



Article

Fast Exclusion Candidate Identification Based on Sparse Estimation for ARAIM Fault Exclusion Process

Hangtian Qi ¹, Xiaowei Cui ¹ and Mingquan Lu ^{1,2,*}

¹ Department of Electrical Engineering, Tsinghua University, Beijing 100084, China; qht20@mails.tsinghua.edu.cn (H.Q.); cxw2005@tsinghua.edu.cn (X.C.)

² Beijing National Research Center for Information Science and Technology, Beijing 100084, China

* Correspondence: lumq@tsinghua.edu.cn

Abstract: Advanced receiver autonomous integrity monitoring (ARAIM) is an integrity technique for a global navigation satellite system (GNSS), centered on the multiple hypothesis solution separation (MHSS) test, which assesses the consistency between a subset and the all-in-view solution. Successful fault exclusion (FE) in ARAIM relies on identifying exclusion candidates that ensure no faults among the remaining satellites, a process requiring computationally expensive MHSS tests. The existing methods guide exclusion candidate searches based on the size of the normalized solution separation statistics, i.e., the normalized absolute difference between the subset solution and the all-in-view solution. However, in scenarios involving more than one satellite fault, these statistics can become unreliable due to fault diversity and interactions, perhaps misleading the FE process and causing its failure. To overcome this issue, our study proposes employing sparse estimation to simply identify satellite faults in one go, leveraging the sparsity of faulty satellites compared to the total number of observations in civil aviation GNSSs. Unlike the existing methods, which infer the fault likelihood indirectly through solution separation statistics, our approach represents an improvement that directly indicates potential exclusion candidates. Our experiments demonstrate that this method is fast and accurate. As a fundamentally different approach, it serves as a valuable complement or an alternative to the existing methods, enhancing the success and efficiency of the ARAIM FE process.



Citation: Qi, H.; Cui, X.; Lu, M. Fast Exclusion Candidate Identification Based on Sparse Estimation for Advanced Receiver Autonomous Integrity Monitoring Fault Exclusion Process. *Remote Sens.* **2024**, *16*, 3537. <https://doi.org/10.3390/rs16183537>

Academic Editors: Krzysztof Naus and Mieczysław Bakuła

Received: 12 August 2024

Revised: 15 September 2024

Accepted: 17 September 2024

Published: 23 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: GNSS; integrity; ARAIM; fault exclusion; sparse estimation

1. Introduction

With the increasing demand for global navigation satellite system (GNSS) applications in life safety, integrity research has garnered significant attention. Receiver autonomous integrity monitoring (RAIM), a classic integrity technology, is widely adopted in civil aviation due to its fast alerting and independence from additional equipment [1]. The evolution of GNSS constellations motivates the development of advanced RAIM (ARAIM) to harness multi-constellation and dual-frequency signals for vertical services [2]. The EU-US Cooperative Working Group-C (WG-C) has led the effort to develop ARAIM [3] and formulated the reference ARAIM algorithm description document (ADD) [4]. The core of ARAIM as described in the ADD is the multiple hypothesis solution separation (MHSS) test. It assumes possible satellite faults, known as fault modes, and compares the consistency between the subset (i.e., the remaining satellites after removing a fault mode) and the all-in-view (i.e., incorporating all the satellites used) positioning solution. ARAIM then calculates the protection level (PL) based on the MHSS results, providing a reliability measure for the declared fault-free positioning solution. Overall, the ARAIM algorithm consists of three steps [5]: first, forming a list of the fault modes monitored; second, performing fault detection and exclusion (FDE) via solution separation; and finally, generating integrity indicators.

ARAIM is still in the experimental testing stage, with room for improvement in its baseline algorithm [6]. This paper focuses on the second step of the ARAIM baseline

algorithm—the FDE process—and specifically on the FE operation. The crux of ARAIM’s FE operation is searching for an exclusion candidate, which, when removed, produces a fault-free subset of satellites that pass the MHSS test [4,7]. Each candidate verification involves a complete MHSS test, equivalent to the complexity of one fault detection (FD), which is computationally expensive [8]. Consequently, a prioritized search guide is necessary, as we cannot rely on luck to find an exclusion candidate among the many fault modes. Otherwise, the computational load from blindly searching may prevent users from finding a candidate within the required time, leading to FE failure [9]. Even if an exclusion candidate is found, this may be down to chance, and it may consume significant computational resources. Therefore, providing the search order for the exclusion candidates is crucial for successful and fast FE, which is the research objective of this paper.

There are two existing methods for determining the priority order for the exclusion candidate search. The first method considers the severity of the subset solution separation statistic (i.e., the absolute difference between the subset solution and the all-in-view solution) exceeding the corresponding detection threshold [10]. This statistic is normalized by dividing it by the detection thresholds to eliminate the impact of different subsets’ detection thresholds. The greater the exceedance, the higher the likelihood of the fault mode being real, thus prioritizing it for verification. This is an intuitive approach. The second method determines the verification priority order based on the chi-square statistic of the subset’s pseudorange residuals [4]. The smaller the chi-square statistic of the pseudorange residuals for a subset, the more likely the subset is to be free of faults [11], making its corresponding fault mode the likely exclusion candidate. This method is recommended by many studies [6,7,12]. Indeed, literature [13] shows that these two methods are essentially the same because the chi-square statistic of the subset in the second method is inversely proportional to the normalized solution separation statistic in the first method.

However, we recognize that the existing methods have a shortcoming: using the subset’s solution separation statistic as the basis for determining the exclusion candidate order is effective for single-satellite faults but not for multi-satellite faults. In multi-fault conditions, due to the diversity of the faults and their interactions, the solution separation statistic for the actual fault may not be significant, and it could be small or even the smallest value. Specifying the order according to the existing method and searching for exclusion candidates through MHSS tests can lead to a waste of resources and may even fail to identify the exclusion candidate. This situation means that the current method based on the solution separation statistic can mislead the search for exclusion candidates. Relying solely on this method is insufficient and not robust, which not uncommonly hinders the FE process.

To achieve a faster and more successful FE process and to address the potential for the search for exclusion candidates to be misled in the presence of multi-satellite faults using the current methods, we propose a new approach to aid in finding exclusion candidates. This is the main contribution of this paper. Unlike existing methods that infer the fault likelihood indirectly through solution separation statistics, our proposed method quickly and directly estimates which satellites are faulty in one go. Considering the sparsity of faulted pseudoranges compared to the total pseudorange observations in civil aviation applications and leveraging the consistency of fault-free pseudoranges, this study employs sparse estimation to directly identify faults. Related experiments demonstrate that this method is fast, efficient, and accurate. As a fundamentally different approach, our method is a valuable substitution for or complement to existing methods, facilitating smooth execution of the FE process.

This paper is organized as follows. Section 2 introduces the fundamentals of ARAIM and related works. Section 3 presents the motivation behind this study, detailing why solution separation statistics are unsuitable for searching for exclusion candidates in cases with multiple satellite faults. Section 4 describes the proposed method, which uses sparse estimation to directly identify potential faults. Section 5 evaluates the effectiveness and efficiency of this approach through experiments. Finally, this paper concludes with the key findings.

2. Related Works

This section provides the groundwork for our research and consists of two main subsections. The first subsection briefly outlines the fundamentals of ARAIM, including the MHSS test and the FE process. The second subsection presents the existing methods for determining the search order for exclusion candidates, emphasizing their essential sameness.

2.1. ARAIM Fundamentals

2.1.1. The MHSS Test

ARAIM technology for civil aviation targets the integrity of GNSS standard point positioning (SPP). The classic linearization process for SPP can be represented as [14]

$$\mathbf{y} = \mathbf{G}\mathbf{x} + \mathbf{e} \quad (1)$$

where

- $\mathbf{y} \in \mathbb{R}^{N_{sat}}$ is the pseudorange difference vector (the difference between the measured pseudorange and the pseudorange generated by the position estimate). For ARAIM, pseudoranges are typically obtained through ionosphere-free combinations of carrier-smoothed processing.
- $\mathbf{G} \in \mathbb{R}^{N_{sat} \times (3+N_{const})}$ is the observation geometry matrix indicating the geometric relationship between satellites and the receiver. Each row represents the linearized line-of-sight direction from the satellite to the receiver.
- $\mathbf{x} \in \mathbb{R}^{(3+N_{const})}$ is the state difference vector (the difference between the previous state estimate and the current state estimate), including the receiver's position and clock bias.
- $\mathbf{e} \in \mathbb{R}^{N_{sat}}$ is the pseudorange error vector, which includes various errors mapped to the pseudorange, such as the satellite clock; ephemeris; ionospheric delay; tropospheric, multipath errors; and receiver noise errors [15]. \mathbf{e} is modeled as a Gaussian distribution, and even if it does not conform perfectly, a Gaussian distribution can always be found to overbound it [16,17]. The covariance matrix of \mathbf{e} is a diagonal matrix \mathbf{C} , where the diagonal elements represent the square of user equivalent range errors (UERE), indicating the nominal errors of each pseudorange.
- N_{sat} is the number of satellites, and N_{const} is the number of constellations. Unless otherwise noted, the vectors described in this paper are column vectors.

For SPP, the above model, Equation (1), incorporates a fault bias vector $\mathbf{b} \in \mathbb{R}^{N_{sat}}$ to reflect malfunctions:

$$\mathbf{y} = \mathbf{G}\mathbf{x} + \mathbf{e} + \mathbf{b} \quad (2)$$

The basic principle of solution separation is that the difference between the positioning solution using all satellites and the solution after removing the faulty satellite(s) should be significant. Accordingly, ARAIM first assumes N_{fault} different faults, i.e., fault modes, in carrying out the solution separation test. These fault modes form a monitor list labeled from 1 to N_{fault} . The monitored fault modes must be numerous enough to ensure their combined a priori probability exceeds a predetermined threshold. Additionally, an important concept corresponding to fault modes is the “subset”, which refers to the set of remaining satellites after removing the fault mode(s) from the all-in-view satellites (i.e., all the satellites used for positioning).

The second step of ARAIM is to perform solution separation for each fault mode in the monitor list. For fault mode k , using the weighted least squares (WLSQ) method to solve Equation (1), the positioning solution for subset k , $\hat{\mathbf{x}}^{(k)}$, can be obtained:

$$\hat{\mathbf{x}}^{(k)} = \mathbf{S}^{(k)}\mathbf{y} \quad (3)$$

where

$$\mathbf{S}^{(k)} = (\mathbf{G}^T\mathbf{W}^{(k)}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}^{(k)} \quad (4)$$

In the above, Equations (3) and (4), superscript (k) represents the value corresponding to the removal of fault mode k . Specifically, superscript (0) represents the all-in-view value. Accordingly, $\hat{x}^{(0)}$ is the all-in-view positioning solution. $W^{(0)}$ is the weight matrix used in WLSQ to solve $\hat{x}^{(0)}$, which is the inverse of the covariance matrix C and is a diagonal matrix. $W^{(k)}$ is the weight matrix for solving $\hat{x}^{(k)}$, obtained by setting the diagonal elements corresponding to the faulty satellites of fault mode k in $W^{(0)}$ to zero.

Both the subset solution $\hat{x}^{(k)}$ and the all-in-view solution $\hat{x}^{(0)}$ are vectors, where the first three elements are the position solutions in three directions. The ADD denotes these using subscript q : $q = 1, 2$ for the two horizontal directions, and $q = 3$ for the vertical direction. If

$$|\hat{x}_q^{(k)} - \hat{x}_q^{(0)}| \leq T_q^{(k)} \quad (5)$$

For all the fault modes monitored and all three directions, the measurements are consistent, and no malfunction is detected. The detection threshold for the k th subset is given by

$$T_q^{(k)} = K_{fa,q} \sigma_{ss,q}^{(k)} \quad (6)$$

where $K_{fa,q}$ is a coefficient related to the probability of a false alarm and the number of fault modes N_{fault} , which does not change in the current epoch; $\sigma_{ss,q}^{(k)}$ is the standard deviation model that $\hat{x}_q^{(0)} - \hat{x}_q^{(k)}$ should follow. A description of Equations (3)–(6) and the relevant parameters is detailed in the ADD [4].

If $|\hat{x}_q^{(k)} - \hat{x}_q^{(0)}| > T_q^{(k)}$ occurs, ARAIM detects a fault, and FE can be performed. The process for FE is introduced in Section 2.1.2.

After completing the FDE step, ARAIM finally generates integrity indicators for the fault-free solution, either when no fault is detected or after successfully excluding faults. The integrity indicator can be the positioning solution's protection level or integrity risk.

2.1.2. The FE Process

The prerequisite for fault exclusion to be functional is to find at least one subset that is considered fault-free to provide a reliable positioning solution. Naturally, this fault-free subset must pass the MHSS test. The fault mode corresponding to this subset is defined as an “exclusion candidate”, meaning that this fault mode can be selected as the exclusion target [4]. An exclusion candidate j must satisfy the following condition for all monitored fault modes k :

$$|\hat{x}_q^{(k)} - \hat{x}_q^{(j,k)}| \leq T_q^{(j,k)} \quad (7)$$

where $\hat{x}_q^{(j,k)}$ is the subset solution after removing faults j and k , and $T_q^{(j,k)}$ is the corresponding detection threshold.

Verifying whether a fault mode is an exclusion candidate requires a complete MHSS test, which is computationally expensive. It is important to note that the MHSS test in FD, as shown in Equation (5), and the MHSS test in FE, as shown in Equation (7), are two distinct processes. The former checks for faults in the all-in-view satellites, while the latter verifies whether the remaining satellites after exclusion are fault-free. Ideally, the exclusion process requires just one MHSS test, approximately equivalent to the computational load of FD. However, FE often requires multiple MHSS tests to find a fault-free subset, as identifying an exclusion candidate on the first attempt is quite challenging. As a result, the computational burden of FE is typically several times higher than that of FD, making the FE process more resource-intensive.

Given that the computational load of a one-time MHSS test is substantial [8], it is therefore impractical to perform an MHSS test for each fault mode in the monitoring list one by one or randomly without any strategy. Otherwise, the possible consequence is that ARAIM may not find an exclusion candidate within the required time, and the exclusion process will fail. Even if an exclusion candidate is discovered by chance, such a laborious process is detrimental to the real-time performance of ARAIM. In response, the ARAIM

user algorithm must set a priority order to verify whether a fault mode is an exclusion candidate. This is our study's objective, which will be explored in depth later.

The exclusion candidates may not be unique. Exclusion candidate j is not necessarily the actual fault as long as the subset solution's integrity risk, including the wrong exclusion risk, is correctly calculated because the MHSS test has already confirmed its subset is fault-free. Thus, a simple and common approach is to immediately exclude a candidate once it is found according to the verification order without continuing to search for other candidates, aiming to reduce the computational complexity of ARAIM's fault exclusion process. Furthermore, since the exclusion candidate is obtained based on a prioritized order guided by a specific method, the first candidate found is likely the actual fault among all possible exclusion candidates.

2.2. Existing Methods for Setting the Search Order

To reduce the computational load of FE and increase its success, the ARAIM user algorithm must set a priority order for searching for exclusion candidates. Currently, there are two methods.

(1) The first method considers the severity of the fault mode exceeding its detection threshold. The greater the ratio of the exceedance to the threshold, the higher the likelihood of the fault mode being an exclusion candidate, and the higher its verification priority [10]. This means that the method is based on

$$\frac{|\hat{x}_q^{(0)} - \hat{x}_q^{(k)}|}{T_q^{(k)}} \quad (8)$$

where dividing by threshold $T_q^{(k)}$ eliminates the impact of the different thresholds for different subsets. As shown in Equation (6), $T_q^{(k)}$ is proportional to $\sigma_{ss,q}^{(k)}$, so the above criterion is equivalent to

$$\frac{|\hat{x}_q^{(0)} - \hat{x}_q^{(k)}|}{\sigma_{ss,q}^{(k)}} \quad (9)$$

This method essentially examines the ratio of the actual solution separation statistic to the nominal solution separation standard deviation. Accordingly, it is referred to as the normalized solution separation statistic method.

(2) The other method prioritizes based on the chi-square statistic of the pseudorange residuals for each satellite subset [13]. The chi-square statistic for subset k is calculated as

$$\chi^2 = \mathbf{y}^T \left(\mathbf{W}^{(k)} - \mathbf{W}^{(k)} \mathbf{G} (\mathbf{G}^T \mathbf{W}^{(k)} \mathbf{G})^{-1} \mathbf{G}^T \mathbf{W}^{(k)} \right) \mathbf{y} \quad (10)$$

This method is inspired by the traditional RAIM approach of examining the pseudorange residuals: subsets with lower χ values are more likely to be fault-free, and their corresponding fault modes should be given higher priority for verification as exclusion candidates.

Finally, it is essential to emphasize that Appendix F in Ref. [13] has confirmed that the chi-square statistic size for subset k directly reflects the size of k 's normalized solution separation statistic. Indeed, the two existing methods for quickly delineating the search order are the same, with both grounded in the solution separation statistic.

3. Motivation: Challenges in Solution-Separation-Statistic-Based Searches

Sections 3 and 4 are the core of our work. This section proves that the solution separation statistic is not a good criterion for guiding the exclusion candidate search under multi-satellite failures due to the uncertainties in the satellite geometry and faults. More specifically, when more than one satellite malfunctions, the actual fault's corresponding solution separation statistic—i.e., the absolute difference between its subset solution and the all-in-view solution, influenced by $\mathbf{S}^{(k)}$, $\mathbf{S}^{(0)}$, and \mathbf{b} ,—may not be the largest, may not be

significant, and may even be below the detection threshold. In these cases, it is challenging to find exclusion candidates using the existing methods.

Before an in-depth analysis, we will define some concepts for clarity.

- Single-satellite fault mode: The assumption that only one satellite has a fault.
- Dual-satellite fault mode: The assumption that two satellites have faults simultaneously.
- Multi-satellite fault mode: The assumption that more than one satellite has faults simultaneously. A dual-satellite fault mode is a type of multi-satellite fault mode.
- Single-satellite real fault: A real occurrence of only one satellite having a fault.
- Dual-satellite real fault: A real occurrence of two satellites having faults simultaneously.
- Multi-satellite real fault mode: A real occurrence of more than one satellite having faults simultaneously. A dual-satellite real fault is a type of multi-satellite real fault.

Furthermore, the solution separation statistics discussed in this section are not normalized, as $\sigma_{ss,q}^{(k)}$ is influenced by the error model and introduces significant difficulty into the analysis. Fortunately, the solution separation statistic is the decisive factor affecting the normalized solution separation statistic, and lenient qualitative conclusions about it can be extended to the normalized solution separation statistic. For instance, if the solution separation statistic is large, its normalized value is usually also large; if the solution separation statistic is close to zero, its normalized value is similarly close to zero.

According to Equation (3), $\hat{\mathbf{x}}^{(0)} - \hat{\mathbf{x}}^{(k)}$ can be obtained by

$$\hat{\mathbf{x}}^{(0)} - \hat{\mathbf{x}}^{(k)} = (\mathbf{S}^{(0)} - \mathbf{S}^{(k)})\mathbf{y} \quad (11)$$

Substituting Equation (4) into the above, Equation (11), we have

$$\begin{aligned} \hat{\mathbf{x}}^{(0)} - \hat{\mathbf{x}}^{(k)} &= (\mathbf{S}^{(0)} - \mathbf{S}^{(k)})\mathbf{G}\mathbf{x} + (\mathbf{S}^{(0)} - \mathbf{S}^{(k)})(\mathbf{b} + \mathbf{e}) \\ &= \mathbf{S}^{(0)}\mathbf{G}\mathbf{x} - \mathbf{S}^{(k)}\mathbf{G}\mathbf{x} + (\mathbf{S}^{(0)} - \mathbf{S}^{(k)})(\mathbf{b} + \mathbf{e}) \\ &= \mathbf{x} - \mathbf{x} + (\mathbf{S}^{(0)} - \mathbf{S}^{(k)})(\mathbf{b} + \mathbf{e}) \\ &= (\mathbf{S}^{(0)} - \mathbf{S}^{(k)})(\mathbf{b} + \mathbf{e}) \\ &= \mathbf{S}^k(\mathbf{b} + \mathbf{e}) \end{aligned} \quad (12)$$

where we define $\mathbf{S}^{(0)} - \mathbf{S}^{(k)}$ as \mathbf{S}^k :

$$\mathbf{S}^{(0)} - \mathbf{S}^{(k)} = \mathbf{S}^k \quad (13)$$

It is not hard to see that \mathbf{S}^k has the two following significant characteristics.

- (1) All the columns in \mathbf{S}^k are zero vectors except for those corresponding to the faulty satellite, indicated by fault mode k .
- (2) The non-zero column vectors in \mathbf{S}^k are those in $\mathbf{S}^{(0)}$. For clarity, we will give two examples.

Example 1. Assume a single-satellite fault mode m , where the A th satellite fails. Accordingly, \mathbf{S}^m , i.e., $\mathbf{S}^{(0)} - \mathbf{S}^{(m)}$, has zero elements except for the A th column. The A th column in \mathbf{S}^m is the same as $\mathbf{S}^{(0)}$'s A th column. Then, \mathbf{S}^m is expressed as

$$\mathbf{S}^m = [\mathbf{0} \dots \mathbf{0}, \underset{\substack{\downarrow \\ A\text{th}}}{\mathbf{S}_A^{(0)}}, \mathbf{0} \dots \mathbf{0}] \quad (14)$$

where $\mathbf{0}$ and $\mathbf{S}_A^{(0)}$ are both column vectors of $N_{\text{const}} + 3$ dimensions; $\mathbf{S}_A^{(0)}$ is the A th column vector of $\mathbf{S}^{(0)}$, located in the A th column of \mathbf{S}^m .

Example 2. Assume a dual-satellite fault mode n , where the A th and B th satellites simultaneously fail. Accordingly, \mathbf{S}^n , i.e., $\mathbf{S}^{(0)} - \mathbf{S}^{(n)}$, has zero elements except for the A th and B th columns. The A th and B th columns in \mathbf{S}^n are the same as those in $\mathbf{S}^{(0)}$. Then, \mathbf{S}^n is expressed as

$$\mathbf{S}^n = [\mathbf{0} \dots \mathbf{0}, \underset{\downarrow \text{Ath}}{\mathbf{S}_A^{(0)}}, \mathbf{0} \dots \mathbf{0}, \underset{\downarrow \text{Bth}}{\mathbf{S}_B^{(0)}}, \mathbf{0} \dots \mathbf{0}] \quad (15)$$

where $\mathbf{S}_B^{(0)}$ is the B th column vector of $\mathbf{S}^{(0)}$, located in the B th column of \mathbf{S}^n .

These two characteristics can be explained by the physical principles of solution separation: in $\mathbf{S}^{(k)}$, the corresponding column is zero, meaning the pseudorange of the corresponding satellite does not participate in positioning. Therefore, the distinction between $\mathbf{S}^{(k)}$ and $\mathbf{S}^{(0)}$ is only in the former's corresponding column becoming a zero vector. Mathematically, these characteristics are determined by the difference between $\mathbf{W}^{(k)}$ and $\mathbf{W}^{(0)}$ in $\mathbf{S}^{(k)}$ and $\mathbf{S}^{(0)}$ shown in Equation (4). Detailed proofs can be found in reference [18]. Additionally, they are demonstrated in the generation functions of $\mathbf{S}^{(k)}$ and $\mathbf{S}^{(0)}$ in Stanford University's ARAIM simulation software, MATLAB Algorithm Availability Simulation Tool (MASST) [19].

Next, without a loss of generality, we use the vertical direction ($q = 3$) to show that variation in the solution separation statistic, i.e., $|\hat{x}_3^{(0)} - \hat{x}_3^{(k)}|$, in cases of single- and dual-satellite real faults.

(1) When the real fault is a single-satellite fault

Given only one satellite failure, the fault bias vector \mathbf{b} can be represented as

$$\mathbf{b} = [0 \dots 0, b_{real}, 0 \dots 0]^T \quad (16)$$

where b_{real} is the fault bias of a specific satellite's pseudorange, which can be either positive or negative and typically has a magnitude larger than the nominal error.

For vertical direction $q = 3$ and fault mode m , it is known from Equation (12) that

$$\hat{x}_3^{(0)} - \hat{x}_3^{(m)} = \mathbf{S}_3^m (\mathbf{b} + \mathbf{e}) \quad (17)$$

where \mathbf{S}_3^m is the third row of \mathbf{S}^m , that is,

$$\mathbf{S}_3^m = [0 \dots 0, \underset{\downarrow \text{Ath}}{\mathbf{S}_{3,A}^{(0)}}, 0 \dots 0] \quad (18)$$

Clearly, $\mathbf{S}_{3,A}^{(0)}$ is the third element of $\mathbf{S}_A^{(0)}$, i.e., the element in the third row and the A th column of $\mathbf{S}^{(0)}$. Equations (17) and (18) describe the results for a single-satellite fault mode. Subsequently, a dual-satellite fault mode is tested under the condition that a single real fault is present.

For the dual-satellite fault mode n , it is known that

$$\hat{x}_3^{(0)} - \hat{x}_3^{(n)} = \mathbf{S}_3^n (\mathbf{b} + \mathbf{e}) \quad (19)$$

$$\mathbf{S}_3^n = [0 \dots 0, \underset{\downarrow \text{Ath}}{\mathbf{S}_{3,A}^{(0)}}, 0 \dots 0, \underset{\downarrow \text{Bth}}{\mathbf{S}_{3,B}^{(0)}}, 0 \dots 0] \quad (20)$$

where $\mathbf{S}_{3,B}^{(0)}$ is the third element of $\mathbf{S}_B^{(0)}$, i.e., the element in the third row and the B th column of $\mathbf{S}^{(0)}$.

Then, \mathbf{e} is represented as

$$\mathbf{e} = [e_1 \dots e_A \dots e_B \dots e_i \dots e_{N_{sat}}]^T \quad (21)$$

where e_i is the i th element of \mathbf{e} . Note that the absolute value of e_i is much smaller than that of fault b_{real} since the former represents nominal errors.

Once we know S_3^m , S_3^n , b , and e and that b_{real} is much larger than e_i , the following findings can easily be reached.

- **For the single-satellite fault mode m :** $|\hat{x}_3^{(0)} - \hat{x}_3^{(m)}|$ is maximized to $|S_{3,A}^{(0)}(b_{real} + e_A)|$ when $S_{3,A}^{(0)}$ and b_{real} are at the same location in their respective vectors, i.e., when $S_{3,A}^{(0)}$ and b_{real} are both the A th element. Otherwise, i.e., if $S_{3,A}^{(0)}$ and b_{real} are at different locations, $|\hat{x}_3^{(0)} - \hat{x}_3^{(m)}|$ becomes a small value, $|S_{3,A}^{(0)}e_A|$. This indicates that the solution separation statistic for a single-satellite fault mode is large or maximized when it is precisely the real single-satellite fault.
- **For the dual-satellite fault mode n :** $|\hat{x}_3^{(0)} - \hat{x}_3^{(n)}|$ is generally maximized when b_{real} is the A th or B th element in b , reaching $|S_{3,A}^{(0)}(b_{real} + e_A) + S_{3,B}^{(0)}e_B|$ or $|S_{3,A}^{(0)}e_A + S_{3,B}^{(0)}(b_{real} + e_B)|$. Otherwise, $|\hat{x}_3^{(0)} - \hat{x}_3^{(n)}|$ will be $|S_{3,A}^{(0)}e_A + S_{3,B}^{(0)}e_B|$, which is noticeably decreased. This indicates that the solution separation statistic for the dual-satellite fault mode is large or maximized when it includes the single-satellite real fault.

Given that b is much more significant than e in terms of pseudorange deviation, we approximate $(b + e)$ as b , which does not affect the qualitative analysis. Under this simplification, our discussion is reduced to the locations of non-zero elements in S_3^k relative to b_{real} . Here, k is not just limited to single- or dual-satellite fault modes (m or n) but can also be extended to more than two satellite fault modes. Consequently, the solution separation statistic is not zero only when one of the non-zero elements in S_3^k corresponds to the non-zero element b_{real} . In other words, only when fault mode k contains the single-satellite real fault will $|\hat{x}_3^{(0)} - \hat{x}_3^{(k)}|$ be other than zero.

In summary, if the real fault is a single-satellite fault, the fault mode k with the greatest or maximum value for $|\hat{x}_3^{(0)} - \hat{x}_3^{(k)}|$ is likely to contain (or even be) the real fault. At this point, we have the first key conclusion.

Conclusion 1: When a single-satellite fault occurs, the solution separation statistic is an excellent criterion for determining the search order for exclusion candidates.

(2) When the real fault is a dual-satellite fault

However, our conclusion changes when it comes to a dual-satellite real fault.

Given two satellite failures, vector b is represented as

$$b = [0 \dots 0, b_{real,1}, 0 \dots 0, b_{real,2}, 0 \dots 0]^T \quad (22)$$

where $b_{real,1}$ and $b_{real,2}$ are the fault biases imposed on the two pseudoranges.

Let us assume that the dual-satellite fault mode n corresponds to the real fault, that is,

$$b = [0 \dots 0, \underset{\substack{\downarrow \\ A\text{th}}}{b_{real,1}}, 0 \dots 0, \underset{\substack{\downarrow \\ B\text{th}}}{b_{real,2}}, 0 \dots 0] \quad (23)$$

$S_A^{(0)}$ and $S_B^{(0)}$ in S_3^n correspond to $b_{real,1}$ and $b_{real,2}$, respectively; then,

$$|\hat{x}_3^{(0)} - \hat{x}_3^{(n)}| = |S_{3,A}^{(0)}(b_{real,1} + e_A) + S_{3,B}^{(0)}(b_{real,2} + e_B)| \quad (24)$$

Unfortunately, $|S_{3,A}^{(0)}(b_{real,1} + e_A) + S_{3,B}^{(0)}(b_{real,2} + e_B)|$ may not be large, and it might not even exceed the detection threshold. This is because $S_{3,A}^{(0)}(b_{real,1} + e_A)$ and $S_{3,B}^{(0)}(b_{real,2} + e_B)$ might have opposite signs, i.e., $S_{3,A}^{(0)}(b_{real,1} + e_A) \cdot S_{3,B}^{(0)}(b_{real,2} + e_B) < 0$. Particularly, it is noticed that $b_{real,1}$ and $b_{real,2}$ could be positive or negative due to the variability in faults.

Given the results of Equation (24) and if $S_{3,A}^{(0)}(b_{real,1} + e_A) \cdot S_{3,B}^{(0)}(b_{real,2} + e_B) < 0$, $|S_{3,A}^{(0)}(b_{real,1} + e_A) + S_{3,B}^{(0)}(b_{real,2} + e_B)|$ manifests in the following scenarios.

1. **Smaller than the solution separation statistic for single-satellite fault mode:** The solution separation statistic for single-satellite fault mode where the A th satellite malfunctions is $|S_{3,A}^{(0)}(b_{real,1} + e_A)|$. For the B th satellite malfunctioning, it is $|S_{3,B}^{(0)}(b_{real,2} + e_B)|$. Both of these are likely larger than $|S_{3,A}^{(0)}(b_{real,1} + e_A) + S_{3,B}^{(0)}(b_{real,2} + e_B)|$. However, excluding the A th satellite alone or the B th satellite alone based on the solution separation statistics apparently does not produce a set of fault-free satellites.
2. **Smaller than the solution separation statistic for multi-satellite fault mode:** This scenario refers to a multiple satellite fault involving one of the A th or B th satellites combined with other satellites. For example, the solution separation statistic for dual-satellite fault mode involving the A th satellite and another C th satellite is $|S_{3,A}^{(0)}(b_{real,1} + e_A) + S_{3,C}^{(0)}e_C|$, where $S_{3,C}^{(0)}$ represents the C th element of $S_3^{(0)}$. This can be larger than $|S_{3,A}^{(0)}(b_{real,1} + e_A) + S_{3,B}^{(0)}(b_{real,2} + e_B)|$. However, excluding the A th and C th satellites based on the solution separation statistics is inappropriate since a fault remains, that is, the B th satellite.
3. **Below the detection threshold:** $|S_{3,A}^{(0)}(b_{real,1} + e_A) + S_{3,B}^{(0)}(b_{real,2} + e_B)|$ may be close to zero, thus falling below the detection threshold. It is important to emphasize that this does not mean ARAIM cannot detect faults. Other fault modes monitored, such as single-satellite fault mode where the A th satellite malfunctions, would produce large enough solution separation statistics to exceed the detection thresholds.
4. **Exactly zero:** In the extreme situation that $S_{3,A}^{(0)}(b_{real,1} + e_A)$ equals $-S_{3,B}^{(0)}(b_{real,2} + e_B)$, $|S_{3,A}^{(0)}(b_{real,1} + e_A) + S_{3,B}^{(0)}(b_{real,2} + e_B)|$ is zero. Both scenarios 3 and 4 would pose tough challenges for excluding the real fault, as their solution separation statistics would be quite small.

In summary, under multiple faults, the diversity of the faults and their interactions can cause the size of the solution separation statistic to become irregular. Therefore, we can draw another critical conclusion.

Conclusion 2: When multiple satellites experience faults, the size of the solution separation statistic exhibits randomness and is not a suitable criterion for determining the search order for exclusion candidates. This conclusion is validated in Section 5 through experiments.

Conclusion 2 is the primary motivation for this study. The existing methods may not quickly find exclusion candidates under multi-satellite faults because the real faults or exclusion candidates may be at the end of the search order. This can misleadingly guide the search for exclusion candidates, increasing the difficulty of fault exclusion. Additionally, the current methods limit the exclusion candidates in the monitoring list, making ARAIM unable to exclude real faults outside the monitored list. Although ARAIM involves integrity and continuity for unmonitored fault modes, excluding faults that are outside of the list is still an attractive option which can enhance ARAIM's FDE capability.

4. Proposed Method: Yielding Possible Faults Directly through Sparse Estimation

The fundamental limitation of the existing methods lies in their inability to identify which satellites are failures directly. Instead, they infer potential faults through the solution separation statistic, which becomes random under multiple faults. In view of this, we propose a different approach: directly estimating faulty satellites, i.e., identifying the non-zero values in \mathbf{b} .

Transcribing Equation (2), the positioning model with fault \mathbf{b} can be expressed as

$$\mathbf{y} = \mathbf{G}\mathbf{x} + \mathbf{e} + \mathbf{b} \quad (25)$$

Note a characteristic of \mathbf{b} : in civil aviation applications, \mathbf{b} is sparse [14,20], meaning that most of the elements are zero because the signal propagation and reception environments in aviation are good [21,22], the civil aviation receivers are robust, and the nominal model on the user side is usually accurately established. Additionally, ARAIM hardly ever monitors more than two satellite faults with independent failure causes. Based on the sparsity of \mathbf{b} ,

sparse estimation methods can directly figure out \mathbf{b} independently of ARAIM. Then, we can describe in step-by-step detail how to arrive at an estimate of \mathbf{b} .

A classical estimation method known as linear regression can be mathematically expressed as

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{Y} - \mathbf{H}\boldsymbol{\theta}\|_2^2 \quad (26)$$

where \mathbf{Y} is the observation vector; \mathbf{H} is the known state transition matrix; and $\boldsymbol{\theta}$ is the state vector to be estimated.

Given the sparsity of $\boldsymbol{\theta}$, Equation (26) is regularized by an additional norm ℓ_0 :

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{Y} - \mathbf{H}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_0 \quad (27)$$

$$\|\boldsymbol{\theta}\|_0 = \#\{i | \theta_i \neq 0\} \quad (28)$$

where λ is a constant known as the regularization parameter controlling the strength of regularization, and $\#$ is the counting symbol.

However, Problem (27) is non-convex, and finding its solution requires a computationally challenging combinatorial search. Therefore, approximating it using the ℓ_1 norm instead of the ℓ_0 norm to form a standard LASSO problem [23] is expected.

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{Y} - \mathbf{H}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 \quad (29)$$

$$\|\boldsymbol{\theta}\|_1 = \sum_i |\theta_i| \quad (30)$$

The standard LASSO of Equation (29) may occasionally yield an imprecise solution. In response, an improved estimation method called the reweighted- ℓ_1 LASSO algorithm has been proposed [24]. It utilizes a diagonal weight matrix \mathbf{W}_L to enhance the sparsity and accuracy of the estimated value $\hat{\boldsymbol{\theta}}$:

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{Y} - \mathbf{H}\boldsymbol{\theta}\|_2^2 + \lambda \|\mathbf{W}_L \boldsymbol{\theta}\|_1 \quad (31)$$

A desirable value for \mathbf{W}_L can strongly refine the estimation of $\boldsymbol{\theta}$. Ideally, the weight in \mathbf{W}_L should be reciprocal of the absolute value of $\boldsymbol{\theta}$'s true magnitude, but this is unattainable in practice, as the true value of $\boldsymbol{\theta}$ is unknown. The nature of matrix \mathbf{W}_L is that higher weights encourage the estimated values to be zero, while lower weights encourage non-zero values. The creator of the reweighted- ℓ_1 LASSO algorithm provided a general but complicated approach to setting \mathbf{W}_L . Fortunately, according to the SPP and ARAIM processes, a simple and effective \mathbf{W}_L is apparent: the covariance matrix \mathbf{C} of the nominal pseudorange errors. We have

$$\mathbf{W}_L = \mathbf{C} \quad (32)$$

The larger the UERE of a pseudorange, the more the deviation in that pseudorange should favor the nominal error e rather than the fault bias \mathbf{b} . In other words, the higher the UERE for a pseudorange, the closer the estimated value for \mathbf{b} should be to zero. Therefore, a covariance matrix \mathbf{C} consisting of the square of the UERE is a suitable weight matrix.

For the specific algorithm design, the crux is to transform the estimation of \mathbf{b} into the standard LASSO, according to Equation (29). According to Equations (2), (31), and (32), we can initially formulate the problem of sparsely estimating \mathbf{b} as follows.

$$\arg \min_{\mathbf{b}} \frac{1}{2} \|\mathbf{y} - \mathbf{G}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{C}\mathbf{b}\|_1 \quad (33)$$

Equation (33) is our core model for estimating \mathbf{b} , which is derived via reweighted- ℓ_1 LASSO. In order to make Equation (33) exclusively comprise the unknown variable \mathbf{b} , \mathbf{x} is converted into

$$\mathbf{x} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T (\mathbf{y} - \mathbf{b}) \quad (34)$$

Equation (34) can be obtained from Equation (2) using least squares, which reflects the consistency of the fault-free pseudorange. It can also use weighted least squares. However, to simplify the variable transformation and because the weights already constrain \mathbf{b} , additional weighting for Equation (34) does not significantly improve the results. Therefore, using simple least squares is appropriate.

Substituting Equation (34) into Equation (33), we have

$$\arg \min_{\mathbf{b}} \frac{1}{2} \left\| \mathbf{y} - \mathbf{G}(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T (\mathbf{y} - \mathbf{b}) - \mathbf{b} \right\|_2^2 + \lambda \|\mathbf{C}\mathbf{b}\|_1 \quad (35)$$

After performing variable replacement, Equation (35) is transformed into

$$\arg \min_{\boldsymbol{\theta}} \frac{1}{2} \left\| \mathbf{Y} - \mathbf{H}\boldsymbol{\theta} \right\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 \quad (36)$$

where

$$\mathbf{Y} = (\mathbf{I}_{N_{sat}} - \tilde{\mathbf{H}})\mathbf{y} \quad (37)$$

$$\mathbf{H} = (\mathbf{I}_{N_{sat}} - \tilde{\mathbf{H}})\mathbf{C}^{-1} \quad (38)$$

$$\boldsymbol{\theta} = \mathbf{C}\mathbf{b} \quad (39)$$

$$\tilde{\mathbf{H}} = \mathbf{G}(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \quad (40)$$

and $\mathbf{I}_{N_{sat}}$ is the $N_{sat} \times N_{sat}$ identity matrix.

Equation (36) is precisely the standard LASSO problem. At this point, estimating \mathbf{b} becomes straightforward. We solve Equation (36) to obtain solution $\hat{\boldsymbol{\theta}}$ and yield the estimated bias vector $\hat{\mathbf{b}}$:

$$\hat{\mathbf{b}} = \mathbf{C}^{-1} \hat{\boldsymbol{\theta}} \quad (41)$$

Equation (36) has efficient solution methods, such as coordinate descent (CD) [25] and the least angle regression method [26]. We choose the classical CD method for solving LASSO, which iteratively progresses along the coordinate axes' directions until convergence is reached. Appendix A describes how to solve Equation (36) and illustrates that its computational load is typically low.

In addition, there is no need for extensive tuning of the regularization parameter λ . This is because the primary focus is identifying which elements of the estimated vector $\hat{\mathbf{b}}$ are non-zero rather than precisely quantifying the magnitude of these non-zero values. We have selected the commonly used value of $\lambda = 1$ for simplicity, and research using sparse estimation for GNSSs suggests that $\lambda = 1$ is a robust choice that can effectively ensure the estimation accuracy [14].

As a summary, Table 1 presents the algorithm for directly estimating faults using reweighted- ℓ_1 LASSO. This algorithm is quite simple: it involves "variable substitution to form a standard LASSO" followed by "solving the LASSO". The proposed algorithm serves as a solid complement to the existing normalized solution separation statistic method, and both methods can be applied sequentially. Considering the advantages of sparse estimation principles and the experimental results discussed in Section 5, we recommend initially employing our method for rapid, one-time identification of the exclusion candidates. If this approach fails to identify the candidates, the normalized solution separation statistic method can then be applied. This recommendation is further explained in Section 6.

Table 1. Fault identification for the ARAIM FE process based on reweighted- ℓ_1 LASSO. The FE process only needs to confirm whether the satellites identified in step 4 are fault-free through the MHSS test.

Step	Operation	Description
1	Input parameters	Input y , G , and C .
2	Formulate the standard LASSO problem	Apply Equation (37)–(40) to form the standard LASSO problem, Equation (36).
3	Solve LASSO	Use the CD method to obtain an estimate of the fault bias vector, \hat{b} .
4	Output satellites	Output the satellites corresponding to the non-zero elements in \hat{b} .

5. Experiments and Results

This section evaluates the performance of the proposed algorithm using both real and simulated data. To maintain consistency with most of the domestic and international tests and simulations, this experiment adopts a GPS–Galileo dual-constellation configuration, referring to the Stanford University MAAST program and related literature [19,27,28] to ensure the conservativeness of the integrity support message (ISM) parameters.

The receiver is the mosaic X5 from Septentrio, statically positioned on the roof of the Weiqing Building at Tsinghua University. The true coordinates of the receiver’s location in terms of Latitude, Longitude, and Altitude (LLA) are [40.0014762483°, 116.3302357350°, 84.5217 m]. The data collection window is the entire solar day of 12 January 2024, with a time step of one minute. The specific test configuration is shown in Table 2.

Table 2. Main ISM and ARAIM system parameters. They are referenced from existing studies, and the chosen values are both common and conservative.

Parameters	Description	Values
$p_{sat,i}$	Prior fault probability of satellite i	1×10^{-5} for GPS; 1×10^{-4} for other constellations
p_{const}	Prior fault probability of constellation	1×10^{-5} for all constellations
$\sigma_{URA,i}$	The standard deviation of clock and ephemeris errors of satellite i used for integrity from ISMs	2.4 m for GPS satellites; 4 m for other constellation satellites
$\sigma_{URE,i}$	The standard deviation of clock and ephemeris errors of satellite i used for accuracy from ISMs	$2/3 \sigma_{URA,i}$
$b_{nom,i}$	Maximum nominal bias for satellite i from ISMs	0.75 m for all satellites
$PHMI$	Integrity budget	9.8×10^{-8} for $q = 3$; 2×10^{-9} for $q = 1, 2$
PFA	Continuity budget allocated to disruptions due to false alerts	3.9×10^{-6} for $q = 3$; 9×10^{-7} for $q = 1, 2$
P_{THRES}	The threshold for the risk of unmonitored faults	6×10^{-8}
K	Number of Monte Carlo simulations	200

5.1. Estimation Performance of Our Method

5.1.1. Validation of Our Method’s Effectiveness through Real-Time Data

We first evaluate the reweighted- ℓ_1 LASSO’s capability in identifying faults. Since satellite failures do not frequently occur in real life, this experiment artificially introduces

fault bias \mathbf{b} . In \mathbf{b} , non-zero elements signify faults, which are set to be more than ten times the corresponding UERE. We simultaneously added 40 m and -40 m biases to two pseudoranges, representing dual-satellite real faults. The faulty satellites correspond to the GPS and Galileo satellites with the lowest pseudo-random noise (PRN) numbers, denoted as GPS #1 and Galileo #1, respectively. The results suggest that our method is highly effective in estimating the fault bias vector \mathbf{b} .

Figure 1 shows the real-time estimation of the dual-satellite real faults. The horizontal axis represents 1440 epochs within a 24 h window with a 1-min step size. The blue dotted line represents the fault bias estimated by the proposed algorithm, i.e., non-zero elements in the estimated vector $\hat{\mathbf{b}}$, denoted as \hat{b}_{real} . The red dotted line represents the artificially set fault bias, i.e., non-zero elements b_{real} in \mathbf{b} . The yellow dotted line, the UERE, shows the nominal error of the pseudorange. From Figure 1, it is clear that our method can effectively identify real-time faults in the two channels. The estimated results are quite accurate, with \hat{b}_{real} consistently near b_{real} . Moreover, there were no instances of omission among the 1440 values, meaning no cases were estimated as zero.

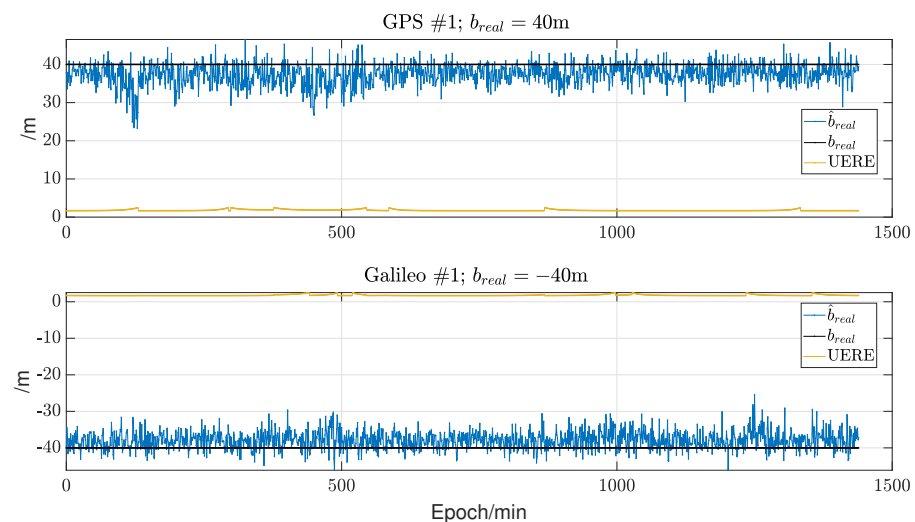


Figure 1. Real-time estimation of dual-satellite real faults using reweighted- ℓ_1 LASSO.

We emphasize that setting the fault magnitudes to 40 m and -40 m is intended to penalize solution-separation-statistic-based searches. As discussed in Section 3, the existing methods struggle with dual-satellite faults, especially when the fault magnitudes have opposite signs. The value of 40 m is a typical choice without a loss of generality, representing the UERE ten times, which ensures that we introduce a detectable fault. Other sufficiently large fault magnitudes (usually greater than 20 m in this experiment) would have yielded similar results as long as the two faults had opposite signs. These changes do not affect the conclusions.

For comparison, Figure 2 illustrates the challenges of using the existing normalized solution separation statistic method for excluding dual-satellite faults with $+40$ m and -40 m biases. Unlike our approach, which directly identifies faulty satellites, this method infers potential faults indirectly based on the magnitude of the normalized solution separation statistics.

In Figure 2, the red dotted line represents the number of fault modes monitored by ARAIM in the current epoch, while the blue dotted line shows the ranking of the real fault modes sorted by the normalized solution separation statistics (i.e., the values in Equation (9)) in descending order. For instance, in the first epoch, the red indicates 54 fault modes being monitored by ARAIM. These fault modes are ranked from largest to smallest according to their solution separation statistics, with the real fault ranked as 9, as shown by the blue. This implies that the traditional method would need about nine MHSS tests to identify the real fault. Therefore, the closer the real fault is to the top of the ranking (i.e.,

the lower the blue dotted line is), the quicker the traditional method can exclude it. If the real fault is ranked as one, only one MHSS test is required. Conversely, if it is ranked at the bottom, this complicates the FE sorting process.

Typically, the real fault ranks are in the middle to the later parts of the list, averaging at around rank 15 (with an average value of 15.46). This suggests that the FE process may need approximately fifteen MHSS tests, equivalent to the complexity of fifteen FD calculations, to find the exclusion candidate. Sometimes, the blue dotted line is extremely high, even close to the red dotted line, which can mislead the FE process in identifying the correct exclusion candidates.

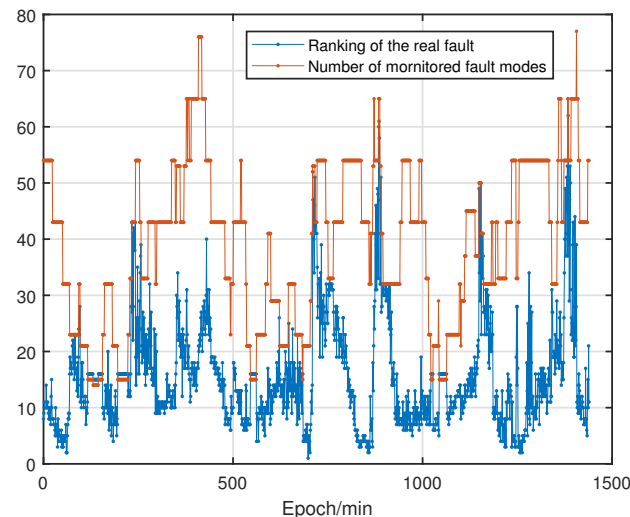


Figure 2. Real fault's ranking sorted by normalized solution separation statistics in descending order. The situation where the blue dotted line is above the red indicates that the real fault is outside the monitored list, and at this point, we set the rank to the possible lowest value, $N_{fault} + 1$.

The results in Figure 2 are consistent with our analysis in Section 3, showing that under multiple faults, normalized solution separation statistics are not a reliable basis for guiding the search for exclusion candidates, especially when two faults with the same magnitude but opposite signs are imposed. This can lead to the failure of FE due to the excessive time required to find an exclusion candidate.

Furthermore, there is another situation where the existing method leads to FE failure: the real fault is outside the monitored list, and an exclusion candidate cannot be found. Although such FE failures are permissible as ARAIM redundantly allocates continuity and integrity budgets for them, being able to exclude out-of-list faults is still desirable as it enhances ARAIM's FDE capability. We include a scenario with a triple-satellite real fault, where biases of +40 m, −40 m, and +40 m are imposed on GPS #1, Galileo #1, and GPS #2 (i.e., the satellite with the second lower PRN number), respectively, which is an unmonitored triple-satellite fault under the experimental conditions. The existing method cannot identify it as an exclusion candidate because the statistic for the unmonitored fault is not calculated. On the contrary, Figure 3 demonstrates the satisfactory results estimated in real time for the three failed satellites, and this is a supplemental FDE capability outside of the ARAIM list. It is learned from Figures 1 and 3 that our method can effectively point out possible exclusion candidates in a single run, fundamentally different from using solution separation statistics to indirectly reflect possible faulty pseudoranges.

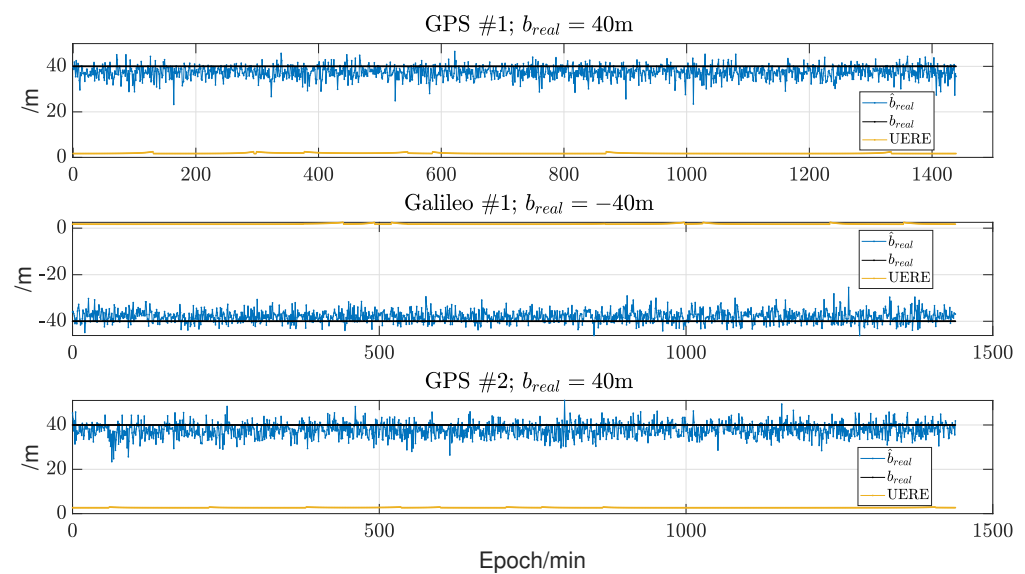


Figure 3. Real-time estimation of triple-satellite real fault using reweighted- ℓ_1 LASSO. Faults involving three satellites are beyond the existing methods' capability due to not all of them being monitored.

5.1.2. Validation of Our Method's Accuracy through Simulated Data

After confirming the effectiveness of the proposed algorithm in real-time satellite fault estimation, we further evaluate the accuracy rate of our method under different constellations, fault numbers, and fault magnitudes. The accuracy rate of estimation is defined as the proportion of estimated faulty satellites estimated among all artificially set faults. When the fault accuracy rate reaches 100%, this means that reweighted- ℓ_1 LASSO can accurately identify all of the faults. Figures 4–6 show the accuracy rates under different numbers and sizes of faults within a 24 h observation window at Tsinghua University.

To present the accuracy rate more comprehensively, this work employs 200 Monte Carlo simulation data points instead of actual data collected at one time. In the simulation, the ephemeris data are real, but the pseudoranges are derived from the actual distances between the satellite and receiver plus simulated errors conforming to the nominal model. Faults are assigned to GPS, Galileo, BeiDou, and GLONASS in that order, with each constellation having as equal a number of faults as possible. Faults within each constellation are prioritized for satellites with lower PRN numbers, and each fault has the same bias magnitude b_{real} . b_{real} is set in the 10 m to 100 m range, with 10 m approximately three times the UERE and a bias smaller than this being unlikely to be considered a fault, while 100 m represents a significant value sufficient to represent severe fault conditions. This range covers a wide range from minor errors to severe faults. Although ARAIM hardly ever monitors more than two satellite faults with independent failure causes, we extended the simulation to include up to ten satellites. This was chosen for two primary purposes: first, to explore the sparsity conditions of sparse estimation and determine when our method becomes less effective as the number of simultaneous satellite faults increases, and second, to demonstrate that our method can identify faults outside the monitoring list.

Figure 4 shows the simulation results under the dual-constellation configuration. For up to three faults above 10 m, the accuracy rate of estimation reaches 100%. For four faults with biases exceeding 30 m, the accuracy rate is about 99%. For five faults with 50 m biases, the accuracy rate can still reach almost 90%. When the number of faults increases to six or more, the performance of reweighted- ℓ_1 LASSO is affected due to the reduced sparsity of the fault vector \mathbf{b} . It is quite satisfactory to accurately estimate a four-satellite failure given that ARAIM does not normally monitor the failure of more than two satellites from independent causes.

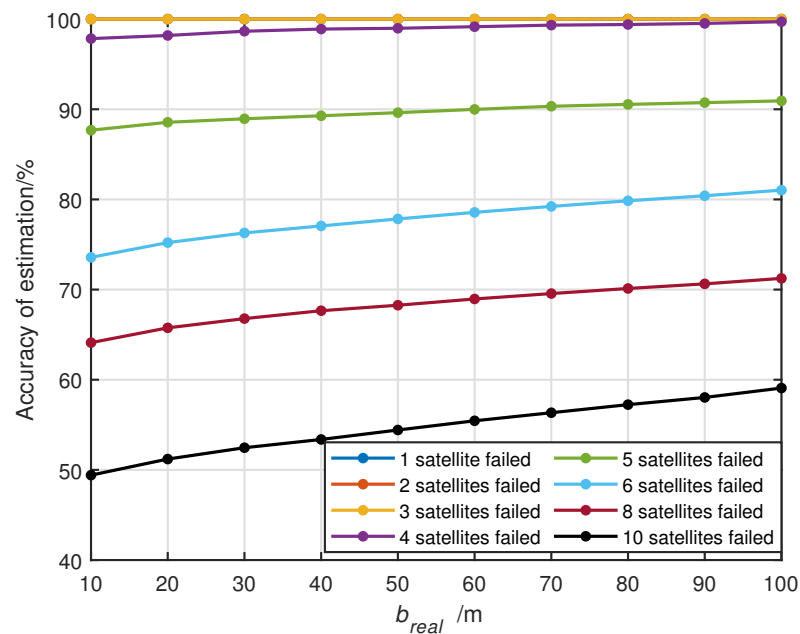


Figure 4. Accuracy rate of the proposed algorithm under two constellations. For this dual-constellation configuration, the minimum, mode, and average of visible satellites are 12, 16, and 15.71, respectively. When the number of failed satellites is within three, all the estimation accuracy reached 100%, causing the overlap of their corresponding lines.

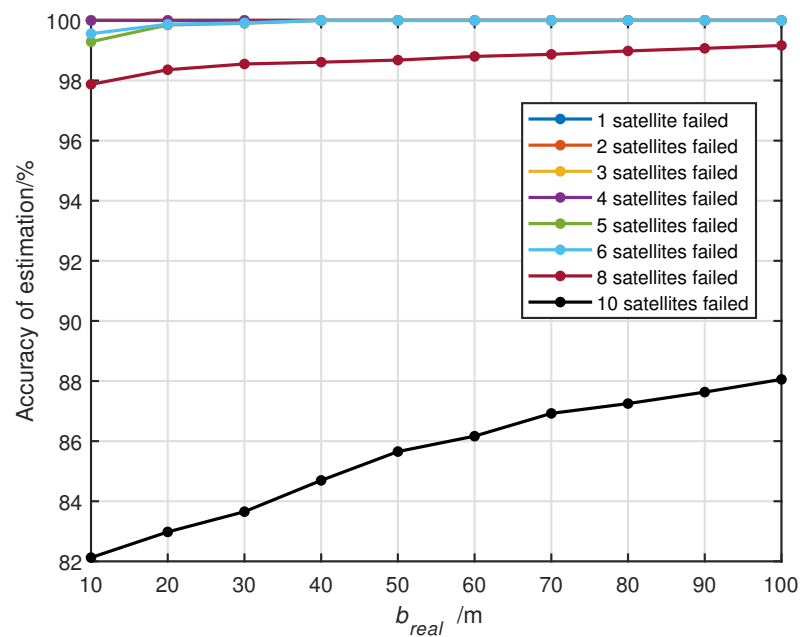


Figure 5. Accuracy rate of the proposed algorithm under three configurations. For this triple-constellation configuration, the minimum, mode, and average number of visible satellites are 23, 28, and 27.84, respectively. When the number of failed satellites is within six and the biases exceed 30 m, all the estimation accuracy reached 100%, causing the overlap of their corresponding lines.

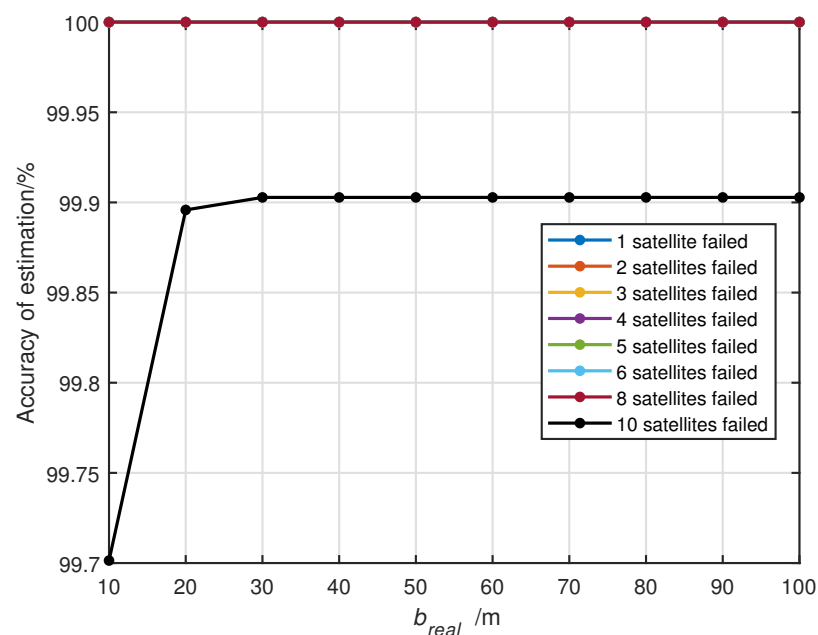


Figure 6. Accuracy rate of the proposed algorithm under four constellations. For this quad-constellation configuration, the minimum, mode, and average number of visible satellites are 28, 36, and 34.92, respectively. When the number of failed satellites is within eight, all the estimation accuracy reached 100%, causing the overlap of their corresponding lines.

The proposed method relies on the number of visible satellites to meet the sparsity requirements. With an average of 27.84 visible satellites for 2 constellations, Figure 4 represents the experimental results under the average visibility conditions. To further illustrate the impact of satellite count on the method's performance, Table 3 presents the estimation accuracy under the minimum and mode visible satellite counts for the dual constellation. The minimum satellite count reflects the worst-case scenario, while the mode satellite count reflects the general scenario. Fault magnitudes of 10 m, 40 m, and 100 m are selected to represent potential non-fault values, typical fault values, and significant fault values, respectively.

Table 3. Estimation accuracy of our method under minimum and mode visible satellite counts for the dual-constellation configuration.

Number of Satellite	Fault Bias	Number of Faults							
		1	2	3	4	5	6	8	10
12 satellites	10 m	100%	100%	100%	95.23%	78.57%	65.17%	48.71%	42.34%
	40 m	100%	100%	100%	96.36%	82.04%	69.23%	51.48%	46.00%
	100 m	100%	100%	100%	97.70%	86.98%	74.02%	56.67%	52.22%
16 satellites	10 m	100%	100%	100%	97.94%	87.78%	73.68%	64.21%	49.52%
	40 m	100%	100%	100%	99.00%	89.38%	77.16%	67.76%	53.38%
	100 m	100%	100%	100%	100%	91.03%	81.13%	73.34%	59.08%

As seen in Table 3, even under the worst-case scenario with 12 visible satellites, our method can still accurately estimate the faults within 4 satellites. In this case, although the estimation performance may be slightly inferior to that for 16 visible satellites or the average performance shown in Figure 4, it remains an effective fault identification method.

Figure 5 shows the accuracy rate under the GPS, Galileo, and BeiDou triple-constellation configuration. In this simulation configuration, the proposed algorithm can recognize up to six faults (with biases above 20 m) with a 100% accuracy rate and up to eight faults

(with biases above 10 m) with an accuracy rate of over 98%. When ten or more faults occur, the accuracy rate begins to decline. In practical applications, this performance is excellent for identifying faults.

Like Table 3, Table 4 presents the estimation accuracy for the minimum and mode visible satellite counts under a triple-constellation scenario. Even in the worst-case scenario with 23 visible satellites, our method achieves an accuracy of over 96% when estimating up to 8 faults.

Table 4. Estimation accuracy of our method under minimum and mode visible satellite counts for the triple-constellation configuration.

Number of Satellite	Fault Bias	Number of Faults							
		1	2	3	4	5	6	8	10
23 satellites	10 m	100%	100%	100%	99.73%	99.05%	98.64%	96.96%	81.85%
	40 m	100%	100%	100%	100%	100%	100%	98.03%	83.55%
	100 m	100%	100%	100%	100%	100%	100%	98.73%	84.12%
28 satellites	10 m	100%	100%	100%	100%	99.56%	99.38%	97.87%	82.13%
	40 m	100%	100%	100%	100%	100%	100%	98.61%	84.69%
	100 m	100%	100%	100%	100%	100%	100%	99.17%	88.06%

Figure 6 shows the accuracy rate under a quad-constellation configuration, where our method continues to exhibit outstanding performance even in extreme situations with ten faults. This indicates that with an increasing number of constellations and visible satellites, the estimation performance is significantly enhanced. Reweighted- ℓ_1 LASSO is particularly suitable for scenarios with multiple constellations and many visible satellites, as the sparsity assumption for pseudorange faults is met better in such environments. Finally, Figures 4–6 also illustrate that the detection capability of LASSO increases with the growth of b_{fault} : the more apparent the fault, the easier it is to recognize.

Table 5 provides the estimation accuracy for the minimum and mode visible satellite counts under a quadruple-constellation scenario. The data in Table 5 show that the accuracy is consistently above 97% under all conditions, demonstrating that our method is highly effective in the case of four constellations.

Table 5. Estimation accuracy of our method under minimum and mode visible satellite counts for the quad-constellation configuration.

Number of Satellite	Fault Bias	Number of Faults							
		1	2	3	4	5	6	8	10
28 satellites	10 m	100%	100%	100%	100%	100%	100%	100%	97.24%
	40 m	100%	100%	100%	100%	100%	100%	100%	98.37%
	100 m	100%	100%	100%	100%	100%	100%	100%	98.86%
36 satellites	10 m	100%	100%	100%	100%	100%	100%	100%	99.72%
	40 m	100%	100%	100%	100%	100%	100%	100%	99.93%
	100 m	100%	100%	100%	100%	100%	100%	100%	99.93%

5.2. Computational Complexity of Our Method

Finally, we examine the complexity of our method. The additional computational complexity induced has two components. One is the five matrix multiplications described in Equations (37)–(41) for building the standard LASSO problem. The other is the CD method for solving the LASSO, which requires iterative computation. In this study, the time complexity of one iteration of CD is approximately $O(N_{sat}^3)$ according to the LASSO-solving process shown in Appendix A. Specifically, the local minimum of the N_{sat} coordinates needs to be calculated in one iteration. Each coordinate requires one matrix multiplication of $N_{sat} \times N_{sat}$ and $N_{sat} \times 1$, two vector multiplications of $1 \times N_{sat}$ and $N_{sat} \times 1$, one vector

multiplication of $1 \times N_{sat}$, and a scalar, along with several additions. Taking the highest power, the computational complexity of one iteration is approximately N_{sat}^3 scalar multiplications, equivalent to the complexity of multiplying two $N_{sat} \times N_{sat}$ matrices. Therefore, incorporating the first component of the complexity, our approach is roughly equivalent to $(5 + M)$ matrix multiplications, where M is the number of iterations in the CD method. Given that the convergence speed of the CD method is fast [25], M is not high, and N_{sat} is the number of satellites, which is generally not more than fifty. Hence, the computational complexity introduced by the LASSO is quite limited.

In addition to the inherent simplicity of the CD method and the low dimensionality of the matrices involved, the low computational complexity of this method is also attributed to the fact that single- and dual-satellite faults occupy a substantial portion of the monitoring list. These faults are pretty sparse, with b being quite close to the coordinate origin, 0 , allowing CD to converge in just a few iterations.

Figure 7 compares the FE runtime using our method with the fastest theoretical FE process time. The “fastest theoretical runtime” refers to a scenario where we know the real fault a priori and can troubleshoot it with only one MHSS exclusion candidate test, representing the shortest possible time. The runtime with our method is only slightly higher than this fastest theoretical runtime, indicating that it consumes modest computational resources.

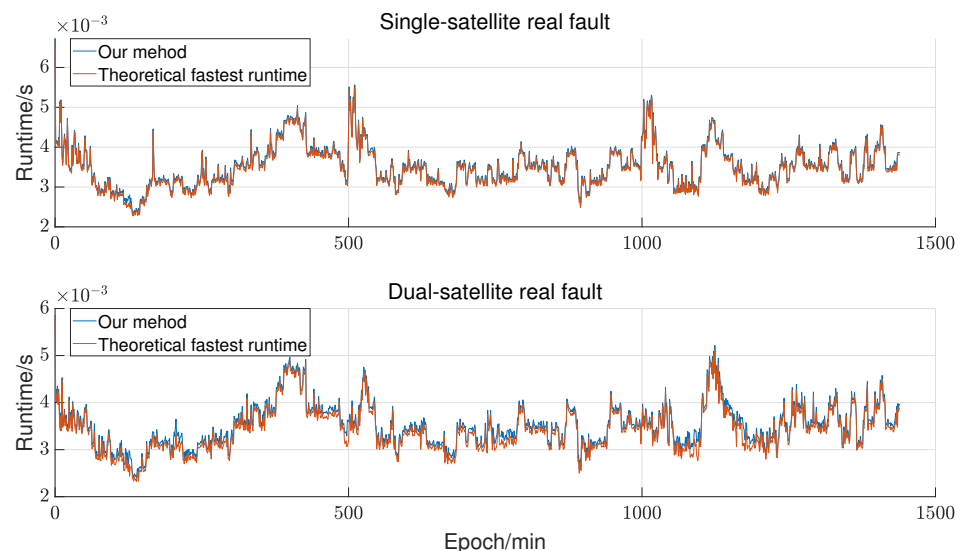


Figure 7. FE runtime of our method compared with the fastest theoretical FE runtime. Data processing is implemented using Matlab 2022a on a personal computer from Lenovo Group Limited, Beijing, China, with a 2.2 GHz i7-8750H CPU and 8 GB RAM.

We then compare the proposed method with the normalized solution separation method in terms of the FE runtime for single-satellite and dual-satellite real faults, as shown in Figure 8. For single-satellite real faults, the runtime of both methods is similar because both can quickly identify the exclusion candidates. The traditional methods compare the solution separation statistic and typically perform one MHSS test to determine the fault. Our proposed LASSO optimization can also achieve this with a single MHSS test. The main difference in the runtime arises from the fact that the traditional methods require calculating the solution separation statistic for each fault mode or indirectly obtaining it through chi-square statistics, while our method involves solving the LASSO. The time difference is quite minimal. Naturally, if solving the LASSO takes longer than calculating the solution separation statistic, solution separation-statistic-based searches would be slightly more efficient than LASSO optimization, as indicated by the arrow in the top panel of Figure 8.

Overall, both methods are highly feasible for single-satellite faults and are close to the fastest theoretical runtime.

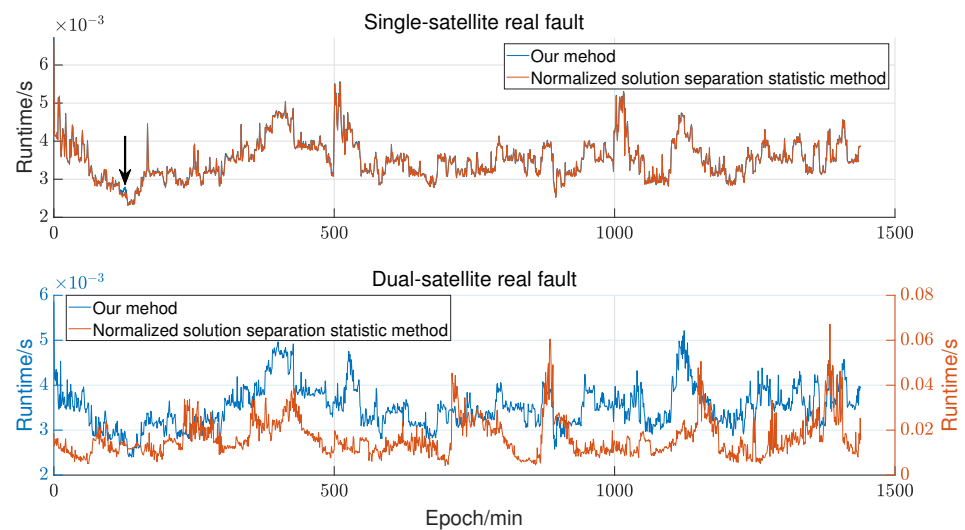


Figure 8. Comparison of FE runtime between our method and normalized solution separation statistic methods. Data processing is implemented using Matlab 2022a on a personal computer from Lenovo Group Limited, Beijing, China, with a 2.2 GHz i7-8750H CPU and 8 GB RAM. Note that the second panel in the figure has two y -axes.

However, for a dual-satellite real fault, the runtime of the normalized solution separation method is significantly longer than that of our method. This discrepancy arises because our method can directly estimate possible faults in just one run, whereas the real fault's solution separation statistics are usually small and have a lower priority in the search order. This results in the need to perform numerous MHSS tests to determine the exclusion candidate. Under the conditions of this experiment, the normalized solution separation method is misguided in the search order, leading to a prolonged runtime. If the maximum allowable FE runtime is 20 ms, the existing methods will fail for most dual-satellite FE processes, which is precisely their serious limitation.

6. Conclusions

Finding exclusion candidates is crucial for the swift and successful execution of the ARAIM process. The existing methods infer the likelihood of specific fault modes being actual faults through normalized solution separation statistics, guiding the search for exclusion candidates. However, our qualitative analysis and experimental results indicate that solution separation statistics are not reliable indicators in scenarios involving multiple faults.

This work presents an alternative supplementary method for identifying possible exclusion candidates. Considering the sparsity of GNSS faults and the consistency of normal observations, we propose using sparse estimation to find faulty pseudoranges directly, thus rapidly and efficiently determining the exclusion candidates in a single run without the need for sorting and sequential searching.

Our experiments demonstrate that the proposed method performs exceptionally well under sparse conditions, being fast and accurate. However, it must be acknowledged that while the sparsity requirements are relatively lenient (e.g., in a dual constellation, it can identify faults among five satellites, and in a quad constellation, it can make estimations for up to ten satellites), this method may struggle with wide faults affecting entire constellations, as such faults are rarely sparse. Therefore, we recommend combining our proposed method with the existing approaches: first, execute our method due to its rapidity. If the exclusion candidates cannot be verified, only the resources for one MHSS test have been expended. Subsequently, the existing approach can be employed, especially focusing on constellation-wide faults. Indeed, the method proposed in this paper is a strong complement to the existing normalized solution separation statistic methods.

Author Contributions: Conceptualization, methodology, software, data analysis, investigation, and original draft preparation, review and editing, H.Q.; methodology and manuscript review, X.C.; supervision, project administration, and funding acquisition, M.L. All authors have read and agreed to the published version of the manuscript

Funding: Funding was provided by the National Key R&D Program of China under Grant No. 2021YFA0716603 and the National Natural Science Foundation of China under Grant No. U2233217.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank all the editors and reviewers for their patience, professionalism, and constructive comments.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADD	Algorithm Description Document
ARAIM	Advanced Receiver Autonomous Integrity Monitoring
CD	Coordinate Descent
FD	Fault Detection
FDE	Fault Detection and Exclusion
FE	Fault Exclusion
GNSS	Global Navigation Satellite System
ISM	Integrity Support Message
LLA	Latitude, Longitude, and Altitude
MAAST	MATLAB Algorithm Availability Simulation Tool
MHSS	Multi-Hypothesis Solution Separation
PL	Protection Level
PRN	Pseudo-Random Noise
RAIM	Receiver Autonomous Integrity Monitoring
SPP	Standard Point Positioning
UERE	User Equivalent Range Error
WG-C	Working Group-C
WLSQ	Weighted Least Squares

Appendix A. Solving for $\hat{\theta}$

Among the many methods for solving the LASSO problem, the CD method and the least angle regression method are two widely adopted techniques. Given the considerations of algorithm efficiency and convergence speed in practical applications, this paper chooses the CD method as the solution method.

The CD method finds the minimum of the objective function by optimizing each target function's coordinate (or parameter) one at a time. Its mathematical foundation relies on a key convex optimization principle: for a differentiable convex function $J(\theta)$, which in this paper is defined as

$$J(\theta) = \frac{1}{2} \|Y - H\theta\|_2^2 + \lambda \|\theta\|_1 \quad (A1)$$

if at a certain point $\bar{\theta}$, the function $J(\theta)$ reaches a local minimum on each coordinate axis $\bar{\theta}_n (n = 1, 2, \dots, N_{sat})$, then $J(\bar{\theta})$ reaches a global minimum at $\bar{\theta}$. Therefore, the optimization goal of the CD method transforms into iteratively and gradually minimizing the loss function on the N_{sat} coordinate axes (or N_{sat} elements) of θ . The specific solving process is as follows.

1. Initial value setting: First, an initial value for the vector θ needs to be chosen. This initial value is usually set based on some prior knowledge.

In this problem, the zero vector $\mathbf{0}$ is directly used as the initial value. For SPP in civil aviation, its service performance is stable, and the actual fault probability is very low.

This means that in most cases, vector \mathbf{b} is sparse and close to a vector of $\mathbf{0}$. Therefore, choosing a vector of zero as the initial value will significantly accelerate the convergence of the solution.

2. p th Iteration: In the p th iteration, the updated value for each coordinate is calculated sequentially from θ_1 to $\theta_{N_{sat}}$. When finding the optimal value θ_i^* for the i th coordinate, the values of all other coordinates are fixed, and the zero point of the partial derivative of $J(\theta)$ with respect to θ_i is calculated:

$$\left. \frac{\partial J(\theta)}{\partial \theta_i} \right|_{\theta_i=\theta_i^*} = 0 \quad (\text{A2})$$

θ_i^* is the local minimum near the current coordinate. The analytical solution of Equation (A2) is given directly as

$$\theta_i^* = \frac{1}{\mathbf{H}_i^T \mathbf{H}_i} \mathbf{S}(\mathbf{H}_i^T (\mathbf{y} - \mathbf{H}_{-i} \theta_{-i}), \lambda) \quad (\text{A3})$$

where \mathbf{H}_i represents the i th column vector of matrix \mathbf{H} ; \mathbf{H}_{-i} and θ_{-i} represent matrix \mathbf{H} with the i th column vector removed and vector θ with the i th element removed, respectively; and \mathbf{S} is the soft thresholding operator, defined as

$$\mathbf{S}(x, \lambda) = \text{sign}(x) \cdot \max(|x| - \lambda, 0) \quad (\text{A4})$$

The sign function $\text{sign}(x)$ preserves the sign (positive or negative) of the original value x , and $\text{sign}(x) \cdot \max(|x| - \lambda, 0)$ ensures that value x is retained only if it exceeds the threshold λ . If the absolute value of a value is less than λ , the soft thresholding operation reduces it to zero. This is the principle by which the LASSO achieves sparse solutions. For elements with absolute values greater than λ , the soft thresholding operation shrinks them to zero by a fixed amount but does not entirely eliminate them, thus retaining the information on those critical features.

3. Convergence determination: After each iteration, the value of the objective function $J_{step,p}$ at step p needs to be compared with the value of the objective function $J_{step,p-1}$ at the previous step. If the change is minimal and reaches a preset threshold, i.e.,

$$|J_{step,p} - J_{step,p-1}| \leq \varepsilon \quad (\text{A5})$$

It can be considered that the algorithm has converged. Otherwise, the iteration continues. In this algorithm, ε is set to 1×10^{-4} .

Appendix B. A Numerical Example of the Proposed Algorithm

To illustrate the pseudorange fault estimation method based on the LASSO proposed in this paper more clearly, a numerical example is provided for reproducibility. We only consider the GPS constellation, ensuring the matrix dimensions are small enough to display in the appendix. On 12 January 2024, at 00:00, at the top of the Weiqing Building, Tsinghua University, a fault offset of +40 m was applied to GPS #1. The pseudoranges received were

$$\mathbf{p} = [21273445.9416, 22601323.0668, 22846185.6945, 22467724.6946, 24452274.7816, 20257154.8892, 25187197.5835, 23549979.0397]^T \quad (\text{A6})$$

Note that the pseudorange vector \mathbf{p} is not \mathbf{y} in our algorithm. GNSS standard single-point positioning employs a Newton–Raphson method, where weighted least squares solves each iteration. \mathbf{y} represents the pseudorange residuals rather than the pseudoranges. More specifically, \mathbf{y} is the pseudorange residual at the last iteration when convergence is achieved. The value of \mathbf{y} is

$$\mathbf{y} = [3.8722, -18.9872, -14.5725, -5.4242, -20.2281, -34.5013, -16.9118, -18.0469]^T \quad (\text{A7})$$

Correspondingly, we have

$$G = \begin{bmatrix} 0.1965 & -0.9763 & -0.0910 & 1 \\ -0.4707 & -0.3269 & -0.8195 & 1 \\ 0.9403 & -0.3238 & 0.1051 & 1 \\ 0.6883 & 0.0077 & -0.7254 & 1 \\ -0.8570 & -0.5007 & -0.1221 & 1 \\ 0.4374 & -0.7258 & -0.5309 & 1 \\ 0.2729 & -0.5134 & 0.8136 & 1 \\ 0.7432 & -0.4703 & 0.4758 & 1 \end{bmatrix} \quad (A8)$$

$$C = \begin{pmatrix} 2.8534 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.9206 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.9509 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.9085 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.7262 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.8391 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6.0783 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.1066 \end{pmatrix} \quad (A9)$$

Then, with y , G , and C as inputs to our algorithm, we can estimate the fault bias b .

Substituting y and G into Equations (37) and (40), we obtain

$$Y = [19.4092, -2.5133, 1.1797, 10.6318, -5.5764, -18.2427, -2.2067, -2.6817]^T \quad (A10)$$

From Equations (38) and (39), it is known that

$$H = \begin{bmatrix} 0.1512 & -0.0191 & 0.0089 & 0.0812 & -0.0333 & -0.1428 & -0.0081 & -0.0192 \\ -0.0196 & 0.1811 & 0.0150 & -0.0986 & -0.0981 & -0.0266 & 0.0150 & 0.0384 \\ 0.0092 & 0.0151 & 0.2280 & -0.0929 & 0.0327 & -0.0460 & -0.0301 & -0.0905 \\ 0.0828 & -0.0982 & -0.0916 & 0.1294 & 0.0066 & -0.0345 & 0.0149 & -0.0234 \\ -0.0434 & -0.1252 & 0.0413 & 0.0085 & 0.1085 & 0.0520 & -0.0383 & 0.0075 \\ -0.1421 & -0.0259 & -0.0443 & -0.0337 & 0.0396 & 0.1611 & 0.0267 & -0.0185 \\ -0.0172 & 0.0313 & -0.0620 & 0.0310 & -0.0624 & 0.0572 & 0.0758 & -0.1095 \\ -0.0209 & 0.0408 & -0.0953 & -0.0250 & 0.0063 & -0.0203 & -0.0559 & 0.2152 \end{bmatrix} \quad (A11)$$

The standard LASSO problem is formulated as

$$\arg \min_{\theta} \frac{1}{2} \|Y - H\theta\|_2^2 + \lambda \|\theta\|_1 \quad (A12)$$

where λ is set to 1. Solving for θ according to Appendix A and knowing that $\theta = Cb$, we obtain

$$\hat{b} = [38.3843, 0, 0, 0, 0, 0, 0, 0]^T \quad (A13)$$

The proposed algorithm estimates a bias of 38.3843 m for GPS #1, which is close to the +40 m fault setting.

In this example, the elements in p , y , Y , and \hat{b} are measured in meters, the elements in C are the variance in the pseudorange error in units of square meters, and G and H are dimensionless.

References

1. FAA. Phase II of the GNSS Evolutionary Architecture Study, FAA Report. February 2010. Available online: https://www.faa.gov/sites/faa.gov/files/about/office_org/headquarters_offices/ato/GEASPhaseII_Final.pdf (accessed on 12 August 2024).
2. Blanch, J.; Walter, T.; Berz, G.; Burns, J.; Clark, B.; Joerger, M.; Mabillean, M.; Martini, I.; Milner, C.; Pervan, B.; et al. Development of advanced RAIM minimum operational performance standards. In Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019), Miami, FL, USA, 16–20 September 2019; pp. 1381–1391.

3. Working Group C, ARAIM Technical Subgroup. Milestone 3 Report. February 2016. Available online: <http://www.gps.gov/policy/cooperation/europe/2016/working-group-c/> (accessed on 12 August 2024).
4. Working Group C, ARAIM Technical Subgroup. Reference Airborne Algorithm Description Document (ADD), (Version 3.1). 2019. Available online: http://web.stanford.edu/group/scpnt/gpslab/website_files/maast/ARAIM_TSG_Reference_ADD_v3.1.pdf (accessed on 12 August 2024).
5. Qi, H.; Wang, X.; Cui, X.; Lu, M. ARAIM based on fault detector reuse for reducing computational load. *GPS Solut.* **2023**, *27*, 78. [CrossRef]
6. Blanch, J.; Walter, T.; Milner, C.; Joerger, M.; Pervan, B.; Bouvet, D. Baseline advanced RAIM user algorithm: Proposed updates. In Proceedings of the 2022 International Technical Meeting of The Institute of Navigation, Long Beach, CA, USA, 25–27 January 2022; pp. 229–251.
7. Blanch, J.; Walter, T.; Enge, P. Exclusion for advanced RAIM: Requirements and a baseline algorithm. In Proceedings of the 2014 International Technical Meeting of the Institute of Navigation, San Diego, CA, USA, 27–29 January 2014; pp. 99–107.
8. Ge, Y.; Wang, Z.; Zhu, Y. Reduced ARAIM monitoring subset method based on satellites in different orbital planes. *GPS Solut.* **2017**, *21*, 1443–1456. [CrossRef]
9. Zhai, Y. Ensuring Navigation Integrity and Continuity Using Multi-Constellation GNSS. Ph.D. Dissertation, Illinois Institute of Technology, Chicago, IL, USA, 2018.
10. Zhai, Y.; Joerger, M.; Pervan, B. Fault exclusion in multi-constellation global navigation satellite systems. *J. Navig.* **2018**, *71*, 1281–1298. [CrossRef]
11. Joerger, M.; Pervan, B. Fault detection and exclusion using solution separation and chi-squared ARAIM. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 726–742. [CrossRef]
12. Blanch, J.; Walter, T.; Enge, P. Protection levels after fault exclusion for advanced RAIM. *Navig. J. Inst. Navig.* **2017**, *64*, 505–513. [CrossRef]
13. Blanch, J.; Walter, T.; Enge, P.; Lee, Y.; Pervan, B.; Rippl, M.; Spletter, A.; Kropp, V. Baseline advanced RAIM user algorithm and possible improvements. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 713–732. [CrossRef]
14. Lesouple, J.; Robert, T.; Sahmoudi, M.; Tourneret, J.Y.; Vigneau, W. Multipath mitigation for GNSS positioning in an urban environment using sparse estimation. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 1316–1328. [CrossRef]
15. Karaim, M.; Elsheikh, M.; Noureldin, A.; Rustamov, R.B. GNSS error sources. *Multifunct. Oper. Appl. GPS* **2018**, *32*, 137–144.
16. Nikiforov, I. From pseudorange overbounding to integrity risk overbounding. *Navigation* **2019**, *66*, 417–439. [CrossRef]
17. Zhao, L.; Zhang, J.; Li, L.; Yang, F.; Liu, X. Position-domain non-Gaussian error overbounding for ARAIM. *Remote Sens.* **2020**, *12*, 1992. [CrossRef]
18. Cassel, R. Real-Time ARAIM Using GPS, GLONASS, and Galileo. Master's Dissertation, Illinois Institute of Technology, Chicago, IL, USA, 2017.
19. Stanford University. Software: MATLAB Algorithm Availability Simulation Tool, (Version 3.1). Available online: <https://gps.stanford.edu/resources/software-tools/maast/> (accessed on 12 August 2024).
20. Blanch, J.; Walter, T.; Enge, P. Fast multiple fault exclusion with a large number of measurements. In Proceedings of the 2015 International Technical Meeting of the Institute of Navigation, Dana Point, CA, USA, 26–28 January 2015; pp. 696–701.
21. Chang, J.; Zhan, X.; Zhai, Y.; Wang, S.; Lin, K. Analysis of BDS GEO satellite multipath effect for GNSS integrity monitoring in civil aviation. *Aerosp. Syst.* **2021**, *4*, 133–141. [CrossRef]
22. Chang, J.B. Modeling and Monitoring of New GNSS Signal Distortions in the Context of Civil Aviation. Ph.D. Dissertation, INPT, Rabat, Morocco, 2016.
23. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1996**, *58*, 237–288. [CrossRef]
24. Candes, E.J.; Wakin, M.B.; Boyd, S.P. Enhancing sparsity by reweighted l1 minimization. *J. Fourier Anal. Appl.* **2008**, *14*, 877–905. [CrossRef]
25. Wu, T.T.; Lange, K. Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat.* **2008**, *2*, 224–244. [CrossRef]
26. Fraley, C.; Hesterberg, T. Least angle regression and LASSO for large datasets. *Istatistical Anal. Data Mining Asa Data Sci. J.* **2009**, *1*, 251–259. [CrossRef]
27. Milner, C.; Pervan, B. Bounding fault probabilities for advanced RAIM. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 2947–2958. [CrossRef]
28. Perea, S.; Meurer, M.; Rippl, M.; Belabbas, B.; Joerger, M. URA/SISA analysis for GPS and Galileo to support ARAIM. *Navig. J. Inst. Navig.* **2017**, *64*, 237–254. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.