix} \tag{26}$$

where x_p is the vehicle's position (m), x_v is the vehicle's velocity (m/s), x_a is the vehicle's acceleration (m/s^2), and $\tau_a = 300$ 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[w(t) w(t+τ)^T] = Q$\delta(\tau)$ and

$$\mathbf{Q} = (1.0 \times 10^{-2})^2 \mathbf{I}_{3\times 3}\ (m^2/s^4) \tag{27}$$

Each vehicle is initialized with unique initial positions according to

$$\hat{\mathrm{x}}_1(0) = \begin{bmatrix} 0 & 0 & 3000 \end{bmatrix}^T \text{ meters,}$$
$$\hat{\mathrm{x}}_2(0) = \begin{bmatrix} -10000 & 3000 & 5000 \end{bmatrix}^T \text{ meters,}$$
$$\hat{\mathrm{x}}_3(0) = \begin{bmatrix} 10000 & 3000 & 5000 \end{bmatrix}^T \text{ meters,}$$
$$\hat{\mathrm{x}}_4(0) = \begin{bmatrix} -10000 & -3000 & 2000 \end{bmatrix}^T \text{ meters,}$$
$$\hat{\mathrm{x}}_5(0) = \begin{bmatrix} 10000 & -3000 & 2000 \end{bmatrix}^T \text{ meters.}$$

Each vehicle is initialized with identical state error covariance matrices according to

$$\mathbf{P}_n(0) = diag\left(\begin{bmatrix} 20^2 & 20^2 & 20^2 & 10^2 & 10^2 & 10^2 & 0.01^2 & 0.01^2 & 0.01^2 & 8000^2 \end{bmatrix}\right). \tag{28}$$

Each aircraft receives discrete measurements from a constellation of 5 stationary satellite vehicles (SV). The constellation is uniformly distributed in azimuth and simulates favorable coverage with elevation angles between approximately 45 degrees and 63.4 degrees. Each vehicle is equipped with dual-frequency GNSS receivers receiving simulated pseudorange measurements from 5xL1 (1575 MHz), and 5xL2 (1227 MHz) SVs.

Individual pseudorange measurements are performed according to (29)

$$\rho_i = \sqrt{(X_{SV,i} - X_u)^2 + (Y_{SV,i} - Y_u)^2 + (Z_{SV,i} - Z_u)^2} + b_u \tag{29}$$

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 is $\mathbf{R_{SV}} = 10^2$ m^2. A receiver clock bias b_u is independently estimated as an additional state in each EKF. Each vehicle is able to augment with collaborative ranging measurements from nearby wingmen according to

$$R_j = \sqrt{(X_{wng,j} - X_r)^2 + (Y_{wng,j} - Y_r)^2 + (Z_{wng,j} - Z_r)^2}. \tag{30}$$

where R_j is a simulated radar range to Wingman j with actual coordinates $(X_{wng}, Y_{wng}, Z_{wng})$ and with actual recipient coordinates (X_r, Y_r, Z_r). Unlike the GNSS sensors, no ephemeris is available, so estimated recipient and wingman solutions are fused with (30) to form the measurement update. The main solution from the user's collaborative instance of ARMAS-SOM provides the estimated user location and covariance. The wingman's proprioceptive instance of ARMAS-SOM is used to provide the estimated wingman location

and associated position covariance. The ranging measurement covariance is $\mathbf{R}_{WING}$ = 10^2 m^2. Collaborative updates are performed only in the collaborative instance of ARMAS-SOM for each recipient. Clock synchronization is assumed for collaborative measurements. Vehicles are equipped with simulated alternative visual navigation sensors which provide noisy velocity updates with $\mathbf{R}_{VIS}$ = $(10^2)\mathbf{I}_{3\times3}$ (m/s)^2. All sensors provide measurement updates at 1Hz. SOM *a posteriori* state estimate covariance parameters are set to β = 5 and $\mathbf{P}_{j,max}$ = $(25\text{m})^2 \times 3 = 6889\text{m}^2$. The vehicles travel at approximately 75 meters/second in the $+Y$ direction in a LTF for a total of 27 minutes (1620 seconds) of run time.

The simulation is divided into 4 separate regions (Figure 5):

1) Permissive (0-5km)
2) L1 Jamming (5-50km)
3) L1 Jamming + L2 Spoofing (50km-100km)
4) Permissive (100km - End of Sim)

All five aircraft initialize with $5 \times L1$ trusted pseudorange sensors, $5 \times L2$ trusted pseudorange sensors, and trusted $1 \times VIS$ sensor in region 1 at $T = 0$ seconds. Once "WNG1" enters region 2 at $T = 60$ seconds, all vehicles encounter $L1$ noise jamming and pare down to $5 \times L2$ pseudorange sensors and $1 \times VIS$ sensor.

In region 2, ARMAS-SOM detects a threat to resilience and each vehicle selectively validates untrusted individual offboard collaborators until the observability warnings are rescinded. Each donor vehicle shares its proprioceptive solution according to the split "stingy collaboration" approach described in section III-3 and each recipient vehicle performs a collection period to ensure measurement residuals are valid for acceptance. Each vehicle augments with approximately 2-3 collaborative relationships for prior to entering region 3. Once "WNG1" enters region 3 at $T = 660$ seconds, all vehicles experience an insidious pseudorange bias on $L2SAT4$ which initializes at 10 meters and grows at 10 m/s.

Once each vehicle detects the growing pseudorange bias and determines the culprit, $L2SAT4$ is excluded from each ARMAS-SOM instance. In response to the $L2SAT4$ exclusion, ARMAS-SOM detects a threat to resilience and forms an additional collaborative relationship to stabilize (See Figure 6). The vehicles transition into region 4 at $T = 1320$ seconds where $L1$ jamming and $L2$ spoofing end. Once each vehicle's proprioceptive ARMAS-SOM instances re-establish resilience, then the collaborative relationships are removed and the vehicles gracefully regain autonomy.

V. Numerical Results

The results of the simulation show that ARMAR-SOM's stingy collaboration scheme stabilizes the all-source GPZ during jamming and spoofing events as shown in Figure 6. A comparison of the GPZ dimensions for both the proprioceptive and collaborative ARMAS-SOM instances is shown in Figure 7. This shows that the SOM algorithm sufficiently augments itself using collaboration to stabilize the all-source position integrity solution during the combined navigation anomalies. Since the GPZ is a visualization of the union of the FDE layer

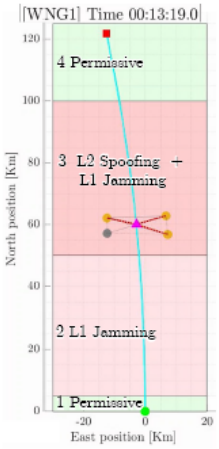

Fig. 5. Overhead view of GNSS Regions with WNG1 in center of formation at T = 13m 19s (799s). Wingman locations are: WNG2 (Upper Left), WNG3 (Upper Right), WNG4 (Lower Left), WNG5 (Lower Right)

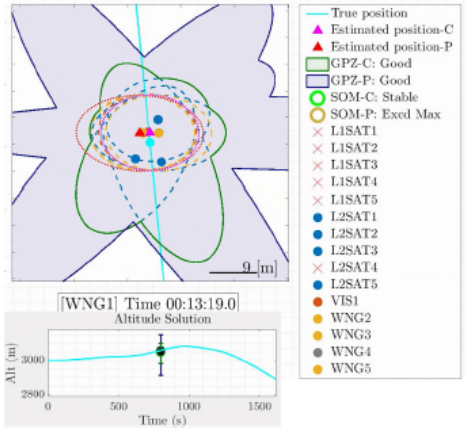

Fig. 6. Combined Overhead and Profile View of WNG1 Split Collaborative "GPZ-C" and Proprioceptive "GPZ-P" ARMAS-SOM Instances with individual Layer 1 Subfilter Solutions at T = 13m 19s (T = 799s) inside the L1 Jamming + L2 Spoofing Region 3

solutions, this is proof that observability layer stabilization is inherited by the FDE and main layer. The end result of this contribution is guaranteed resilience to a single simultaneous failure for 81% of the total simulation time versus only 5% for a non-collaborative ARMAS instance.

Figure 8 shows estimator credibility results in the form of NEES for each wingman. The mean results for 3D-RSS Error and NEES are visible in Table I. This shows that "stingy" collaboration preserves consistent estimation while mititgating the risk of undesired error propagation and avoiding the double-counting problem normally associated with unknown cross-correlation.

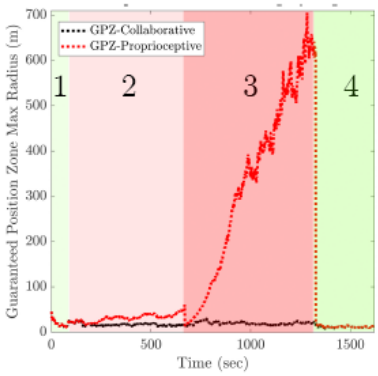

Fig. 7. Size Comparison of Split "Stingy" Collaborative and Proprioceptive All-source Guaranteed Position Zones for WNG1 in GNSS Regions 1-4

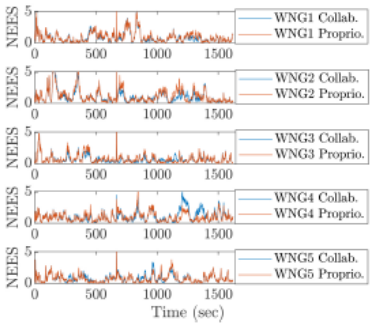

Fig. 8. Estimator Credibility Comparison for Split Collaborative and Proprioceptive Instances of ARMAS-SOM for 5 Vehicles

VI. Conclusion

This research presented the motivation for collaborative all-source navigation and covered a brief background on the fundamental concepts required to understand the resilient ARMAS-SOM framework. We explained how SOM provides intrinsic information awareness for timely selective collaboration to preserve estimator consistency with the assumption of a single simultaneous sensor failure. We introduced a novel split structure which maintains both "stingy" collaborative and proprioceptive frameworks to maximize efficient use of network resources, limit the propagation of unknown corruption to a single donor, prioritize high fidelity donors, and eliminate double counting normally associated with recursive Bayesian fusion of collaborative information. We also demonstrated how the same intrinsic information awareness used by SOM can be used to determine when it is safe to transition back to autonomy from collaboration. This structure demonstrated effective stabilization of the all-source GPZ integrity solution, a 11% reduction in RSS error, preservation of consistent estimation, and the ability to gracefully recover autonomy. Follow on work in this area involves lab testing with real sensors with the eventual goal of flight test.

Instance	Veh. 1	Veh. 2	Veh. 3	Veh. 4	Veh. 5	Grand Mean
Proprioceptive Mean RSS Error (m)	9.86	13.39	9.90	11.13	9.52	10.76
"Stingy" Collaborative Mean RSS Error (m)	8.89	10.39	8.37	10.82	9.22	9.54
Proprioceptive ANEES	0.94	1.16	0.74	0.98	0.78	0.92
"Stingy" Collaborative ANEES	0.88	1.04	0.69	1.10	0.81	0.90

TABLE I
Comparison of Proprioceptive and Collaborative ARMAS-SOM Instances

References

[1] M. J. Veth, "Fusion of imaging and inertial sensors for navigation," Ph.D. dissertation, Air Force Institute of Technology, 2006.

[2] J. Curro and J. Raquet, "Navigation using vlf environmental features," in *Position, Location and Navigation Symposium (PLANS), 2016 IEEE/ION*. IEEE, 2016, pp. 373–379.

[3] A. Canciani and J. Raquet, "Absolute positioning using the earth's magnetic anomaly field," *Navigation*, vol. 63, no. 2, pp. 111–126, 2016.

[4] K. Fisher and J. Raquet, "Precision position, navigation, and timing without the global positioning system," *Navigation*, no. 2, pp. 24–44, 2010.

[5] J. Raquet and R. Martin, "NON-GNSS RADIO FREQUENCY NAVIGATION," *Signals*, pp. 5308–5311, 2008.

[6] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.

[7] D. Temper. (2010) Darpa adaptable navigation systems (ans). [Online]. Available: https://www.darpa.mil/program/adaptable-navigation-systems

[8] J. Jurado, J. F. Raquet, C. M. Schubert-Kabban, and J. S. Gipson, "Residual-Based Multi-Filter Methodology for All-Source Fault Detection, Exclusion, and Performance Monitoring," *ION*, 2019.

[9] J. Blanch, T. Walker, P. Enge, Y. Lee, B. Pervan, M. Rippl, A. Spletter, and V. Kropp, "Baseline advanced RAIM user algorithm and possible improvements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 713–732, 2015.

[10] J. Jurado, J. F. Raquet, and C. M. Schubert-Kabban, "Autonomous and Resilient Management of All-source sensors for Navigation," *ION*, 2018.

[11] K. Kauffman, D. Marietta, J. Raquet, D. Carson, R. C. Leishman, A. Canciani, A. Schofield, and M. Caporellie, "Scorpion : A Modular Sensor Fusion Approach for Complementary Navigation Sensors," in *ION/IEEE PLANS*. Portland, OR: IEEE, 2020.

[12] J. S. Gipson and R. C. Leishman, "Resilience for Multi-filter All-source Navigation Framework with Integrity," *AFIT Faculty Preprint*, 2021.

[13] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, p. 35, 1960. [Online]. Available: https://doi.org/10.1115/1.3662552

[14] C. F. Van Loan, "Computing integrals involving the matrix exponential," in *IEEE Transactions on Automatic Control*, vol. 23:3, 6 1978, pp. 395–404.

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

[16] Z. Chen, C. Heckman, S. Julier, and N. Ahmed, "Weak in the NEES?: Auto-Tuning Kalman Filters with Bayesian Optimization," *2018 21st International Conference on Information Fusion, FUSION 2018*, pp. 1072–1079, 2018.

[17] S. J. Julier and J. K. Uhlmann, "Non-divergent estimation algorithm in the presence of unknown correlations," *Proceedings of the American Control Conference*, vol. 4, pp. 2369–2373, 1997.

[18] S. J. Julier, "Fusion without independence," *IET Seminar Digest*, vol. 2008, no. 12273, pp. 1–4, 2008.

[19] A. Bahr, M. R. Walter, and J. J. Leonard, "Consistent cooperative localization," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3415–3422, 2009.

[20] L. Sepulveda, R. Leishman, K. Kauffman, and J. Gipson, "Optimizing a Bank of Kalman Filters for Navigation Integrity using Efficient Software Design." *Proceedings of the ION GNSS+*, 2021.