Bayesian Integrity Monitoring for Cellular Positioning - A Simplified Case Study

Bayesian Integrity Monitoring for Cellular Positioning — A Simplified Case Study

Liqin Ding*, Gonzalo Seco-Granados†, Hyowon Kim*,
Russ Whiton‡, Erik G. Ström*, Jonas Sjöberg*, Henk Wymeersch*
*Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden
†Department of Telecommunications and Systems Engineering, Universitat Autonoma de Barcelona, Barcelona, Spain
‡Volvo Car Corporation, Gothenburg, Sweden

***Abstract*—Bayesian receiver autonomous integrity monitoring (RAIM) algorithms are developed for the snapshot cellular positioning problem in a simplified one-dimensional (1D) linear Gaussian setting. They allow for position estimation, multi-fault detection and exclusion, and protection level (PL) computation by the efficient and exact computation of the position posterior probabilities via message passing along a factor graph. Numerical simulations show that the proposed Bayesian RAIM algorithms achieve significant performance improvement over a baseline advanced RAIM algorithm by providing tighter protection levels (PLs) that meet the target integrity risk (TIR) requirements.**

***Index Terms*—Cellular Positioning, Positioning Integrity, Bayesian Inference, Factor Graph**

I. Introduction

With widespread deployment, large bandwidth, and massive antenna arrays, 6G cellular networks promise to provide reliable positioning services for vertical industries with stringent performance requirements, such as factory automation and autonomous driving [1], [2]. In addition to traditional requirements such as high accuracy and low latency, safety-critical applications place new demands on *positioning integrity*, i.e., the level of confidence in the correctness of the positioning results [3]–[6]. The rigorous quantification of integrity is typically done through the formulation of an upper bound of instantaneous position error, termed protection level (PL), to meet the required confidence level, often given in the form of one minus the so-called target integrity risk (TIR). PLs can be computed for each dimension of user position, or jointly for two or three dimensions. Together with the position estimates, they describe a three-dimensional (3D) geometry that contains the true user position with the desired confidence (probability). An application can then decide whether the needed safety margins are provided. Therefore, it is expected that the computed PLs are tight enough so that their availability in safety-critical applications can be maximized [7].

In global navigation satellite systems (GNSSs), integrity assurance mechanisms and receiver autonomous integrity monitoring (RAIM) algorithms have been employed for a long time, primarily for aviation needs [8], [9]. Integrity support for GNSS assistance has been incorporated into the latest 3GPP Release-17 standard [10], whereas radio standalone positioning with integrity guarantee is expected from Release-18 onward [11]. The main task of RAIM is to detect and exclude faulty measurements, which can arise from many different causes [6] and often deviate from the expected values under nominal conditions, in order to avoid large errors in position estimates [12]. RAIM methods can be grouped into *traditional RAIM algorithms* and *Bayesian RAIM methods*.

Traditional RAIM algorithms detect and exclude faulty measurements by performing rounds of consistency checks on the statistics (e.g., range residuals and position estimates) associated with the fault patterns (i.e., the alternative hypotheses in the language of statistical hypothesis testing). This frequentist approach relies on redundant measurements and does not directly lead to instantaneous position error probability distributions. To avoid underestimating tail risk, the formulation of solvable PL equations has to conservatively overbound the error distribution. Consequently, the computed PLs tend to be loose.

In contrast, Bayesian RAIM methods [13]–[16] aim to find the posterior probability distribution of position error directly based on the information contained in the measurement models (prior) and actual measurements (evidence). It can be expected that the computed PLs are tight, since, in theory, all the position information contained in the measurements can be preserved in the posterior. The downside of Bayesian RAIM is its potentially high complexity associated with posterior computation, particularly when the number of unknown parameters is large and the problem model admits no closed-form expressions (which, unfortunately, is the general situation). Therefore, a major challenge of Bayesian methods is to find computationally efficient implementations. To date, research work on Bayesian RAIM methods is limited, mainly based on Monte Carlo algorithms, such as particle filters [15], [16] or Gibbs samplers [14], for posterior distribution computation.

In this paper, we consider the RAIM problem for snapshot[1] cellular positioning, providing three distinct contributions:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101006664. The authors would like to thank all partners within Hi-Drive for their cooperation and valuable contribution. This work is also supported in part by Spanish R+D Grant PID2020-118984GB-I00, the Catalan ICREA Academia Programme, and MSCA-IF grant 101065422 (6G-ISLAC).

[1]The connection of user position between epochs is ignored, although it can be included naturally through the prediction step in the proposed Bayesian (filter) framework, and we assume that all measurements are taken at the same time instance in one snapshot.

(i) we propose a novel factor graph-based Bayesian RAIM method to compute position and PL, including multi-fault detection and exclusion; (ii) we evaluate the method in a simplified one-dimensional (1D) linear Gaussian scenario and compare its performance with a baseline advanced RAIM algorithm [9] using Monte-Carlo simulation; (iii) we demonstrate that, while fulfilling the TIR requirement and incurring comparable computational complexity, the resultant PLs are much tighter than the baseline algorithm due to the exact posterior probability density computation by the new method, thus greatly improving the availability of the system. These pave the way for the development of the Bayesian RAIM method in 3D scenarios.

II. Problem Formulation

In this section, we describe the snapshot 3D positioning and integrity monitoring problem and its simplified 1D version on which we test the proposed Bayesian RAIM method.

A. Snapshot Integrity Monitoring Problem

Consider a downlink/user-centric positioning scenario with M time-synchronized base stations (BSs) with known locations $\boldsymbol{x}_i \in \mathbb{R}^3$, $i \in \mathcal{I} \triangleq \{1, \ldots, M\}$, and a single mobile user equipment (UE) with unknown position $\boldsymbol{x} = [x_{\mathrm{X}}, x_{\mathrm{Y}}, x_{\mathrm{Z}}]^{\mathrm{T}} \in \mathbb{R}^3$ and clock bias $C \in \mathbb{R}$ (expressed in meters). At each positioning epoch, the BSs send coordinated positioning reference signals (PRSs), and the UE estimates M times-of-arrival (ToAs) from the received PRSs over the line-of-sight (LoS) paths, which are converted to pseudo-ranges as follows: For $i \in \mathcal{I}$,

$$y_i = \|\boldsymbol{x}_i - \boldsymbol{x}\| + C + b_i + n_i, \tag{1}$$

where n_i is the independent measurement noise, and b_i is the range bias that accounts for any possible faults, such as synchronization errors or non-line-of-sight (NLoS) biases. We further consider that an initial estimate of the UE position is available, so that (1) can be linearized [17]–[19] to yield (after removing unnecessary terms)

$$y_i = \boldsymbol{a}_i^{\mathrm{T}} \begin{bmatrix} \boldsymbol{x} \\ C \end{bmatrix} + b_i + n_i, \tag{2}$$

where $\boldsymbol{a}_i \in \mathbb{R}^{4\times 1}$ is a known vector. The objective of integrity monitoring is to calculate a position estimate $\hat{\boldsymbol{x}}$ and a PL for each dimension n in $\{\mathrm{X}, \mathrm{Y}, \mathrm{Z}\}$ such that

$$\Pr(|x_n - \hat{x}_n| > \mathrm{PL}_n) \leq \mathrm{TIR}_n, \tag{3}$$

where $\Pr(|x_n - \hat{x}_n| > \mathrm{PL}_n)$ is the actual integrity risk (IR) and TIR_n the TIR for dimension n. A schematic illustration of the problem is given in Fig. 1.

B. Simplified Problem

Inspired by the observation from (3), that the PL is computed per dimension, we propose a simplified observation model, with the purpose of understanding the possible gains of Bayesian integrity monitoring over conventional frequentist approaches. In its most simple and nontrivial version, that is, a

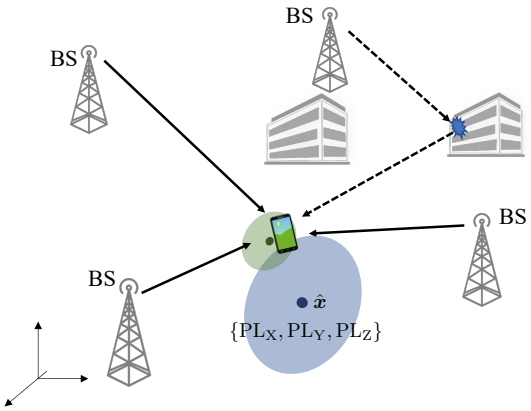


Fig. 1. Illustration of the cellular positioning with integrity monitoring problem. In the displayed scenario, one BS is in NLoS condition due to blockage and causes a large bias to the pseudo-range measurement. When a RAIM algorithm failed to exclude this faulty measurement, the computed position estimate $\hat{\boldsymbol{x}}$ and PLs $\{\mathrm{PL_X}, \mathrm{PL_Y}, \mathrm{PL_Z}\}$ are represented by the blue ellipsoid are obtained; when it excludes the faulty measurement successfully, better results represented by the green ellipsoid are obtained.

1D positioning problem without clock bias and with Gaussian error, the 1D model analogy to (2) is[2]

$$y_i = x + b_i + n_i, \tag{4}$$

where x, $\{b_i\}$, and $\{n_i\}$ are treated as realizations of independent random variables X, $\{B_i\}$, and $\{N_i\}$, respectively, from a Bayesian perspective. The measurement noise is modeled as $N_i \sim \mathcal{N}(n_i; 0, \sigma_{n,i}^2)$, where the notation $Z \sim \mathcal{N}(z; m, \sigma^2)$ represents the Gaussian distribution for the random variable Z with mean m and variance σ^2. A latent variable Λ_i, which takes value $\lambda_i \in \{0, 1\}$ following a Bernoulli probability mass function (PMF) given by $p_{\Lambda_i}(\lambda_i) = \theta_i^{\lambda_i}(1-\theta_i)^{(1-\lambda_i)}$, where $0 < \theta_i \ll 1$ is a known prior probability, is adopted as an indicator of whether the measurement y_i is faulty or not. When $\lambda_i = 0$, $b_i = 0$ with probability 1 and y_i is free from fault; when $\lambda_i = 1$, B_i is modeled as a Gaussian random variable $B_i \sim \mathcal{N}(b_i; m_{b,i}, \sigma_{b,i}^2)$, whose variance $\sigma_{b,i}^2$ is considered considerably greater than $\sigma_{n,i}^2$. Consequently, the prior probability density function (PDF) of B_i is given by

$$\begin{aligned} p_{B_i}(b_i) &= p_{B_i|\Lambda_i}(b_i \mid 0)\, p_{\Lambda_i}(0) + p_{B_i|\Lambda_i}(b_i \mid 1)\, p_{\Lambda_i}(1) \\ &= (1-\theta_i)\delta(b_i) + \theta_i \mathcal{N}(b_i; m_{b,i}, \sigma_{b,i}^2), \end{aligned} \tag{5}$$

where $\delta(\cdot)$ denotes the Dirac delta distribution. The objective is then to find a position estimate $\hat{x}$ and a PL such that

$$\Pr(|x - \hat{x}| > \mathrm{PL}) \leq \mathrm{TIR}. \tag{6}$$

III. Baseline Advanced RAIM

For performance benchmarking purposes, we modify the hypothesis testing-based advanced RAIM algorithm [9] to the 1D problem described above. The original algorithm was developed in a multi-fault and multi-constellation GNSS setting. The algorithm has four components: (i) computation of the so-called all-in-view solution based on all M measurements; (ii)

[2]The focus on the 1D model is without any significant loss of generality since the method can be applied directly to a 3D scenario, computing the PL for each dimension separately.

identification of the fault modes that will be monitored, that is, which combination of BSs will be considered; (iii) detection of the faults and computation of PL if no faults were detected; (iv) exclusion of possibly faulty measurements if faults were detected and computation of PL.

A. All-in-View Solution

We rewrite the measurement model (4) in vector form:

$$\boldsymbol{y} = \mathbf{1}_M x + \boldsymbol{z}, \tag{7}$$

where $\boldsymbol{y} = [y_1, \ldots, y_M]^{\mathrm{T}}$, $\mathbf{1}_M$ is a $M \times 1$ all-one vector and $\boldsymbol{z} = [z_1, \ldots, z_M]^{\mathrm{T}}$ with elements $z_i = b_i + n_i$, $i = 1, \ldots, M$. When all measurements are free of fault, $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is the diagonal covariance matrix with diagonals $\boldsymbol{\Sigma}_{i,i} = \sigma_{n,i}^2$, $i = 1, \ldots, M$, and the weighted least squares (WLS) estimate for x, which in this case also is the maximum likelihood (ML) estimate, is given by

$$\hat{x}^{(0)} = \left(\mathbf{1}_M^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \mathbf{1}_M\right)^{-1} \mathbf{1}_M^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \boldsymbol{y} = W_0 \boldsymbol{c}_0^{\mathrm{T}} \boldsymbol{y}, \tag{8}$$

where $W_0 \triangleq \left(\mathbf{1}_M^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \mathbf{1}_M\right)^{-1} = \left(\sum_{i=1}^{M} \sigma_{n,i}^{-2}\right)^{-1}$, and $\boldsymbol{c}_0^{\mathrm{T}} \triangleq \mathbf{1}_M^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} = [\sigma_{n,1}^{-2}, \ldots, \sigma_{n,M}^{-2}]$.

B. Fault Modes Identification

First, the baseline RAIM algorithm determines the set of *fault modes* to be monitored, based on prior knowledge of the probability of each measurement being faulty. A fault mode is a specific subset of measurements that are simultaneously faulty, while the rest of the measurements are fault-free. The baseline algorithm lists all fault modes that contain up to $M-2$ measurements[3] and computes the corresponding probabilities of occurrence. The total number of fault modes is given by $N_{\mathrm{FM}} = \sum_{j=1}^{M-2} \binom{M}{j}$. For convenience, the fault-free case is denoted as the fault mode 0. We denote by $\mathcal{I}_k$ the set of faulty measurement indices contained in the fault mode $k \in \{0, 1, \ldots, N_{\mathrm{FM}}\}$. In particular, $\mathcal{I}_0 = \emptyset$.The probability that this fault mode occurs is thus given by

$$p_{\mathrm{FM},k} = \prod_{i \in \mathcal{I}_k} \theta_i \prod_{i \in \mathcal{I} \setminus \mathcal{I}_k} (1 - \theta_i). \tag{9}$$

Fault modes are sorted in decreasing order of their probability of occurrence. Namely, $p_{\mathrm{FM},k} \geq p_{\mathrm{FM},k+1}$, for $k < N_{\mathrm{FM}}$. Note that since $0 < \theta_i \ll 1$, $p_{\mathrm{FM},0} > p_{\mathrm{FM},1}$ is considered true.

To ensure the best possible performance, *all* fault modes that can be monitored are listed in this baseline RAIM algorithm. It is also possible to skip fault modes that contain a large number of faulty measurements but with a very small probability of occurrence for computational complexity reduction at the expense of an extra TIR margin [9].

C. Fault Detection

For fault mode k, we define $W_k \triangleq \left(\sum_{i \in \mathcal{I} \setminus \mathcal{I}_k} \sigma_{n,i}^{-2}\right)^{-1}$ and let $\boldsymbol{c}_k^{\mathrm{T}}$ to be the vector given by replacing elements of $\boldsymbol{c}_0^{\mathrm{T}}$ with indices in $\mathcal{I}_k$ by 0. It can be easily verified that

$$\hat{x}^{(k)} = W_k \boldsymbol{c}_k^{\mathrm{T}} \boldsymbol{y} \tag{10}$$

[3]To be able to perform fault detection, as will be soon detailed, there should be at least $N_{\mathrm{unknown}} + 1$ measurements available, where N_{unknown} stands for the number of unknown position variables, which is 1 in the 1D model.

is the WLS estimate of x using the remaining measures after excluding those contained in fault mode k from $\boldsymbol{y}$.

The baseline RAIM algorithm determines whether the measurements contain faults by performing a list of *solution separation (SS) tests* for each fault mode. When all measurements are fault-free, it can be shown that the test statistic $\Delta\hat{x}^{(k)} \triangleq \hat{x}^{(0)} - \hat{x}^{(k)}$ is a Gaussian random variable with zero mean and variance given by

$$\sigma_{ss}^{(k)2} \triangleq \left(W_k \boldsymbol{c}_k^{\mathrm{T}} - W_0 \boldsymbol{c}_0^{\mathrm{T}}\right) \boldsymbol{\Sigma} \left(W_k \boldsymbol{c}_k^{\mathrm{T}} - W_0 \boldsymbol{c}_0^{\mathrm{T}}\right)^{\mathrm{T}}. \tag{11}$$

To be able to identify the test thresholds, a false alarm probability P_{FA} is also required as input to the algorithm. The false alarm budget is evenly allocated to the N_{FM} fault modes that contain fault(s), leading to the following SS test threshold for fault mode k, $k = 1, \ldots, N_{\mathrm{FM}}$,

$$T_k = \sigma_{ss}^{(k)} Q^{-1}\left(\frac{P_{\mathrm{FA}}}{2N_{\mathrm{FM}}}\right), \tag{12}$$

where $Q^{-1}(\cdot)$ is the inverse of the Q function, $Q(u) = \frac{1}{\sqrt{2\pi}} \int_u^{+\infty} e^{-\frac{t^2}{2}} \mathrm{d}t$. To be specific, $|\Delta\hat{x}^{(k)}|$ will be compared with T_k, and if $|\Delta\hat{x}^{(k)}| \leq T_k$, $\forall k = 1, \ldots, N_{\mathrm{FM}}$, all measurements are considered fault-free and the baseline RAIM algorithm outputs $\hat{x}^{(0)}$ as the position estimate. The choice of P_{FA}, therefore, affects the continuity and availability performance of the positioning system: the larger the value, the more likely the algorithm is to warn upon a potential fault and will put more computational effort into fault exclusion attempts.

A PL is computed by solving the following equation:

$$2Q\left(\frac{\mathrm{PL}}{\sigma^{(0)}}\right) + \sum_{k=1}^{N_{\mathrm{FM}}} p_{\mathrm{FM},k}\, Q\left(\frac{\mathrm{PL} - T_k}{\sigma^{(k)}}\right) = \mathrm{TIR}, \tag{13}$$

where $\sigma^{(k)} \triangleq W_k \sqrt{\boldsymbol{c}_k^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{c}_k}$, and $Q\left(\frac{\mathrm{PL}}{\sigma^{(0)}}\right)$ and $Q\left(\frac{\mathrm{PL}-T_k}{\sigma^{(k)}}\right)$ are in fact upper bounds for the two tails of the actual IR when fault mode k occurs but the SS test passes. The formulation process of the above equation can be found in [9, Appendix H], and a solution of (13) is found using the bisection search method detailed in [9, Appendix B].

D. Fault Exclusion

When any SS tests fail, that is, $|\Delta\hat{x}^{(k)}| > T_k$ for some k, the baseline RAIM algorithm will try to exclude faulty measurements that cause failure. Starting from $k = 1$, the algorithm performs the *complete* process of fault modes identification and fault detection on the *new* problem formed after removing the measurements contained in fault mode k. The remaining number of measurements in this new problem is $M - |\mathcal{I}_k|$. If the SS tests pass successfully, the fault exclusion process is terminated and a PL can be computed in the same way as described above (but using the newly obtained values for $p_{\mathrm{FM},k}$, T_k, N_{FM}, etc.). Otherwise, the algorithm continues to check the next fault mode $(k + 1)$. If the SS tests fail for all fault modes, then the fault exclusion attempt is considered failed. In this case, the algorithm terminates without being able to compute a PL and declares integrity unavailable.

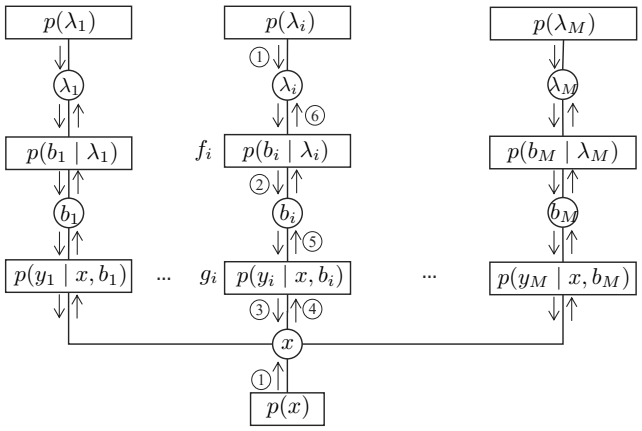


Fig. 2. Factor graph of the Bayesian RAIM problem, corresponding to the factorization (14). The order of message computation and passing is given by the numbers in circles (shown only on the ith branch, but is the same for all branches), while the arrows indicate the passing direction.

IV. Proposed Bayesian Methods

For each epoch of the integrity monitoring problem, the proposed Bayesian RAIM computes the marginal posterior distributions $p_{X|\boldsymbol{Y}}(x|\boldsymbol{y})$ and $p_{\Lambda_i|\boldsymbol{Y}}(\lambda_i|\boldsymbol{y})$, $i \in \mathcal{I}$, via message passing along a factor graph built according to the model described in Section II-B. Based on the obtained posterior distributions, different methods can be selected to compute the position estimate $\hat{x}$ and PL to meet the TIR requirement. In the description of the message passing algorithm, the prior probability distribution of X, $p_X(x)$, which would be passed on from the previous epoch(s) in practice, is assumed known. In the absence of prior knowledge, $p_X(x)$ can be set to a uniform distribution over a large region, as will be assumed in our numerical study.

A. Message Passing Algorithm

Based on the assumptions made in Section II, the joint posterior probability of x, $\boldsymbol{b} \triangleq [b_1, \ldots, b_M]^{\mathrm{T}}$ and $\boldsymbol{\lambda} \triangleq [\lambda_1, \ldots, \lambda_M]^{\mathrm{T}}$ can be factorized as

$$\begin{aligned} p(x, \boldsymbol{b}, \boldsymbol{\lambda} \mid \boldsymbol{y}) &\propto p(\boldsymbol{y} \mid x, \boldsymbol{b}, \boldsymbol{\lambda}) p(x, \boldsymbol{b}, \boldsymbol{\lambda}) \\ &= p(x) \prod_{i=1}^{M} p(y_i \mid x, b_i) p(b_i \mid \lambda_i) p(\lambda_i). \end{aligned} \tag{14}$$

For clarity, the subscripts of the probability distributions are omitted in (14) and all subsequent expressions and the variables to which they belong should be clear from the context.

Despite the seemingly simple form, the task of computing the desired posterior probabilities from (14) is not trivial. To do that, a cycle-free factor graph representation of (14), shown in Fig. 2, is adopted. Each term of (14) is represented by a factor node in a rectangle and connected to the variable nodes that appear in parentheses, represented by circles in the graph. For simplicity, factor nodes $p(b_i|\lambda_i)$ and $p(y_i|x, b_i)$ are denoted by f_i and g_i respectively. A message-passing schedule is applied following the sum-product algorithm [20].

The following notations and operations are involved. $\mathrm{GM}(z, L)$ stands for a Gaussian mixture (GM) distribution for Z with L components: $\sum_{l=1}^{L} w_l \mathcal{N}(z; m_l, \sigma_l^2)$ with $\sum_{l=1}^{L} w_l = 1$. Many messages passed on the factor graph are of this form, and their weights $\{w_l\}$, means $\{m_l\}$, and standard deviations $\{\sigma_l\}$ are what actually needs to be sent. Given $\mathrm{GM}(z, L_1) = \sum_{l_1=1}^{L_1} w_{l_1} \mathcal{N}(z; m_{l_1}, \sigma_{l_1}^2)$ and $\mathrm{GM}(z, L_2) = \sum_{l_2=1}^{L_2} w_{l_2} \mathcal{N}(z; m_{l_2}, \sigma_{l_2}^2)$, their product is given by

$$\sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} w_{l_1} w_{l_2} s_{l_1 l_2} \mathcal{N}(z; m_{l_1 l_2}, \sigma_{l_1 l_2}^2) \propto \mathrm{GM}(z, L_1 L_2), \tag{15}$$

where $s_{l_1 l_2}$, $m_{l_1 l_2}$ and $\sigma_{l_1 l_2}$ are obtained by solving the following simple equations:

$$\begin{cases} 1/\sigma_{l_1 l_2}^2 = 1/\sigma_{l_1}^2 + 1/\sigma_{l_2}^2, \\ m_{l_1 l_2}/\sigma_{l_1 l_2}^2 = m_{l_1}/\sigma_{l_1}^2 + m_{l_2}/\sigma_{l_2}^2, \\ s_{l_1 l_2} = \mathcal{N}(m_{l_1}; m_{l_2}, \sigma_{l_1}^2 + \sigma_{l_2}^2). \end{cases} \tag{16}$$

A limited number of arithmetic operations are needed to solve these equations. Therefore, the computational complexity associated with this product is given by $\mathcal{O}(L_1 L_2)$.

The message-passing schedule is described below for the ith branch. It proceeds in parallel along all M branches.

① The prior $p(\lambda_i)$ is sent from the leaf node to variable nodes λ_i, and then directly to f_i. Meanwhile, $p(x)$ is sent to the variable node x.

② Factor node f_i sends the following message to variable node b_i:

$$\mu_{f_i \to b_i}(b_i) = \sum_{\lambda_i} p(b_i \mid \lambda_i) p(\lambda_i), \tag{17}$$

which is a GM distribution of b_i, given by $\mu_{f_i \to b_i}(b_i) = (1 - \theta_i) \mathcal{N}(b_i; 0, 0) + \theta_i \mathcal{N}(b_i; m_{b,i}, \sigma_{b,i}^2)$. This message will be passed directly to factor node g_i.

③ Factor node g_i sends the following message to variable node x:

$$\mu_{g_i \to x}(x) = \int p(y_i \mid x, b_i) \, \mu_{f_i \to b_i}(b_i) \, \mathrm{d}b_i. \tag{18}$$

Since $p(y_i|x, b_i) = \mathcal{N}(b_i; y_i - x, \sigma_{n,i}^2)$, based on (15) and (16), the integration results in a $\mathrm{GM}(x, 2)$ density:

$$(1 - \theta_i) \mathcal{N}(x; y_i, \sigma_{n,i}^2) + \theta_i \mathcal{N}(x; y_i - m_{b,i}, \sigma_{n,i}^2 + \sigma_{b,i}^2). \tag{19}$$

④ Variable node x sends the following message back to g_i:

$$\mu_{x \to g_i}(x) = p(x) \prod_{j \in \mathcal{M} \setminus i} \mu_{g_j \to x}(x). \tag{20}$$

Under the assumption of Gaussian or uniform distribution of $p(x)$, $\mu_{x \to g_i}(x) \propto \mathrm{GM}(x, 2^{M-1})$.

⑤ Factor node g_i sends the following message back to b_i:

$$\mu_{g_i \to b_i}(b_i) = \int p(y_i \mid x, b_i) \, \mu_{x \to g_i}(x) \, \mathrm{d}x, \tag{21}$$

which is then passed on to factor node f_i directly. Since $p(y_i|x, b_i)$ can also be regarded as Gaussian PDF of x, i.e., $\mathcal{N}(x; y_i - b_i, \sigma_{n,i}^2)$, following the same procedure as

in step ③, it is clear that $\mu_{g_i\to b_i}(b_i) \propto \mathrm{GM}(b_i, 2^{M-1})$ and the parameters are easy to compute.

⑥ Finally, the factor node f_i sends a message back to λ_i, which is given by

$$\mu_{f_i\to\lambda_i}(\lambda_i) = \int p(b_i \mid \lambda_i)\, \mu_{g_i\to b_i}(b_i)\, \mathrm{d}b_i. \tag{22}$$

The message is computed separately for $\lambda_i = 0$ and $\lambda_i = 1$. In particular, $\mu_{f_i\to\lambda_i}(\lambda_i = 0) = \mu_{g_i\to b_i}(b_i = 0)$, and the computation of $\mu_{f_i\to\lambda_i}(\lambda_i = 1)$ requires 2^{M-1} terms of product of two Gaussian in the integrand.

The message-passing process terminates after the above six steps. The desired marginal posteriors are then calculated following

$$p(x \mid \boldsymbol{y}) \propto p(x) \prod_{i\in\mathcal{M}} \mu_{g_i\to x}(x), \tag{23}$$

which leads to a GM distribution with 2^M terms, and

$$p(\lambda_i \mid \boldsymbol{y}) = \begin{cases} 1-\theta_i' \propto (1-\theta_i)\cdot \mu_{f_i\to\lambda_i}(\lambda_i = 0), & \text{if } \lambda_i = 0, \\ \theta_i' \quad\;\; \propto \theta_i \cdot \mu_{f_i\to\lambda_i}(\lambda_i = 1), & \text{if } \lambda_i = 1, \end{cases} \tag{24}$$

where θ_i' can be obtained after normalization. Note that since the factor graph is cycle-free, convergence is guaranteed.

B. Fault Exclusion

In theory, all the information about the user position contained in the measurements and priors has been collected in $p(x|\boldsymbol{y})$ given by (23), and a position estimate and a PL can be computed directly based on it. However, the bias term in a faulty measurement will introduce a large uncertainty in the posterior, which will lead to a large PL. To counteract this effect, fault exclusion can be performed based on the marginal posterior PMFs of the indicators given by (24), before computing the marginal posterior PDF of x using the remaining measurements. To be specific, a threshold θ_T is selected, and if $\theta_i' > \theta_T$, the measurement y_i is considered faulty, the corresponding branch will be pruned from the factor graph and the message $\mu_{g_i\to x}(x)$ passed to variable node x along it will be discarded. No other additional message computation/passing is required.

Denoting the set of indices of excluded measurements is by $\mathcal{I}_\mathrm{F}$, the marginal posterior of x after fault exclusion is computed by

$$p_\mathrm{FE}(x \mid \boldsymbol{y}) \propto p(x) \prod_{i\in\mathcal{I}\setminus\mathcal{I}_\mathrm{F}} \mu_{g_i\to x}(x), \tag{25}$$

which leads to a GM posterior distribution of $2^{M-|\mathcal{I}_\mathrm{F}|}$ terms.

C. Position Estimation and PL Computation

With or without fault exclusion, the marginal posterior of x can be written in the following form:

$$p_X^\mathrm{pos}(x) = \sum_{l=1}^{L} w_{x,l}\, \mathcal{N}(x; m_{x,l}, \sigma_{x,l}^2), \tag{26}$$

where $\{w_{x,l}, l = 1,\dots,L\}$ is sorted in decreasing order and L is either given by 2^M following (23), or by $2^{M-|\mathcal{I}_\mathrm{F}|}$ following (25). For position estimation, the weighted mean (WM) method is adopted, so that $\hat{x} = \sum_{l=1}^{L} w_{x,l}\, m_{x,l}$.

Based on (26), the IR associated with a position estimate $\hat{x}$ and a said PL can be formulated as

$$\begin{aligned} \mathrm{IR} &= \Pr(x < \hat{x} - \mathrm{PL}) + \Pr(x > \hat{x} + \mathrm{PL}) \\ &= \sum_{l=1}^{L} w_{x,l} \Big[\Phi_{\mathcal{N},l}(\hat{x} - \mathrm{PL}) + 1 - \Phi_{\mathcal{N},l}(\hat{x} + \mathrm{PL})\Big], \end{aligned} \tag{27}$$

where $\Phi_{\mathcal{N},l}(\cdot)$ stands for the cumulative density function (CDF) of the lth Gaussian term in (26). Using the notation of Q function as in (13), the goal is to find the smallest value for PL such that the following inequality holds:

$$\begin{aligned} &\sum_{l=1}^{L} w_{x,l} \left[1 - Q\Big(\frac{\hat{x}-\mathrm{PL}-m_{x,l}}{\sigma_{x,l}}\Big) + Q\Big(\frac{\hat{x}+\mathrm{PL}-m_{x,l}}{\sigma_{x,l}}\Big)\right] \\ &\le \mathrm{TIR}. \end{aligned} \tag{28}$$

For this, a bisection search is again employed by the Bayesian RAIM algorithms.

Depending on whether to perform fault exclusion or not, two variations of the Bayesian RAIM algorithm have resulted under the same framework (labeled using FE or NFE in the numerical study). Their performance will be compared with the baseline RAIM algorithm in the following section. We also remark that other position estimation methods, e.g., maximum a posterior (MAP) estimation, can also be adopted, which can be expected to have an impact on the computed PL according to (28). The comparison of different position estimation methods will be left for future work.

D. Computational Complexity Comparison

For all other steps except PL computation, the Bayesian RAIM algorithm has a fixed complexity scaling of $\mathcal{O}(3M \times 2^{M-1})$, mainly due to the computation of (23) via message passing. The baseline RAIM has variable complexity (as the number of SS test varies), between $\mathcal{O}(N_\mathrm{FM})$ (when all SS tests pass in the fault detection step) and approximately $\mathcal{O}(N_\mathrm{FM}^2)$ (when every fault mode is examined in the fault exclusion step), where $N_\mathrm{FM} = \sum_{j=1}^{M-2} \binom{M}{j} = 2^M - M - 2$.

Both methods require a bisection search to compute the PL, which is an iterative algorithm. For a given number of iterations, say $N_\mathrm{it} > 0$, the complexity of the PL computation in Bayesian RAIM is $\mathcal{O}(N_\mathrm{it} 2L)$, where $L = 2^M$ without fault exclusion and $L = 2^{M-|\mathcal{I}_\mathrm{F}|}$ with fault exclusion. For baseline RAIM, this complexity is $\mathcal{O}(N_\mathrm{it} N_\mathrm{FM})$.

Remark 1. *In cellular positioning, there are usually few connected BSs ($M \le 10$). Therefore, the computational complexities of the baseline and proposed Bayesian RAIM algorithms can be said in a similar order for the 1D positioning problem. We note that in GNSS, the number of observable signals can be much higher, which could entail high computational complexities of the Bayesian algorithms.*

V. Numerical Study

A. Simulation Setting

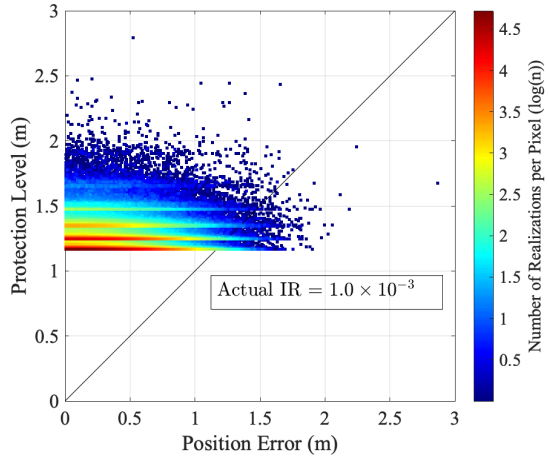
We conduct a numerical study with one UE and M BSs, where M is set to either 5 or 8. Without loss of generality, the UE's true position is fixed at $x = 0$. Although measurement noise levels ($\sigma_{n,i}$) in practice can vary across BSs and possibly depend on their locations, we assume a single value (σ_n) for all BSs in this simulation for validation purposes. We evaluate the performance as σ_n increases from 1 to 9 meters with 2-meter increments. We also assume that the probability of each BS generating a faulty measurement is the same ($\theta_i = 0.05$, for all $i = 1, \ldots, M$). Gaussian biases in the faulty measurements have the same variance ($\sigma_{b,i} = 50$ meters) but different mean values, which are randomly generated following $m_{b,i} \sim \text{uni}[-50, 50]$ meters in each set of simulations (corresponding to a certain pair of values for M and σ_n).

The TIR is set to 10^{-3}, and we simulate at least 5×10^6 independent realizations in each set of simulations. For the Bayesian RAIM algorithm with fault exclusion, we set the threshold $\theta_T = 0.5$. For the baseline RAIM algorithm, we select $P_{\text{FA}} = 5 \times 10^{-2}$.

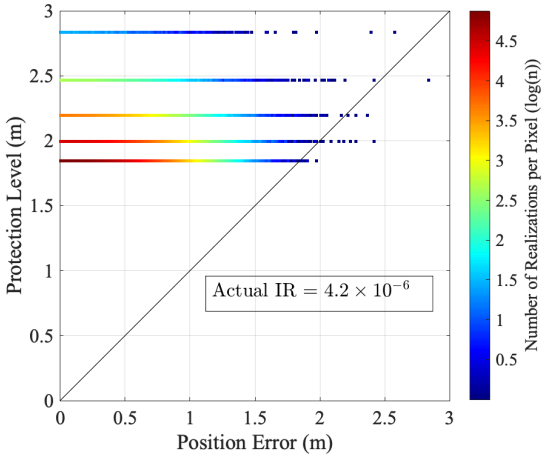
B. Results and Discussion

In each set of simulations, the simulated IR is obtained, that is, the ratio of realizations when the actual position error $|x - \hat{x}|$ exceeds the computed PLs among all realizations. It is found that the simulated IRs resultant from the Bayesian RAIM algorithms, with or without fault exclusion, all converged to the TIR after running enough realizations. This in turn demonstrates that the posterior probabilities computed via the factor graph are exact. On the other hand, the simulated IR results of the baseline RAIM algorithm are in the order of 10^{-6}, much lower than the TIR. For visualization, the results given by the Bayesian RAIM algorithm (with fault exclusion) and the baseline RAIM algorithm (also with fault exclusion) in the setting with $M = 8$ BSs and $\sigma_n = 1$ meter over 5×10^6 realizations are presented in the form of Stanford diagram [3] in Fig. 3. As can be seen, the proposed Bayesian RAIM provides much tighter PLs while meeting the TIR requirement, compared to the baseline RAIM. The latter is conservative in PL computation, so the computed PLs are larger and the actual IR is significantly smaller than the TIR. One can also see that the PL results the baseline RAIM algorithms returns are discrete. This is because, unlike the Bayesian methods, the actual measurements are not utilized in the PL computation, as can be seen from (13). In particular, in (13), N_{FM} is determined by the number of measurements excluded, and $p_{\text{FM},k}$, T_k and $\sigma^{(k)}$ can only take a limited number of values, since the measurement models are assumed to be the same for all BSs.

For each set of simulations, empirical complementary cumulative distribution function (CCDF) curves of the computed PLs are obtained. Four sets of the CCDF curves are presented in Fig. 4, from which the significant performance improvement of the proposed Bayesian RAIM algorithms over the baseline



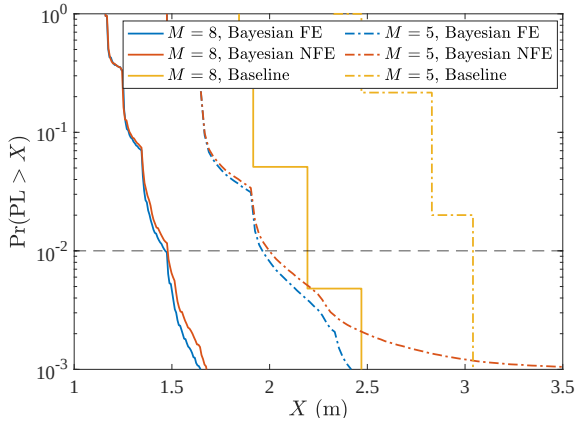
(a) Bayesian RAIM with fault exclusion



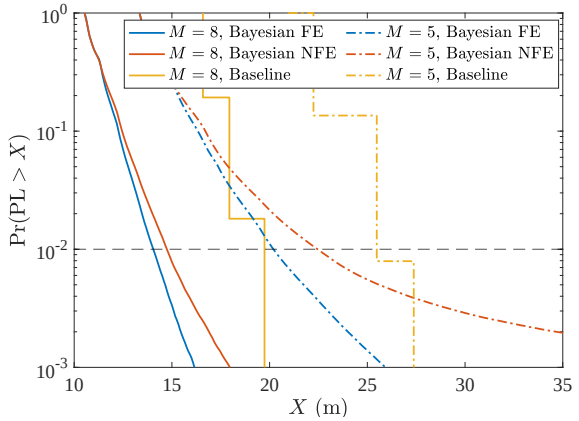
(b) Baseline RAIM

Fig. 3. Stanford diagrams of the proposed Bayesian RAIM algorithm with fault exclusion in (a) and the baseline RAIM algorithm in (b). Plotted for $M = 8$ and $\sigma_n = 1$ over 5×10^6 realizations, and with a 1 cm $\times$1 cm pixel size. A dot that appears on the bottom right side of the diagonal line represents an integrity failure (i.e., $|x - \hat{x}| > \text{PL}$); the simulated IR of the Bayesian RAIM (a) is much tighter to the desired bound $\leq$ TIR (10^{-3}).

RAIM algorithm in obtaining tighter PLs can be clearly seen. In addition, the importance of fault exclusion before PL computation is revealed. Comparing the PL results given by the Bayesian RAIM algorithms with and without fault exclusion, we see that the large uncertainty introduced by the potentially faulty measurements in the posteriors leads to larger PL results (in a statistical sense), and the gap increases when σ_n increases or/and when M decreases, because the uncertainty can be reduced by providing more fault-free measurements or by reducing the measurement noises. To better illustrate this effect, in Fig. 5, the PL values at 99% percentile (given by the intersections of the horizontal line at 10^{-2} with the CCDF curves) obtained under all sets of simulations are shown. Another interesting observation from Fig. 5 is that the 99% percentile PL results given by any evaluated RAIM algorithms seem to increase linearly with σ_n.



(a) $\sigma_n = 1$ meter



(b) $\sigma_n = 9$ meters

Fig. 4. CCDF curves of PLs at two measurement noise levels.

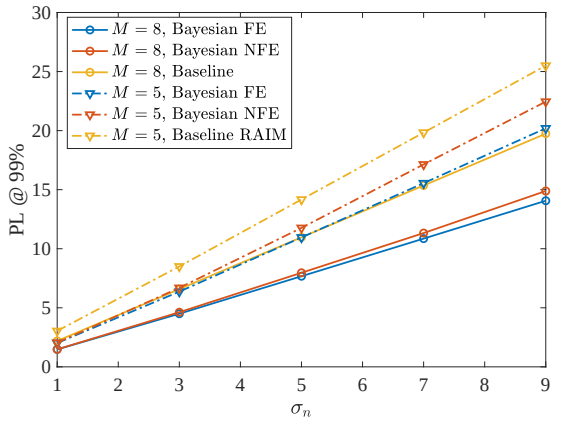


Fig. 5. PL at 99% percentile as a function of the measurement noise level.

VI. Conclusion

In this paper, we have developed Bayesian RAIM algorithms for snapshot-type positioning problems in a 1D linear Gaussian setting, which serves as a methodological validation. These algorithms enable position estimation, multi-fault detection and exclusion, and computation of PL based on exact posterior distributions obtained through message passing along a factor graph. Monte-Carlo simulations demonstrate that our proposed algorithms achieve tight PLs while satisfying the given TIR requirement, resulting in a significant performance improvement over the baseline advanced RAIM algorithm. Based on these promising results, we plan to extend the algorithms to the ToA-based 3D positioning problem within the same framework.

References

[1] C. De Lima, D. Belot, R. Berkvens *et al.*, "Convergent communication, sensing and localization in 6G systems: An overview of technologies, opportunities and challenges," *IEEE Access*, vol. 9, pp. 26 902–26 925, 2021.

[2] A. Behravan, V. Yajnanarayana, M. F. Keskin *et al.*, "Positioning and sensing in 6G: Gaps, challenges, and opportunities," *IEEE Veh. Technol. Mag.*, 2022, early access.

[3] R. Whiton, "Cellular localization for autonomous driving: A function pull approach to safety-critical wireless localization," *IEEE Veh. Technol. Mag.*, vol. 17, no. 4, pp. 2–11, 2022.

[4] T. G. Reid, S. E. Houts, R. Cammarata *et al.*, "Localization requirements for autonomous vehicles," *SAE Int. J. Connected Autom. Veh.*, vol. 2, no. 12-02-03-0012, pp. 173–190, 2019.

[5] S. Bartoletti, H. Wymeersch, T. Mach *et al.*, "Positioning and sensing for vehicular safety applications in 5G and beyond," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 15–21, 2021.

[6] M. Maaref and Z. M. Kassas, "Autonomous integrity monitoring for vehicular navigation with cellular signals of opportunity and an IMU," *IEEE Trans. Intell. Transp. Syst.*, 2021.

[7] J. Larson and D. Gebre-Egziabher, "Conservatism assessment of extreme value theory overbounds," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 3, pp. 1295–1307, 2017.

[8] T. Walter, P. Enge, and B. DeCleene, "Integrity lessons from the WAAS integrity performance panel (WIPP)," in *Proc. ION NTM*, 2003, pp. 183–194.

[9] J. Blanch, T. Walker, P. Enge *et al.*, "Baseline advanced RAIM user algorithm and possible improvements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 713–732, 2015.

[10] 3GPP, "Study on NR positioning enhancements," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.857, March 2021, version 17.0.0.

[11] ——, "Study on expanded and improved NR positioning," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.859, Aug 2022, version 0.1.0.

[12] N. Zhu, J. Marais, D. Bétaille, and M. Berbineau, "GNSS position integrity in urban environments: A review of literature," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2762–2778, 2018.

[13] H. Pesonen, "A framework for Bayesian receiver autonomous integrity monitoring in urban navigation," *Navigation*, vol. 58, no. 3, pp. 229–240, 2011.

[14] Q. Zhang and Q. Gui, "A new Bayesian RAIM for multiple faults detection and exclusion in GNSS," *J. Navig.*, vol. 68, no. 3, pp. 465–479, 2015.

[15] S. Gupta and G. X. Gao, "Particle RAIM for integrity monitoring," in *Proc. 32nd ION GNSS*, 2019, pp. 811–826.

[16] J. Gabela, A. Kealy, M. Hedley, and B. Moran, "Case study of Bayesian RAIM algorithm integrated with spatial feature constraint and fault detection and exclusion algorithms for multi-sensor positioning," *Navigation*, vol. 68, no. 2, pp. 333–351, 2021.

[17] M. Koivisto, M. Costa, J. Werner *et al.*, "Joint device positioning and clock synchronization in 5G ultra-dense networks," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 5, pp. 2866–2881, 2017.

[18] I. Guvenc, S. Gezici, and Z. Sahinoglu, "Fundamental limits and improved algorithms for linear least-squares wireless position estimation," *Wirel. Commun. Mob. Comput.*, vol. 12, no. 12, pp. 1037–1052, 2012.

[19] S. Zhu and Z. Ding, "A simple approach of range-based positioning with low computational complexity," *IEEE Trans. Wirel. Commun.*, vol. 8, no. 12, pp. 5832–5836, 2009.

[20] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, 2001.