TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 12/24 pp457–468
DOI: 10.26599/TST.2023.9010017
Volume 29, Number 2, April 2024

A GNSS Anti-Spoofing Technique Based on the Spatial Distribution Characteristic of the Residual Vectors

Qi'an Wu, Xiaowei Cui*, Mingquan Lu, Pengxiang Yang, and Peng Wu

Abstract: Anti-spoofing is becoming a crucial technique for applications with high navigation accuracy and reliability requirements. Anti-spoofing technique based on Receiver Autonomous Integrity Monitoring (RAIM) is a good choice for most Global Navigation Satellite System (GNSS) receivers because it does not require any change to the hardware of the receiver. However, the conventional RAIM method can only detect and mitigate a single spoofing signal. Some improved RAIM methods can deal with more spoofing signals, but the computational complexity increases dramatically when the number of satellites in view increase or need additional information. This paper proposes a new RAIM method, called the SRV-RAIM method, which has a very low computation complexity regardless of the number of satellites in view and can deal with any number of spoofing signals. The key to the new method is the spatial distribution characteristic of the Satellites' Residual Vectors (SRV). In replay or generative spoofing scenarios, the pseudorange measurements of spoofing signals are consistent, the residual vectors of real satellites and those of spoofing satellites have good separation characteristics in spatial distribution. Based on this characteristic, the SRV-RAIM method is proposed, and the simulation results show that the method can separate the real signals and the spoofing signals with an average probability of 86.55% in the case of 12 visible satellites, regardless of the number of spoofing signals. Compared to the conventional traversal-RAIM method, the performance is only reduced by 3.59%, but the computational cost is reduced by 98.3%, so most of the GNSS receivers can run the SRV-RAIM algorithm in time.

Key words: Global Navigation Satellite System (GNSS); anti-spoofing; Receiver Autonomous Integrity Monitoring (RAIM); Satellites' Residual Vectors (SRV)

1 Introduction

Spoofing is becoming a serious threat to the Global Navigation Satellite System (GNSS) receivers and is regarded as being more malignant than jamming. Compared to jamming, spoofing intends to deceive the receiver to generate false Positioning, Velocity, and Time (PVT) solutions, while jamming intends to make the receiver unable to generate PVT solutions. The spoofer can transmit or generate Global Navigation Satellite System (GNSS) signals with higher power to let the receiver catch and track the fault signals. The pseudorange measurements of the spoofing signals are different from the real ones, which can cause the receiver to get wrong PVT solutions. When the satellites in the receiver tracks are all spoofing satellites, the spoofer can control the PVT solutions of the receiver. Therefore, a spoofer can control the vehicles like unmanned aerial vehicles, which rely heavily on the GNSS

• Qi'an Wu, Xiaowei Cui, and Mingquan Lu are with Department of Electronic Enginnering, Tsinghua University, Beijing 100084, China. E-mail: wu_qi_an@ 163.com; cxw2005@tsinghua.edu.cn; lumq@tsinghua. edu.cn.
• Pengxiang Yang and Peng Wu are with Xi'an Modern Control Technology Research Institute, Xi'an 710065, China. E-mail: 28422673@qq.com; wupengrock@163.com.
* To whom correspondence should be addressed.
Manuscript received: 2022-10-27; revised: 2023-03-06; accepted: 2023-03-17

receiver. For those applications which require navigation information to be extremely accurate and reliable, spoofing interference must be taken into consideration, and the anti-spoofing technique becomes essential. Although many researchers have proposed plenty of anti-spoofing techniques, there does not exist a technique that is fit for all spoofing scenarios. Most of the anti-spoofing techniques require hardware support, such as methods based on noise level monitoring[1, 2], Signal Quality Monitoring (SQM) techniques[3–5], methods based on multi-correlator[6], methods based on multi-antenna[7, 8], methods based on Inertial Navigation System (INS)[9], and so on. Compared to these anti-spoofing methods, anti-spoofing techniques based on Receiver Autonomous Integrity Monitoring (RAIM)[10–17] are fit for most GNSS receivers, for they do not require any change to the hardware or require additional devices like INS. Therefore, RAIM is a good choice for most general receivers.

The least-square residual method, parity vector method, and range-comparison method are three basic RAIM methods and are proved to be almost equivalent[12]. The main idea of these methods is to compare the test statistic with the threshold. If the test statistic exceeds the threshold, the existence of fault pseudorange is declared. If there only exists one single spoofing signal, the pseudorange with the characteristic bias line nearest to the parity vector is regarded as the fault pseudorange[13]. If there is more than one spoofing signal, the conventional RAIM can only detect the existence of spoofing, but cannot exclude spoofing signals. To deal with multi-spoofing situations, some improved methods were proposed, such as NIORAIM[14], MHSS[15], Random Traversal-RAIM[17], and so on. The NIORAIM algorithm can reduce integrity levels and improve RAIM availability by lowering the slope. The algorithm calculates the optimal weights used in a conventional weighted least-squares algorithm to minimize the integrity level. To calculate the optimal weights, all the possible combinations of fault satellites need to be considered. The MHSS algorithm also considers all the fault-free cases and needs to calculate the partial position solutions of each fault-free case. The Random Traversal-RAIM can be considered as an advanced use of the conventional RAIM (cRAIM) algorithm. The Random Traversal-RAIM uses cRAIM to test all possible combinations of fault-free satellites until the cRAIM is effective. All these three algorithms consider every possible combination of fault satellites, so their computations can increase rapidly when the number of satellites in view or the number of fault satellites increases.

In the previous studies, only the numerical value of the pseudorange residual is considered. However, the pseudorange residual of each satellite is not a scalar but a vector, which is called the Satellites' Residual Vector (SRV) in this paper. This paper focuses on the spatial distribution of the pseudorange residual vector of each satellite and based on this, a new anti-spoofing method, called the SRV-RAIM method, is proposed. In replay or generative spoofing scenarios, the pseudorange measurements of spoofing signals are consistent, the residual vectors of real satellites and those of spoofing satellites have good separation characteristics in spatial distribution according to the simulation. Based on this, the SRV-RAIM algorithm can quickly find five signals that are all real or spoofing, and based on these five signals, the rest of the signals can be easily divided into two groups: the real group and the spoofing group. The SRV-RAIM makes no assumptions about the number of spoofing signals and does not traverse all the possible combinations of spoofing signals, so the algorithm has a very low computational complexity, regardless of the number of satellites in view or the number of spoofing signals. According to the simulations, the computational cost of the SRV-RAIM method is reduced by 98.7%, compared with the conventional traversal-RAIM, and with only a 3.59% of performance loss. The low computational complexity allows the SRV-RAIM algorithm to be able to run in time on most of the GNSS receivers.

The remainder of this paper is organized in the following way. In Section 2, the definition of the SRV is introduced and the spatial distribution characteristic of SRVs is studied. In Section 3, the SRV-RAIM method is proposed, and simulations are done to evaluate the performance of the SRV-RAIM method. Section 4 summarizes the research and puts forward some prospects for future research.

2 Spatial Distribution Characteristic of Residual Vectors

2.1 Definition of SRV

In this subsection, the definition of the SRV is proposed. Firstly, the mathematical expression of the pseudorange residual of each satellite in the spoofing situation is derived. Then, the expression of the SRV is given.

The linear positioning equation can be expressed as

$$\Delta y = G\Delta x + \varepsilon \tag{1}$$

where Δy is the pseudorange residual vector, Δx is the positioning error, ε is the measurement error, and G is the geometric matrix defined as

$$G = \begin{bmatrix} -\frac{x_1^s - x_p}{r_1} & -\frac{y_1^s - y_p}{r_1} & -\frac{z_1^s - z_p}{r_1} & 1 \\ -\frac{x_2^s - x_p}{r_2} & -\frac{y_2^s - y_p}{r_2} & -\frac{z_2^s - z_p}{r_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{x_N^s - x_p}{r_N} & -\frac{y_N^s - y_p}{r_N} & -\frac{z_N^s - z_p}{r_N} & 1 \end{bmatrix} \tag{2}$$

where $\left(x_i^s, y_i^s, z_i^s\right)$ is the position of the i-th satellite; $\left(x_p, y_p, z_p\right)$ is the estimated receiver position; r_i is the geometric distance between the i-th satellite and the estimated receiver position; N is the number of satellites. For simplicity, the measurement error ε is ignored in the following mathematical derivation.

Equation (1) can be solved according to the iterative least squares method. In normal situations, the position can be exactly solved, so Δy is a zero vector. In the spoofing situation, the position is deceived to a false solution, and Δy is no longer a zero vector, defined as

$$\begin{cases} \Delta y = \begin{bmatrix} \Delta y_a \\ \Delta y_s \end{bmatrix} = \begin{bmatrix} G_a \Delta x_{ap} \\ G_s \Delta x_{sp} \end{bmatrix} = \begin{bmatrix} G_a \left(x_a - x_p\right) \\ G_s \left(x_s - x_p\right) \end{bmatrix}, \\ G = \begin{bmatrix} G_a \\ G_s \end{bmatrix} \end{cases} \tag{3}$$

where subscript a represents authentic signals and subscript s represents spoofing signals. x_a is the positioning result (including the user clock bias) solved from real pseudorange measurements. x_s is the positioning result solved from spoofing pseudorange measurements. x_p is solved from real and spoofing pseudorange measurements. $\Delta x_{ap} = x_a - x_p$ is the relative position between the real position x_a and the estimated position x_p. $\Delta x_{sp} = x_s - x_p$ is the relative position between the spoofing position x_s and the estimated position x_p.

Equation (1) is usually solved by the iteration method. The estimate of Δx is expressed as

$$\Delta \hat{x} = \left(G^{\mathrm{T}} G\right)^{-1} G^{\mathrm{T}} \Delta y \tag{4}$$

Then, we have

$$G^{\mathrm{T}} \Delta y = G_a^{\mathrm{T}} \Delta y_a + G_s^{\mathrm{T}} \Delta y_s = G^{\mathrm{T}} G \Delta \hat{x} \tag{5}$$

According to Eqs. (3) and (5), Δx_{ap} and Δx_{sp} can be expressed as

$$\begin{cases} \Delta x_{ap} = \left(G^{\mathrm{T}} G\right)^{-1} G_s^{\mathrm{T}} G_s \Delta x_{as} + \Delta \hat{x}, \\ \Delta x_{sp} = \left(G^{\mathrm{T}} G\right)^{-1} G_a^{\mathrm{T}} G_a \Delta x_{sa} + \Delta \hat{x} \end{cases} \tag{6}$$

where $\Delta x_{as} = -\Delta x_{sa} = x_a - x_s$ is the relative position between real position and spoofing position. $\Delta \hat{x}$ is usually solved by iteration method, and when the iteration converges, $\Delta \hat{x}$ is approximately a zero vector or the magnitude of $\Delta \hat{x}$ is very small comparing to Δx_{ap} or Δx_{sp}. Therefore, Eq. (6) can be approximated as

$$\begin{cases} \Delta x_{ap} \approx \left(G^{\mathrm{T}} G\right)^{-1} G_s^{\mathrm{T}} G_s \Delta x_{as}, \\ \Delta x_{sp} \approx \left(G^{\mathrm{T}} G\right)^{-1} G_a^{\mathrm{T}} G_a \Delta x_{sa} \end{cases} \tag{7}$$

According to Formula (7), Δy_a and Δy_s can be expressed as

$$\begin{cases} \Delta y_a = G_a \left(G^{\mathrm{T}} G\right)^{-1} G_s^{\mathrm{T}} G_s \Delta x_{as}, \\ \Delta y_s = G_s \left(G^{\mathrm{T}} G\right)^{-1} G_a^{\mathrm{T}} G_a \Delta x_{sa} \end{cases} \tag{8}$$

Equation (8) shows that the pseudorange residual vectors Δy_a and Δy_s depend on the constellation of the real satellites and spoofing satellites, and also depend on the relative position Δx_{as}. Although Eq. (2) shows that G depends on the estimated position x_p, G is not sensitive to x_p, that is, the value of G at x_p and the value of G at x_a are considered approximately equal.

In the previous studies, Δy is usually called pseudorange residual vector. In this paper, SRV is defined as follows:

$$\begin{cases} SRV_{ai} = \Delta y_{ai} g_{ai} = g_{ai} g_{ai}^{\mathrm{T}} \left(G^{\mathrm{T}} G\right)^{-1} G_s^{\mathrm{T}} G_s \Delta x_{as}, \\ SRV_{sj} = \Delta y_{sj} g_{sj} = g_{sj} g_{sj}^{\mathrm{T}} \left(G^{\mathrm{T}} G\right)^{-1} G_a^{\mathrm{T}} G_a \Delta x_{sa} \end{cases} \tag{9}$$

where SRV_{ai} and SRV_{sj} are the SRVs of the i-th real satellite and the j-th spoofing satellite, respectively; Δy_{ai} is the i-th element of the Δy_a, refers to the least-square residual of the i-th real satellite; Δy_{sj} is the j-th element of the Δy_s, refers to the least-square residual of the j-th spoofing satellite; g_{ai}^{T} is the i-th row of matrix G_a, and g_{sj}^{T} is the j-th row of matrix G_s.

The SRV of the i-th satellite, defined as $SRV_i = \Delta y_i g_i$, indicates that there exists a pseudorange estimation error of Δy_i in the direction of g_i. If we do not take into account of all the measurement errors, the SRV of all satellites (SRVs) are zero vector. If the measurement errors are Gaussian errors, Δy_i is also a Gaussian error, usually small. In this condition, the SRVs are small vectors. If there exists spoofing pseudorange, the value of $|\Delta y_i|$ is usually very large, which means that SRV_i is a large vector.

2.2 Spatial distribution characteristic of the SRV

In this subsection, the spatial distribution of the SRVs is studied, and it is proved that the pseudorange residual vector Δy in previous papers is actually just the projection value of SRVs in the dimension of clock bias.

According to Eq. (9), and considering that $\Delta\hat{x}$ is a small vector comparing to the SRVs, Eq. (5) can be rewritten as

$$G^{\mathrm{T}}\Delta y = \sum_{i=1}^{N_a} SRV_{ai} + \sum_{j=1}^{N_s} SRV_{sj} \approx 0 \quad (10)$$

where N_a and N_s are the numbers of real satellites and spoofing satellites, respectively.

Formula (10) shows that the sum of the SRV_a (the residual vectors of real satellites) is opposite to the sum of the SRV_s (the residual vectors of spoofing satellites). The sum of SRV_a can be expressed as $G_a^{\mathrm{T}}G_a\Delta x_{ap}$, and it is obviously not a zero vector. When there exists spoofing pseudorange, the positioning result will usually be deceived to a false one (x_p), which is far from the real position x_a, that is, $\Delta x_{ap}^{\mathrm{T}}\Delta x_{ap}$ is usually large. Therefore, the sum of SRV_a is usually a large vector. The sum of a series vector usually reflects the direction in which these vectors are concentrated. Therefore, Formula (10) indicates that the SRV_a and the SRV_s may be distributed in different regions of space. Note that Formula (10) does not guarantee that the SRV_a and the SRV_s are separated, but there may exist a high probability that the SRV_a and the SRV_s are separated. If there are no spoofing signals, the sum of SRV_a is a zero vector, and the SRV of each satellite is also a small vector. This indicates that the SRV_a are distributed near the zero vector.

The ideal condition is that the SRV_a and the SRV_s can be divided by a plane, that is, there exists a unit vector v that meets the following Formula:

$$\min_i SRV_{ai}^{\mathrm{T}}v > \max_j SRV_{sj}^{\mathrm{T}}v \quad (11)$$

Define function $P(SRV_{ai}, SRV_{sj}, v)$ as

$$P\left(SRV_{ai}, SRV_{sj},\ v\right) = SRV_{ai}^{\mathrm{T}}v - SRV_{sj}^{\mathrm{T}}v = v^{\mathrm{T}}Q_{i,j}\Delta x_{as} \quad (12)$$

where

$$Q_{i,j} = g_{ai}g_{ai}^{\mathrm{T}}\left(G^{\mathrm{T}}G\right)^{-1}G_s^{\mathrm{T}}G_s + g_{sj}g_{sj}^{\mathrm{T}}\left(G^{\mathrm{T}}G\right)^{-1}G_a^{\mathrm{T}}G_a \quad (13)$$

Formula (11) can be transformed into the following expression:

$$\forall i, j, \exists v,\ \text{s.t.},\ P\left(SRV_{ai}, SRV_{sj}, v\right) > 0 \quad (14)$$

Formula (14) indicates that all of the vectors $Q_{i,j}\Delta x_{as}$ are distributed at a side of a plane. The distribution of $Q_{i,j}\Delta x_{as}$ depends on the constellation of the real satellites and spoofing satellites, and also depends on Δx_{as}. To be more precise, the length of Δx_{as} (the 2-norm of vector Δx_{as}) has no effect on the distribution of $Q_{i,j}\Delta x_{as}$, instead, the direction of Δx_{as} has a big effect on the distribution.

It is difficult to describe the distribution of vectors $Q_{i,j}\Delta x_{as}$ in 4-dimensional space. In this paper, a metric is proposed to describe the distribution of $Q_{i,j}\Delta x_{as}$.

The projection of SRV_i (the SRV of the i-th satellite) onto the vector v is defined as follows:

$$p_i(v) = v^{\mathrm{T}}SRV_i, i = 1, 2, \ldots, N \quad (15)$$

Assume that $p_1(v) > p_2(v) > \cdots > p_N(v)$, and the minimum projection of real satellites is $p_{m_a}(v)$, the maximum projection of spoofing satellites is $p_{m_s}(v)$. The Number of Hybrids (NH) at direction v is defined as

$$NH(v) = m_a - m_s + 1 \quad (16)$$

where m_a is the subscript of the minimum projection of real satellites, and m_s is the subscript of the maximum projection of spoofing satellites. The metric is defined as

$$NH_0 = \min_v NH(v) \quad (17)$$

When $NH_0 = 0$, the distribution of $Q_{i,j}\Delta x_{as}$ is considered excellent, that is, the SRV_a and the SRV_s are spatially separated. The distribution of $Q_{i,j}\Delta x_{as}$ is considered becoming worse as the value of NH_0 increases.

NH_0 is a function of G_a (the constellation of real satellites), G_s (the constellation of spoofing satellites), and Δx_{as} (the relative position), but it is hard to mathematically express it. In this paper, simulations are done to analyze the statistical properties of NH_0. The simulation conditions are shown in Table 1. The simulation tries to go through all the possible values of G_a, G_s, and Δx_{as} to cover all the possible spoofing scenarios. The spoofing pseudorange measurement differs from the real one by more than 100 m, considering it is a common condition in spoofing scenarios. Measurement error, ionospheric error, tropospheric error, etc., are not considered here. The simulation results are shown in Figs. 1 and 2, and

Table 1 Simulation conditions.

Simulation condition	Value
Number of satellites in view	12
Number of spoofing satellites	4094 situations
Distance between real position and spoofing position	Range from 100 m to 4 km; 100 random samples
Measurement error, ionospheric error, tropospheric error, etc.	0 m
Spoofing pseudorange minus real pseudorange	>100 m

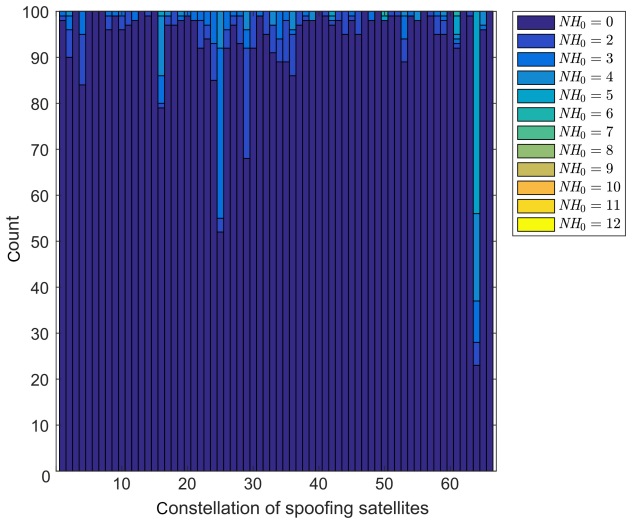

Fig. 1 Distribution of NH_0 in different spoofing satellite constellations when there are two spoofing satellites.

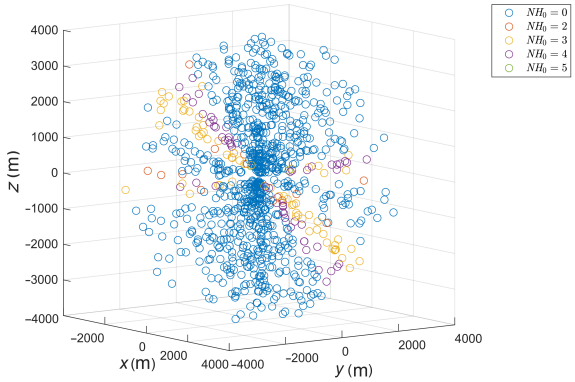

Fig. 2 Distribution of NH_0 versus the relative position between spoofing and real in a certain spoofing constellation.

Table 2. Note that it is not advisable to solve the Eq. (17) by traversing the vector v in the whole 4-dimentional space. In the simulations, Eq. (17) is solved by traversing

Table 2 Relationship between the probability distribution of NH_0 and the number of spoofing satellites.

(%)

Number of spoofing satellites	NH_0				
	0	2	3	4	$\geqslant 5$
1	100.0	0	0	0	0
2	94.5	2.4	1.7	0.8	0.6
3	87.1	5.0	3.6	2.7	1.6
4	81.9	6.9	5.3	3.6	2.3
5	78.9	7.7	6.4	4.1	2.9
6	77.6	8.4	6.3	4.3	3.4
7	78.2	8.0	6.4	4.2	3.2
8	80.5	7.4	5.8	3.7	2.6
9	84.9	6.0	4.3	3.0	1.8
10	92.7	2.1	2.4	1.1	1.7
11	100.0	0	0	0	0
Average	80.4	7.3	5.7	3.8	2.8

the vector v in a subspace S_0,

$$S_0 = \left\{ \sum_{i=1}^{N} a_i SRV_i \right\}, a_i \in \{0, 1\} \tag{18}$$

It is difficult to give a clear figure of the distribution of NH_0 of every spoofing constellation, because there are too much. Figure 1 only shows the distribution of NH_0 in different spoofing satellite constellations with two spoofing satellites. The spoofing satellite constellations are labeled from 1 to 66. The result shows that although the number of spoofing signals is the same, the distribution of NH_0 changes with G_s. Note that NH_0 depends on G_a and G_s, but here only G_s is considered because in the simulation G (the constellation of all satellites in view) is definite, that is, once G_s is determined, G_a is also determined.

NH_0 also depends on Δx_{as}. Figure 2 shows the distribution of NH_0 versus Δx_{as} (the clock bias is ignored in the figure) in a certain spoofing constellation. The result shows that NH_0 only depends on the direction of Δx_{as}, but does not depend on the length of Δx_{as}. According to the results shown in Figs. 1 and 2, it is verified that the spoofer can deliberately set the value of G_s,G_a, and Δx_{as} to make the SRV_a and the SRV_s can not be spatially separated, but in most scenarios, the SRV_a and the SRV_s are spatially separated.

Table 2 shows the probability distribution of NH_0 with the different number of spoofing satellites, and shows two main pieces of information: the one is that, when the number of spoofing signals approaches 6 (half the number of satellites in view), the distribution of NH_0 becomes worse, indicating that the spatial separation condition become worse; the other is that the average probability of $NH_0 = 0$ is 80.37%, which means that in most of the spoofing conditions, the SRV_a and the SRV_s are spatially separated. This spatial distribution characteristic of SRV makes the receiver able to separate real satellites from spoofing satellites. Another thing to point out here is that, the distribution of NH_0 improves when the number of spoofing satellites in more than 6. This is because the pseudorange of spoofing signals are also consistent, and the GNSS receiver cannot tell which group of signals is real or spoofing (only by the pseudorange measurements), but can tell they belong to different groups.

In the previous studies, only the residual vector Δy is considered. In this paper, according to the definition of SRV in Eq. (9), the residual vector Δy is equivalent to the time-dimension of SRV, as shown in the following:

$$\Delta y = [SRV_1, SRV_2, \ldots, SRV_N]^{\mathrm{T}} e_4 \tag{19}$$

where $e_4 = [0, 0, 0, 1]^{\mathrm{T}}$.

According to Eqs. (16) and (19), the distribution of Δy can be measured by $NH(e_4)$. Table 3 shows the simulation result of $NH(e_4)$.

Table 3 shows that, if only considering the value of Δy (the time-dimension of the SRV), there are only 43.31% of spoofing conditions, in which the residuals of real satellites and the residuals of spoofing satellites are separated. However, Table 3 shows that, in the 4-dimensional space, there are 80.37% of spoofing conditions, in which the SRV_a and the SRV_s are spatially separated. This suggests that when viewed in higher dimensions, the pseudorange residuals will have better spatial separation properties. This opens up more possibilities for separating real satellites from spoofing satellites.

3 SRV-RAIM Method

In this section, the main idea of the SRV-RAIM method is introduced and a basic algorithm based on this idea is proposed. Simulations are also done to evaluate the performance of the SRV-RAIM algorithm and compared with the performance of the conventional traversal-RAIM.

According to the previous section, the SRV_a and the SRV_s are spatially separated in most of the spoofing conditions. However, in order to make full use of this characteristic to achieve the purpose of separating the real and spoofing signals, two problems remain to be solved: the one is that the best projection vector $v_0 = \arg\min_v NH(v)$ is unable to be solved mathematically because there is no prior knowledge about G_s and $x_a s$, only G is known; the other is that even though the v_0 is given, it is also difficult to directly separate real satellites from spoofing satellites because the number of spoofing satellites is unknown.

SRV_a and SRV_s are discretely distributed in the space, which means that the set of vectors that satisfies Eq. (17) is the union of continues subspaces and v_0 is the vector in this set. Therefore, the basic strategy to solve the first problem is searching the value of v in a subspace S_v with a finite number of vectors, instead of the whole space, to find v_0. This strategy may reduce the probability to find $NH_0 = 0$, but it is a necessary sacrifice to reduce the amount of computation. If v_0 is a uniform distributed random vector in space, then a good strategy for choosing a subspace S_v is as follows: the vectors in the subspace should be as uniformly distributed in space as possible.

Table 3 Average probability distribution of $NH(e_4)$. (%)

$NH(e_4)$	Average probability
0	43.3
2	9.2
3	8.0
4	7.3
$\geqslant 5$	32.2

To solve the second problem, instead of finding all real satellites at once, finding 5 real satellites or 5 spoofing satellites is easier. If v_0 is given and $NH_0 = 0$, the biggest five projections $D_1 = \{p_i(v_0)\}_{i\leqslant 5}$ correspond to five real satellites ($N_a \leqslant 5$) or the smallest five projections $D_2 = \{p_i(v_0)\}_{i\geqslant N-4}$ correspond to five spoofing satellites ($N_s \geqslant 5$). After the five real (or spoofing) satellites are found, the rest of real (or spoofing) satellites can be found easily, because the pseudorange measurements of real (or spoofing) satellites are consistent, but the pseudorange measurements of real satellites are not consistent with those of spoofing satellites. Therefore, if a vector v on which the biggest five projections $D_1 = \{p_i(v)\}_{i\leqslant 5}$, or the smallest five projections $D_2 = \{p_i(v)\}_{i\geqslant N-4}$ correspond to five satellites, whose pseudorange measurements are consistent, can be found, the real satellites and the spoofing satellites can be separated. Based on this idea, it is not necessary the find the best vector v_0, which is impossible. That also means that even if the SRV_a and the SRV_s are not spatially separated ($NH_0 > 0$), it may also be possible to separate the SRV_a from the SRV_s. That gives another chance to reduce the number of vectors in the subspace S_v.

Based on the above analysis, the problem of separating the SRV_a and the SRV_s can be simplified as a hypothesis test problem as follows:

(1) H_0: $\exists\ v \in S_v$, s.t., $D_1 \subseteq S_a, D_1 \subseteq S_s, D_2 \subseteq S_a$, or $D_2 \subseteq S_s$;

(2) H_1: $\nexists\ v \in S_v$, s.t., $D_1 \subseteq S_a, D_1 \subseteq S_s, D_2 \subseteq S_a$, or $D_2 \subseteq S_s$.

S_a is the set of projections of real satellites, and S_s is the set of projections of spoofing satellites.

In this paper, four projection vector subspaces (S_v) are considered, and $p(H_0)$ of each subspace is given by simulations. The simulation conditions are shown in Table 1, and simulation results are shown in Table 4.

Table 4 Simulation results of $p(H_0)$ about the four subspaces. $p_0(H_0) = p(H_0; S_v = S_0)$, $p_1(H_0) = p(H_0; S_v = S_1)$, $p_2(H_0) = p(H_0; S_v = S_2)$, and $p_3(H_0) = p(H_0; S_v = S_3)$.

(%)

Number of spoofing satellites	$p_0(H_0)$	$p_1(H_0)$	$p_2(H_0)$	$p_3(H_0)$
1	100.00	100.00	100.00	100.00
2	100.00	99.27	99.95	100.00
3	99.89	97.60	99.28	99.83
4	99.66	94.83	97.67	99.10
5	98.57	89.01	93.55	96.49
6	97.42	85.58	90.69	94.15
7	98.27	89.25	93.62	96.53
8	99.55	94.63	97.52	99.06
9	99.81	97.26	99.26	99.70
10	100.00	99.00	99.91	100.00
11	100.00	100.00	100.00	100.00
Average	98.72	90.98	94.74	97.08

Assume that v_0 is evenly distributed in the 4-dimentional space, in this case, the vectors in S_v is suggested to be also evenly distributed in the 4-dimentional space, but it is not easy to construct such a perfect subspace S_v. An easier way to construct a subspace is that the coordinates of the vectors in the subspaces are lattice points (the value of coordinate axis is -1, 0, or 1) near the origin, and the lattice points are as far away as possible. Except for S_0, this paper gives three subspaces, defined as follows:

$$\begin{cases} S_1 = \left\{ e_1, e_2, e_3, e_4, \sum_{i=1}^{4} a_i e_i \right\}, \\ S_2 = S_1 \cup \left\{ \sum_{i=1}^{4} b_i e_i \right\}, \\ S_3 = S_2 \cup \left\{ \sum_{i=1}^{4} c_i e_i \right\} \end{cases} \tag{20}$$

where $e_i (i = 1, 2, 3, 4)$ is a 4×1 vector, whose i-th element is 1 and other elements are 0; $a_1 = 1, a_i \in \{-1, 1\} (i = 2, 3, 4)$; $[b_1, b_2, b_3, b_4]^{\mathrm{T}}$ is the column of the matrix $B = [B_1, B_2]$; $[c_1, c_2, c_3, c_4]^{\mathrm{T}}$ is the column of the matrix $C = [C_1, C_2]$,

$$B_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix} \tag{21}$$

$$C_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix} \tag{22}$$

Table 4 shows that the average $p(H_0; S_v = S_0)$ is 98.72%, much larger than 80.37% (the average $p(NH_0 = 0)$ in Table 2). That is because even if $NH_0 > 0$, H_0 could also be possibly true. S_0 contains 4094 vectors, so it will take a large amount of computation to find the right v_0. S_1 only contains 12 vectors, but the average $p_1(H_0)$ is reduced to 90.98%, comparing to $p_0(H_0)$. By expanding S_1 to S_2, the average $p(H_0)$ can arise to 94.74% ($p_2(H_0)$), but the number of vectors in subspace only increases 12. By expanding S_2 to S_3, the average $p(H_0)$ arises to 97.08% ($p_3(H_0)$), and the number of vectors increases to 40. Table 4 shows that by expanding the set S_v, the probability of $p(H_0)$ will increase, but the computational cost will also increase. $p_3(H_0)$ is very close to $p_0(H_0)$, so the performance gained from expanding the S_3 is no longer significant, but the computational cost will continue to increase. Therefore, in this paper, S_3 is considered a good balance between performance and computational complexity, and is chosen for further simulations.

According to the simulation results, the two problems mentioned at the beginning are almost solved. However, in the previous discussion, a question is ignored: How to test $D_1 \subseteq S_a$, $D_1 \subseteq S_s$, $D_2 \subseteq S_a$, or $D_2 \subseteq S_s$. In this paper, we assume that the pseudorange measurements of real (or spoofing) satellites are consistent, but the pseudorange measurements of real satellites are not consistent with those of spoofing satellites. Based on this assumption, the question is translated to test if the pseudorange measurements of five satellites, whose projections are D_1 or D_2, are consistent. A common method to test the consistency of pseudorange measurements is to use SSE as the test statistic, which is defined as

$$\mathrm{SSE} = \sqrt{\frac{\Delta y^{\mathrm{T}} \Delta y}{n - 4}} \tag{23}$$

where Δy is the least squares residuals, n is the number of pseudorange measurements.

Based on the above discussion, an SRV-RAIM algorithm is proposed, as shown in Algorithm 1.

Note that the method only divides the pseudorange measurements into two groups. Extra information is needed to identify which group is real, and in this paper, the identification is ignored. Step 39 in Algorithm 1 is a detailed supplement of the SRV-RAIM algorithm, but not the core idea of it. Therefore, in this paper, only steps before Step 39 of the method are evaluated by simulations. The simulation conditions are basically the same as in Table 1, except that the measurement error is considered as a Gauss noise with mean being 0 and standard deviation being 4 m. According to Ref. [12], the test threshold TH in the SSE method is calculated in the alarm rate of 95%. The simulation result is shown in Table 5.

Algorithm 1 SRV-RAIM algorithm

Define:

v_k: the k-th vector of S_v;

N_{S_v}: number of vectors in S_v;

$TH(n)$: thresholds of SSE with different satellite number n;

SSE(S): SSE test statistic calculated according to the satellites in set S;

1: $Flag \leftarrow 0$;
2: **for** $k = 1, 2, \ldots, N_{S_v}$ **do**
3: **for** $n = 1, 2, \ldots, N$ **do**
4: $p_i \leftarrow SRV_i^{\mathrm{T}} v_k$;
5: **end for**
6: $(p_{\rho_1}, p_{\rho_2}, \ldots, p_{\rho_N}) \leftarrow$ sorted $(p_1, p_2, \ldots, p_N)$;
7: $G_1 \leftarrow \{\rho_i\}_{i \leqslant 5}$;
8: $G_2 \leftarrow \{\rho_i\}_{i > 5}$;
9: $m \leftarrow 2$;
10: **while** $m > 0$ **do**
11: **if** SSE(G_1) $< TH(5)$ **then**
12: **for** each $\rho_j \in G_2$ **do**
13: **if** SSE($G_1 \cup \{\rho_j\}$) $< TH(N_{G_1} + 1)$ **then**
14: $G_1 \leftarrow G_1 \cup \{\rho_j\}$;
15: $N_{G_1} \leftarrow N_{G_1} + 1$;
16: Throw out ρ_j from G_2;
17: $N_{G_2} \leftarrow N_{G_2} - 1$;
18: **end if**
19: **end for**
20: **if** $N_{G_2} < 5$ or SSE(G_2) $< TH(N_{G_2})$ **then**
21: $Flag \leftarrow 1$;
22: $m \leftarrow 0$;
23: **else**
24: $G_1 \leftarrow \{\rho_i\}_{i \geqslant N-4}$;
25: $G_2 \leftarrow \{\rho_i\}_{i < N-4}$;
26: $m \leftarrow m - 1$;
27: **end if**
28: **else**
29: $G_1 \leftarrow \{\rho_i\}_{i \geqslant N-4}$;
30: $G_2 \leftarrow \{\rho_i\}_{i < N-4}$;
31: $m \leftarrow m - 1$;
32: **end if**
33: **end while**
34: **if** $Flag = 1$ **then**
35: **break**;
36: **end if**
37: **end for**
38: **if** $Flag = 1$ **then**
39: Check if the pseudorange measurements in G_1 and G_2 keep consistent for some seconds, respectively;
40: **else**
41: Algorithm is failed;
42: **end if**

The method is considered "success" if real signals and spoofing signals are successfully separated. The method is considered "false" if the *Flag* is 1, but there is a group that contains both real signals and spoofing signals. The method is considered "fail" if *Flag* is 0. Table 5 shows that the SRV-RAIM algorithm has an average of 86.55% success rate, 3.01% of false rate, and 10.44% of failure. Comparing to the ideal performance 97.08%, shown in Table 4, the practical performance is decreased by 10.53%. The main reason for the obvious performance reduction is the high miss detection and false alarm rate of the SSE method. When a miss detection of the SSE method happens, the SSE test statistic may still be smaller than the threshold even if there exist both spoofing signals and real signals. When the SSE method arises a false alarm, the SRV-RAIM algorithm may loss the right projection vector v_k or fail in dividing the real signals and spoofing signals (Steps 9–33 in Algorithm 1). From Algorithm 1, it is obvious

Table 5 Successful rate, false rate, and failed rate of the SRV-RAIM algorithm.

(%)

Number of spoofing satellites	Success rate	False rate	Fail rate
1	93.08	6.92	0
2	93.83	5.86	0.30
3	93.61	5.35	1.04
4	92.01	1.34	6.65
5	84.55	1.93	13.52
6	82.22	2.61	15.17
7	83.24	2.84	13.92
8	90.21	3.41	6.38
9	91.85	6.92	1.24
10	91.91	7.95	0.14
11	92.75	7.25	0
Average	86.55	3.01	10.44

that the SSE method takes a very important part, so the performance of SSE will greatly affect the performance of the algorithm. Both the miss detection and the false alarm of the SSE method can cause a false separation or a failure of the SRV-RAIM algorithm. The performance of the SSE method decreases significantly in the following two scenarios:

(1) The real pseudorange is close to the spoofing one;

(2) The quality of satellite geometry is inferior[18, 19].

The first case can be avoided because, in normal spoofing situations, the spoofing pseudorange will not always be close to the real one over time. Therefore, Step 39 in Algorithm 1 can correct the false separation. The second case happens because, in the SRV-RAIM method, the first step is testing the consistency of only five pseudorange measurements. The quality of satellite geometry may be inferior when the number of satellites is small. In this case, the false separation and failure probability can be reduced only when the difference between the spoofing pseudorange and the real pseudorange is large enough. Therefore, Step 39 in Algorithm 1 is very important for the reliability of the SRV-RAIM method, but in this paper, it will not be discussed further.

The computational effort of the method is mainly focused on the search for the correct vector $v_k \in S_v$, and frequent positioning calculations to separate the pseudorange measurements into two groups. In this paper, the amount of computation is measured by the number of positioning calculations (n_s) needed to separate real signals from spoofing signals, because it is the most computationally expensive part of Algorithm 1. The amount of computation of the SRV-RAIM is shown in Table 6. The result shows that the computational cost of Algorithm 1 is very low, regardless of the number of spoofing satellites. The computational cost only arises a little when the spoofing number grows to 6. This is because the separation conditions of SRV worsens when the spoofing number increases, as shown in Table 2, leading to the need to search more vectors $v_k \in S_v$ to find the first five signals that have consistent pseudorange measurements.

Table 6 Amount of computation of the SRV-RAIM method.

Number of spoofing satellites	Mean (n_s)
1	11.6
2	15.0
3	18.8
4	22.0
5	24.2
6	26.9
7	24.1
8	22.3
9	18.7
10	15.2
11	11.5
Average	23.2

In order to further analyze the advantages and disadvantages of the SRV-RAIM algorithm, the performance of the conventional traversal-RAIM is also proposed. The main idea of the conventional traversal-RAIM method to solve multi-spoofing situations is to go through all the possible fault-free cases, and use SSE method to test if there exist fault signals until find the right one that pass the SSE test. Simulations are done to evaluate the conventional traversal-RAIM method, and the results are shown in Tables 7 and 8.

Table 7 shows that the conventional traversal-RAIM has an average success probability of 90.14%, an average

Table 7 Successful rate, false rate, and failed rate of the conventional traversal-RAIM.

(%)

Number of spoofing satellites	Success rate	False rate	Fail rate
1	94.92	5.08	0
2	95.15	4.85	0
3	94.80	5.13	0.08
4	95.10	0.33	4.58
5	90.24	0.38	9.38
6	86.29	4.00	9.71
7	87.32	3.14	9.54
8	91.99	3.29	4.72
9	93.05	6.84	0.12
10	93.70	6.30	0
11	94.50	5.50	0
Average	90.14	2.87	6.99

Table 8 Amount of computation of the conventional traversal-RAIM method.

Number of spoofing satellites	Mean (n_s)
1	8.4
2	52.1
3	205.0
4	549.5
5	1590.0
6	2784.0
7	1549.4
8	537.2
9	200.6
10	52.2
11	8.2
Average	1361.1

false separation probability of 2.87%, and an average failure probability of 6.99%. The reasons that cause the false separation and failure of the traversal-RAIM are almost the same as SRV-RAIM method. Table 8 shows the amount of computation of the conventional traversal-RAIM method. It shows that the amount of computation increases rapidly when the number of spoofing satellites grews to 6, because of the increase number of possible combinations of fault satellites. The average computational cost of the traversal-RAIM method is 1361.1.

According to Tables 6 and 8, the amount of computation of the conventional traversal-RAIM is smaller than that of the SRV-RAIM method when the spoofing satellite number is 1 or 11. In this case, the SRV-RAIM method firstly tries to find 5 real (or spoofing) satellites, which requires several times of positioning calculations. After finding 5 real (or spoofing) satellites, the SRV-RAIM method needs 6 more times of positioning calculations (Step 12–19 in Algorithm 1) to find other 5 real (or spoofing) satellites. Comparing to the SRV-RAIM method, the conventional traversal-RAIM only needs to traverse 12 fault conditions. If lucky enough, the conventional traversal-RAIM only needs 1 time of positioning calculation, but the SRV-RAIM method needs 7 times of positioning calculation. When the number of spoofing satellites increases, the amount of computation of the SRV-RAIM method only increases a little, but the amount of computation of the conventional traversal-RAIM increases rapidly. The average computational cost of the SRV-RAIM method is only 1.7% of the conventional traversal-RAIM.

By comparing the simulation results of the conventional traversal-RAIM method and the SRV-RAIM method, it can be found that the effective probability of the SRV-RAIM method is 3.59% lower than that of the conventional traversal-RAIM, but the computational cost is reduced by 98.3%. This low computational complexity allows the algorithm to be run in-time on most of the GNSS receivers, and it greatly improves the availability of the GNSS receiver in spoofing environment.

4 Conclusion

In this paper, the SRV is defined for the first time and a metric, called the number of hybrids, is defined to reflect its spatial distribution. Simulations are done to analyze the spatial distribution characteristic of SRV in some spoofing scenarios, in which the pseudorange measurements of spoofing signals are consistent. The results show that, when there are 12 satellites (including spoofing satellites and real satellites) in view, there is 80.37% of spoofing situations, in which the SRV of spoofing satellites and the SRV of real satellites can be spatially separated by a plane. Based on this spatial distribution characteristic, the SRV-RAIM method is proposed, and simulation results show that its' performance is only 3.59% lower than that of the conventional traversal-RAIM method, but the computational cost is only 1.7% of the conventional one, and the computational cost is almost independent of the number of spoofing satellites and the number of all satellites. The low computational complexity of the SRV-RAIM algorithm allows it to be run in-time on most GNSS receivers, and only 3.59% performance loss is acceptable.

The main advantage of the SRV-RAIM algorithm is the extremely low computational complexity, with only little performance loss, comparing to the traversal-RAIM. However, there are two main limitations of the SRV-RAIM method. The first is that the simulations only verify the effectiveness of the SRV-RAIM method in specific spoofing scenarios, in which the pseudorange measurements of spoofing signals are consistent. Therefore, when the number of spoofing signals is more than 4, the spoofing pseudorange measurements may be not consistent, and in this condition, the SRV-RAIM method may not work. The second limitation is that the SRV-RAIM method is strongly affected by SSE method, which means that the SRV-RAIM method inherits all the weaknesses of the SSE method. The SRV-RAIM tries to find out 5 real or spoofing signals, and SSE method is used to check if their pseudorange measurements are consistent. However, when there are only 5 satellites, the quality of satellite geometry is likely to be inferior, that is, the Position Dilution Of Precision (PDOP) may be very large. In this condition, the SSE method is not so trustful. The SRV-RAIM method will perform better in dynamic scenarios, in which the relative position between the real position and the spoofing position changes over time. That is because in dynamic scenarios, the spatial distribution of SRVs changes over time, and if the

method fails in the current epoch, it may success in the next epoch. Beside this, the false separation probability will also decrease in dynamic scenarios, because the difference between the authentic pseudorange and the spoofing pseudorange will increase with time.

Although the SRV-RAIM method proposed in this paper has an excellent performance, the research is still insufficient. The following points need further study:

(1) The simulation results show that the SRV of authentic satellites and the SRV of spoofing satellites have good spatial separation characteristics, but the reason for this characteristic is still unclear.

(2) The influence of the number of satellites in view and the test threshold on the SRV-RAIM method needs to be further studied. In addition, the performance of the method in other spoofing scenarios, in which the pseudorange measurements are not consistent or there exist multiple groups of spoofing satellites, also needs to be further studied.

(3) In this paper, simulation experiments are quite ideal, and the effects of multi-path, ionospheric anomalies, etc., are ignored. The performance of SRV-RAIM method in harsher and more realistic environment needs to be further studied.

Acknowledgment

This work was supported by the National Key R&D Program of China (No. 2021YFA0716603).

References

[1] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle, Pre-despreading authenticity verification for GPS L1 C/A signals, *Navigation*, vol. 61, no. 1, pp. 1–11, 2014.

[2] Y. Hu, S. Bian, K. Cao, J. Cao, and B. Ji, GNSS spoofing detection based on new signal quality assessment model, *GPS Solut.*, vol. 22, no. 1, p. 28, 2018.

[3] A. Broumandan, A. Jafarnia-Jahromi, S. Daneshmand, and G. Lachapelle, Effect of tracking parameters on GNSS receivers vulnerability to spoofing attack, in *Proc. 29th Int. Technical Meeting of the Satellite Division of the Institute of Navigation* (*ION GNSS + 2016*), Portland, OR, USA, 2016, pp. 3033–3043.

[4] S. Chao, J. W. Cheong, A. G. Dempster, H. Zhao, L. Demicheli, and W. Feng, A new signal quality monitoring method for anti-spoofing, in *Proc. China Satellite Navigation Conf.*, Singapore, 2018. pp. 221–231.

[5] B. M. Ledvina, W. J. Bencze, B. Galusha, and I. Miller, An in-line anti-spoofing device for legacy civil GPS receivers, in *Proc. 2010 Int. Technical Meeting of the Institute of Navigation*, San Diego, CA, USA, 2010, pp. 698–712.

[6] X. Shang, F. Sun, L. Zhang, J. Cui and Y. Zhang, Detection and mitigation of GNSS spoofing via the pseudorange difference between epochs in a multicorrelator receiver, *GPS Solut.*, vol. 26, no. 2, p. 37, 2022.

[7] L. He, H. Li, and M. Lu, Dual-antenna GNSS spoofing detection method based on Doppler frequency difference of arrival, *GPS Solut.*, vol. 23, no. 3, p. 78, 2019.

[8] W. Bai, H. Li, Y. Yang, and M. Lu, Motion state monitoring based GNSS spoofing detection method for repeater spoofing attack, in *Proc. 2016 Int. Technical Meeting of the Institute of Navigation*, Monterey, CA, USA, 2016, pp. 732–738.

[9] M. Majidi, A. Erfanian, and H. Khaloozadeh, A new approach to estimate true position of unmanned aerial vehicles in an INS/GPS integration system in GPS spoofing attack conditions, *Int. J. Automat. Comput.*, vol. 15, no. 6, pp. 747–760, 2018.

[10] S. Han, D. Luo, W. Meng, and C. Li, Antispoofing RAIM for dual-recursion particle filter of GNSS calculation, *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 2, pp. 836–851, 2016.

[11] Y. Wei, H. Li, C. Peng, and M. Lu, Time domain differential RAIM method for spoofing detection applications, in *Proc. China Satellite Navigation Conf.*, Singapore, 2019, pp. 606–614.

[12] R. G. Brown, A baseline GPS RAIM scheme and a note on the equivalence of three RAIM methods, *Navigation*, vol. 39, no. 3, pp. 301–316, 1992.

[13] A. Brown and M. Sturza, The effect of geometry on integrity monitoring performance, in *Proc. 46th Annual Meeting of the Institute of Navigation*, Atlantic City, NJ, USA, 1990, pp. 121–129.

[14] P. Y. Hwang and R. G. Brown, NIORAIM integrity monitoring performance in simultaneous two-fault satellite scenarios, in *Proc. 18th Int. Technical Meeting of the Satellite Division of the Institute of Navigation*, Long Beach, CA, USA, 2005, pp. 1760–1771.

[15] B. S. Pervan, S. P. Pullen, and J. R. Christie, A multiple hypothesis approach to satellite navigation integrity, *Navigation*, vol. 45, no. 1, pp. 61–71, 1998.

[16] H. Tao, H. Li, W. Zhang, and M. Lu, A recursive receiver autonomous integrity monitoring (recursive-RAIM) technique for GNSS anti-spoofing, in *Proc. 2015 Int. Technical Meeting of the Institute of Navigation*, Dana Point, CA, USA, 2015, pp. 738–744.

[17] J. Li, H. Li, C. Peng, J. Wen, and M. Lu, Research on the random traversal RAIM method for anti-spoofing applications, in *Proc. China Satellite Navigation Conf.*, Singapore, 2019, pp. 593–605.

[18] M. A. Sturza and A. K. Brown, Comparison of fixed and variable threshold RAIM algorithms, in *Proc. 3rd Int. Technical Meeting of the Satellite Division of the Institute of Navigation*, Colorado Spring, CO, USA, 1990, pp. 437–443.

[19] R. G. Brown, G. Y. Chin, and J. H. Kraemer, Update on GPS integrity requirements of the RTCA MOPS, in *Proc. 4th Int. Technical Meeting of the Satellite Division of the Institute of Navigation*, Albuquerque, NM, USA, 1991, pp. 761–772.

Qi'an Wu received the BEng degree from Tsinghua University, Beijing, China in 2020. He is currently a master student at Department of Electronic Engineering, Tsinghua University, Beijing, China. His research interests include GNSS security and GNSS software receiver design.

Xiaowei Cui received the BEng and PhD degrees in electronic engineering from Tsinghua University, China in 1999 and 2005, respectively. He is currently an associate professor at Department of Electronic Engineering, Tsinghua University, China. He is a member of the Expert Group of China BeiDou Navigation Satellite System. His research interests include robust GNSS signal processing, multipath mitigation techniques, and high-precision positioning.

Pengxiang Yang received the PhD degree from Northwestern Polytechnical University, China in 2011. He is a researcher at Xi'an Modern Control Technology Research Institute, Xi'an, China; a member of China Mechanical Engineering Society; and an editorial board member of *Journal of Chinese Inertial Technology*. His research interests include GNSS anti-jamming, integrated navigation system, and high-precision positioning.

Mingquan Lu received the MEng and PhD degrees from University of Electronic Science and Technology of China, China in 1993 and 1999, respectively. He is currently a professor at Department of Electronic Engineering, Tsinghua University, China. He is the director of Tsinghua Position, Navigation and Timing Center, and a member of the Expert Group of China BeiDou Navigation Satellite System. His current research interests include GNSS signal design and analysis, GNSS signal processing and receiver development, and GNSS system modeling and simulation.

Peng Wu received the PhD degree from Northwestern Polytechnical University, China in 2017. From October 2012 to March 2014, he was a visiting student at Department of Electrical Engineering, State University of New York at Buffalo, USA. He is now an engineer at Xi'an Modern Control Technology Research Institute, China. His research interests are in signal processing and GNSS positioning.