An ARAIM adaptation for Kalman Filter

An ARAIM Adaptation for Kalman Filter

Tran Trung Hieu, Gustavo Belforte
Polytechnic of Turin
24 Corso Duca degli Abruzzi, Turin, Italy. Email: {hieu.tran; gustavo.belforte}@polito.it

Abstract – **This work propose a new Kalman filter-based method for integrity monitoring, following the solution separation approach of the Advanced Receiver Autonomous Integrity Monitoring (ARAIM) algorithm. This method evaluates the separation of state correction using different subsets of measurement to detect abnormalities as well as potential faulty satellites for exclusion. This approach differs from existing Kalman filter-based methods, which use innovation vector or residual vector for stochastic evaluation.**

Keywords – GNSS, Integrity monitoring, ARAIM, Solution Separation, Kalman Filter

I. INTRODUCTION

Integrity, as defined in [1], is the capability of a GNSS system to provide timely warning to users when the system becomes unreliable and should not be used for navigation purposes. In other words, integrity evaluates the level of trust a user can assign on the information provided by a system [2]. Integrity concerns with the faults within the navigation system, such as satellites faults, incorrect ephemeris parameters, and any fault that could lead to hazardous outcome if the wrong information is used.

Defined since the early days of GPS, the concept of integrity was designed for civil aviation use, specifically in the Communication, Navigation and Surveillance / Air Traffic Management (CNS/ATM) system [3]. Integrity has always been an important aspect of the navigation system, since the validity of navigation data could affect lives of many people. To realize the integrity assessment process, some concepts should be mentioned [2]:

- Protection Level (PL): the statistical bound surrounding the calculated position, ensuring to cover the true position at a certain level of confidence. PL is defined in horizontal and vertical direction, namely Horizontal Protection Level (HPL) and Vertical Protection Level (VPL), respectively. The probability of positioning error exceeding the PL is less than or equal the integrity risk.
- Integrity risk: the probability that actual position error exceeds PL. In literatures, integrity risk is also denoted Probability of Hazardous Misleading Information (PHMI).

There are several approaches for the integrity assessment process. The first approach utilizes the Satellite-based Augmentation System (SBAS), which involves external monitoring systems and additional satellite systems to broadcast the integrity monitoring data to users. SBASes are regional systems, such as the WAAS covering the North America area, or the EGNOS covering the EU and North Africa region. Other SBAS includes MSAS and QZSS in Japan, GAGAN in India. The second approach uses the Ground-based Augmentation System (GBAS) which, as the name suggests, usually consists of a monitoring system and a transmitter on the ground (instead of satellites) to relay information directly to the users. GBASes are often implemented in airfields to assist precision landing of civil aircraft, since they can provide real-time and accurate integrity assessment, especially under bad weather or when SBAS is not available. Despite high performance, both SBAS and GBAS suffer from coverage problem, since each approach can only cover certain areas of services. The third approach is called Aircraft-based Augmentation System (ABAS), which is a set of internal algorithms that utilize solely the navigation data (and optionally, data from external sensors) available on board for integrity assessment. Although having lower performance than SBAS and GBAS, ABAS is still a favorable choice due to its versatility and autonomous nature. The most prominent set of ABAS is the Receiver Autonomous Integrity Monitoring (RAIM) algorithms [4], typically used during phases of flight without stringent requirement.

The integrity assessment process using RAIM algorithms usually consists of two main steps. The first step is called Fault Detection and Exclusion (FDE), in which RAIM performs consistency test to check for faults in the input data, then the algorithm may attempt to exclude the faulty input if any. If the input is deemed consistent, RAIM proceeds to the next step, which is to calculate the PL. In short, the goals of the integrity assessment process are to protect users from excessive positioning errors by detecting and mitigating (if possible) the faults, and alert users in the worst case. There are two main approaches for RAIM methods: residual-based approach [4], which operates in pseudorange domain, and solution separation approach [5], which runs in solution domain. While residual-based RAIM is simple and fast, solution separation RAIM can have better performance due to its customization [6]. RAIM algorithms operate under several assumptions [4]:

- Only one constellation is used (although it's possible to extend the algorithm to multi-constellation),
- There is at most one satellite fault at a time,
- The PL depends mostly on the satellite geometry.

On the other hand, recent works in navigation integrity [7][8][9][10] for aviation have led to the next generation of RAIM algorithm, called Advanced RAIM (ARAIM) [7][11], based on the solution separation approach. ARAIM improves the traditional RAIM in various ways, such as multi-constellation capability (traditional RAIM was developed solely for GPS), generalized satellite fault hypothesis instead of single-fault hypothesis. ARAIM aims at providing better availability, lower PL [12][13][14], making it suitable for more stringent phases of flight which are usually protected by SBAS and GBAS [11].

While the RAIM (and ARAIM) algorithms mentioned above are developed for least-square (LS) positioning method, there are also RAIM methods designed when using Kalman Filter (KF) approach. Most methods follow residual-based approach, using the innovation vector [15] or the ranging residual vector [16][17] for the consistency test, as well as corresponding methods to calculate the PL. It's worth mentioning that, the traditional RAIM for LS is a snapshot method and the LS itself is memoryless, consequently the whole LS-based integrity assessment process using traditional RAIM is memoryless. In contrast, the authors of [16] pointed out the sequential nature of KF and consequently took this into account when proposing the batch-processing version of RAIM for KF. As many other works, these works were developed in aviation context, utilizing the well-founded standards and models for civil aviation [19].

Taking into account the previously presented literature, this paper proposes a solution separation RAIM method for standalone KF as an adaptation of the ARAIM method in [11] for KF. The proposed method combines the advantages of the underlying techniques: the smoothness and high accuracy of KF, the good integrity performance and flexibility of solution separation ARAIM approach. On the other hand, the solution separation approach also performs well against sudden changes in the measurement. The paper is organized as follows. Section II describes the KF model used for navigation and subsequent integrity assessment. Section III presents in details the derivation of the PL equations for KF-based ARAIM. From this, Section IV explains the proposed ARAIM method. Section V concludes the paper.

II. KALMAN FILTER MODEL

Kalman filter is widely applied in many engineering fields, it was developed in the 1960s to recursively estimate the state vector of dynamic systems. In GNSS receivers, KF is utilized in different ways, to integrate inertial sensors with the GNSS receiver at different levels [20], or it is embedded in the tracking part of the receivers with vector tracking loop architecture [21]. Commonly, the KF uses a state vector containing the receiver position coordinates and clock bias along with corresponding derivatives [20][22]. In this paper, the receiver is designed to work with GPS L1 and Galileo E1b signal, therefore the state vector consists of 9 states as follows:

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{u}_k \\ -c\Delta t_{GPS,k} \\ -c\Delta t_{Gal,k} \\ \dot{\boldsymbol{u}}_k \\ -c\dot{\Delta t} \end{bmatrix} \tag{1}$$

where $\boldsymbol{u}_k \in \mathbb{R}^3$ is the position coordinate vector; $\Delta t_{GPS,k}$ and $\Delta t_{Gal,k}$ are the difference between the receiver's clock and system time of GPS and Galileo, respectively; c is the speed of light; $\dot{\Delta t}$ is the clock drift.

With N_{sat} being the total number of satellites, the notation used for the filter description at epoch t_k is:

- $\boldsymbol{x}_k \in \mathbb{R}^9$ is the process state vector.
- $\boldsymbol{\Phi}_k \in \mathbb{R}^{9\times 9}$ is the state transition matrix relating $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k+1}$.
- $\boldsymbol{w}_k \in \mathbb{R}^9$ is the process noise vector, assumed to be a white sequence with known covariance matrix $\mathbf{Q}_k$.
- $\boldsymbol{z}_k \in \mathbb{R}^{2N_{sat}}$ is the measurement vector containing the pseudorange and Doppler measurement for each satellite.
- $\boldsymbol{H}_k \in \mathbb{R}^{2N_{sat}\times 9}$ is the matrix giving the linear connection between the measurement and the state vector.
- $\boldsymbol{v}_k \in \mathbb{R}^{2N_{sat}}$ is the measurement error vector, which is assumed white with known covariance $\boldsymbol{R}_k$ and zero crosscorrelation with $\mathbf{w}_k$.
- $\boldsymbol{R}_k$ is the covariance matrix of $\boldsymbol{v}_k$, it is assumed to be diagonal.

The linear connection matrix $\mathbf{H}_k$ is given by [2]:

$$\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{h}_{u,GPS} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{h}_{u,Gal} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{h}_{u,GPS} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{h}_{u,Gal} & \mathbf{1} \end{bmatrix} \tag{2}$$

where each row of $\boldsymbol{h}_{u,GPS} \in \mathbb{R}^{N_{sat}\times 3}$ is the unit vector pointing from the linearization point to the location of each satellite [22], in which $\boldsymbol{h}_{u,GPS}$ and $\boldsymbol{h}_{u,Gal}$ contain the vectors corresponding to satellites of GPS and Galileo constellation, respectively.

In this work, the alternative form of Kalman filter is used [22]. The state covariance matrix $\boldsymbol{P}_k$ is calculated from the apriori covariance matrix $\boldsymbol{P}_k^-$ as:

$$\boldsymbol{P}_k^{-1} = (\boldsymbol{P}_k^-)^{-1} + \boldsymbol{H}_k^T \boldsymbol{R}_k^- \boldsymbol{H}_k \tag{3}$$

Then the result of (3) is inverted to obtain $\boldsymbol{P}_k$. The apriori covariance matrix $\boldsymbol{P}_k^-$ is the predicted error covariance of the state estimate $\hat{\boldsymbol{x}}_k$ of $\boldsymbol{x}_k$, can be propagated from the previous epoch:

$$\boldsymbol{P}_k^- = \boldsymbol{\Phi}_{k-1} \boldsymbol{P}_{k-1} \boldsymbol{\Phi}_{k-1}^T + \boldsymbol{Q}_{k-1} \tag{4}$$

$\boldsymbol{P}_k$ can be initialized by either assuming to be diagonal with large value [25], or estimated using Least-square results of several initializing steps.

The Kalman Gain $\boldsymbol{K}_k$ is given by:

$$\boldsymbol{K}_k = \boldsymbol{P}_k \boldsymbol{H}_k^T \boldsymbol{R}_k^{-1} \tag{5}$$

The state estimate $\hat{\boldsymbol{x}}_k$ can be calculated from the predicted state vector $\hat{\boldsymbol{x}}_k^-$ as:

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_{k-1}^-) \tag{6}$$

$\hat{\boldsymbol{x}}_k^-$ is propagated from the previous epoch by $\hat{\boldsymbol{x}}_k^- = \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k-1}$.

III. PROTECTION LEVEL EQUATION DERIVATION

As introduced earlier, the Protection Level is an overbound

The derivation for the Protection Level (PL) equation follows a similar approach to one described in [11] and [27], considering the KF model. In [27], the integrity risk is defined to cover both fault detection and fault exclusion.

For fault detection, the event of integrity risk occurrence can be defined when there is a large error (larger than the PL) but no fault was detected. For each fault mode q, the integrity risk for the s-coordinate is defined as a joint probability:

$$P\left(\left|\boldsymbol{e}_{k,s}\right| > PL, \left|\Delta \boldsymbol{x}_{k,s}^{(q)}\right| < T_{k,s}^{(q)}\right) \tag{7}$$

where $\boldsymbol{e}_{k,s} = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^{(0)}$ is the true error of $\hat{x}_k^{(0)}$.

For simplicity, the derivation in this section will omit the notation of k, since all the Kalman filterrelated vectors and matrices are all correspond to discrete time t_k. That said, (7) can be rewritten as:

$$P\left(|\boldsymbol{e}_s| > PL, \left|\Delta \boldsymbol{x}_s^{(q)}\right| < T_s^{(q)}\right) \tag{8}$$

Defining $\boldsymbol{e}^- = \boldsymbol{x} - \widehat{\boldsymbol{x}}^-$ as the error of the apriori state estimation $\widehat{\boldsymbol{x}}^-$, the difference between the updated and true state vector can be expressed as:

$$\begin{aligned}
\widehat{\boldsymbol{x}}^{(q)} - \boldsymbol{x} &= \widehat{\boldsymbol{x}}^- - \boldsymbol{x} + \boldsymbol{K}^{(q)}(\boldsymbol{z} - \boldsymbol{H}\widehat{\boldsymbol{x}}^-) \\
&= \widehat{\boldsymbol{x}}^- - \boldsymbol{x} + \boldsymbol{K}^{(q)}(\boldsymbol{H}\boldsymbol{x} + \boldsymbol{v} - \boldsymbol{H}\widehat{\boldsymbol{x}}^-) \\
&= \left(\boldsymbol{K}^{(q)}\boldsymbol{H} - \boldsymbol{I}\right)\boldsymbol{e}^- + \boldsymbol{K}^{(q)}\boldsymbol{v} \\
&= \boldsymbol{S}^{(q)}\boldsymbol{e}^- + \boldsymbol{K}^{(q)}\boldsymbol{v}
\end{aligned} \tag{9}$$

where

$$\begin{aligned}
\boldsymbol{S}^{(q)} &= \boldsymbol{K}^{(q)}\boldsymbol{H} - \boldsymbol{I} \\
\boldsymbol{S}^{(0)} &= \boldsymbol{K}\boldsymbol{H} - \boldsymbol{I}
\end{aligned} \tag{10}$$

From (7) we have:

$$\begin{aligned}
&P\left(\left|\boldsymbol{e}_{k,s}\right| > PL, \left|\Delta \boldsymbol{x}_{k,s}^{(q)}\right| < \boldsymbol{T}_{k,s}^{(q)}\right) \\
&= P\left(\left|\widehat{\boldsymbol{x}}_s^{(0)} - \boldsymbol{x}_s\right| > PL, \left|\widehat{\boldsymbol{x}}_s^{(q)} - \widehat{\boldsymbol{x}}_s^{(0)}\right| < T_s^{(q)}\right) \\
&= P\begin{pmatrix} \left|\widehat{\boldsymbol{x}}_s^{(0)} - \widehat{\boldsymbol{x}}_s^{(q)} + \widehat{\boldsymbol{x}}_s^{(q)} - \boldsymbol{x}_s\right| > PL, \\ \left|\widehat{\boldsymbol{x}}_s^{(q)} - \widehat{\boldsymbol{x}}_s^{(0)}\right| < T_s^{(q)} \end{pmatrix} \\
&\le P\begin{pmatrix} \left|\widehat{\boldsymbol{x}}_s^{(0)} - \widehat{\boldsymbol{x}}_s^{(q)}\right| + \left|\widehat{\boldsymbol{x}}_s^{(q)} - \boldsymbol{x}_s\right| > PL, \\ \left|\widehat{\boldsymbol{x}}_s^{(q)} - \widehat{\boldsymbol{x}}_s^{(0)}\right| < T_s^{(q)} \end{pmatrix} \\
&\le P\begin{pmatrix} \left|\widehat{\boldsymbol{x}}_s^{(q)} - \boldsymbol{x}_s\right| > PL - T_s^{(q)}, \\ \left|\widehat{\boldsymbol{x}}_s^{(q)} - \widehat{\boldsymbol{x}}_s^{(0)}\right| < T_s^{(q)} \end{pmatrix} \\
&\le P\left(\left|\widehat{\boldsymbol{x}}_s^{(q)} - \boldsymbol{x}_s\right| > PL - T_s^{(q)}\right) \\
&= Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right)
\end{aligned} \tag{11}$$

where Q(·) is the complement of the Gaussian cumulative distribution function, with zero mean and unit variance, and $\sigma_s^{(q)}$ is defined as:

$$\begin{aligned}
\sigma_s^{(q)2} &= \left[\boldsymbol{S}^{(q)}\boldsymbol{P}^-\boldsymbol{S}^{(q)T} + \boldsymbol{K}^{(q)}\boldsymbol{R}\boldsymbol{K}^{(q)T}\right]_{s,s} \\
&= \left[\boldsymbol{P}_{alt}^{(q)}\right]_{s,s}
\end{aligned} \tag{12}$$

Let $p_{\mathrm{fault}}^{(q)}$ be the fault probability for fault mode q, the boundary for the integrity risk in case of detection can be written as:

$$\begin{aligned}
P_{HMI,D} &= \sum_{q=0}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right) \\
&= p_{fault}^{(0)} Q\left(\frac{PL}{\sigma_s^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right) \\
&\le \mathrm{Q}\left(\frac{\mathrm{PL}}{\sigma_{\mathrm{s}}^{(0)}}\right) + \sum_{\mathrm{q}=1}^{\mathrm{N_{fault}}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right)
\end{aligned} \tag{13}$$

The first term of (13) represents the event of no detection under fault-free condition, while the second terms represents the missed detection when subset q is faulty.

In case of exclusion, the event of integrity risk occurrence can be defined when there is a large error (larger than the PL), a fault is detected and an exclusion candidate j is actually excluded. Note that for a subset j to be excluded, all fault hypothesis q on the remaining satellite set must pass the solution separation test. For each exclusion candidate j and each fault mode q, the integrity risk for exclusion along the s-coordinate is defined as:

$$P\begin{pmatrix} \left|\boldsymbol{e}_s^{(j)}\right| > PL, \ \left|\Delta \boldsymbol{x}_s^{(j)}\right| > T_s^{(j)}, \\ \left|\Delta \boldsymbol{x}_s^{(j,q)}\right| < T_s^{(j,q)} \end{pmatrix} \tag{14}$$

where $\Delta \boldsymbol{x}_s^{(j,q)} = \boldsymbol{x}_s^{(j,q)} - \boldsymbol{x}_s^{(j)}$ is the solution separation for fault mode q after excluding candidate j, i.e. considering the remaining satellites as all in view; $\boldsymbol{e}_{\mathrm{s}}^{(\mathrm{j})}$ is the positioning error after having removed subset j; $T_s^{(j,q)}$ is the test threshold for the corresponding solution separation. Notice that subset j is detected to be a fault due to $\left|\Delta \boldsymbol{x}_s^{(j)}\right| > T_s^{(j)}$, then is removed because $\left|\Delta \boldsymbol{x}_s^{(j,q)}\right| < T_s^{(j,q)}$. Following similar derivation as for detection case, (14) can be written as:

$$\begin{aligned}
&P\begin{pmatrix} \left|\boldsymbol{e}_s^{(j)}\right| > PL, \left|\Delta \boldsymbol{x}_s^{(j)}\right| > T_s^{(j)}, \\ \left|\Delta \boldsymbol{x}_s^{(j,q)}\right| < T_s^{(j,q)} \end{pmatrix} \\
&\le P\begin{pmatrix} \left|\boldsymbol{e}_s^{(j)}\right| > PL, \\ \left|\Delta \boldsymbol{x}_s^{(j,q)}\right| < T_s^{(j,q)} \end{pmatrix}
\end{aligned} \tag{15}$$

$$= P\left(\begin{matrix}\left|\widehat{\boldsymbol{x}}_s^{(j)} - \boldsymbol{x}_s\right| > PL, \\ \left|\widehat{\boldsymbol{x}}_s^{(j,q)} - \boldsymbol{x}_s^{(j)}\right| < T_s^{(j,q)}\end{matrix}\right)$$
$$\leq P\left(\left|\widehat{\boldsymbol{x}}_s^{(j,q)} - \boldsymbol{x}_s\right| > PL - T_s^{(j,q)}\right)$$
$$= Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right)$$

Considering all exclusion hypothesis j and all fault mode q, the integrity risk in case of exclusion can be expressed as:

$$P_{HMI,E}$$
$$= \sum_{q=0}^{N_{fault}} \sum_{j=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right)$$
$$= p_{fault}^{(0)} \sum_{j=1}^{N_{fault}} Q\left(\frac{PL}{\sigma_s^{(j)}}\right)$$
$$+ \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL}{\sigma_s^{(q)}}\right)$$
$$+ \sum_{q=1}^{N_{fault}} \sum_{\substack{j=1 \\ j\neq q}}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right)$$
$$\leq \sum_{\mathrm{j=1}}^{\mathrm{N_{fault}}} Q\left(\frac{PL}{\sigma_s^{(j)}}\right) + \sum_{\mathrm{q=1}}^{\mathrm{N_{fault}}} p_{fault}^{(q)} Q\left(\frac{PL}{\sigma_s^{(q)}}\right)$$
$$+ \sum_{q=1}^{N_{fault}} \sum_{\substack{j=1 \\ j\neq q}}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right) \tag{16}$$

The first term in the last line of (16) represents wrong exclusion of subset j under fault free condition. The second term represents the case where $j = q, q > 0$ which is the correct exclusion case. Note that with $j = q$, subset (j, q) simplifies to (q) since they are the same subset. The third term represents the wrong exclusion case with $j \neq q$, i.e. excluding subset j while the fault is in fact in subset q.

From (13) and (16), the PL equation for both detection and exclusion can be formally expressed as:

$$P_{HMI} = P_{HMI,D} + P_{HMI,E} \tag{17}$$
$$= Q\left(\frac{PL}{\sigma_s^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right)$$
$$+ \sum_{j=1}^{N_{fault}} Q\left(\frac{PL}{\sigma_s^{(j)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL}{\sigma_s^{(q)}}\right)$$
$$+ \sum_{q=1}^{N_{fault}} \sum_{\substack{j=1 \\ j\neq q}}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right)$$

(17) is the formal equation to evaluate the PL, which takes into account the integrity risks from both detection and exclusion. However, this form should be considered as a predictive form to calculate PL before either the events of detection and exclusion occurs. In practice, only certain terms of the equation should be used in certain cases, to avoid suboptimal allocation of the integrity risk and thus ensure sufficient availability. Therefore, in case of simple detection without exclusion implementation, the practical equation should eliminate the exclusion terms (since there are no exclusion candidate to be considered):

$$P_{HMI,0} = Q\left(\frac{PL}{\sigma_s^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right) \tag{18}$$

In case of fault detection and exclusion of a subset j, the missed detection terms (the first two terms) of (17) should be eliminated since at least a fault was detected on subset j, and the exclusion terms should only consider the excluded subset j:

$$P_{HMI,j}$$
$$= Q\left(\frac{PL}{\sigma_s^{(j)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{PL}{\sigma_s^{(q)}}\right)$$
$$+ \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right) \tag{19}$$

Equation (17) and the other forms of (18), (19) are the generalized equations for PL along the direction of s coordinate. From this, the specific equation for VPL and HPL can be defined for each case accordingly. In case of no detection, the equation for VPL can be written as:

$$P_{HMI,VERT,0} = Q\left(\frac{VPL}{\sigma_3^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{VPL - T_3^{(q)}}{\sigma_3^{(q)}}\right) \quad (20)$$

The equations for HPL can be defined as:

$$\frac{1}{2} P_{HMI,HOR,0} = Q\left(\frac{HPL_1}{\sigma_1^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{HPL_1 - T_1^{(q)}}{\sigma_1^{(q)}}\right)$$

$$\frac{1}{2} P_{HMI,HOR,0} = Q\left(\frac{HPL_2}{\sigma_2^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{HPL_2 - T_2^{(q)}}{\sigma_2^{(q)}}\right)$$

$$HPL = \sqrt{HPL_1^2 + HPL_2^2} \quad (21)$$

In case of exclusion, the equation for VPL can be defined as:

$$P_{HMI,VERT,j} = Q\left(\frac{VPL}{\sigma_3^{(j)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{VPL}{\sigma_3^{(q)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{VPL - T_3^{(j,q)}}{\sigma_3^{(j,q)}}\right) \quad (22)$$

Similarly, the equations for HPL can be defined as:

$$\frac{1}{2} P_{HMI,HOR,j} = Q\left(\frac{HPL_1}{\sigma_1^{(j)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{HPL_1}{\sigma_1^{(q)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{HPL_1 - T_1^{(j,q)}}{\sigma_1^{(j,q)}}\right) \quad (23)$$

$$\frac{1}{2} P_{HMI,HOR,j} = Q\left(\frac{HPL_2}{\sigma_2^{(j)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{HPL_2}{\sigma_2^{(q)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{HPL_2 - T_2^{(j,q)}}{\sigma_2^{(j,q)}}\right)$$

$$HPL = \sqrt{HPL_1^2 + HPL_2^2}$$

IV. ARAIM FOR KALMAN FILTER

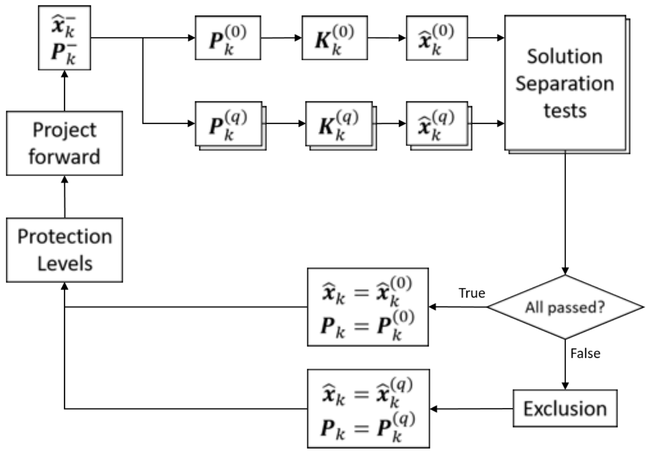

Figure 1. Algorithm Scheme

A. Concept

The overall scheme of the Kalman filter-based ARAIM algorithm is proposed on Figure 1. Instead of predicting forward using the state transition matrix, the algorithm calculates the *fault-tolerant* Kalman gains $\boldsymbol{K}_k^{(q)}$, state corrections $\hat{\boldsymbol{x}}_k^{(q)}$ and the error covariance matrices $\boldsymbol{P}_k^{(q)}$, using subsets of satellites. Here, q denotes a case corresponding to a subset of satellites, called *fault mode*, or *fault hypothesis*. The initial all-in-view state vector $\hat{\boldsymbol{x}}_k^{(0)}$ and fault-tolerant state vectors $\hat{\boldsymbol{x}}_k^{(q)}$ are used as inputs for the Solution Separation (SS) tests, which measure the consistency of the measurement based on the deviation between the fault-tolerant and the all-in-view states. If any of the tests fails, exclusion will be attempted on potential faulty satellites. Otherwise, the algorithm will proceed to calculate the Protection Levels, before projecting the state and error covariance matrix ahead. The whole process is then repeated for the next epoch.

B. Fault tolerant derivation

The SS approach of the algorithm involves running the Kalman filter repeatedly on various subsets of satellites, or fault mode. The subset of satellites of fault mode q is denoted $S^{(q)}$, while $S^{(q)}$ denotes the set containing all the satellites in view. For each fault mode q, a diagonal matrix $\boldsymbol{W}^{(q)}$ is defined as:

$$\boldsymbol{W}^{(q)}(i,i) = \boldsymbol{W}^{(q)}(i+N_{sat}, i+N_{sat}) = \begin{cases} 1 & \text{if } i \in S^{(q)} \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

With this, the fault tolerant observation matrix $\boldsymbol{H}_k^{(q)}$, measurement vector $\boldsymbol{z}_k^{(q)}$, and measurement error covariance $\boldsymbol{R}_k^{(q)}$ are given by:

$$\begin{aligned} \boldsymbol{H}_k^{(q)} &= \boldsymbol{W}^{(q)}\boldsymbol{H}_k \\ \boldsymbol{z}_k^{(q)} &= \boldsymbol{W}^{(q)}\boldsymbol{z}_k \\ \boldsymbol{R}_k^{(q)} &= \boldsymbol{W}^{(q)}\boldsymbol{R}_k\boldsymbol{W}^{(q)} \\ &= \boldsymbol{W}^{(q)}\boldsymbol{W}^{(q)}\boldsymbol{R}_k = \boldsymbol{W}^{(q)}\boldsymbol{R}_k \end{aligned} \quad (25)$$

Noting that, both $\boldsymbol{R}_k$ and $\boldsymbol{W}^{(q)}$ are diagonal matrices, therefore their multiplication are commutative. Let $\boldsymbol{R}_k^{(q)+}$ be the Moore-Penrose pseudoinverse of $\boldsymbol{R}_k^{(q)}$. In fact, since the pseudoinverse of $\boldsymbol{W}^{(q)}$ is itself and $\boldsymbol{R}_k^{-1}$ is also diagonal, $\boldsymbol{R}_k^{(q)+}$ is given by:

$$\boldsymbol{R}_k^{(q)+} = \boldsymbol{W}^{(q)}\boldsymbol{R}_k^{-1} \quad (26)$$

Replacing (25) and (26) into (3), the fault tolerant state covariance matrix $\boldsymbol{P}_k^{(q)}$ can be obtained from:

$$\begin{aligned} \left(\boldsymbol{P}_k^{(q)}\right)^{-1} &= (\boldsymbol{P}_k^{-})^{-1} + \boldsymbol{H}_k^{(q)T}\,{\boldsymbol{R}_k^{(q)}}^{-1}\,\boldsymbol{H}_k^{(q)} \\ &= (\boldsymbol{P}_k^{-})^{-1} + \boldsymbol{H}_k^{T}\boldsymbol{W}^{(q)}\boldsymbol{R}_k^{-1}\,\boldsymbol{H}_k^{(q)} \end{aligned} \quad (27)$$

Similarly, replacing (25) and (26) into (5), the fault tolerant Kalman Gain $\boldsymbol{K}_k^{(q)}$ can be written as:

$$\boldsymbol{K}_k^{(q)} = \boldsymbol{P}_k^{(q)}\,\boldsymbol{H}_k^{(q)T}\boldsymbol{W}^{(q)}\boldsymbol{R}_k^{-1} \quad (28)$$

The fault tolerant state update $\hat{\boldsymbol{x}}_k^{(q)}$ can be expressed as:

$$\begin{aligned} \hat{\boldsymbol{x}}_k^{(q)} &= \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{K}_k^{(q)}\left(\boldsymbol{z}_k^{(q)} - \boldsymbol{H}_k^{(q)}\hat{\boldsymbol{x}}_k^{-}\right) \\ &= \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{K}_k^{(q)}(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^{-}) \end{aligned} \quad (29)$$

C. Proposed ARAIM algorithm

With the analysis and derivation from previous sections, the detailed algorithm is presented here step by step. The proposed algorithm uses the apriori estimate $\hat{\boldsymbol{x}}_{\mathrm{k}}^{-}$ and the apriori error covariance $\boldsymbol{P}_k^{-}$ as inputs. Their initial estimates can be obtained processing several initialization epochs using least-square positioning method. Once initialized, the algorithm consists of six steps, detailed as follows.

Step 1. All-in-view state update

In this step, the state vector and error covariance are updated following the aforementioned Kalman filter loop, using all available measurements. Note that the observation matrix $\boldsymbol{H}_k$ has to be defined in ENU coordinate, using the apriori estimate $\hat{\boldsymbol{x}}_k^{-}$ as the origin.

The output state vector and error covariance matrix are denoted $\hat{\boldsymbol{x}}_k^{(0)}$ and $\boldsymbol{P}_k^{(0)}$, respectively.

Step 2. Number of fault modes

This step follows the formula of conventional ARAIM, presented in [11]. Output of this step is the maximum number of concurrent satellite faults $N_{sat,max}$ and the total number of fault modes N_{fault}. In other word, N_{fault} represents the total number of possible satellite combinations by excluding from 1 to $N_{sat,max}$ satellites:

$$N_{fault} = \sum_{g=1}^{N_{sat,max}} \binom{N_{sat}}{N_{sat} - g} \quad (30)$$

Step 3. Fault tolerant state update

Fault tolerant state is the state obtained by doing the correction step of KF using a subset of measurements, excluding potential faulty satellites. For each q from 1 to N_{fault}, the algorithm determines:

- The fault tolerant state $\hat{\boldsymbol{x}}_k^{(q)}$,
- The difference $\Delta\hat{\boldsymbol{x}}_k^{(q)}$ between $\hat{\boldsymbol{x}}_k^{(q)}$ and $\hat{\boldsymbol{x}}_k^{(0)}$
- The covariance of $\Delta\hat{\boldsymbol{x}}_k^{(q)}$, denoted $P_{\Delta\hat{x}_k}^{(q)}$

The posteriori state covariance $P_k^{(q)}$ is obtained from (27), rewritten here for completeness:

$$\left(\boldsymbol{P}_k^{(q)}\right)^{-1} = \left(\boldsymbol{P}_k^{-}\right)^{-1} + \boldsymbol{H}_k^T \boldsymbol{W}^{(q)} \boldsymbol{R}_k^{-1} \boldsymbol{H}_k^T \tag{31}$$

The fault tolerant Kalman gain is computed as (28):

$$\boldsymbol{K}_k^{(q)} = \boldsymbol{P}_k^{(q)} \boldsymbol{H}_k^T \boldsymbol{W}^{(q)} \boldsymbol{R}_k^{-1} \tag{32}$$

The fault tolerant state estimation update can be obtained as (29):

$$\hat{\boldsymbol{x}}_k^{(q)} = \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{K}_k^{(q)}\left(\boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^{-}\right) \tag{33}$$

The difference between fault tolerant and all in view state updates is thus:

$$\begin{aligned}\Delta\hat{\boldsymbol{x}}_k^{(q)} &= \hat{\boldsymbol{x}}_k^{(q)} - \hat{\boldsymbol{x}}_k \\ &= \left(\boldsymbol{K}_k^{(q)} - \boldsymbol{K}_k\right)\left(\boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^{-}\right)\end{aligned} \tag{34}$$

The covariance matrix $\boldsymbol{P}_{\Delta\hat{x}_k}^{(q)}$ of $\Delta\hat{\boldsymbol{x}}_k^{(q)}$ is calculated as:

$$\begin{aligned}\boldsymbol{P}_{\Delta\hat{x}_k}^{(q)} &= \boldsymbol{P}_k^{(0)} + \boldsymbol{P}_k^{(q)} \\ &-2\left((\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^{-}\left(\boldsymbol{I} - \boldsymbol{K}_k^{(p)}\boldsymbol{H}_k\right)^T \right. \\ &\qquad \left. + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^{(q)T}\right)\end{aligned} \tag{35}$$

The alternative state covariance matrix $\boldsymbol{P}_{k,alt}^{(q)}$, which will be used to calculate the Protection Level, is given by:

$$\begin{aligned}\boldsymbol{P}_{k,alt}^{(q)} = &\left(\boldsymbol{I} - \boldsymbol{K}_k^{(p)}\boldsymbol{H}_k\right)\boldsymbol{P}_k^{-}\left(\boldsymbol{I} - \boldsymbol{K}_k^{(p)}\boldsymbol{H}_k\right)^T \\ &+ \boldsymbol{K}_k^{(q)}\boldsymbol{R}_k\boldsymbol{K}_k^{(q)T}\end{aligned} \tag{36}$$

Step 4. Solution separation threshold test

Let $s = 1,2,3$ corresponds to east, north, up component, respectively. For each fault mode q, the solution separation (SS) test is executed on all 3 components of coordinates: north, east and up. For each component, the test threshold is calculated as:

$$T_{k,s}^{(q)} = K_{fa,s}\,\sigma_{\Delta x,s} \tag{37}$$

where:

$$\sigma_{\Delta x,s} = \left(\sqrt{P_{\Delta\hat{x}_k}^{(q)}}\right)_{s,s} \tag{38}$$

The coefficients $K_{fa,1}$ follow the same formula as for ARAIM [11]:

$$\begin{aligned}K_{fa,1} = K_{fa,2} &= Q^{-1}\left(\frac{P_{FA\_HOR}}{4N_{fault}}\right) \\ K_{fa,3} &= Q^{-1}\left(\frac{P_{FA\_VERT}}{2N_{fault}}\right)\end{aligned} \tag{39}$$

Where $Q^{-1}(p)$ is the $(1-p)$-quantile of a Normal distribution; P_{FA_HOR} and P_{FA_VERT} are the probability of false alarm allocated along the horizontal and vertical direction, respectively, and are usually chosen based on the intended application.

The SS test is considered passed if for all q and s we have:

$$\tau_{k,s}^{(q)} = \frac{\left|\hat{\boldsymbol{x}}_{k,s}^{(q)}\right|}{T_{k,s}^{(q)}} \leq 1 \tag{40}$$

If any of the test fail, the fault mode with failed test becomes a candidate for exclusion.

Step 5. Exclusion

The failed fault modes are sorted in decreasing order of $\tau_{k,s}^{(j)}$, with j denotes any failed fault mode, or exclusion candidate. For each j, the satellites set $S^{(j)}$ are treated as all-in-view set, then the algorithm (from Step 1 to Step 4) is executed again, using all fault-tolerant solution $\hat{\boldsymbol{x}}_k^{(j,q)}$ within the candidate set $S^{(j)}$. If all the SS test are passed, i.e. $\tau_{k,s}^{(j,q)} < 1\ \forall q \neq j$, $S^{(j)}$is considered a consistent satellite set, the remaining satellites (those in $S^{(0)} - S^{(j)}$) will be excluded.

When a good satellite set is found (with all SS tests passed), the algorithm will proceed to PL calculation.

Step 6. Protection Level

The PL equations used here are derived earlier in Section III. Let:

$$\sigma_s^{(q)} = \sqrt{\left[\boldsymbol{P}_{k,alt}^{(q)}\right]_{s,s}} \forall q = 0..N_{fault} \tag{41}$$

Let $P_{HMI,VERT}$ and $P_{HMI,HOR}$ denote the integrity risk allocated to the vertical and horizontal, respectively. Similar to P_{FA_HOR} and P_{FA_VERT}, the allocated integrity risks are chosen based on the intended application. In case of no exclusion, the VPL is the solution to the equation:

$$Q\left(\frac{VPL}{\sigma_3^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{VPL - T_{k,3}^{(q)}}{\sigma_3^{(q)}}\right) = P_{HMI,VERT} \tag{42}$$

Similarly, the HPL is calculated on 2 directions of the horizontal plane, namely HPL_1 and HPL_2, as follows (with $s = 1,2$):

$$Q\left(\frac{HPL}{\sigma_3^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{HPL - T_{k,s}^{(q)}}{\sigma_s^{(q)}}\right) = P_{HMI,HOR} \tag{43}$$

The final HPL is then computed as:

$$HPL = \sqrt{HPL_1^2 + HPL_2^2} \tag{44}$$

On the other hand, if exclusion was attempted and subset candidate j was excluded from the satellite set, then the PL equations are defined to account for the event. In this case, the VPL is the solution to the equation:

$$P_{HMI,VERT} = 2Q\left(\frac{VPL}{\sigma_3^{(j)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{VPL}{\sigma_3^{(q)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{VPL - T_{k,3}^{(j,q)}}{\sigma_3^{(j,q)}}\right) \tag{45}$$

where $N_{fault,j}$ is the number of fault in the remaining set of satellites after excluding exclusion candidate j.

The HPL is calculated in similar manner as in the no detection case:

$$\frac{1}{2}P_{HMI,HOR} = 2Q\left(\frac{HPL_1}{\sigma_1^{(j)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(H\frac{HPL_1}{\sigma_1^{(q)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{HPL_1 - T_1^{(j,q)}}{\sigma_1^{(j,q)}}\right)$$

$$\frac{1}{2}P_{HMI,HOR} = 2Q\left(\frac{HPL_2}{\sigma_2^{(j)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(H\frac{HPL_2}{\sigma_2^{(q)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{HPL_2 - T_1^{(j,q)}}{\sigma_2^{(j,q)}}\right)$$

$$HPL = \sqrt{HPL_1^2 + HPL_2^2} \tag{46}$$

V. DISCUSSION

The proposed algorithm has many points that can be further exploited to suit different applications. The first possible derivation is the allocation of integrity risk budget. For example, the conventional ARAIM algorithm [11] (LS-based) was developed for aviation, hence the integrity risk is allocated with higher priority for vertical error. On the other hand, applications in urban environment can put higher budget in horizontal error, or start with a different total integrity risk budget.

The noise model also allows potential modification. In RAIM algorithm in general, the noise model acts as a (hypothetical) overbound of the measurement error, which then contributes to the consistency tests and the PL calculation step. While different operating environment may have cause measurement error with different nature, [30] has shown that incorrect assumption of the measurement error may lead to unnecessary loss of availability. Therefore it is possible to modify the underlying noise model when adapting the algorithm for different scenarios.

VI. CONCLUSION

This paper proposed an adaptation of the conventional ARAIM algorithm for KF. A detailed derivation of the PL equations was presented, taking into account the KF model, covering both detection and fault exclusion scenario. The proposed algorithm can serve as a generic model, a starting point for modification based on application requirements.

REFERENCES

[1] DoD, U.; DHS, U.; DoT, U. Federal Radio navigation Plan. Technical report, DOT-TSC-RSPA-84.8, 2012.

[2] Kaplan, E.D.; Hegarty, C.J. Understanding GPS: principles and applications; Artech House: Norwood, MA, USA, 2005.

[3] Andrés, S.; Daniel, C. Integrity monitoring applied to the reception of GNSS signals in urban environments. PhD thesis, 2012.

[4] B. W. Parkinson and J. J. Spilker. Chapter 5: Receiver Autonomous Integrity Monitoring Global Positioning System: Theory and Applications; American Institute of Aeronautics and Astronautics, USA, 1996.

[5] BROWN, R.G.; McBURNEY, P.W. Self-Contained GPS Integrity Check Using Maximum Solution Separation. NAVIGATION 1988, 35, 41–53. [CrossRef].

[6] Joerger, M.; Chan, F.C.; Pervan, B. Solution Separation Versus Residual-Based RAIM. Navigation 2014, 61, 273–291. NAVI-2014-027.

[7] Working Group C - ARAIM Technical Subgroup. Interim Report 2012.

[8] FAA GEAS Panel. Phase II of the GNSS evolutionary architecture study. Report. GNSS Evolutionary Architecture Study Panel 2010.

[9] Blanch, J.; Walter, T.; Enge, P.; Wallner, S.; Amarillo Fernandez, F.; Dellago, R.; Ioannides, R.; Fernandez Hernandez, I.; Belabbas, B.; Spletter, A.; Rippl, M. Critical Elements for a Multi-Constellation Advanced RAIM. Navigation 2013, 60, 53–69.

[10] Blanch, J.; Ene, A.; Walter, T.; Enge, P. An Optimized Multiple Hypothesis RAIM Algorithm for Vertical Guidance. Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007), 2007.

[11] Blanch, J.;Walter, T.; Enge, P.; Lee, Y.; Pervan, B.; Spletter, A. Advanced RAIM user Algorithm Description: Integrity Support Message. Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012), 2012, pp. 2828–2849.

[12] Tran, H.T.; Presti, L.L. Demonstration of Multi-GNSS Advanced RAIM Algorithm Using GPS and Galileo Signals. ICSANE 2013 (International Conference on Space, Aeronautical and Navigational Electronics 2013, 113, 191–196.

[13] El-Mowafy, A.; Arora, B.S. The Current ARAIM Availability According to LPV-200 Using GPS and BeiDou in Western Australia. IGNSS2013 Symposium - The International Global Navigation Systems Society (IGNSS), 2013, pp. 1–16.

[14] Mei, H.; Zhan, X.; Zhang, X. GNSS vulnerability assessment method based on ARAIM user algorithm. 2017 Forum on Cooperative Positioning and Service (CPGPS, 2017, pp. 111–115.

[15] Wieser, A.; Petovello, M.G.; Lachapelle, G. Failure scenarios to be considered with kinematic high precision relative GNSS positioning. Proceedings ION GNSS. Citeseer, 2004, p. 6.

[16] Joerger, M.; Pervan, B. Kalman filter-based Integrity Monitoring Against Sensor Faults. Journal of Guidance, Control, and Dynamics 2013, 36, 349–361.

[17] Bhattacharyya, S.; Gebre-Egziabher, D. Kalman filter based RAIM for GNSS receivers. IEEE Transactions on Aerospace and Electronic Systems 2015, 51, 2444–2459.

[18] YOUNG, R.S.Y.; MCGRAW, G.A. Fault Detection and Exclusion Using Normalized Solution Separation and Residual Monitoring Methods. Navigation 2003, 50, 151–169.

[19] ICAO. Annex 10, Amendment 84. Aeronautical Telecommunications, 1 (Radio Navigation Aids).

[20] Falletti, E.; Rao, M.; Savasta, S., The Kalman Filter and Its Applications in GNSS and INS. In Handbook of

Position Location; John Wiley and Sons, Inc., 2011; pp. 709–751.
[21] Tang, X.; Falco, G.; Falletti, E.; Presti, L.L. Practical implementation and performance assessment of an Extended Kalman Filter-based signal tracking loop. 2013 International Conference on Localization and GNSS (ICL-GNSS), 2013, pp. 1–6.
[22] Brown, R.G.; Hwang, P.Y. Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, third ed.; Vol. 1, John Wiley and Sons, 1997.
[23] Kuusniemi, H. User-Level Reliability and Quality Monitoring in Satellite-Based Personal Navigation; Tampere University of Technology, 2005.
[24] WAAS Minimum Operational Performance Specification (MOPS). RTCA document DO-229D.
[25] Linderoth, M.; Soltesz, K.; Robertsson, A.; Johansson, R. Initialization of the Kalman filter without assumptions on the initial state. 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 4992–4997.
[26] Falco, G.; Nicola, M.; Falletti, E. Constellation-aware method for computing the covariance matrix of GNSS measurements. 2016 European Navigation Conference (ENC), 2016, pp. 1–8.
[27] Joerger, M.; Pervan, B. Fault detection and exclusion using solution separation and chi-squared ARAIM. IEEE Transactions on Aerospace and Electronic Systems 2016, 52, 726–742.
[28] Pervan, B.S.; Pullen, S.P.; Christie, J.R. A Multiple Hypothesis Approach to Satellite Navigation Integrity. Navigation 1998, 45, 61–71.
[29] Blanch, J.; Walter, T.; Enge, P. RAIM with Optimal Integrity and Continuity Allocations Under Multiple Failures. IEEE Transactions on Aerospace and Electronic Systems 2010, 46, 1235–1247.
[30] Trung, H.T.; Hai, T.T.; Presti, L.L. Adaptation of ARAIM Algorithm for Urban Environment Applications. Proceeding of International Symposium on GNSS 2015, 2015.

AUTHOR' BIOGRAPHY

Tran Trung Hieu is a PhD student at the Polytechnic of Turin, Italy. He received his Master degree (2013) from Polytechnic of Turin, Italy. He is also a researcher at NAVIS Centre, Hanoi University of Science and Technology. His researches focus on next generation algorithms for GNSS integrity monitoring.