IMPLEMENTATION OF THE REFERENCE ADVANCED RAIM USER ALGORITHM

Jianming She, Kathleen Misovec, *MITRE*
Juan Blanch, *Stanford University*
Natali Caccioppoli, David Duchet, *EUROCONTROL*
Enrique Domínguez Tijero, *GMV*
Fan Liu, *FAA*
Danielle Racelis, Mathieu Joerger, *Virginia Tech*
Matteo Sgammini, *European Commission, Joint Research Centre (JRC)*

BIOGRAPHY

Jianming She is a Principal System Architect at The MITRE Corporation. He specializes in the aviation applications of GNSS. Research areas include GNSS integrity modeling and monitoring, ADS-B dependency on GNSS, RAIM/ARAIM algorithms, GNSS related jamming and spoofing countermeasures, numerical and symbolic algorithms, and data visualization. He holds a BS in Mathematics from Wuhan University, China, and an MS degree in Mathematics, and an MS degree and a Ph.D. in Electrical Engineering from West Virginia University.

Kathleen Misovec is a Lead Systems Engineer at The MITRE Corporation. She specializes in advanced controls, estimation and applied mathematical techniques. She holds a BS and MS in Aeronautics and Astronautics and PhD in Mechanical Engineering, from the Massachusetts Institute of Technology.

Juan Blanch is a senior research engineer at Stanford University, where he works on integrity algorithms for Space-based Augmentation Systems and on Receiver Autonomous Integrity Monitoring. A graduate of Ecole Polytechnique in France, he holds an MS in Electrical Engineering and a Ph.D. in Aeronautics and Astronautics from Stanford University. He received the 2004 Institute of Navigation (ION) Parkinson Award for his doctoral dissertation and the 2010 ION Early Achievement Award.

Natali Caccioppoli received a M.Sc. with specialization in radio-electronic navigation in 2003, and a B.Sc. in telecommunication engineering in 2007, both from University of Naples Parthenope (Italy). He also received the AvMP designation issued by Stanford University (USA) and International Air Transport Association (IATA). In 2003, he was appointed as fellow researcher at the G. Latmiral Engineering School granted by the Italian aerospace research centre (CIRA). Since 2008, he has been working as senior GNSS operational validation consultant at EUROCONTROL Innovation HUB (France) contributing to the development and validation of PEGASUS tool for GNSS aviation applications (A/RAIM, SBAS, GBAS and RFI).

David Duchet graduated in Computer Science at Université de Nancy Henri Poincaré in 1996. He then joined the French General Delegation for Armament Programmes in the nuclear safety service. In 1997, he joined the Val-Informatique company to develop paraclinical device software. Since April 2001, he joined the PEGASUS team at EUROCONTROL Innovation HUB (France) working on development and maintenance of the PEGASUS tool.

Enrique Domínguez Tijero is a GNSS and Multisensor Navigation Expert at GMV. He received a Master in Telecommunication Engineering in 2000 and a Master in Space Technologies in 2009, both from the Polytechnic University of Madrid (UPM). He joined GMV in 2000 working first in the development of EGNOS and Galileo and since 2009 in GNSS software receivers, multi-sensor fusion algorithms, integrity algorithms, localization systems for autonomous driving vehicles and 5G positioning.

Fan Liu is an Aerospace Engineer at Federal Aviation Administration (FAA) in Aircraft Certification Service. She is involved in the development of international standards for use of Dual-Frequency Multi-Constellation (DFMC) Satellite Based

Augmentation System (SBAS). She received a Bachelor of Science and Master of Science in Electrical Engineering from Huazhong University of Science and Technology, and a Ph.D. degree in Electrical Engineering from Ohio University.

Danielle Racelis obtained a bachelor's degree (2014) in mechanical engineering from University of the Philippines, and a bachelor's (2017) and master's degree (2019) in aerospace engineering from University of Arizona. She is a PhD candidate at Virginia Tech, working with Dr. Mathieu Joerger on navigation integrity using multi-constellation GNSS and LEO satellite constellations.

Mathieu Joerger is an assistant professor at Virginia Tech. He obtained a Master's in Mechatronics from INSA Strasbourg, France (2002) and MS (2002) and PhD (2009) in Aerospace Engineering from the Illinois Institute of Technology, Chicago. He received the ION Bradford Parkinson Award in 2009, the ION Early Achievement Award in 2014, the ION Burka Award and Thurlow Award in 2023. He is the technical editor on navigation of IEEE Transactions on Aerospace and Electronic Systems. Dr. Joerger is a technical lead for the FAA on ARAIM for the EU/US Cooperation on Satellite Navigation.

Matteo Sgammini is a technical officer at the Joint Research Centre (JRC) of the European Commission. He was a system and software engineer at MTU Aero Engines from 2006 to 2008. From September 2008 to April 2017 he was a research associate in the Navigation group of the Institute of Communication and Navigation at the German Aerospace Center (DLR), Germany. His research includes signal processing and estimation theory for GNSS. Currently he focuses on GNSS integrity, Galileo Safety-of-Life services, and Galileo system performance verification.

ABSTRACT

Advanced Receiver Autonomous Integrity Monitoring (ARAIM) is a proposed evolution of Receiver Autonomous Integrity Monitoring (RAIM) to multi-constellation and dual frequency implementations. Six different groups have coded the ARAIM Algorithm Description Document (ADD)'s reference receiver algorithm in their GNSS availability simulation tools, including ARTEX by the Joint Research Centre (JRC) of the European Commission, Global Navigation Satellite System (GNSS) Performance Analysis Tool (GPAT) by MITRE, MATLAB Algorithm Availability Simulation Tool (MAAST) by Stanford University, PEGASUS by EUROCONTROL, Service Volume Simulator (SVS) by GMV, and AVA (Assured Vehicle Autonomy laboratory) Tool for ARAIM Analysis (ATAA) by Virginia Tech. This paper describes their implementations of the baseline ARAIM user algorithm. First, the paper provides flowcharts of the ARAIM algorithm: it explains the ARAIM algorithm's input, processes, and output with practical considerations relevant to different algorithm implementations. Next, the paper describes the lessons learned from cross-checking the tools among the six groups. Finally, a worldwide ARAIM evaluation with complete numerical value description is given to illustrate the implementation of the ARAIM algorithm, which can serve as a practical verification methodology for other ARAIM algorithm implementations. With the flowcharts, practical observations and findings from collaborative groups that have been working together for years, and an illustrative ARAIM numerical example, this paper can be seen as a companion to the ARAIM ADD.

I. INTRODUCTION

This paper describes a validation and verification (V&V) methodology for the reference Advanced Receiver Autonomous Integrity Monitoring (ARAIM) algorithm described in the "Advanced RAIM Reference Airborne Algorithm Description Document" (WG-C ARAIM TSG, 2023). Validation is achieved through cross-checks of multiple ARAIM configurations and scenarios using six implementations independently coded by U.S. and European researchers in academia, government, and industry. This paper provides insights and best-practices for implementing the reference ARAIM algorithm and presents a numerical example of the algorithm.

ARAIM is a proposed evolution of Receiver Autonomous Integrity Monitoring (RAIM) to dual frequency multi-constellation implementations (Blanch et al., 2015; Blanch et al., 2022). The ARAIM algorithm addresses more complex scenarios than RAIM at the cost of a higher computational load. For example, ARAIM enables the integration of multiple constellations with diverse signal and performance characteristics, which will be communicated to receivers in the Integrity Support Data (ISD). Depending on ISD satellite and constellation parameter values, ARAIM may be required to address the mitigation of different fault combinations. Among those fault modes, multiple simultaneous faults are more challenging to mitigate than single-satellite faults. When undetected, bounding the impacts of these faults on position solutions increases algorithm complexity.

To account for the risks of loss of integrity and loss of continuity while improving availability, a reference ARAIM user algorithm has been developed since 2012. This algorithm is a key part of the ARAIM standardization effort (Working Group C, 2016). Although the algorithm is not mandatory for receiver manufacturers, it serves as a reference approach to assess ARAIM coverage.

The ARAIM reference algorithm is designed to have the following two attributes. First, the algorithm must have an analytical proof of safety applicable to any possible user-to-satellite geometry, any nominal error characteristics (including temporal correlation when evaluating risks over time), and any value of the probability of satellite and constellation faults. Second, the algorithm must be computationally efficient.

The reference algorithm guarantees a statistical upper-bound on positioning errors called the Protection Level (PL). The ARAIM PL conservatively accounts for the receiver's lack of knowledge on (a) whether an undetected fault is occurring, (b) which satellites may be faulted, (c) the fault's ranging error distribution, and (d) the nominal error time-correlation. In addition, the PL equation limits the number of fault hypotheses for which a positioning error bound needs to be computed – the risk caused by rarely-occurring faults can simply be bounded by their prior probability of occurrence.

As part of the ARAIM standards development, six different groups coded the ARAIM ADD's reference algorithm in their Global Navigation Satellite System (GNSS) availability simulation tools, including ARTEX by the Joint Research Centre (JRC) of the European Commission, GPAT by MITRE, MAAST by Stanford University, Pegasus by EUROCONTROL, SVS by GMV, and ATAA by Virginia Tech. This effort aims at finding a common understanding of the ADD and at ensuring that the availability simulation tools consistently predict identical levels of performance. Through the collaborative work of the multiple groups, we have developed a set of effective algorithm verification and comparison mechanisms.

This paper describes the implementation of the reference ARAIM user algorithm in three parts. First, the paper provides an illustration of ARAIM ADD v4.2 through flowcharts of the ARAIM algorithm including fault detection (FD) and fault detection and exclusion (FDE). These flowcharts identify individual ARAIM algorithm inputs, processes, and outputs with practical considerations relevant to different algorithm implementations. Second, the paper describes the lessons learned based on challenges encountered when cross-checking the tools among the six groups. Third, a numerical example is detailed, including simulation settings and performance results for scenarios of increasing complexity, which can serve as a practical verification methodology for other ARAIM algorithm implementations.

The paper is organized as follows. In Section II of the paper, flowcharts describe ARAIM implementations by breaking the reference algorithm down into key functional blocks. Section III presents the main findings drawn from a detailed comparison between independently coded implementations. Section IV provides a numerical example with dual-frequency ARAIM using nominal GPS/Galileo constellations. Concluding remarks are given in Section V.

II. ARAIM ALGORITHM FLOWCHARTS AND DESCRIPTIONS

This section is a brief and introductory summary of the reference ARAIM algorithm. It supplements the current Algorithm Description Document (WG-C-ARAIM TSG, 2023). The first part of the section overviews the inputs to the ARAIM algorithm. The second part of the section overviews the key algorithms. A flowchart is presented for the fault detection algorithm. And the flowchart from the ADD 4.2 (WG-C-ARAIM TSG, 2023) is presented for fault detection and exclusion algorithms with slight updates to include inputs and outputs. References are selected from the literature where more detail can be found, and we note that many other excellent references are available, and it is not possible to include them all here. This section is geared towards a technical audience that may be new to ARAIM details but needs to do further work or implementation with ARAIM.

A. Overall Algorithm Inputs

1. FD-I1 ISD Inputs

ARAIM includes the provision of ISD. In contrast with RAIM (which can be thought of as using "static ISD"), ARAIM enables dynamic ISD where integrity parameters can be adapted to better represent the actual satellite and constellation performance. ISD will be broadcast in core satellite constellation navigation data messages. Messages containing ISD are

called integrity support messages (ISM). If a core satellite constellation does not provide ISM, then default ISD values backed by constellation service provider commitments can be used in ARAIM.

The ISD contains parameters that describe a Gaussian overbound of the fault-free ranging signal errors as well as parameters describing the likelihood that the satellite signal is faulted and may not be adequately characterized by the fault-free Gaussian overbounds. The ARAIM algorithm uses the ISD to perform fault monitoring and integrity tests and to calculate the protection level(s), which are used to assess if the integrity performance meets operational requirements. The following table shows the parameters that are derived from the ISD.

As reported in the ADD 4.2, (WG-C-ARAIM TSG,2023), the parameters included in the Table II-1 might depend on the frequency combination (single-frequency or dual-frequency), or on the mode of operations horizontal ARAIM (H-ARAIM) for en route, terminal, and required navigation performance (RNP) to Lateral Navigation (LNAV) minima, or H-ARAIM/vertical ARAIM (V-ARAIM) for RNP Approach to localizer performance with vertical guidance (LPV) minima.

Table II-1: List of Parameters Derived from the ISD

ISD Parameter	Description	Source
$\sigma_{URA,i}$	standard deviation of the clock and ephemeris error of satellite i used for integrity	ISD + navigation data
$\sigma_{URE,i}$	standard deviation of the clock and ephemeris error of satellite i used for accuracy and continuity	ISD + navigation data
$\boldsymbol{b_{nom,i}}$	maximum nominal bias for satellite i used for integrity	ISD
$\boldsymbol{P_{sat,i}}$	probability of a fault in satellite i	ISD
$\boldsymbol{P_{const,j}}$	probability of a fault affecting more than one satellite in constellation j due to a common cause	ISD
$\boldsymbol{R_{sat,i}}$	Rate of a fault in satellite i	ISD
$\boldsymbol{R_{const,j}}$	Rate of a fault affecting more than one satellite in constellation j due to a common cause	ISD
$\boldsymbol{MFD_{sat,i}}$	Mean Fault Duration for satellite i	ISD
$\boldsymbol{MFD_{const,j}}$	Mean Fault Duration for constellation j	ISD

The ISD parameters σ_{URA} and σ_{URE} characterize the fault-free ranging signal errors caused by clock and ephemeris errors. The b_{nom} term characterizes bias-like errors that cannot be reliably modeled using Gaussian functions but can be bounded in magnitude. These are mainly due to the receiver-dependent impacts of pseudorange signal deformations and can include contributions from other error sources. The b_{nom} term can also be used, as part of the overbounding process, to bound asymmetry and non-unimodality in the observed clock and ephemeris error distributions. b_{nom}, σ_{URA} and σ_{URE} do not account for errors caused by signal propagation through the troposphere and ionosphere.

The ISD fault rate (R_{sat}, R_{const}), mean fault duration (MFD_{sat}, MFD_{const}), and fault probability (P_{sat}, P_{const}) describe the likelihood that the satellite or constellation is faulted. Only two out of these three parameter pairs are needed: for example, if (MFD_{sat}, MFD_{const}) and (R_{sat}, R_{const}) are provided then (P_{sat}, P_{const}) can be derived at the receiver. The ISD aims at supporting horizontal and vertical aircraft navigation during en route operation, initial approach, intermediate approach, non-precision approach and departure.

2. FD-I2 Navigation Parameters

Additional parameters are derived from the navigation requirements such as the ones reported in Table II-2.

Table II-2: Navigation Requirement Parameters

Name	Description
$PHMI$	Total Integrity budget
P_{Alert}	Continuity budget allocated to disruptions due to false alert and failed exclusions
HAL	Horizontal alert limit
T_{exp}	Exposure window length
$N_{es,INT}$	Number of effective samples for integrity purposes
$N_{es,CONT}$	Number of effective samples for continuity purposes

3. FD-I3 Design Parameters

Table II-3 provides the ARAIM algorithm design parameters which define:

- the allocation of the integrity budget between vertical and horizontal,
- the false alert rate allocation to the monitors in the vertical and horizontal domain, and
- the parameter used to limit the number of fault modes that are monitored by the airborne algorithm.

Table II-3: Design Parameters (Tunable)

Name	Description
$PHMI_{VERT}$	Integrity budget allocation for the vertical component
$P_{FA_{VERT}}$	Probability of false alert allocated to the vertical direction
$P_{FA_{HOR}}$	Probability of false alert allocated to the horizontal direction
P_{THRES}	Threshold for the integrity risk allocation to not-monitored faults
F_C	Threshold used for fault consolidation
$N_{ITER_{MAX}}$	Maximum number of iterations to compute the protection level (PL)
TOL_{PL}	Tolerance for the computation of the PL

The parameters in Table II-3 can be adjusted as a function of the ISD content, and of the targeted operation (WG-C ARAIM TSG, 2023). For instance, to address horizontal operations, the full integrity budget is set to the horizontal dimension. Similarly, the threshold for the integrity risk allocation to unmonitored faults can be adjusted to remove most of the fault modes, but setting this threshold too low may cause some neglectable fault modes to be triple counted (WG-C ARAIM TSG, 2023). The F_C parameter helps refine the PL bounding process by hypothesis grouping: it sets the maximum probability of independent satellite fault within one constellation that can be grouped within the corresponding constellation wide fault mode (WG-C ARAIM TSG, 2023).

4. FD-I4 Parameters

Additional inputs include almanac data and error model data.

B. Fault Detection and Exclusion Flowcharts

This section presents an overview and high-level flowcharts for the FD and FDE algorithms as described in sections 4 and 5 of the ARAIM Algorithm Description Document (ADD v4.2 2023) with the inputs described in the previous section. The goal of the flowcharts is to focus on the processing steps of the algorithms and on the key input and output data directly used by the functional processes within the algorithm.

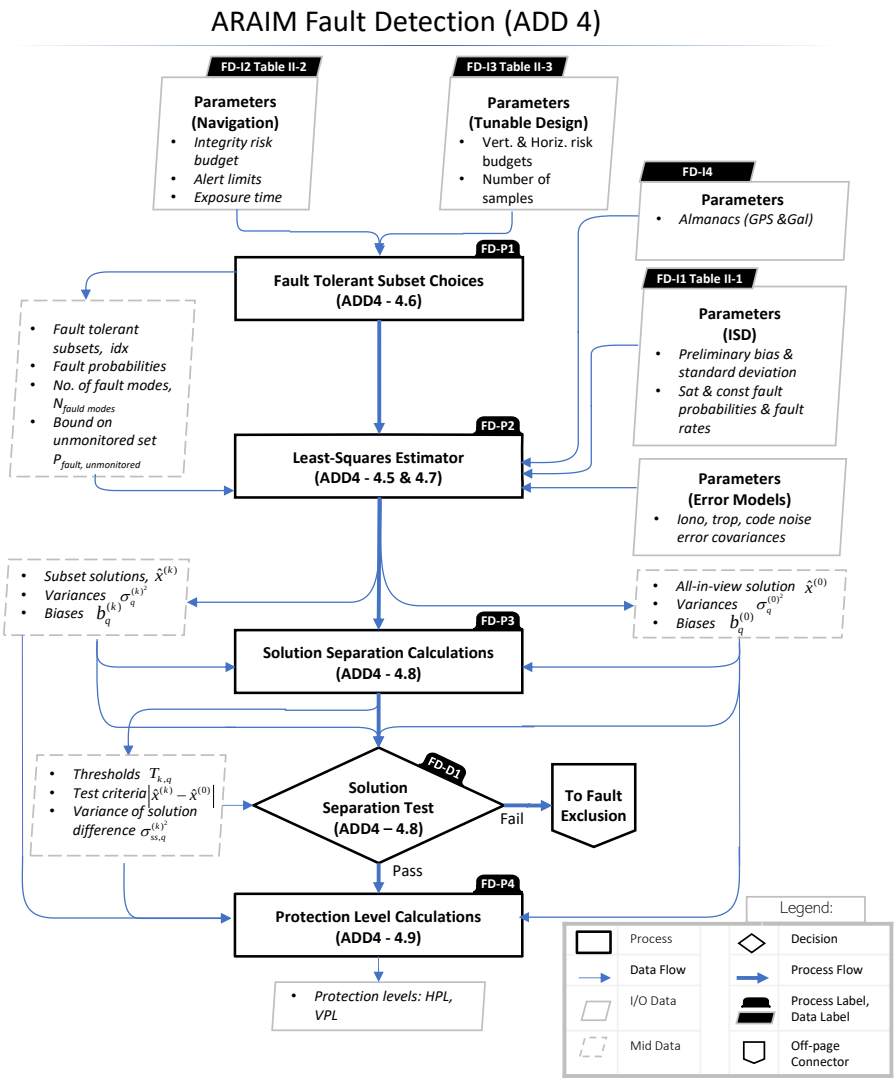

Figure II-1 Fault Detection Flowchart

1. Fault Detection

For fault detection, integrity risk is defined as the joint probability that the estimated state error is larger than a specified alert limit and that the detection test statistic is less than the detection threshold (WG-C ARAIM TSG, 2023; Blanch et al., 2015; Zhai et al, 2016; Cassel, 2017; Joerger et al., 2013). The alert limit is specified such that it defines hazardously misleading information (HMI). The test statistic and threshold are calculated in ARAIM as discussed below. The thresholds are calculated such that specified false alarm and detection probabilities satisfy the requirements given the assumptions, and Blanch et al. (2015) contains a proof of safety. Protection levels are an important set of outputs produced by ARAIM fault detection. They can be calculated iteratively, and details are in the ADD 4.2 (WG-C ARAIM TSG, 2023). Horizontal and vertical protection levels are distances that define the regions within which the true solution lies at a specified probability level (WG-C ARAIM TSG, 2023; Blanch et al., 2015). These derived measures are a particularly useful way for users to consider the integrity risk.

Note that for fault detection and exclusion, integrity risk is defined differently, and we will discuss that in the Fault Detection and Exclusion section later.

The flowchart in Figure II-1 is based on the ARAIM fault detection that is presented in step-by-step detail in section 4 of ADD 4.2 (WG-C ARAIM TSG, 2023).

Note that the first algorithm process in the flowchart FD-P1 which pertains to subset choices is presented last in the subsection.

FD-P2 Least Squares Solutions

A key algorithm (process) is the least squares estimator. When the input to this estimator includes measurements from all of the satellites, the estimator produces the 'all-in-view' position solution $\hat{x}^{(0)}$. This is described in ADD subsection 4.5 and the superscript '0' indicates that it is the all-in-view solution. When the input consists of measurements from the subset of the satellites excluding those possibly affected under the k^{th} fault mode, the estimator produces an output called the k^{th} subset position solution, $\hat{x}^{(k)}$ (section 4.7 ADD). Each subset is given a numerical identifier, denoted by k in the superscript. The estimator also produces associated variances, $\sigma_q^{2(k)}$, and biases, $b_q^{(k)}$, with these estimates. The subsets and the number of subsets, $N_{fault\ modes}$, are determined in the 'fault tolerant subset choices' box of the flowchart.

FD-P3 Solution Separation Calculations

Differences between the all-in-view solution and the subset solutions are solution separations. If a subset solution and the all-in-view solutions are different enough, then a fault may be present. For each subset, the estimator also produces a variance on the difference between each subset solution and the all-in- view solution $\sigma_{ss}^{2(k)}$. These are used in the solution separation threshold decision box and is discussed next.

FD-D1 Solution Separation Threshold Tests

In the solution separation threshold test decision box, $\sigma_{ss}^{2(k)}$ a Gaussian characterizes the overbounding distribution of the solution separation test statistic, assuming zero bias. Given this distribution and a specified false alarm probability, a threshold for a decision, $T_{k,q}$, is computed, where k is the fault mode and q is the dimension (horizontal dimensions 1 or 2 and vertical dimension 3) (section 4.8 of the ADD 4.2, (WG-C ARAIM TSG, 2023)). The test statistic, $\left|\hat{x}_q^{(k)} - \hat{x}_q^{(0)}\right|$, is the absolute value of the difference between the subset solutions and the all-in-view solution. For any k and any q, if the test statistic, $\left|\hat{x}_q^{(k)} - \hat{x}_q^{(0)}\right|$, is greater than the threshold $T_{k,q}$, the outcome of the test is that a fault might be present (but not necessarily). If a fault is detected, then exclusion can be attempted to improve continuity. The fault exclusion algorithm is described later in this section.

FD-P4 Protection Levels

If all of the test criteria indicate no fault or if successful exclusion is performed, then Protection Levels, *HPL* and *VPL* for horizontal and vertical, respectively, are computed (section 4.9 and appendices of the ADD). In the proof of safety in (Blanch et al., 2015), terms in the protection level definition are related to the components in the integrity risk definition – missed detection and errors larger than the levels associated with HMI.

FD-P1 Monitored Faults and Associated Probabilities, Fault Tolerant Subset Choices

The ARAIM ADD outlines a sufficient method to choose subsets (4.6 of the ADD 4.2, (WG-C ARAIM TSG, 2023)). Two sets of fault modes, a monitored set and an unmonitored set are developed. The objective is to ensure that the sum of the probabilities of the unmonitored set does not exceed a predefined threshold, p_{Thresh} (WG-C ARAIM TSG, 2023). Fault modes are considered in an order detailed in the ADD. Following the specified order, a fault mode probability is calculated, and the fault mode is moved to the monitored set. As fault modes are added one-by-one to the monitored set, the sum of probabilities of faults in the monitored set increases. The strategy is to build the monitored set until the sum of the probabilities of the monitored set is large enough so that the sum of probabilities of faults in the unmonitored set must be less than p_{Thresh}, utilizing the fact that the probability of the sample space of all possible events is one. More details about fault probability calculations and bounds and the general strategy of creating the monitored and unmonitored sets can be found (Blanch et al., 2015), however some newer details may vary, particularly in the ordering of the fault modes.

The ADD also presents a 'fault consolidation sub-algorithm' that may be used to consolidate multiple satellite faults into a constellation-wide fault. Also, a 'filtering the subsets' algorithm is used to eliminate unobservable sets from the monitored set. In addition, the ADD describes an alternative estimator for fault detection that focuses on reducing the protection levels (although it may be suboptimal in some cases). These details and alternatives are beyond the scope of this overview.

2. *Fault Detection and Exclusion*

The next part of this section is an overview of the ARAIM exclusion algorithm and is described more fully in section 5 of the ADD 4.2 (WG-C ARAIM TSG, 2023). Exclusion is initiated when the subset threshold tests detect the possibility of a fault. The idea is that if exclusion can remove a fault that is making one or more solution separations too large and if protection levels can be computed, then continuity of service can be preserved (Cassel, 2017).

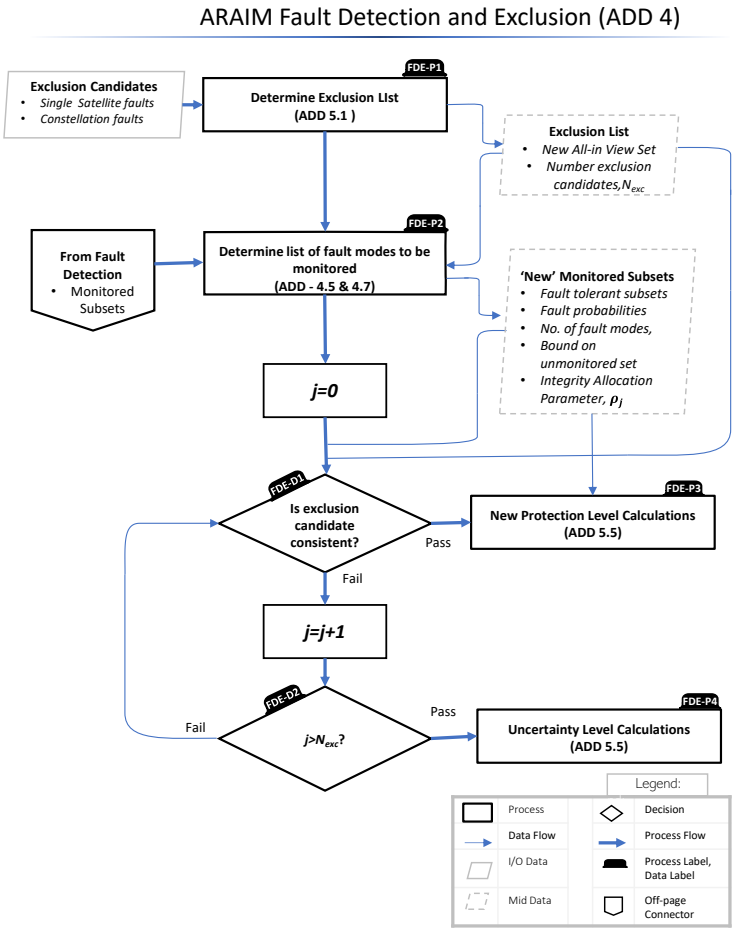

Figure II-2 Fault Detection and Exclusion Flowchart

Note that continuity is the ability of the system to perform its intended operation without unscheduled interruption. These interruptions happen when a detection or false alarm occurs that cannot be excluded (Cassel, 2017). Availability is an additional ARAIM performance measure that is the fraction of time that operational requirements are met (Cassel, 2017). Precise definitions can be found in the ADD and its references.

Essentially, the exclusion process is similar to detection but can be considered to be a 'new' problem such that a 'new' all-in-view solution set does not include an exclusion candidate. 'New' solution separations associated with a 'new' set of monitored subsets would then be considered 'consistent' if they pass the 'new' solution threshold tests. If consistent, protection level calculations can be made.

FDE-P1 Exclusion List

Figure II-2 is a flowchart from the ADD of the exclusion process. The first step, in the top box of the flowchart, is to define a list of exclusion candidates. Based on the current ISD and ADD, we consider all single satellite faults and the constellation-wide fault modes that need to be monitored.

FDE-P2 Monitored Fault List

The list of fault modes that need to be monitored are input from the detection algorithm and this is depicted in the box second from the top of the flowchart. As an exclusion candidate is considered, the set of monitored faults may contain elements that are identical. If so, these elements can be identified and the set of monitored faults and their probabilities can be adjusted.

FDE-D1 Consistency Check for Exclusion Candidate

Next the consistency decision is made on if the set of monitored faults can pass the threshold detection tests. If so, protection levels can be computed.

FDE-D2 Next Exclusion Candidate

If the consistency checks are not met, the next exclusion candidate can be tried and the loop iterated until there are no more candidates.

In the exclusion algorithm, the integrity risk upper bound is guaranteed by making sure that the sum of the integrity risk of all the possible exclusion options is bounded. For this reason, we need to pre-allocate the integrity risk to each of the possible exclusion options. The set of parameters ρ_j regulates the allocation to each of the exclusion options, and details are in ADD 4.2 (WG-C ARAIM TSG, 2023). The formulas for protection level are adjusted using the variable, ρ_j, that accounts for proper allocation of risk for the 'new' set of exclusion options.

Note that ADD outlines an exclusion method that uses pseudorange measurements. In our work, we are interested in calculating predictive risk metrics both temporally, typically over a day and spatially, typically around the globe. Therefore, we use an exclusion method that is modified such that we are not using measurements. This can be done because the thresholds are functions of variances $\sigma_{ss}^{2(k)}$, false alarm requirements, and number of fault modes and other constants. Also, we process the entire list of exclusion candidates and compute associated protection levels. If the worst-case protection level is of interest, this is the maximum of the calculated protection levels.

FDE-P3 Protection Uncertainty Levels

If consistency does not occur for any of the exclusion candidate list, alternative horizontal and vertical protection uncertainty levels are computed which are detailed in the ADD.

Related definitions and bounds for integrity and continuity risks for detection and exclusion are developed in Joerger et al. (2019). Related work in Zhai et al. (2016) explores the trade-off between the integrity and continuity risk for detection and exclusion and the tightness of the bounds. These related definitions account for risk that is related to wrong or correct exclusions. Probability bounds are developed in Zhai et al., (2016) and are useful to quantifying trade-offs between performance metrics, particularly integrity and continuity risks. An analytical proof of integrity is presented in Blanch et al. (2017) for a related fault exclusion algorithm.

C. Section Summary

In summary, this section presents high-level flowcharts for the fault detection and fault detection and exclusion algorithms of ARAIM. The purpose is to focus on the sequence of processing steps and the key input and output data. The flowchart ties closely with sections 4 and 5 of the ADD. A key idea in solution separation ARAIM is to compare a position solution using measurements from all the satellites that are in view to solutions using subsets of satellite measurements. The subsets are created such that they are tolerant to certain faults. If all of these solutions are close enough to all-in-view solution, then the associated faults can be declared not detected with specifications on the false alarm probability. ARAIM fault detection produces protection levels that define the regions within which the true solution lies at a specified probability level. Fault

exclusion can be used to explore excluding faults and reaching a consistent set of subsets and corresponding protection levels. If fault exclusion cannot reach a consistent set, alternative uncertainty levels may be computed.

III. CROSS-CHECK OF INDEPENDENT ARAIM ADD IMPLEMENTATIONS TOOLS

A. Purpose and Method of the Cross-Check

The purpose of the cross-check is to verify the implementation of the ARAIM algorithm across six groups' availability simulation tools. The groups aimed to identify and address potential discrepancies, inaccuracies, and inefficiencies. A systematic, collaborative, and streamlined approach was devised to facilitate comprehensive scrutiny and comparison among these diverse tools. The involvement of multiple groups and tools highlights the importance of a clear and consistent approach, as this reduces the complexity of comparisons and promotes understanding by all involved.

In order to ensure the validity of the cross-checks and obtain meaningful results, we have taken a series of specific steps. Baseline simulation configurations and simulation scenarios are defined. These elements provide a basis for comparison and create a consistent comparison environment for all relevant tools. Note that the ISD settings do not coincide with the default ISD as defined in the ARAIM SARPS. The following tables give the ADD base configuration items:

Table III-1: Simulation Parameters

Parameter	Single User Case	Global Grid Case
Time Grid	0:1:1439 minutes (1 day)	0:5:1435 minutes (1 day)
User Grid	(lat, lon, alt) = (0,0,0)	10-degree grid
Almanacs	GPS: almanac week 906 Galileo: almanac week 906 (EUROCAE ED-259A, 2023)	

Table III-2: ARAIM Design Parameters

Item	LPV	RNPx
pHMI_V	9.8e-8	eps
pHMI_H	= pHMI - pHMI_V, where pHMI=1e-7	
pFA_V	3.9e-6	1e-9
pFA_H	9e-8	5e-7
pTHRES	8e-8	6e-8
Fc	0.01	0.01
nITER_MAX	10	10
Tol_PL	0.05	0.05
K_acc	1.96	
K_FF	5.33	
T_{EXP} (sec)	15	1|3600
$N_{ES,Int}$	25	1|450
$N_{ES,Cont}$	25	1|450

Note: Only RNPx parameters are used for this study

Table III-3: ARAIM Algorithm Setting

Item	Value
FD/FDE Mode	FD | FDE
Fault Mode Consolidation	On
Double Counting of Integrity Risk Accounting	off
Alg. Optimization	Off
Fault Mode Ordering Scheme	ADD4.0

Table III-4: ISD

Item	GPS	Galileo
σ_{URA}	2.4 m	6.0 m
σ_{URE}	2.4 m	6.0 m
b_{NOM}	0.75 m	0.75 m
R_{sat}	1.00E-05	3.00E-05
R_{con}	1.00E-08	2.00E-04
MFD_{sat}	1	-
MFD_{con}	1	-
P_{sat}	1.00E-05	3.00E-05
P_{const}	1.00E-08	2.00E-04

Note: The ISD set is not identical to the default ISDs included in the ARAIM SARPS

Table III-5: Ranging Error Models for Dual Frequency Users

Type	Source	Model
URA	ISD Specified	See the ISD table above
URE	ISD Specified	See the ISD table above
MP	ED-259A DF	$\sigma_{MP\&AGDV}(i) = 0.34 + 0.40\exp(-El_{\deg}[i]/14)$ [m]
CN	ED-259A DF	$\sigma_{noise}[i] = 0.4$ [m]
Troposphere	ED-259A DF	$\sigma_{tropo}[i] = 0.12\frac{1.001}{\sqrt{0.002001 + \sin^2(\frac{\pi El_{\deg}[i]}{180})}}$ [m]
Ionosphere	ED-259A DF	$\sigma_{UIRE}[i] = \frac{40.0}{261.0 + El_{\deg}^2[i]} + 0.018$ [m]

Each tool provides datasets in a predefined data file format including structured file names, data headers, and data body formats, to further facilitate efficient processing and evaluation of data. This not only improves the efficiency of the comparison work but also prevents errors that may arise from dataset conversion by individuals other than the tool owner. By adopting a unified data format, we encourage seamless sharing within the group. The following agreed-upon dataset formats exemplify the benefits of this approach. All data files are defined in CSV format.

- The file naming structure is as shown below:

Tool Name Date Version Algorithm User Time Scenario ID

$$model\text{-}yymmdd\text{-}ARAIM\_ADDx.x\text{-}HPL\text{-}FD|E\text{-}u(lat,lon)\_spacing\text{-}tsDTm\text{-}s\#$$

- The data files for single-user (lat, lon, alt) follow the following specifications: the data records start at t=0 sec and end at t=86340 sec, and the length of the data is 1440.

Table III-6: Data File for a Single User

Time (sec)	HPL (m)
0	Data Body
60	
….	
86340	

- The data files for a global grid should start at the coordinates (-90, -180). Retain the latitude value and enhance the longitude by 10 until you reach 170. Upon achieving this, amplify the latitude by 10 and repeat the longitude incrementation step. This procedure should be finalized when you arrive at the coordinates (90, 170). The columns of the data body represent time epochs, with the total count of these time epochs amounting to 288.

Table III-7: Data File for a Global Grid

lat	lon	Time epoch (sec)			
		0	300	…	86100
-90	-180	Data Body			
-90	-170				
…	…				
-90	170				
-80	-180				
-80	-170				
…	…				
-80	170				
…	…				
90	170				

After collecting datasets from each tool, the process enters a pairwise data comparison step. This comparison step is key, focusing on several key statistics of the pairwise difference dataset. These statistics include absolute differences for each point (the point is defined by user location and time epoch), relative differences, histogram of differences, max/min of differences, and RRSQ of differences.

These calculated statistics from the pairwise comparisons were then used to classify the comparisons into three categories - 'good', 'fair' and 'poor'. These labels are defined by group consensus and serve as an index for the agreement assessment for each paired dataset comparison. This categorization simplifies interpretation of results and enables participants to communicate more quickly and effectively about the consistency of their respective tools.

- Pairwise Comparison Categories: For a pairwise comparison, let's denote $data_1$ for data of dataset of the ARAIM tool 1, and $data_2$ for data of dataset of the ARAIM tool 2. $data_1$ and $data_2$ are both $m \times n$ data arrays. Now, we define

$$diff_{(d1,d2)} = data_1 - data_2 \qquad \text{and} \qquad \mu(diff_{(d1,d2)}, a) = \frac{count(diff_{(d1,d2)} \in [-a, a])}{count(diff_{(d1,d2)})}$$

$$diff^{R}_{(d1,d2)} = \frac{data_1 - data_2}{data_1} \qquad \text{and} \qquad \mu(diff^{R}_{(d1,d2)}, a) = \frac{count(diff^{R}_{(d1,d2)} \in [-a, a])}{count(diff^{R}_{(d1,d2)})}$$

where the superscript R stands for 'relative'.

Using the given notations, we can establish the categories 'good,' 'fair,' and 'poor' in the following manner:

Table III-8: Comparison Categories

Cat	Cat Definition	Case Definition
Good	Case 1 | Case 2 | Case 3 | Case 4	• Case 1: $\mu(diff_{(d1,d2)}, 0.1) \geq 99.9\%$ • Case 2: $max(abs(diff_{(d1,d2)})) < 0.05m$ • Case 3: $\mu(diff_{(d1,d2)}, 0.1) \geq 99\%$ & $\mu(diff_{(d1,d2)}, 0.5) \geq 99.9\%$ & $max(abs(diff_{(d1,d2)})) < 5m$ • Case 4: $\mu(diff^{R}_{(d1,d2)}, 0.25\%) \geq 99.9\%$
Fair	Case 1 | Case 2 | Case 3	• Case 1: $\mu(diff_{(d1,d2)}, 0.1) \geq 90\%$ & $\mu(diff_{(d1,d2)}, 0.5) \geq 95\%$ & $max(abs(diff_{(d1,d2)})) < 100m$ • Case 2: $\mu(diff_{(d1,d2)}, 0.1) \geq 85\%$ & $\mu(diff_{(d1,d2)}, 0.5) \geq 90\%$ & $max(abs(diff_{(d1,d2)})) < 50m$ • Case 3: $\mu(diff^{R}_{(d1,d2)}, 0.25\%) \geq 95\%$
Poor		[~Good & ~Fair]

Where "|" stands for logic operator OR, and "&" stands for logic operator "AND"

To provide a clearer understanding of the pair-wise comparison statistics and the resulting classification categories, the tables below offer some numerical examples for illustrative purposes:

Table III-9: $\mu(diff_{(d1,d2)}, 0.1)$

	ARTEX	ATAA	GPAT	MAAST	PEGASUS	SVS
ARTEX	NaN	9.9999e-01	1.0000e+00	1.0000e+00	1.0000e+00	9.9901e-01
ATAA	9.9999e-01	NaN	9.9999e-01	9.9998e-01	9.9997e-01	9.9768e-01
GPAT	1.0000e+00	9.9999e-01	NaN	1.0000e+00	1.0000e+00	9.8482e-01
MAAST	1.0000e+00	9.9998e-01	1.0000e+00	NaN	1.0000e+00	9.9934e-01
PEGASUS	1.0000e+00	9.9997e-01	1.0000e+00	1.0000e+00	NaN	9.9934e-01
SVS	9.9901e-01	9.9768e-01	9.8482e-01	9.9934e-01	9.9934e-01	NaN

Table III-10: $\mu(diff^{R}_{(d1,d2)}, 0.25\%)$

	ARTEX	GPAT	MAAST	PEGASUS	SVS	ATAA

ARTEX	NaN	9.8200e-01	1.0000e+00	1.0000e+00	9.9939e-01	9.9998e-01
GPAT	9.8159e-01	NaN	9.7552e-01	9.7552e-01	8.4911e-01	9.8782e-01
MAAST	1.0000e+00	9.7593e-01	NaN	1.0000e+00	9.9978e-01	9.9997e-01
PEGASUS	1.0000e+00	9.7595e-01	1.0000e+00	NaN	9.9978e-01	9.9997e-01
SVS	9.9939e-01	8.5193e-01	9.9979e-01	9.9979e-01	NaN	9.9747e-01
ATAA	9.9998e-01	9.8806e-01	9.9997e-01	9.9997e-01	9.9740e-01	NaN

Table III-11: $max(diff_{(d1,d2)})$

	ARTEX	GPAT	MAAST	PEGASUS	SVS	ATAA
ARTEX	NaN	6.8264e-02	4.6278e-02	4.5813e-02	4.4553e-01	1.1358e-01
GPAT	6.8264e-02	NaN	7.1623e-02	7.0290e-02	4.5423e-01	1.0488e-01
MAAST	4.6278e-02	7.1623e-02	NaN	4.1034e-02	3.9925e-01	1.5985e-01
PEGASUS	4.5813e-02	7.0290e-02	4.1034e-02	NaN	3.9972e-01	1.5939e-01
SVS	4.4553e-01	4.5423e-01	3.9925e-01	3.9972e-01	NaN	5.5911e-01
ATAA	1.1358e-01	1.0488e-01	1.5985e-01	1.5939e-01	5.5911e-01	NaN

Table III-12: Pairwise Comparison Flags

	ARTEX	GPAT	MAAST	PEGASUS	SVS	ATAA
ARTEX	NA	Good	Good	Good	Good	Good
GPAT	Good	NA	Good	Good	Fair	Good
MAAST	Good	Good	NA	Good	Good	Good
PEGASUS	Good	Good	Good	NA	Good	Good
SVS	Good	Fair	Good	Good	NA	Good
ATAA	Good	Good	Good	Good	Good	NA

After the classification process, an additional step is taken, identifying the data point with the largest difference. This was suggested as a potential starting point for a collaborative analysis of the two corresponding tools. Working together, they can examine the source of the differences, making it easier to identify any problems or inconsistencies in their respective tools. This collaboration corrects any identified issues, leading to improvements in each tool's overall task performance.

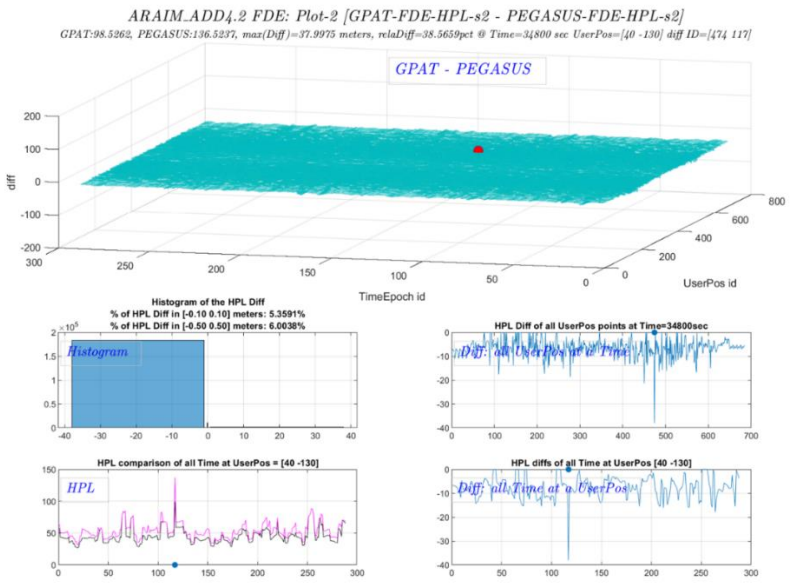

Figure III-1: Tools Pairwise Comparison

To complete the cross-check for a given baseline definition and scenario, the process needs to repeat until all indices resulting from pairwise comparisons were considered acceptable by the group. This iterative process ensures that all tools are fully cross-checked, and any discrepancies are identified and resolved.

B. Overview of Independently Implemented ARAIM Tools

The cross-check presented in this paper is based on the results obtained from six independent implementations of the ARAIM algorithm. Table III-13 provides a description of these tools. These ARAIM tools are able to compute HPL and VPL values according to ARAIM ADD v4.2. All of them have the capability of generating simulated scenarios in order to test ARAIM and assess its performance. The configuration of these scenarios requires setting the Simulation Parameters, ARAIM Design Parameters, ARAIM Algorithm Settings, ISM Parameters and Ranging Error Models described in section IIIA.

Table III-13: Main Characteristics of the ARAIM Tools Employed in the Cross-Check.

Tool Name & Developer	**Programming Language (ARAIM Ver.)**	**Description**
MAAST for ARAIM By Stanford University GPS Lab	MATLAB - ADD v3.0 & ADD v3.1 & ADD v4.2	MAAST (MATLAB Algorithm Availability Simulation Tool) is a toolset for simulating WAAS and ARAIM confidence estimation algorithms and evaluating their effects on service availability
ARTEX By JRC (EC)	MATLAB - ADD v3.1 & ADD v4.2	The tool was developed within the ARTEX project of the European Commission. The tool serves as an instrument to the Galileo programme to evaluate H-ARAIM performance and to run sensitivity analysis on the different integrity parameters.
GPAT By MITRE	MATLAB - ADD v3.1 & ADD v4.2	The GNSS Performance Analysis Tool (GPAT) integrates RAIM, ARAIM, and SBAS services into a consistent analysis environment. GPAT provides flexible configuration setup for different constellations, carrier frequencies, error models, SV outage representations, flight operations, and other simulation conditions. GPAT delivers efficient and interactive simulation and root-cause analysis capability.
PEGASUS By EUROCONTROL	C++ MPFR library - ADD v3.1 & ADD v4.2	The EUROCONTROL PEGASUS tool supports the validation of principal augmentations to GPS and Galileo such as SBAS, GBAS and it is now updated to A-RAIM. It integrates civil aviation user MOPS requirements, ensuring that developed systems are operationally acceptable and meet the safety regulations
ATAA By Virginia Tech	MATLAB - ADD v4.2	AVA Tool for ARAIM Analysis (ATAA) was developed at Virginia Tech's Assured Vehicle Autonomy laboratory (AVA) by PhD candidate Danielle Racelis, advised by Dr. Mathieu Joerger. The global availability simulation tool was written on MATLAB initially for snapshot ARAIM, but has since evolved to allow simulation of sequential ARAIM with LEO satellites and time-correlated error models.
SVS-ARAIM By GMV	Octave & MATLAB - ADD v3.1 & ADD v4.2	The SVS-ARAIM platform is a Service Volume Simulator (SVS) that provides flexible multi-constellation and multi-user setup allowing to assess the availability performances of the ARAIM integrity user algorithm

C. Aspects of Analysis Not Fully Elaborated in the ADD

1. Determination of the List of Fault Tolerant Subsets

When selecting the fault modes to be monitored there are some aspects to be considered regarding the way in which they are ordered. This ordering method is important because it conditions the fault modes that will be monitored (the sorted fault modes are included into the list of monitored modes until the total probability of the not monitored modes is below the defined threshold). ADD v4.2 states in section 4.6 that for FD, the choice of modes is not critical, and the order included in previous ADD versions can be adequate (i.e., from smallest degree to larger and, within one degree, from larger to smaller $p_{fault,k}$); for FDE the choice of monitored faults can have a large impact on performance, so it proposes a way of ordering the fault modes because it works well with the ISD defined in the draft ICAO SARPS Annex 10.

It Is important to consider that different ordering methods will lead to different results where all of them are valid. It is also important to understand that the ARAIM algorithm does not impose any way of ordering the fault modes when selecting them. This means that, as long as the total probability of the not monitored modes is below the defined threshold (P_{THRES}), any method for ordering/selecting the monitored modes is valid, and the ADD is just recommending ways of doing it. Since, when fault modes with weak geometries are included in the list of monitored modes there is an impact on the size of the computed protection levels, the objective is to employ a simple ordering method (not requiring a high computational load) that avoids these cases as much as possible, thus providing better availability performances than other ordering methods (i.e., having lower protection level values/peaks with respect to other methods).

In the tests that have been carried out it has been detected that the ordering method proposed in ADD v4.2 for FDE provides lower protection level values/peaks than the one proposed in previous ADD versions, even for the FD. The main reason is that the order proposed in ADD v4.2 relegates the fault modes that can have higher impact on the size of the protection levels (according to ISM parameters and also taking into account weaker geometries), as it can be seen in Figure III-2. For example, the order proposed in previous ADD versions always monitors the GPS constellation fault mode, while the order proposed in ADD v4.2 relegates it and usually ends up not monitoring it. Also, the order proposed in ADD v4.2 considers the GPS Satellite Vehicles (SV) and GAL (Galileo will be abbreviated as GAL when needed here after) SV dual faults before the GPS SV and GAL constellation faults, this prioritizes stronger geometries and potentially leads to lower protection levels at the cost of monitoring a greater number of fault modes. Note that if the “Accounting for possible double counting of integrity risk (IR)” check is applied then some modes (e.g., the GPS constellation fault mode) may trigger the check and be moved to not monitored.

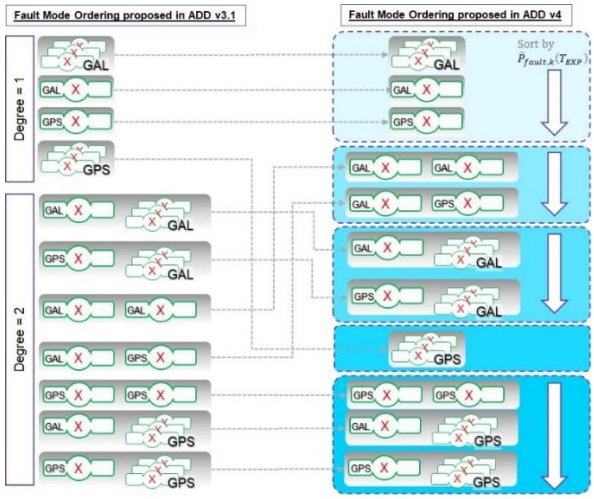

Figure III-2: Proposed Fault Mode Ordering: ARAIM ADD v3.1 vs ARAIM ADD v4.2 Schemes

Figure III- shows the impact on the computed HPLs in the FD case with ADD v4.2 by comparing the way of ordering faults proposed in previous ADD versions with the one recommended for exclusion in ADD v4.2 (without applying fault consolidation or the accounting for possible double counting of IR check).

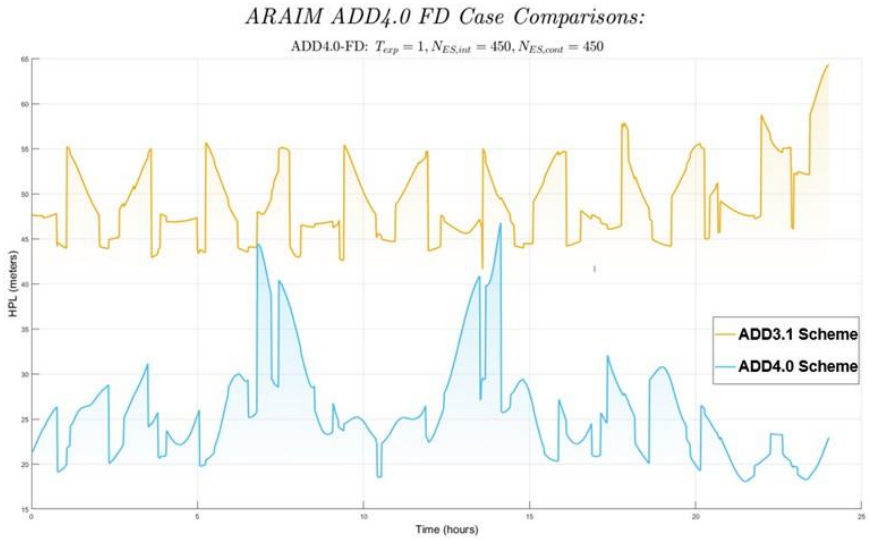

Figure III-3: Fault Mode Ordering: ARAIM ADD v3.1 vs ARAIM ADD v4.2 HPLs (without applying fault consolidation or the accounting for possible double counting of IR check)

Therefore, as all ARAIM tools must employ the same ordering method to be able to compare their results, the ordering method proposed in ADD v4.2 is the one that has been employed in the cross-check tests for both, FD and FDE. The method proposed in ADD v4.2 leaves some ordering aspects undefined (because they may be decided arbitrarily with minimal impact on performance), so the following Table III- describes in detail the fault order followed in the cross-check tests.

Table III-14: Fault Order Followed in the Cross-Checks
(Ordering Method Proposed in ARAIM ADD v4.2)

Order	Fault Modes included in each group	Order within each group
1st	The modes for which $\bar{P}_{fault,k}(T_{EXP})>PHMI$. These modes need to be monitored. For the more likely ISD settings, this will include all the single satellite fault modes, and the constellation fault modes such that $\bar{P}_{fault,k}(T_{EXP})>PHMI$	Sorted by $\bar{P}_{fault,k}(T_{EXP})$ For the same $\bar{P}_{fault,k}(T_{EXP})$ sort them by the fault satellite ID in ascending order.
2nd	The dual satellite faults (except GPS-GPS)	Sorted by $\bar{P}_{fault,k}(T_{EXP})$ For the same $\bar{P}_{fault,k}(T_{EXP})$ sort them first by the fault Galileo satellite ID in ascending order and then, for GAL-GPS dual faults having the same GAL SV ID, by the GPS satellite ID in ascending order (e.g., first GPS1&GAL1, GPS1&GAL2, GPS1&GAL3, …, then GPS2&GAL1, GPS1&GAL2, GPS1&GAL3, …).
3rd	The satellite-constellation wide fault modes corresponding to the constellation wide fault modes that are included in step 1 (GPS satellite – Galileo constellation, in the case of the default ISD parameters)	*NOTE: This group includes GAL SV – GAL Const and GPS SV – GAL Const fault modes. The last sentence in parenthesis can be misleading.* Sorted by $\bar{P}_{fault,k}(T_{EXP})$ For the same $\bar{P}_{fault,k}(T_{EXP})$ sort them by the fault satellite ID in ascending order.
4th	The remaining constellation wide fault modes with $\bar{P}_{fault,k}(T_{EXP})>0$	Sorted by $\bar{P}_{fault,k}(T_{EXP})$ *(GPS constellation fault is the only mode)*
5th	The remaining fault modes	Sorted by $\bar{P}_{fault,k}(T_{EXP})$ For the same $\bar{P}_{fault,k}(T_{EXP})$ sort them by the fault satellite ID in ascending order.

An important clarification regarding the 3rd group is that it includes GAL SV – GAL Const and GPS SV – GAL Const fault modes. The last sentence "(GPS satellite – Galileo constellation, in the case of the default ISD parameters)" can be misunderstood as if only those faults need to be considered.

In the tests ran with the ARAIM tools for FDE, another detail about the monitored fault modes that needs to be clarified is that, although after excluding one SV there will be repeated monitored fault modes, these repeated modes were not grouped into one single fault mode.

2. Peak Values in Protection Levels for Weak Satellite Geometries

Some monitored fault modes can have very weak satellite geometries and this can lead to a near singular G^TWG matrix, which in turn would make the ARAIM algorithm provide a very high protection level. The lower the number of satellites the higher the probability of having weaker geometries in the monitored fault modes, so the probability of having spikes in the computed HPLs caused by very weak geometries depends on the number of satellites in view. The tested scenarios included assessing the protection levels for a grid of users distributed all around the world, so many different possible geometries were evaluated. Under these circumstances, when assessing ARAIM for FD, no monitored fault modes with a very weak satellite geometry were found at any epoch. However, when assessing ARAIM for FDE and computing the worst HPL for the single exclusion of any of the SVs, some users at isolated epochs had monitored fault modes with very weak satellite geometries, thus causing the computed HPL to reach values in the order of several km at those epochs.

The cross-check carried out between the different ARAIM tools showed differences of a few meters in the HPL values computed at those epochs when monitored fault modes had very weak geometries. See Figure III-. Although these differences

are in the order of meters, when the HPLs are in the order of km, they are not problematic. The cause is due to minor numeric differences in:

- The generation of the simulated scenario (SV orbits);
- The method/library employed for inverting matrices;
- The implementation of the modified Q function;
- The method employed to solve the HPL equation (see equation 57 in section 5.5 of ARAIM ADD v4.2). While most of the tools follow the iterative method described in Appendix A of the ARAIM ADD v4.2, some use fzero function to solve the equation.

These minor differences between the ARAIM tools are always present and their effect in the HPLs is negligible, because, in cases/epochs having very weak geometries, the poor condition number of the G^TWG matrix causes the numerical results to be highly sensitive to these minor differences between the ARAIM tools thus causing noticeable differences in the HPLs in the order of meters.

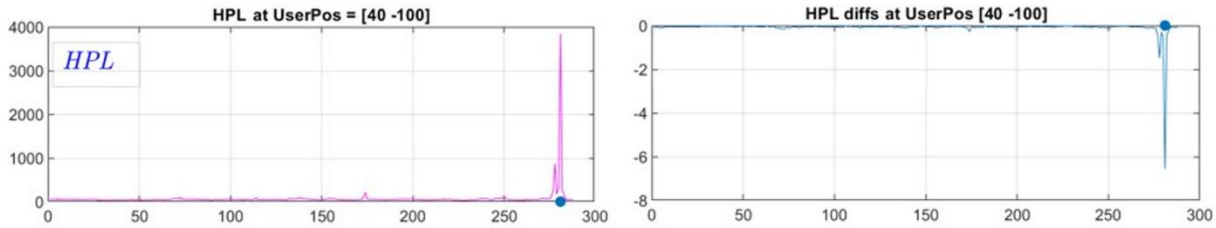

Figure III-4 HPL Difference Between Two Tools – FDE Scenario with Very Weak Geometry at Some Epochs

3. Fault Consolidation for Complexity Reduction

The ARAIM reference algorithm described in the ADD (WG-C ARAIM TSG, 2023) proposes a method to determine the list of fault modes that need to be monitored. The ISD does not specify explicitly which fault modes need to be monitored or their corresponding probabilities. Therefore, the determination of faults that need to be monitored and the associated probabilities of fault must be made based on the content of ISD, which specifies the probabilities of events that can be treated as independent. Then, for each of the determined fault modes, a subset solution is computed and compared to the all-in-view solution. The list of fault modes is dependent on the probabilities of satellite fault and constellation fault P_{sat} and P_{const} (derived from ISD). In addition to being dynamic, this list can become long for large values of P_{sat} and P_{const}. The list of fault modes that needs to be monitored as described in the ADD (WG-C ARAIM TSG, 2023) is only sufficient. As a consequence, redundant faults modes are also included which contributes to the increased computational complexity of the reference ARAIM algorithm.

Once the fault modes that need to be monitored are determined, fault consolidation can be used to reduce the number of monitored modes without a significant impact on the estimated protection level. The following Table III-15 shows the faults modes determined for a single observer when 15 satellites (GPS and Galileo) were usable for H-ARAIM Fault Detection only mode. This scenario has been selected given the choice of modes is not critical, and their order does not have a significant impact on performance, unlike for Fault Detection and Exclusion mode. Table III-15 presents the determined fault modes with correspondent probability of fault. Moreover the "Subsets" column reports the subsets of satellites belonging to GPS (PRN$\in$[1,24]) and Galileo (PRN$\in$[71,94]). The empty cells indicate the faulty satellite for a given constellation in the related fault mode (EUROCONTROL, 2022).

Table III-15: List of Fault Modes Before Consolidation

Num	Subsets (C_j)															$P_{fault,k}$
1	1	6	8	12	13	19	22	74	75	79	80	81	92	93	94	1
2	1	6	8	12	13	19	22									1,9993800681986E-04
3	1	6	8	12	13	19	22		75	79	80	81	92	93	94	2,9985602450848E-05
4	1	6	8	12	13	19	22	74		79	80	81	92	93	94	2,9985602450848E-05
5	1	6	8	12	13	19	22	74	75		80	81	92	93	94	2,9985602450848E-05
6	1	6	8	12	13	19	22	74	75	79		81	92	93	94	2,9985602450848E-05
7	1	6	8	12	13	19	22	74	75	79	80		92	93	94	2,9985602450848E-05
8	1	6	8	12	13	19	22	74	75	79	80	81		93	94	2,9985602450848E-05
9	1	6	8	12	13	19	22	74	75	79	80	81	92		94	2,9985602450848E-05
10	1	6	8	12	13	19	22	74	75	79	80	81	92	93		2,9985602450848E-05
11		6	8	12	13	19	22	74	75	79	80	81	92	93	94	9,9950009109338E-06
12	1		8	12	13	19	22	74	75	79	80	81	92	93	94	9,9950009109338E-06
13	1	6		12	13	19	22	74	75	79	80	81	92	93	94	9,9950009109338E-06
14	1	6	8		13	19	22	74	75	79	80	81	92	93	94	9,9950009109338E-06
15	1	6	8	12		19	22	74	75	79	80	81	92	93	94	9,9950009109338E-06
16	1	6	8	12	13		22	74	75	79	80	81	92	93	94	9,9950009109338E-06
17	1	6	8	12	13	19		74	75	79	80	81	92	93	94	9,9950009109338E-06
18	1	6	8	12	13	19	22			79	80	81	92	93	94	8,9959506137727E-10
19	1	6	8	12	13	19	22		75		80	81	92	93	94	8,9959506137727E-10
20	1	6	8	12	13	19	22		75	79		81	92	93	94	8,9959506137727E-10
21	1	6	8	12	13	19	22		75	79	80		92	93	94	8,9959506137727E-10
22	1	6	8	12	13	19	22		75	79	80	81		93	94	8,9959506137727E-10
23	1	6	8	12	13	19	22		75	79	80	81	92		94	8,9959506137727E-10
24	1	6	8	12	13	19	22		75	79	80	81	92	93		8,9959506137727E-10
25	1	6	8	12	13	19	22	74			80	81	92	93	94	8,9959506137727E-10
26	1	6	8	12	13	19	22	74		79		81	92	93	94	8,9959506137727E-10
27	1	6	8	12	13	19	22	74		79	80		92	93	94	8,9959506137727E-10
28	1	6	8	12	13	19	22	74		79	80	81		93	94	8,9959506137727E-10
29	1	6	8	12	13	19	22	74		79	80	81	92		94	8,9959506137727E-10
30	1	6	8	12	13	19	22	74		79	80	81	92	93		8,9959506137727E-10
31	1	6	8	12	13	19	22	74	75			81	92	93	94	8,9959506137727E-10
32	1	6	8	12	13	19	22	74	75		80		92	93	94	8,9959506137727E-10
33	1	6	8	12	13	19	22	74	75		80	81		93	94	8,9959506137727E-10
34	1	6	8	12	13	19	22	74	75		80	81	92		94	8,9959506137727E-10
35	1	6	8	12	13	19	22	74	75		80	81	92	93		8,9959506137727E-10
36	1	6	8	12	13	19	22	74	75	79			92	93	94	8,9959506137727E-10
37	1	6	8	12	13	19	22	74	75	79		81		93	94	8,9959506137727E-10
38	1	6	8	12	13	19	22	74	75	79		81	92		94	8,9959506137727E-10
39	1	6	8	12	13	19	22	74	75	79		81	92	93		8,9959506137727E-10
40	1	6	8	12	13	19	22	74	75	79	80			93	94	8,9959506137727E-10
41	1	6	8	12	13	19	22	74	75	79	80		92		94	8,9959506137727E-10
42	1	6	8	12	13	19	22	74	75	79	80		92	93		8,9959506137727E-10
43	1	6	8	12	13	19	22	74	75	79	80	81			94	8,9959506137727E-10
44	1	6	8	12	13	19	22	74	75	79	80	81		93		8,9959506137727E-10
45	1	6	8	12	13	19	22	74	75	79	80	81	92			8,9959506137727E-10
46		6	8	12	13	19	22		75	79	80	81	92	93	94	2,9985902309871E-10
47		6	8	12	13	19	22	74		79	80	81	92	93	94	2,9985902309871E-10
48		6	8	12	13	19	22	74	75		80	81	92	93	94	2,9985902309871E-10

The fault consolidation has the objective to group faults so that one subset solution can monitor multiple faults. As described in the ADD (WG-C ARAIM TSG, 2023), after establishing the initial list the algorithm consolidates multiple satellite faults from

the same constellation with constellation wide fault. The result of consolidation is that the fault modes in C_j are removed from the list and the probability of fault mode k_j is updated as shown in the following table.

Table III-16: List of Fault Modes After Consolidation

Num	Subsets ($C_j^{(updated)}$)															$P_{fault,k_j}^{(updated)}$
1	1	6	8	12	13	19	22	74	75	79	80	81	92	93	94	1
2	1	6	8	12	13	19	22									1,9996319548158E-04
3	1	6	8	12	13	19	22		75	79	80	81	92	93	94	2,9985602450848E-05
4	1	6	8	12	13	19	22	74		79	80	81	92	93	94	2,9985602450848E-05
5	1	6	8	12	13	19	22	74	75		80	81	92	93	94	2,9985602450848E-05
6	1	6	8	12	13	19	22	74	75	79		81	92	93	94	2,9985602450848E-05
7	1	6	8	12	13	19	22	74	75	79	80		92	93	94	2,9985602450848E-05
8	1	6	8	12	13	19	22	74	75	79	80	81		93	94	2,9985602450848E-05
9	1	6	8	12	13	19	22	74	75	79	80	81	92		94	2,9985602450848E-05
10	1	6	8	12	13	19	22	74	75	79	80	81	92	93		2,9985602450848E-05
11		6	8	12	13	19	22	74	75	79	80	81	92	93	94	9,9950009109338E-06
12	1		8	12	13	19	22	74	75	79	80	81	92	93	94	9,9950009109338E-06
13	1	6		12	13	19	22	74	75	79	80	81	92	93	94	9,9950009109338E-06
14	1	6	8		13	19	22	74	75	79	80	81	92	93	94	9,9950009109338E-06
15	1	6	8	12		19	22	74	75	79	80	81	92	93	94	9,9950009109338E-06
16	1	6	8	12	13		22	74	75	79	80	81	92	93	94	9,9950009109338E-06
17	1	6	8	12	13	19		74	75	79	80	81	92	93	94	9,9950009109338E-06
18		6	8	12	13	19	22		75	79	80	81	92	93	94	2,9985902309871E-10
19		6	8	12	13	19	22	74		79	80	81	92	93	94	2,9985902309871E-10
20		6	8	12	13	19	22	74	75		80	81	92	93	94	2,9985902309871E-10

4. Performance Sensitivity to ARAIM Design Parameters

Increasing P_{THRES} from $6x10^{-8}$ to $9x10^{-8}$ reduces the number of monitored fault hypotheses at the expense of an increase in P_{NM}. Increasing P_{THRES} could also: (1) decrease the number of monitored hypotheses with strong geometries, thus increasing PL (this occurs for FD Scenario 1), and (2) decrease the number of monitored hypotheses with weak geometries (this occurs for FDE scenario 2), thus decreasing PL. In other words, for the cases where increasing P_{THRES} removes strong geometries, there might be a slight increase in PL, and for the cases where it removes weak geometries (as it tends to be the case in FDE), there is generally a reduction in PL. In terms of availability for FDE, it is advantageous to use $9x10^{-8}$ for our current scenarios.

We also found that transitioning to the new Galileo ISDs resulted in smaller HPLs, and better performance in general.

IV. NUMERICAL EXAMPLES OF THE REFERENCE ARAIM ALGORITHM

A. Purpose of Providing Numerical Examples

The joint working group between the EUROCAE WG-62 and RTCA SC-159 WG-2 aims to promote cooperation and to create the necessary regulatory framework, developing minimum operational performance standards (MOPS) for the use of core GNSS systems and Satellite Based Augmentation System (SBAS) in civil aviation navigation applications. To date, the group develops the requirements specifications for airborne equipment detailed in (EUROCAE, 2023), will include H-ARAIM requirements in a future version. To this end, the working group established the ARAIM tools verification sub-group to support the validation of the requirements, and to enhance the clarity and consistency of the standards with the objective to improve common understanding among aviation equipment manufacturers, regulators, and users. The ARAIM verification sub-group establishes a shared framework of knowledge involving experts from various expertise and perspectives developing their tools in an independent manner. Demonstration of independent tools compliance with standards ensures a baseline level of quality and provides assurance to stakeholders. This joint commitment to meeting the designed standards promotes a shared understanding of what is expected in terms of navigation performance, as well as what needs to be done to achieve compliance with standards. Therefore, in the following section we provide a numerical example and illustrations

demonstrating the application of the standards offering practical guidance on how to interpret and implement the specified requirements. The numerical example may serve as a verification reference for implementing the ARAIM algorithm, aid in understanding, benchmarking, debugging, and customizing the algorithm for specific applications. By breaking down the algorithm into functional steps accompanied by numerical values, it can enhance understanding of each of the steps of the algorithm and provide assistance for verifying satisfaction at each step before moving to the next.

B. Step-by-Step Results of the Single-User ARAIM Algorithm Process

In the following, we will adhere to the ARAIM ADD (WG-C ARAIM TSG, 2023), utilizing the parameters and configuration agreed upon by the participating groups. We will provide the values of its primary variables for every step of the algorithm in order to serve the stated purpose above.

Upon the selection of GPS and Galileo almanacs, defined in Appendix F of EUROCASE ED-259 (EUROCAE, 2023), as well as determining the simulation time range, step size, and user location, the satellite-user geometry will be calculated. In order to focus on the ARAIM-related processing, we will exclude the detailed intermediate variables for the geometry calculation. Instead, we give the pseudorange covariance matrices C_{int}, C_{acc}, and the all-in-view geometry matrix $G_{all-in-view}$ which are necessary for ARAIM computation:

5. Pseudorange Covariance Matrices

Table IV-1: C_{int}, C_{acc}, and $G_{all-in-view}$

Sat	PRN	$diag(C_{\text{int}})$	$diag(C_{acc})$	$G_{all-in-view}$				
GPS	1	6.510343738	6.510343738	-0.608264367	0.76586296	-0.208490736	1	0
GPS	2	6.099745143	6.099745143	0.37865138	-0.7117265	-0.591665887	1	0
GPS	6	6.167822033	6.167822033	-0.725535499	-0.540961597	-0.425392512	1	0
GPS	8	6.613678273	6.613678273	-0.9806843	-0.064841099	-0.184537086	1	0
GPS	9	6.274484992	6.274484992	-0.288639732	-0.9052158	-0.311883729	1	0
GPS	10	6.410542825	6.410542825	0.935543489	0.258690111	-0.240494921	1	0
GPS	15	6.053473315	6.053473315	-0.26655246	-0.122049152	-0.956061604	1	0
GPS	18	6.222469483	6.222469483	0.167252792	0.919254193	-0.356368115	1	0
GPS	20	6.140898726	6.140898726	-0.414986131	0.776201176	-0.474655923	1	0
GPS	21	6.091785122	6.091785122	0.602138124	0.496642077	-0.62512105	1	0
Galileo	71	36.30966316	16.30966316	-0.228711141	-0.608479765	-0.759897092	0	1
Galileo	72	36.32791201	16.32791201	-0.754233402	0.130369486	-0.643533816	0	1
Galileo	78	36.67008644	16.67008644	0.401086297	-0.885834842	-0.233294697	0	1
Galileo	83	36.32934082	16.32934082	0.676025603	0.371195179	-0.636555985	0	1
Galileo	84	36.29766889	16.29766889	0.227545162	-0.397473253	-0.888953436	0	1
Galileo	85	36.43853348	16.43853348	-0.351326897	-0.854397481	-0.382850305	0	1
Galileo	88	36.71440439	16.71440439	0.80983206	0.544349129	-0.21876028	0	1
Galileo	89	36.43310732	16.43310732	0.080014866	0.917513722	-0.389571806	0	1
Galileo	90	36.9076092	16.9076092	-0.694042147	0.698471066	-0.17448114	0	1

6. All-in-View Position Solution

The geometry matrix $G_{all-in-view}$ (hereafter referred to as G) is a matrix with N_{sat} rows and 3+ $N_{constel}$ columns. Here, N_{sat} represents the combined number of GPS and Galileo satellites in view, while $N_{constel}$ indicates the number of independent constellations. The computation of the first three columns of G follows the procedure outlined in Appendix D

of ED-259A (EUROCAE, 2023) Each of the remaining columns in G corresponds to the clock reference of each respective constellation.

For this example, $N_{sat} = N_{GPS} + N_{Galileo}$ = 10 + 9 = 19.

When W is defined as $W = C_{\text{int}}^{-1}$, the values of the expression $(G^T WG)^{-1}$ are provided as follows:

Table IV-2: $(G^T WG)^{-1}$

1.710316798	-0.244187298	0.775613819	0.560080633	0.33974262
-0.244187298	1.440821848	-0.768184784	-0.489850252	-0.35079851
0.775613819	-0.768184784	11.03658748	5.041651743	5.298340879
0.560080633	-0.489850252	5.041651743	2.963730371	2.415094275
0.33974262	-0.35079851	5.298340879	2.415094275	6.598909293

7. Determination of the Faults that Need to be Monitored and the Associated Probabilities of Fault

The ARAIM ADD (WG-C ARAIM TSG, 2023) presents a possible way for establishing a list of fault modes to be monitored. At t=0, when considering the user position (0,0) specified in this example, a total of 157 fault modes were identified. To save space while preserving the ability to provide significant fault modes for numerical verification, we chose to select representative fault modes for each type of fault modes. The fault modes we listed in the tables include the single Galileo constellation fault, a single Galileo satellite fault, two simultaneous Galileo satellites faults, a simultaneous GPS and a Galileo satellite fault, a combination of faults of a Galileo satellite and the Galileo constellation, and a combination of a GPS satellite fault and the Galileo constellation fault.

It is worth noting that the probabilities associated with these fault modes remain the same within each fault mode type, assuming that the probability of satellite fault (P_{sat}) is equally assigned to each satellite within a given constellation.

Table IV-3: Fault Modes and Related Probability of Fault

Fault Mode Type	**ID**	idx_k	$fault_k$	$P_{fault}(T_{\text{exp}} = 0)$	$P_{fault}(T_{\text{exp}} = T)$
GAL_{const}	1	1,2,6,8,9,10,15,18,20,21	1002	1.99926011E-04	2.99805053E-04
GAL	2	1,2,6,8,9,10,15,18,20,21,72,78, 83,84,85,88,89,90	71	2.99838034E-05	4.99550164E-05
GPS	11	2,6,8,9,10,15,18,20,21,71,72, 78,83,84,85,88,89,90	1	9.99440122E-06	1.99814071E-05
GAL-GAL	21	1,2,6,8,9,10,15,18,20,21,78,83, 84,85,88,89,90	71,72	8.99541087E-10	2.49787571E-09
GPS-GAL	57	2,6,8,9,10,15,18,20,21,72,78, 83,84,85,88,89,90	1,71	2.99841032E-10	9.99120310E-10
GAL- GAL_{const}	147	1,2,6,8,9,10,15,18,20,21	71,1002	5.99796026E-09	1.49910022E-08
GPS- GAL_{const}	156	2,6,8,9,10,15,18,20,21	1,1002	1.99928010E-09	5.99622099E-09

where, fault '1001' denotes GPS constellation fault, and fault '1002' denotes Galileo constellation fault.

Upon implementing the fault consolidation process as described in paragraph III.C.3, the total number of fault modes are reduced to 112:

Table IV-4: Consolidated Fault Modes and Updated Probability of Fault

Fault Mode Type	ID	idx_k	$fault_k$	$P_{fault}(T_{\exp}=0)$	$P_{fault}(T_{\exp}=T)$
GAL_{const}	1	1,2,6,8,9,10,15,18,20,21	1002	2.00012376E-04	3.00029896E-04
GAL	2	1,2,6,8,9,10,15,18,20,21,72,78 ,83,84,85,88,89,90	71	2.99838034E-05	4.99550164E-05
GPS	11	2,6,8,9,10,15,18,20,21,71,72, 78,83,84,85,88,89,90	1	9.99440122E-06	1.99814071E-05
GAL-GAL	21	2,6,8,9,10,15,18,20,21,72,78, 83,84,85,88,89,90	1,71	2.99841032E-10	9.99120310E-10
GPS- GAL_{const}	111	2,6,8,9,10,15,18,20,21	1,1002	1.99928010E-09	5.99622099E-09

8. Fault-Tolerant Positions and Associated Standard Deviations and Biases

For all-in-view positioning, the weighted least-squares estimation matrix $S^{(0)}$ is given as follows:

Table IV-5: Weighted Least-Square Estimation Matrix $S^{(0)}$

Sat	PRN	$S^{(0)}$				
GPS	1	**-0.127330671**	0.14166866	0.258131221	0.183824001	0.128276233
GPS	2	0.151249726	-0.189069288	-0.106216874	0.088769566	-0.055975563
GPS	6	-0.142458598	-0.124084795	0.032360399	0.109873674	0.016941625
GPS	8	-0.188170076	-0.03054964	0.346882568	0.229200565	0.170392292
GPS	9	0.007260536	-0.236519565	0.330070074	0.266648162	0.156524894
GPS	10	0.298017876	-0.02508792	0.454611605	0.335150886	0.213393176
GPS	15	-0.100362259	0.022106212	-0.928886766	-0.321452535	-0.445726021
GPS	18	0.055486276	0.171563208	0.085517553	0.130241635	0.041990471
GPS	20	-0.115189211	0.178227196	-0.181580064	-0.006834102	-0.083550197
GPS	21	0.161496401	0.091745931	-0.290889716	-0.015421851	-0.142266911
Galileo	71	-0.013556434	-0.016191825	-0.077067542	-0.034317998	0.074593541
Galileo	72	-0.040773159	0.014192057	-0.068520813	-0.036216564	0.079477981
Galileo	78	0.028936127	-0.042155762	0.101312455	0.051744376	0.158435681
Galileo	83	0.025092605	0.013981629	-0.040955233	-0.016444023	0.0915425
Galileo	84	0.003760304	-0.008159471	-0.11104987	-0.048062408	0.058011185
Galileo	85	-0.009590085	-0.032985513	0.039980235	0.019392981	0.130378423
Galileo	88	0.038737258	0.010998659	0.084270198	0.040831462	0.150459212
Galileo	89	-0.001361648	0.034334151	0.009772379	0.001273	0.116381703
Galileo	90	-0.031244968	0.025986075	0.062258191	0.021799172	0.140719774

The monitor is designed to protect against the fault modes identified in Table IV-4. For a given fault mode idx_k, the weighted least-squares estimation matrix $S^{(k)}$ can be simply obtained by utilizing the relationship between $S^{(0)}$ and $S^{(k)}$ through rank one updates. In order to save space, we will not list the $S^{(k)}$ here one by one.

Next, the variances of the fault-tolerant position, and the variance of the difference between the all-in-view and the fault tolerant position solutions, for q from 1 to 3, are given by:

Table IV-6: Variances $\sigma_q^{(k)}$ and $\sigma_{ss,q}^{(k)}$

ID	$\sigma_q^{(k)}$ q=1,2,3			$\sigma_{ss,q}^{(k)}$ q=1,2,3		
1	1.3999005	1.2954596	3.6210605	0.3625543	0.3550718	1.0601415
2	1.3107751	1.2049773	3.3599325	0.0604682	0.0722233	0.3437581
11	1.3655123	1.2774336	3.4162480	0.3879872	0.4316763	0.7865475
21	1.3668899	1.2846709	3.4616847	0.3883900	0.4456969	0.8901124
111	1.4707203	1.4007246	3.7456090	0.5785693	0.6402264	1.4287783

The worst-case impact of the nominal biases for k[th] position solution are as follows:

Table IV-7 : $b_q^{(k)}$

ID	$b_q^{(k)}$		
1	1.162990	1.052701	2.658787
2	1.014692	0.909021	2.306466
11	0.967384	0.855614	2.216559
21	0.970928	0.857996	2.264672
111	1.126820	1.023000	2.639009

9. Solution Separation Threshold Tests

For each fault mode, there are three solution separation threshold tests $T_{k,q}$, one for each coordinate:

Table IV-8: $T_{k,q}$

ID	$T_{k,q}$		
1	2.5040569	2.4523775	2.5040569
2	0.4176361	0.4988252	0.4176361
11	2.6797150	2.9814626	2.6797150
21	2.6824967	3.0782989	2.6824967
111	3.9960097	4.4218577	3.9960097

10. Protection Levels

Finally, we conclude the numerical example by giving the HPL value. For the HPL computation, we first compute PL_q for q=1 and 2, and the HPL is given by $\sqrt{PL_1^2 + PL_2^2}$:

Table IV-9: Computed Horizontal components of Protection Level

PL_1	PL_2	HPL
16.2300	12.7259	20.6243

C. Numerical Illustrations for Special Cases

The PLs heavily depend on the number of satellites available and the resulting geometry. In case two constellations are being used (i.e., GPS and Galileo in this case), and exclusion is attempted on a constellation wide fault, the number of available satellites on the remaining constellation is likely to result in a poor geometry leading to a PL above the alert limit.

The following plot shows the minimum number of visible satellites over one day and considering both GPS and Galileo in baseline configuration of 24 satellites each.

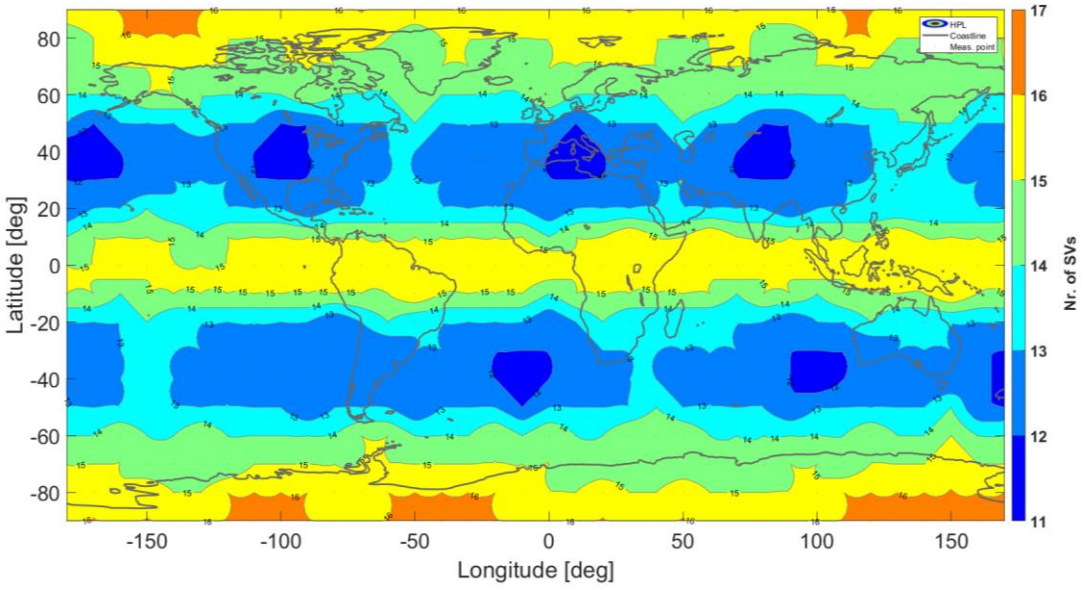

Figure IV-1: Minimum Number of GPS+GAL Visible Satellites Over One Day

Looking at the maximum HPL obtained over the same day using FDE and depicted in Figure III-2, the analogy with the previous plot is evident. The largest HPLs are located around latitudes of ±40deg and ±70deg. In particular, the two worst HPLs (magnitude of 5719.35m and 3815.83m as indicated in the plot) are located at ±40deg latitude and 170deg and -100deg longitude, respectively. The two locations match with the minima of the minimum number of satellites in Figure III-1.

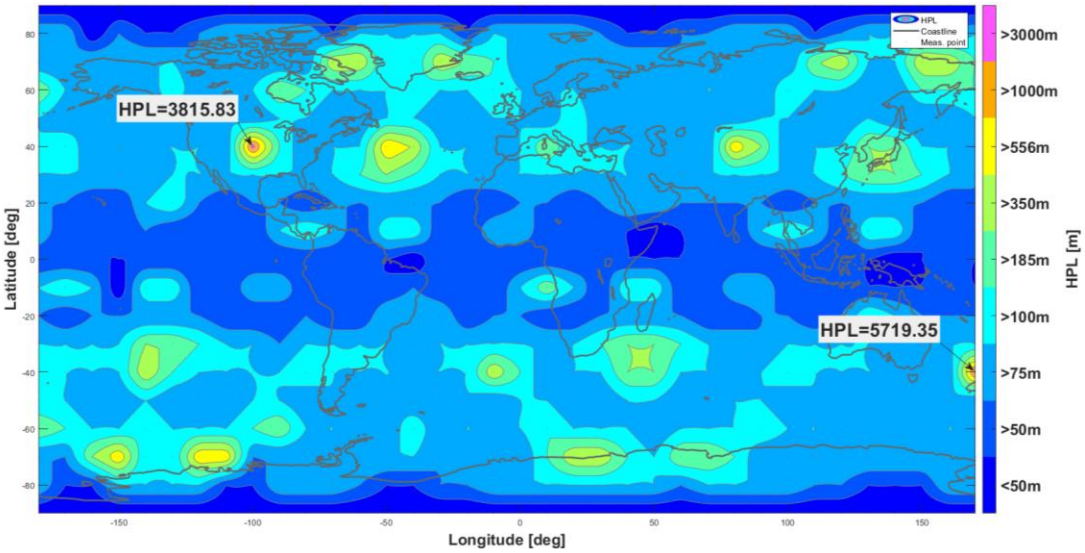

Figure IV-2: Maximum HPL Over One Day

When comparing the HPLs output by the different tools, a very good match within the PL tolerance could be obtained on every point of the global grid, but the two locations mentioned before, resulting in a mismatch between 1m and 3m for these latter, as indicated in the following plot showing the HPL comparison of two exemplary tools.

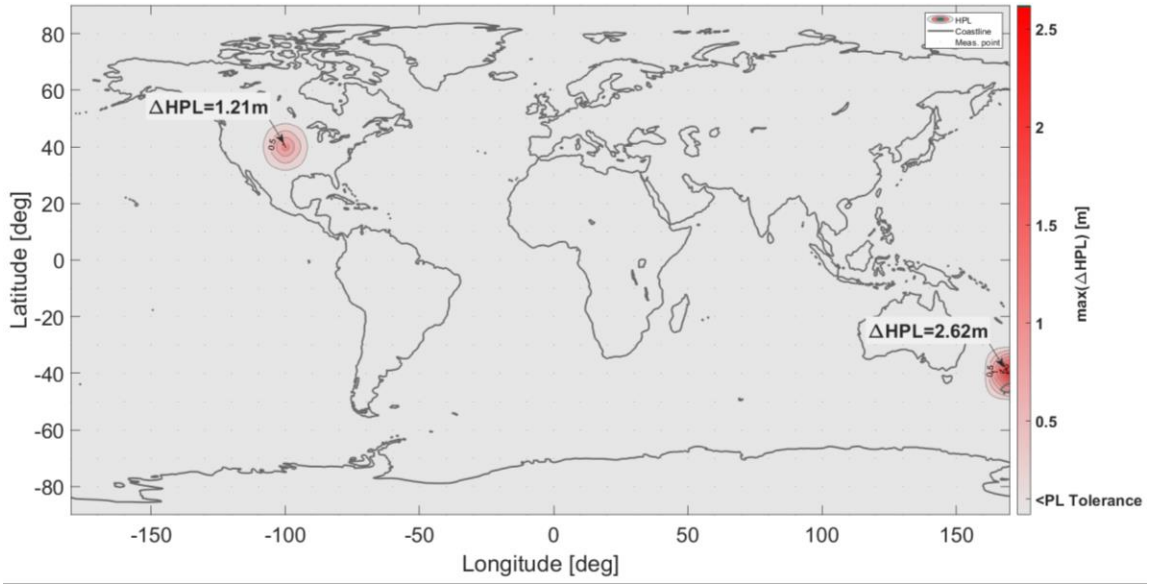

Figure IV-3: Maximum HPL difference between two exemplary ARAIM tools

It is worth looking closer at the epoch of the worst-case HPL located at [-40deg, 170deg] to understand the reason for the mismatch. At that particular epoch there were 5 GPS and 7 Galileo satellites. Under the assumption that the Galileo constellation fault was excluded, the remaining 5 GPS satellites have been used for the solution separation test. In particular, when a single GPS satellite is further excluded, only four satellites remain available for the PVT and HPL computation, resulting in the sky plot shown in Figure IV-4.

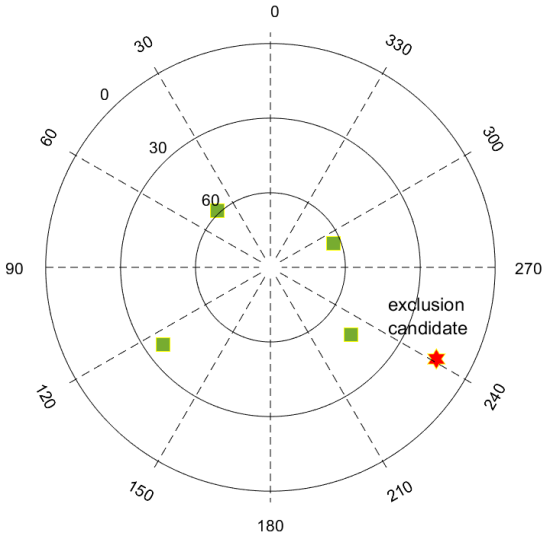

Figure IV-4: Worst-Case HPL - Sky-Plot After Exclusion of the GAL Constellation Fault

For that particular geometry, The matrix $G^T W G$ becomes ill-conditioned with a condition number of 4.3266 x 10^6, making computing the inverse matrix of $G^T W G$ extremely numerically sensitive to small perturbations and to the programming language (e.g., MATLAB, C++) and the methodology used for the inversion, thus resulting in the mismatch of the final PL computed by the different tools.

It is worth noting that such a mismatch, given the peculiar conditions described above, can be considered acceptable. Moreover, a variation of 3m over an HPL of about 5719.35m corresponds to about 0.05% relative difference, which can be reasonably considered negligible.

V. CONCLUSION

This paper describes the lessons learned based on the challenges we have encountered while cross-checking six independently coded software tools implementing a reference ARAIM user algorithm. Compared to RAIM, ARAIM is considerably more complex because it can account for constellations and satellites with heterogeneous probabilities of fault. This flexibility leads to important differences with RAIM and posed challenges in our verification. Two of these challenges concerned:

- the list of monitored faults. As opposed to RAIM, it is the user that forms the list of monitored modes based on the received (or default) ISD. In addition, this list is dynamic to a certain extent. One of the challenges we encountered while comparing the outputs of the tools was the fact that the list of monitored modes is not uniquely defined in the ADD. To successfully cross-check the tools, we needed to further constrain the choice of the monitored modes.
- the integrity allocation to exclusion candidates. Another important difference with RAIM is the explicit allocation of the integrity budget among the exclusion options (including the all-in-view). This feature, which is new compared to RAIM, initially led to important differences in the output HPLs.

To help clarify these and other points that may not be elaborated in the ADD, the paper includes a step-by-step numerical example of an HPL computation. For the participating groups from academia, industry, and government contractors that independently developed their ARAIM tools, the tasks detailed in this paper helped not only in effective ADD verification, but also in identifying potential areas for improvement in the ARAIM ADD. This verification effort encouraged collaboration and information sharing among ARAIM researchers and practitioners. We hope that through the description of the challenges we encountered and the numerical example, this paper will help in the implementation of ARAIM. In this sense, this paper can be seen as a companion to the ADD.

REFERENCES

[1] Working Group- Advanced RAIM Technical Subgroup (WG-C-ARAIM TSG), (2023) Advanced RAIM Reference Airborne Algorithm Description Document (ARAIM ADD), version 4.2, to be published by ARAIM Technical Subgroup

[2] Blanch, J., Walter, T., Enge, P., Lee, Y., Pervan, B., Rippl, M., Spletter, & A., Kropp, V., (2015) "Baseline Advanced RAIM User Algorithm and Possible Improvements," IEEE Transactions on Aerospace and Electronic Systems, Volume 51, No. 1, pp. 713-732, doi: 10.1109/TAES.2014.130739.

[3] Blanch, J., Walter, T., Milner, C., Joerger, M., Pervan, B., & Bouvet, D., (2022) "Baseline Advanced RAIM User Algorithm: Proposed Updates," *Proceedings of the 2022 International Technical Meeting of The Institute of Navigation*, Long Beach, California, pp. 229-251. https://doi.org/10.33012/2022.18254

[4] Working Group C, ARAIM Technical Subgroup (2016), Milestone 3 Report, February 26, 2016. Available at: http://www.gps.gov/policy/cooperation/europe/2016/working-group-c

[5] PEGASUS Interface Control Document, Issue S, March 2022, https://www.eurocontrol.int/tool/pegasus

[6] EUROCAE ED-259A Minimum Operational Performance Standards for Dual Frequency Multi Constellation Satellite Based Augmentation System Airborne Equipment (2023), revision A

[7] Zhai, Y., Joerger, M., Pervan, B. (2016) "H-ARAIM Exclusion: Requirements and Performance" Proceedings of ION GNSS+ Portland, OR. https://doi.org/10.33012/2016.14860

[8] Cassel, R., (2017) "Real-Time ARAIM Using GPS, GLONASS and Galileo", Master of Science in Mechanical and Aerospace Engineering, Illinois Institute of Technology

[9] Joerger, M., Stevanovic, S., Chan, F.-C., Langel, S., &Pervan, B., (2013) "Integrity Risk and Continuity Risk for Fault Detection and Exclusion Using Solution Separation ARAIM," *Proceedings of the 26th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+*, pp. 2702-2722 Nashville, TN

[10] Blanch, J., Walter, T., and Enge, P. (2017) Protection Levels after Fault Exclusion for Advanced RAIM. J Inst Navig, 64: 505–513. doi: https://doi.org/10.1002/navi.210

[11] RTCA Special Committee 159, "Minimum Operational Performance Standards for Global Positioning System/Satellite-Based Augmentation System Airborne Equipment," Document No. RTCA/DO-229E. Washington, DC., 2016

NOTICE

This work was produced for the U.S. Government under Contract 693KA8-22-C-00001 and is subject to Federal Aviation Administration Acquisition Management System Clause 3.5-13, Rights In Data-General, Alt. III and Alt. IV (Jan 2009).

The contents of this document reflect the views of the author and The MITRE Corporation and do not necessarily reflect the views of the Federal Aviation Administration (FAA) or the Department of Transportation (DOT). Neither the FAA nor the DOT makes any warranty or guarantee, expressed or implied, concerning the content or accuracy of these views.

For further information, please contact The MITRE Corporation, Contracts Management Office, 7515 Colshire Drive, McLean, VA 22102-7539, (703) 983-6000.