IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 19, 2018, accepted February 7, 2018, date of publication February 13, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2805841

Analytical Approximation-Based Machine Learning Methods for User Positioning in Distributed Massive MIMO

K. N. R. SURYA VARA PRASAD[1], EKRAM HOSSAIN[2], (Fellow, IEEE), VIJAY K. BHARGAVA[1], (Life Fellow, IEEE), AND SHANKHANAAD MALLICK[3]

[1]Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[2]Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB R3T 5V6, Canada
[3]Skycope Technologies Inc., Burnaby, BC V5G 4W7, Canada

Corresponding author: K. N. R. Surya Vara Prasad (surya@ece.ubc.ca)

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

ABSTRACT We propose a machine learning approach, based on analytical inference in Gaussian process regression (GP), to locate users from their uplink received signal strength (RSS) data in a distributed massive multiple-input-multiple-output setup. The training RSS data is considered noise-free, while the test RSS data is assumed to be noisy due to shadowing effects of the wireless channel. We first apply an analytical moment matching-based GP method, namely, the Gaussian approximation GP (GaGP), and make the necessary extensions to suit the problem under study. The GaGP method learns from the stochastic nature of the test RSS data to provide more realistic 2σ error-bars on the estimated locations than the conventional GP (CGP) method. Despite the improvement in 2σ error-bars, simulation studies reveal that the GaGP method achieves similar root-mean-squared estimation error (RMSE) performance as the CGP method. To address this concern, we propose a new GP method, namely the reconstruction-cum-Gaussian-approximation GP (RecGaGP) method. RecGaGP not only achieves lower RMSE values than the CGP and GaGP methods, but also provides realistic 2σ error-bars on the estimated locations. This ability is achieved by first reconstructing the test RSS from a low-dimensional principal subspace of the noise-free training RSS and then learning from the statistical properties of the residual noise present. For both the GaGP and RecGaGP methods, closed-form expressions are derived for the estimated user locations and the associated 2σ error-bars. Numerical studies reveal that the GaGP and RecGaGP methods indeed provide realistic 2σ error-bars on the estimated user locations and their RMSE performances are very close to the Cramer–Rao lower bounds. Also, their RMSE performances saturate beyond a certain point when the number of BS antennas and/or the number of training locations are increased.

INDEX TERMS Machine learning, Gaussian process regression, massive MIMO, user positioning.

I. INTRODUCTION

The ability to position cellular users from their radio signal information can be useful for telecommunication operators because a variety of context-aware services become possible, for example, delivering custom-made advertisements and offloading data to nearby Wi-Fi hotspots. Traditionally, global positioning systems (GPSs) are used to locate users remotely, but the GPS estimates are unreliable for users in cluttered environments [1]. Also, GPS sensors can quickly drain the user's battery. Other positioning techniques rely on radio signal information from the users, such as the angle-of-arrival (AOA), time-of-arrival (ToA), and/or received signal strength (RSS) [2]. AoA methods provide coarse estimates under non-line-of-sight conditions. ToA methods require expensive hardware at the base station (BS) because they rely on tight timing synchronization. Whereas, the RSS information can be readily measured using existing hardware at the BSs and is therefore chosen in our work as the signal property for user positioning.

The massive multiple-input multiple-output (MIMO) [3] technology allows us to employ supervised machine learning (ML) to estimate user locations from their uplink RSS

data. Whenever a user transmits on the uplink, we can obtain a vector comprising RSS values measured at each base station (BS) antenna. An ML model can then be trained with a database comprising RSS vectors at several known user locations. The trained ML model, when input with the RSS vector of a test user, provides the user's location coordinates as the output.

In this work, we propose a machine learning approach based on Gaussian process regression (GP) to estimate user locations from their uplink RSS vectors in a distributed massive MIMO system. Our approach is built on the Gaussian process regression (GP) framework, so as to obtain both the location estimates and their 2σ error-bars in closed-form. A challenging aspect of using RSS vectors for user positioning is that the recorded RSS values may be noisy due to channel impairments in the form of small-scale fading and shadowing. For the training phase, we consider noise-free RSS vectors because they are easy to generate. For this, we would only require knowledge of the training users' transmission power, their location coordinates, the BS antenna locations, and the path-loss exponent. For the test phase, we consider the RSS data as noisy due to shadowing. This is because (i) the small-scale fading effects can be temporally averaged out [4] and (ii) the shadowing effects need to be spatially averaged out, but this is not possible due to unknown test user locations.

It is known that the conventional GP (CGP) method treats the noisy test RSS vectors as noise-free and therefore provides unrealistically small 2σ error-bars on the estimated locations [6]. To address this concern, we first consider the use of an analytical moment-matching based GP method, namely, the Gaussian approximation GP (GaGP) method, for user positioning. In the GaGP method, we first derive the true predictive distribution of the test user locations by taking the stochastic nature of the test RSS vectors into account. Since the derived distribution cannot be obtained in a closed-form, we approximate it analytically as a Gaussian with the same first and second order moments. Despite the improvement in 2σ error-bars, we find through simulations that the GaGP method achieves similar root-mean-squared prediction error (RMSE) performance as the CGP method. To address this concern, we propose a new GP method, namely, the reconstruction-cum-Gaussian approximation GP (RecGaGP) method. RecGaGP estimates the test user locations by firstly reconstructing their RSS data from a low-dimensional principal subspace of the noise-free training RSS and then learning from the statistical properties of the noise present in the reconstructed RSS. While the reconstruction step allows the RecGaGP method to achieve better RMSE performance than the CGP and GaGP methods, the statistical learning step allows it to provide realistic 2σ error-bars on the estimated locations. Below are the main contributions of this work.

(i) For the specific machine learning problem of positioning users in distributed massive MIMO with their uplink RSS, ours is the first work to derive analytical expressions for the mean and 2σ error-bars of the test user locations. We do so by applying an analytical moment-matching based GP method, namely, the GaGP method. A similar method was proposed earlier in [29] for time-series analysis when a squared exponential GP covariance function is used. We make the necessary extensions here to accommodate our weighted-sum GP covariance model of squared exponential, inner product, and delta terms.

(ii) When the GaGP method is employed with the weighted-sum GP covariance model mentioned above, we derive closed-form expressions for the mean and variance of the test user locations and derive two key insights. First, by making the necessary mathematical abstractions, we show that the derived mean and variance expressions in GaGP are similar in structure to those obtained from the conventional GP method, but with several additive and multiplicative correction factors that account for the noisy nature of the test RSS data. Second, for the special case where the test RSS data is noise-free, we show that the mean and variance expressions from the GaGP and the conventional GP methods are exactly the same.

(iii) We propose a new GP method, namely, the RecGaGP, which achieves better RMSE performance than the GaGP method, and also provides realistic 2σ error-bars on the estimated locations. The superior RMSE performance is because the test RSS vectors are reconstructed from a principal subspace of the noise-free training RSS, before being used as inputs for location prediction. Realistic 2σ error-bars are obtained for the estimated locations because RecGaGP learns from the statistical properties of the noise present in the reconstructed test RSS.

(iv) We also present a comprehensive analysis of the computational complexities of the two GP methods under study. Our analysis shows that both the GaGP and RecGaGP methods incur similar cost in providing the location estimates as the conventional GP method and that they are suitable for operation in the massive MIMO regime.

(v) We provide insights on the impact of the number of training locations and the number of BS antennas on the root-mean-squared-error (RMSE) performance of the GaGP and RecGaGP methods. For both the GP methods, we observe that the RMSE performance improves initially, followed by saturation beyond a certain point, when the number of training locations or BS antennas is increased.

Following is the outline for the rest of the paper. We present the literature review in Section II, the system model and the machine learning setup in Section III, the GaGP method in Section IV, the RecGaGP method in Section V, the computational complexities of GaGP and RecGaGP in Section VI, numerical studies in Section VII, and few concluding remarks in Section VIII.

II. RELATED WORK

A. MACHINE LEARNING TECHNIQUES FOR USER POSITIONING

Ranging-based location prediction techniques [4]–[7] have been widely investigated in the wireless networks literature. These techniques use TOA or RSS measurements to estimate the position of a user by firstly estimating the range to three or more base stations (BSs) and then applying trilateration. While the TOA-based ranging methods require tight synchronization and high signal bandwidth for accurate positioning, the RSS-based ranging methods typically provide coarse estimates due to non-line-of-sight (NLoS) effects. Few papers, such as [8]–[10], first identify and mitigate non-line-of-sight (NLoS) effects in the wireless signals and then apply ranging methods for user positioning. These methods rely on the comparison of certain statistical features of the signal measurements, such as the mean, variance, and skewness, for NLoS identification. Consequently, accurate estimates of the statistical distributions are required, which may not always be possible because a large number of measurements would be required for different distances. With these shortcomings in mind, we focus on the use of machine learning for user positioning.

Both unsupervised machine learning techniques, for example k-nearest neighbors [11], [12], and supervised machine learning techniques, for example, support vector machines [13], [14], GP methods [5], [18], and more recently, deep learning methods [15], [16], have been explored in the literature for user positioning applications. We choose to work with GP methods for two reasons. First, GP methods can provide us with closed-form expressions for the estimated user locations and also their 2σ error-bars. Second, GP methods lend themselves to Cramer-Rao type lower bounds on the prediction performance. This is unlike most other machine learning methods, including the recently popular deep learning methods [17].

B. GP METHODS FOR USER POSITIONING

Most of the existing GP works [18]–[21] obtain user locations in an inverse fashion. During the training phase, GP models are trained with location estimates as input and their RSS values as output. During the test phase, the location estimates are obtained via maximum-likelihood of the RSS values. Schwaighofer *et al.* [18] consider RSS from multiple BSs and pursue the above approach for indoor user positioning. In [19], a smoothing approach is proposed to overcome the highly-peaked nature of the joint likelihood of the RSSs. In [20], additional GPs are trained in order to provide coarse position estimates for initializing the RSS likelihood-maximization problem. In [21], Cramer-Rao lower bounds are derived for the hyperparameter estimation error resulting from the GP training procedure.

The above works adopt the conventional GP method for user positioning, which provides unrealistically small 2σ error-bars on the location estimates. Also, since the focus is on the downlink, the users are required to compute their own locations. In contrast, we consider the use of an analytical moment-matching GP method, namely, the GaGP method, for user positioning. GaGP learns from the statistical properties of the noise present in the RSS inputs to derive realistic 2σ error-bars on the estimated locations. We also propose a new GP method, namely RecGaGP, which achieves lower prediction error than both the conventional GP and GaGP methods. Moreover, we make use of the uplink RSS for user positioning and are thus able to offload the computational cost of location prediction to the BS.

C. USER POSITIONING IN MASSIVE MIMO

Recent works have investigated the problem of positioning users from their radio signal information in massive MIMO systems. In [22], signals received at multiple massive MIMO BSs are directly used to estimate the user locations via compressed sensing. A convex search space is first obtained from coarse TOA estimates at each BS and then, an optimization problem is solved over this search space to obtain the location estimates. The works in [23] and [24] use AoA information at the BSs to position users, while [25] jointly uses the time delay, angle of departure (AoD), and AoA information to position a user. In [26], necessary conditions are derived to position users in a millimeter wave massive MIMO system from the AoD and AoA information of uplink LoS signals. The conventional GP method is employed in [5] to position users from their uplink RSS in a distributed massive MIMO setup. The method in [5] considers noisy RSS for both training and prediction, whereas in this work, we consider noise-free training RSS and noisy test RSS. A reconstruction GP method was proposed in [27] to achieve lower RMSE performance than the conventional GP method, but similar to the conventional GP method, the method in [27] provides unrealistically small 2σ error-bars on the estimated user locations.

Realistic 2σ error-bars are derived in [28] using a numerical approximation technique which does not lend itself to any analytical insights on the derived location estimates and their 2σ error-bars. We improve upon [28] in two major ways. First, by employing GaGP, we take an analytical approximation approach to estimate the test user locations and their 2σ error-bars. The derived expressions lend themselves to the following analytical insights: (i) the mean and variance expressions in GaGP are similar in structure to those from the conventional GP and (ii) for the special case of the test RSS being noise-free, the mean and variance expressions for the GaGP and the conventional GP methods turn out to be the same. Second, the proposed RecGaGP method achieves lower RMSE performance than the method in [28], while also providing realistic 2σ error-bars on the estimated locations.

D. GP METHODS WITH NOISY INPUTS

The GP method proposed recently in [29]–[36] and references therein have considered noisy inputs, but for both the

training and prediction phases. The GP methods proposed in [29]–[31] provide realistic 2σ error-bars on time series data. These methods have been extended in [32]–[35] for channel prediction in wireless networks. Closed-form expressions are derived in [29] and [30] for the mean and variance of the predicted parameter when the true predictive distribution is analytically approximated as a Gaussian and the GP covariance function is modeled as a squared exponential.

However, for location prediction with uplink RSS in distributed massive MIMO, ours is the first work to employ an analytical moment-matching based GP method, namely, the GaGP, to derive and approximate the true predictive distribution as a Gaussian. When GaGP is employed and the GP covariance function is modeled as a weighted sum of squared exponential, inner product, and delta terms, we derive analytical closed-form expressions for the location estimates and their realistic 2σ error-bars. Moreover, we propose a new GP method, namely, the RecGaGP, which achieves lower RMSE than the conventional GP and the GaGP methods.

General Notation: Scalars, vectors, and matrices are denoted using regular font small letters (e.g., a), boldface small letters (e.g., $\mathbf{a}$), and boldface capital letters for matrices (e.g. $\mathbf{A}$), respectively. Element i in vector $\mathbf{a}$ is referred to as $[\mathbf{a}]_i$, column i in matrix $\mathbf{A}$ as $[\mathbf{A}]_i$, and the element in row i and column j of $\mathbf{A}$ as $[\mathbf{A}]_{ij}$, respectively. We use the overhead symbols $\widetilde{(.)}$ and $\widehat{(.)}$ to refer to the training data and test data, respectively. An extra superscript $(.)^*$ is added to the overhead symbols to refer to the noise-free components in the data. The notations $\nabla_{\mathbf{a}}(.)$ and $\nabla^2_{\mathbf{a}}(.)$ refer to the gradient and the Hessian with respect to the vector $\mathbf{a}$. Also, $\nabla_{[\mathbf{a}]_i}(.)$ refers to the partial derivative with respect to the element i of vector $\mathbf{a}$. The symbol $\approx$ denotes approximation of the p.d.f. The trace of the matrix $\mathbf{A}$ is denoted as $\mathrm{Tr}(\mathbf{A})$. Tables 1-2 present additional notation pertaining to the system model and the GP methods.

TABLE 1. Mathematical notations: System model.

Symbol	Meaning
M, m	Number of RRHs, RRH index
K, k	Number of scheduled users, user index
$\boldsymbol{\omega}_k$	Symbol vector transmitted by user k
ρ	Uplink transmission power of each user
$\mathbf{r}_m$	Received symbol vector at RRH m (eq. (1))
$\boldsymbol{\vartheta}_m$	Additive white Gaussian noise in $\mathbf{r}_m$
$\sigma^2_{\vartheta}\mathbf{I}$	Covariance of $\boldsymbol{\vartheta}_m$
h_{mk}	Flat-fading uplink channel between user k and RRH m (eq. (2))
q_{mk},	Small-scale fading coefficient in h_{mk}
g_{mk}	Large-scale fading coefficient in h_{mk}
d_{mk}	Distance between user k and RRH m
η	Path-loss exponent
b_0	Path-loss at a reference distance d_0
z_{mk}	Shadowing noise term in h_{mk}
σ^2_z	Variance of the shadowing noise z_{mk}
p_{mk}	RSS of user k at RRH m
$\mathbf{p}_k$	Uplink RSS vector of user k (eq. (5))
x_k, y_k	2D location coordinates (x, y) of user k
f_x, f_y	Functions which map RSS vectors of users to their x and y coordinates (eq. (6))
$\widetilde{L}$	Number of training locations
$\widetilde{\mathbf{x}}$	$\widetilde{L} \times 1$ vector of training $x-$coordinates
$\widetilde{\mathbf{P}}$	Training RSS matrix (size $\widetilde{L} \times M$) whose rows are the noise-free RSS vectors of users with $x-$coordinates in $\widetilde{\mathbf{x}}$ (eq. (10))
$\widetilde{\mathbf{p}}_l$	Noise-free training RSS vector at row l in $\widetilde{\mathbf{P}}$ (corresponds to the $x-$coordinate $[\widetilde{\mathbf{x}}]_l$)
$\widehat{L}$	Number of test points
$\widehat{\mathbf{x}}$	$\widehat{L} \times 1$ vector of test $x-$coordinates
$\widehat{\mathbf{P}}$	Test RSS matrix of size $\widehat{L} \times M$ with rows being the noisy RSS vectors of test users whose $x-$coordinates are in $\widehat{\mathbf{x}}$ (eq. (13))
$\widehat{\mathbf{p}}_l$	Noisy test RSS vector at row l in $\widehat{\mathbf{P}}$ (corresponds to the test $x-$coordinate $[\widehat{\mathbf{x}}]_l$)
$\widehat{\mathbf{p}}^*_l$	Noise-free term in the test RSS $\widehat{\mathbf{p}}_l$ (eq. (15))
$\widehat{\mathbf{z}}_l$	Shadowing noise term in $\widehat{\mathbf{p}}_l$ (eq. (15))
$\widehat{\boldsymbol{\Sigma}}_l$	Covariance of $\widehat{\mathbf{z}}_l$
$\mathbf{a} \sim \mathcal{N}(\mathbf{u}, \mathbf{A})$	A random vector $\mathbf{a}$ that is Gaussian distributed with mean $\mathbf{u}$ and covariance $\mathbf{A}$
$\mathcal{N}(\mathbf{a}; \mathbf{u}, \mathbf{A})$	The p.d.f of a Gaussian random vector $\mathbf{a}$ with mean $\mathbf{u}$ and covariance $\mathbf{A}$
$N(\mathbf{a}; \mathbf{u}, \mathbf{A})$	Shorthand for the expression $\{(2\pi)^{-n/2} \lvert\mathbf{A}\rvert^{-1/2} e^{-\frac{1}{2}(\mathbf{a}-\mathbf{u})^T\mathbf{A}^{-1}(\mathbf{a}-\mathbf{u})}\}$, when all of $\mathbf{u}$, $\mathbf{a}$, and $\mathbf{A}$ are deterministic and are of size $n \times 1$, $n \times 1$ and $n \times n$ respectively.

III. SYSTEM DESCRIPTION

We study user positioning in a distributed multiuser massive MIMO system comprised of M single-antenna remote radio heads (RRHs) which serve K single-antenna users (UEs) simultaneously on the same time-frequency resource. High-speed fronthaul links connect the RRHs to a central computing unit (CU) (c.f. Fig. 1). The CU collects the RSS values from each RRH and forms an $M \times 1$ RSS vector, whenever the users transmit on the uplink. The CU is also equipped with a machine learning model which takes the uplink RSS vectors as input and provides the location coordinates of the K transmitting users as the output.

A. UPLINK TRANSMISSIONS AND CHANNEL MODEL

Let $\boldsymbol{\omega}_k$ be the symbol vector transmitted by the user k, with a transmission power of ρ. When h_{mk} is the flat-fading channel gain between user k and RRH m, the sum symbol vector $\mathbf{r}_m$ received at the RRH m is given by

$$\mathbf{r}_m = \sqrt{\rho}\sum_{k=1}^{K} h_{mk}\boldsymbol{\omega}_k + \boldsymbol{\vartheta}_m,$$

where

$$h_{mk} = q_{mk}\sqrt{g_{mk}}. \tag{1}$$

TABLE 2. Mathematical notations: GP methods.

Symbol	Meaning
$\phi(.,.)$	Function to model the covariance between $x-$coordinates of users k and k' in terms of their RSS vectors $\mathbf{p}_k$ and $\mathbf{p}_{k'}$ (eq. (7)).
σ_{er}^2	$x-$coordinate measurement error variance.
$\delta_{kk'}$	Delta function $\{=1$ if $k=k'$, 0 otherwise.
α, γ	Free parameters in the covariance function
$\mathbf{B}$	$\phi(.,.)$ (eq. (7)).
α'	Parameter introduced in place of α in the function $\phi(.,.)$ for notational ease (eq. (8)).
$\boldsymbol{\theta}$	Vector comprising all the free parameters in the covariance function $\phi(.,.)$ (eq. (9)).
$\bar{\boldsymbol{\theta}}$	Learned vector $\boldsymbol{\theta}$ after GP training (eq. (11)).
$\widetilde{\boldsymbol{\Phi}}$	Matrix of evaluations of $\phi(.,.)$ between the training RSS vectors in $\widetilde{\mathbf{P}}$ (c.f. (12)).
M_0	Number of principal components of $\widetilde{\mathbf{P}}$ used to reconstruct $\widehat{\mathbf{P}}$ (eq. (21)).
$\mathbf{V}^{[M_0]}$	$M \times M_0$ matrix whose columns are the first M_0 right singular vectors of $\widetilde{\mathbf{P}}$.
$\widehat{\mathbf{P}}^{(\text{rec})}$	Reconstructed form of the test RSS matrix $\widehat{\mathbf{P}}$.
$\widehat{\mathbf{p}}_l^{(\text{rec})}$	Row l in $\widehat{\mathbf{P}}^{(\text{rec})}$, i.e., the reconstructed form of test RSS vector $\widehat{\mathbf{p}}_l$.
$\widehat{\mathbf{p}}_l^{(\text{rec})*}$	Noise-free component in $\widehat{\mathbf{p}}_l^{(\text{rec})}$ (eq. (22)).
$\widehat{\mathbf{z}}_l^{(\text{rec})}$	Shadowing noise in $\widehat{\mathbf{p}}_l^{(\text{rec})}$(eq. (22)).
$\widehat{\boldsymbol{\Sigma}}_l^{(\text{rec})}$	Covariance of $\widehat{\mathbf{z}}_l^{(\text{rec})}$ (eq. (23)).
$\widehat{\boldsymbol{\mu}}_x^{(\text{CGP})}$, $\widehat{\boldsymbol{\mu}}_x^{(\text{GaGP})}$, $\widehat{\boldsymbol{\mu}}_x^{(\text{RGP})}$	Estimates of the test $x-$coordinate vector $\widehat{\mathbf{x}}$, as given by the CGP, GaGP, and RecGaGP methods respectively (eq. (14), (19), and (24)).
$\widehat{\mathbf{C}}_x^{(\text{CGP})}$, $\widehat{\mathbf{C}}_x^{(\text{GaGP})}$, $\widehat{\mathbf{C}}_x^{(\text{RGP})}$,	Predictive variances around the estimates $\widehat{\boldsymbol{\mu}}_x^{(\text{CGP})}$, $\widehat{\boldsymbol{\mu}}_x^{(\text{GaGP})}$, and $\widehat{\boldsymbol{\mu}}_x^{(\text{RGP})}$ of the test vector $\widehat{\mathbf{x}}$, as given by CGP, GaGP, and RecGaGP methods respectively (eq. (14), (19), and (24)).
$\widehat{\boldsymbol{\mu}}_x^{(.)}$	Estimate of the test $x-$coordinate vector $\widehat{\mathbf{x}}$, as given by any of the studied GP methods.
$\widehat{\boldsymbol{\mu}}_y^{(.)}$	Estimate of the test $y-$coordinate vector $\widehat{\mathbf{y}}$, as given by any of the studied GP methods.
$\widehat{\mathbf{C}}_x^{(.)}$	Predictive variance around the estimate $\widehat{\boldsymbol{\mu}}_x^{(.)}$, as given by the chosen GP method.
$\widehat{\mathbf{C}}_y^{(.)}$	Predictive variance around the estimate $\widehat{\boldsymbol{\mu}}_y^{(.)}$, as given by the chosen GP method.
$\boldsymbol{\psi}$, $\boldsymbol{\Upsilon}$, $\boldsymbol{\xi}$, $\boldsymbol{\Upsilon}^{(\text{rec})}$	Matrices introduced to ease notation in the expressions for $\widehat{\boldsymbol{\mu}}_x^{(\text{CGP})}$ (eq. (14)), $\widehat{\mathbf{C}}_x^{(\text{GaGP})}$ (eq. (20)), and $\widehat{\mathbf{C}}_x^{(\text{RGP})}$ (eq. (25)) respectively.
λ_i, λ_{ij} $\boldsymbol{\nu}_i$, $\boldsymbol{\Gamma}$	Latent terms introduced to express $\widehat{\boldsymbol{\mu}}_x^{(\text{GaGP})}$ and $\widehat{\mathbf{C}}_x^{(\text{GaGP})}$ in terms of $\widehat{\boldsymbol{\mu}}_x^{(\text{CGP})}$ and $\widehat{\mathbf{C}}_x^{(\text{CGP})}$ (c.f. (38) and (40))

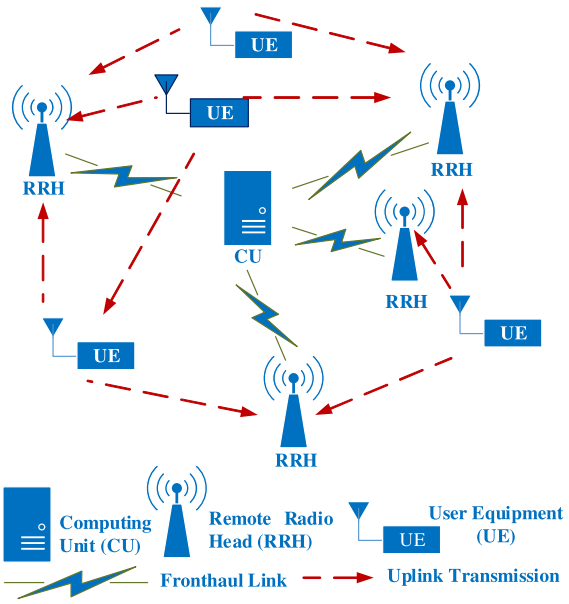


FIGURE 1. System model for location prediction in distributed massive MIMO: during any scheduled timeslot, the M RRHs receive uplink transmissions from K users on the same time-frequency resource. The high-speed fronthaul links forward the RSS values recorded at each RRH to the central unit, which then extracts the per-user RSS values and forms an $M \times 1$ RSS vector for each user. A supervised machine learning model, hosted at the CU, takes the RSS vector of each user as input and yields the user's location coordinates as output.

In (1), q_{mk} and g_{mk} are the small-scale and large-scale fading coefficients and $\vartheta_m \sim \mathcal{N}(\mathbf{0}, \sigma_\vartheta^2 \mathbf{I})$ represents the additive white Gaussian noise. Let η be the path-loss exponent, d_{mk} be the distance between user k and RRH m, b_0 be the path-loss at a reference distance d_0, z_{mk} be the log-normal shadowing noise coefficient, and σ_z^2 be the shadowing noise variance. We then model the large-scale fading coefficients g_{mk} as

$$g_{mk} = b_0 \, d_{mk}^{-\eta} 10^{\frac{z_{mk}}{10}}, \quad \text{where } z_{mk} \sim \mathcal{N}(0, \sigma_z^2). \tag{2}$$

The small-scale fading coefficients q_{mk} are modeled as independent and identically distributed (i.i.d) complex Gaussian variables with zero mean and unit variance, i.e., $q_{mk} \sim \mathcal{CN}(0, 1)$.

B. PRE-PROCESSING MULTIUSER RSS FOR PER-USER RSS VALUES

From (1), we note that the RSS $||\mathbf{r}_m||^2$ at RRH m corresponds to the multiuser RSS because the received vector $\mathbf{r}_m$ is the sum of symbol vectors received from all the K users. We cannot directly use the multiuser RSS $||\mathbf{r}_m||^2$ to position any given user k because we would then be unable to distinguish among the K users that are transmitting simultaneously. Instead, the RRH m should extract the per-user RSS p_{mk} of each user k from $\mathbf{r}_m$ and use it for positioning the user k. This can be done if the symbol vectors $\{\boldsymbol{\omega}_k\}$ in (1) are mutually orthogonal and are already known at the RRH. For example, $\{\boldsymbol{\omega}_k\}$ can be mutually orthogonal pilot vectors transmitted during the channel estimation phase [38]. The RRH m can then project

its received vector $\mathbf{r}_m$ onto the pilot vector $\boldsymbol{\omega}_k$ of user k to obtain $\boldsymbol{\omega}_k^H \mathbf{r}_m = \sqrt{\rho} h_{mk} + \boldsymbol{\omega}_k^H \boldsymbol{\vartheta}_m$, which only contains the received component from user k and an additional noise term. By setting a sensitivity threshold to distinguish between the signal and noise components, the RSS p_{mk} of user k is obtained from $\boldsymbol{\omega}_k^H \mathbf{r}_m$ as

$$p_{mk} = \rho g_{mk} |q_{mk}|^2. \tag{3}$$

Observe from (3) that the extracted per-user RSS values can be noisy due to small-scale fading and shadowing effects of the wireless channel. The small-scale fading effects are assumed to be averaged out over multiple time slots.[1] In contrast, the shadowing effect is assumed to exist because spatial averaging, which requires *a priori* access to the user location, needs to be employed to mitigate it. Taking these assumptions into account, the RSS obtained from (2) and (3), when converted to dB scale, is given by

$$p_{mk}^{\text{dB}} = p_0^{\text{dB}} - 10\eta \log_{10}(d_{mk}) + z_{mk}, \tag{4}$$

where $p_0^{\text{dB}} = 10 \log_{10}(\rho b_0)$ is the uplink RSS at the reference distance d_0. Once the per-user RSS values p_{mk}, $\forall m = 1, \ldots, M$, and $k = 1, \ldots, K$, are extracted as above, the CU accumulates the M RSS values of each user k into an $M \times 1$ uplink RSS vector $\mathbf{p}_k$ such that $[\mathbf{p}_k]_m = p_{mk}^{\text{dB}}$, i.e.,

$$\mathbf{p}_k = [p_{1k}^{\text{dB}} \quad p_{2k}^{\text{dB}} \quad \ldots \quad p_{Mk}^{\text{dB}}]^T. \tag{5}$$

C. MACHINE LEARNING SETUP

1) MATHEMATICAL MODEL

We wish to learn the functions $f_x(.)$ and $f_y(.)$ which take the uplink RSS vector $\mathbf{p}_k$ of a given user k as input and provide the user's location coordinates x_k and y_k as output respectively, i.e.,

$$x_k = f_x(\mathbf{p}_k) \quad \text{and } y_k = f_y(\mathbf{p}_k) \quad \forall x_k, y_k. \tag{6}$$

We follow a supervised machine learning approach, with a training phase and a test phase, to learn $f_x(.)$ and $f_y(.)$. During the training phase, a machine learning model is trained with a database comprising uplink RSS vectors at several known user locations. During the test phase, we feed as input to the trained machine learning model, the RSS vector of a test user whose location coordinates are unknown. The trained model provides as output, an estimate of the test user's location coordinates, along with the associated 2σ error-bars. Unlike previous works [5], we consider noise-free RSS vectors for training purposes because they are easy to generate. For this, we would only need prior knowledge of the RRH locations, the training user locations, the uplink transmission power ρ, and the path loss exponent η. In contrast, the test RSS vectors are assumed to contain shadowing noise because we are unable to employ spatial averaging on the test RSS vectors (this is in turn, because we do not have knowledge of the test users' locations).

[1]Fading due to signal self-interference may be space-dependent, but it can be averaged out over multiple subcarriers [4].

First, we employ the Gaussian approximation GP (GaGP) method as the machine learning method to estimate the test user locations and their 2σ error-bars. Next, we build on the underlying principles of the GaGP method to propose a new GP method, namely, the reconstruction-cum-Gaussian approximation GP (RecGaGP) method. Both the GP methods follow the same procedure in the training phase, but are different in their approaches in the test phase. We will now present the details on the training phase with focus on the $x-$coordinates.[2]

2) TRAINING PHASE

Both the GP methods under study are built on the standard assumption in Gaussian process regression [37]. That is, any finite set of realizations of the function to be learned (e.g. $f_x(.)$ in the case of $x-$coordinates) is Gaussian distributed with mean zero and covariance matrix $\boldsymbol{\Phi}$, the entries of which are given by a user-defined function $\phi(.,.)$. We model $\phi(.,.)$ as a weighted-sum of squared-exponential (SE), inner product (IP) and delta terms, defined for any two users k and k' with RSS vectors $\mathbf{p}_k$ and $\mathbf{p}_{k'}$, respectively, as

$$\phi(\mathbf{p}_k, \mathbf{p}_{k'}) = \alpha e^{-\frac{1}{2}(\mathbf{p}_k - \mathbf{p}_{k'})^T \mathbf{B}^{-1} (\mathbf{p}_k - \mathbf{p}_{k'})} + \gamma \mathbf{p}_k^T \mathbf{p}_{k'} + \sigma_{er}^2 \delta_{kk'},$$

where $\mathbf{B} = \text{diag}\{\beta_m\}$, $m = 1, \ldots, M$, and

$$\delta_{kk'} = \{1 \text{ if } k = k', \ 0 \text{ if otherwise}\}. \tag{7}$$

The SE term $\alpha e^{-\frac{1}{2}(\mathbf{p}_k - \mathbf{p}_{k'})^T \mathbf{B}^{-1} (\mathbf{p}_k - \mathbf{p}_{k'})}$ serves as the stationary component because it captures the correlation between the $x-$coordinates of the users k and k' as a function of the distance between their RSS vectors. The IP term $\gamma \mathbf{p}_k^T \mathbf{p}_{k'}$ serves as the non-stationary component because it captures the correlation among $x-$coordinates as a function of the actual RSS vectors. The delta term $\sigma_{er}^2 \delta_{kk'}$ serves as the measurement error component. For notational ease, we transform the covariance model in (7) as follows:

$$\phi(\mathbf{p}_k, \mathbf{p}_{k'}) = \alpha' N(\mathbf{p}_k; \mathbf{p}_{k'}, \mathbf{B}) + \gamma \mathbf{p}_k^T \mathbf{p}_{k'} + \sigma_{er}^2 \delta_{kk'}, \quad \forall \mathbf{p}_k, \mathbf{p}_{k'}, \tag{8}$$

where we introduce a new parameter $\alpha' = \alpha (2\pi)^{M/2} |\mathbf{B}|^{1/2}$ to convert the SE term in (7) into a Gaussian term.

The free parameters $\alpha, \beta_1, \ldots, \beta_M$, and γ introduced by the covariance model in (7) are to be learned during the training phase. We accumulate them into an $(M+2) \times 1$ vector $\boldsymbol{\theta}$ defined such that

$$\boldsymbol{\theta} = [\alpha \quad \beta_1 \ldots \beta_M \quad \gamma]^T. \tag{9}$$

In order to learn the vector $\boldsymbol{\theta}$, we specify the training database as follows. Let there be a total of $\widetilde{L}$ training user locations, whose $x-$coordinates are accumulated into an $\widetilde{L} \times 1$ vector $\widetilde{\mathbf{x}}$

[2]The proposed methods can be extended in a straightforward manner for the $y-$coordinates as well.

and their noise-free training RSS vectors into an $\widetilde{L} \times M$ matrix $\widetilde{\mathbf{P}}$, defined such that

$$\begin{aligned} \widetilde{\mathbf{x}} &= [\widetilde{x}_1 \quad \widetilde{x}_2 \quad \dots \quad \widetilde{x}_{\widetilde{L}}]^T, \\ \widetilde{\mathbf{P}} &= [\widetilde{\mathbf{p}}_1 \quad \widetilde{\mathbf{p}}_2 \quad \dots \quad \widetilde{\mathbf{p}}_{\widetilde{L}}]^T. \end{aligned} \tag{10}$$

Note from (10) that the RSS vector $\widetilde{\mathbf{p}}_l$ in $\widetilde{\mathbf{P}}$ corresponds to the training $x-$coordinate $\widetilde{x}_l$ in $\widetilde{\mathbf{x}}$, $\forall\, l = 1, \dots, \widetilde{L}$. The vector $\boldsymbol{\theta}$ can then be obtained via maximum-likelihood of the training $x-$coordinate vector $\widetilde{\mathbf{x}}$ as

$$\begin{aligned} \bar{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} \log(p(\widetilde{\mathbf{x}}|\widetilde{\mathbf{P}}, \boldsymbol{\theta})), \\ &\overset{(a)}{=} \arg\max_{\boldsymbol{\theta}} \log(N(\widetilde{\mathbf{x}}; \mathbf{0}, \widetilde{\boldsymbol{\Phi}})), \end{aligned} \tag{11}$$

where $\bar{\boldsymbol{\theta}}$ is the learned parameter vector, (a) follows from the standard GP assumption, i.e., the training $x-$coordinates are jointly Gaussian with mean zero and covariance $\widetilde{\boldsymbol{\Phi}}$, whose elements are given by

$$[\widetilde{\boldsymbol{\Phi}}]_{ll'} = \phi(\widetilde{\mathbf{p}}_l, \widetilde{\mathbf{p}}_{l'}), \quad \forall\, l, l' = 1, \dots, \widetilde{L}. \tag{12}$$

The optimization problem in (11), to be solved during the training phase, is well-known to be non-convex in the GP literature [18]–[21], [37]. Nevertheless, we can obtain the first-order gradients with respect to $\boldsymbol{\theta}$ in a closed-form and can, therefore, employ gradient ascent methods, such as the conjugate gradient [41], to obtain a local optimum. The training phase is complete upon obtaining the optimum vector $\bar{\boldsymbol{\theta}}$.

3) PREDICTION PHASE

Let us say that we need to predict the locations of $\widehat{L}$ test users from their noisy RSS vectors. The test users' $x-$coordinates are accumulated into an $\widehat{L} \times 1$ vector $\widehat{\mathbf{x}}$ and their RSS vectors are accumulated into an $\widehat{L} \times M$ matrix $\widehat{\mathbf{P}}$ such that

$$\begin{aligned} \widehat{\mathbf{P}} &= [\widehat{\mathbf{p}}_1 \quad \widehat{\mathbf{p}}_2 \quad \dots \quad \widehat{\mathbf{p}}_{\widehat{L}}]^T, \\ \widehat{\mathbf{x}} &= [\widehat{x}_1 \quad \widehat{x}_2 \quad \dots \quad \widehat{x}_{\widehat{L}}]^T. \end{aligned} \tag{13}$$

Note from (13) that the test RSS vector $\widehat{\mathbf{p}}_l$ belongs to the test user l with $x-$coordinate $[\widehat{\mathbf{x}}]_l = \widehat{x}_l, \forall = 1, \dots, \widehat{L}$.

The conventional GP method [28], [37] naively treats the noisy test RSS vectors as noise-free and makes use of the standard GP assumption to give the following predictive distribution for the test $x-$coordinate $[\widehat{\mathbf{x}}]_l$

$$[\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l \sim \mathcal{N}([\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l, [\widehat{\boldsymbol{C}}_x^{(\mathrm{CGP})}]_{ll}),$$

where

$$\begin{aligned} [\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l &= \sum_{i=1}^{\widetilde{L}} \phi(\widehat{\mathbf{p}}_l, \widetilde{\mathbf{p}}_i)[\widetilde{\boldsymbol{\Phi}}^{-1}\widetilde{\mathbf{x}}]_i, \\ [\widehat{\boldsymbol{C}}_x^{(\mathrm{CGP})}]_{ll} &= \phi(\widehat{\mathbf{p}}_l, \widehat{\mathbf{p}}_l) - \sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}} \phi(\widehat{\mathbf{p}}_l, \widetilde{\mathbf{p}}_i)[(\widetilde{\boldsymbol{\Phi}})^{-1}]_{ij}\phi(\widetilde{\mathbf{p}}_j, \widehat{\mathbf{p}}_l). \end{aligned} \tag{14}$$

Since the predictive distribution in (14) is Gaussian, the term $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l$ denotes the maximum-a-posteriori (MAP) estimate of the test $x-$coordinate $[\widehat{\mathbf{x}}]_l$. The term $[\widehat{\boldsymbol{C}}_x^{(\mathrm{CGP})}]_{ll}$ denotes the associated variance, with $\pm 2\sqrt{[\widehat{\boldsymbol{C}}_x^{(\mathrm{CGP})}]_{ll}}$ being the 2σ error-bars on viewing $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l$ as the estimate of $[\widehat{\mathbf{x}}]_l$.

The conventional GP method provides unrealistically small 2σ error-bars on the estimated locations because it naively treats the noisy test RSS data as noise-free. To address this concern, we now present two GP methods, namely, the GaGP and the RecGaGP methods. Both the methods provide realistic 2σ error-bars on the estimated locations by accounting for the noisy nature of the test RSS data.

IV. LOCATION PREDICTION WITH GAUSSIAN APPROXIMATION GP (GaGP) METHOD

The Gaussian approximation GP (GaGP) method is a GP method based on analytical moment-matching. For each test user l, GaGP firstly derives the true predictive distribution $p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l)$ by taking the noisy nature of the test RSS $\widehat{\mathbf{p}}_l$ into account. The derived predictive distribution cannot be expressed in closed-form and is therefore approximated analytically as Gaussian with the same first and second order moments. Specifics are given below.

We begin with the observation from (4) that any noisy RSS value recorded at the RRHs can be expressed as the sum of a noise-free component and a shadowing noise component. Any noisy test RSS vector $\widehat{\mathbf{p}}_l$ can, therefore, be expressed as

$$\widehat{\mathbf{p}}_l = \widehat{\mathbf{p}}_l^* + \widehat{\mathbf{z}}_l, \quad \text{such that } \widehat{\mathbf{z}}_l \sim \mathcal{N}(\mathbf{0}, \widehat{\boldsymbol{\Sigma}}_l), \tag{15}$$

where $\widehat{\mathbf{p}}_l^*$ is the noise-free component in $\widehat{\mathbf{p}}_l$ and $\widehat{\mathbf{z}}_l$ is the shadowing noise. $\widehat{\boldsymbol{\Sigma}}_l$ is the covariance of $\widehat{\mathbf{z}}_l$. We assume for simplicity that the shadowing noise terms in the M uplink channels of the test user l are mutually independent and that their variances are known at the CU through prior propagation studies. In other words, we assume that the shadowing covariance matrix $\widehat{\boldsymbol{\Sigma}}_l$ is diagonal in nature and that its M diagonal elements are known to the CU. We can then use (15) to express the conditional distribution of the noise-free component $\widehat{\mathbf{p}}_l^*$ as

$$\widehat{\mathbf{p}}_l^*|\widehat{\mathbf{p}}_l, \widehat{\boldsymbol{\Sigma}}_l \sim \mathcal{N}(\widehat{\mathbf{p}}_l, \widehat{\boldsymbol{\Sigma}}_l). \tag{16}$$

The GaGP method first treats $\widehat{\mathbf{p}}_l^*$ as a latent variable and makes use of the conventional GP method (c.f. (14)) to obtain an initial estimate of the test user location $[\widehat{\mathbf{x}}]_l$ in terms of $\widehat{\mathbf{p}}_l^*$. The conditional distribution of the latent variable $\widehat{\mathbf{p}}_l^*$, obtained from (16), is then used to obtain the true predictive distribution of $[\widehat{\mathbf{x}}]_l$ in terms of the noisy test RSS $\widehat{\mathbf{p}}_l$ as follows[3]:

$$p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l) = \int p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l^*)p(\widehat{\mathbf{p}}_l^*|\widehat{\mathbf{p}}_l, \widehat{\boldsymbol{\Sigma}}_l)d\widehat{\mathbf{p}}_l^*. \tag{17}$$

In (17), the term $p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l^*)$ is obtained from (14) and the term $p(\widehat{\mathbf{p}}_l^*|\widehat{\mathbf{p}}_l, \widehat{\boldsymbol{\Sigma}}_l)$ from (16), respectively.

[3]For notational convenience, all the integrals from here on are expressed as indefinite integrals. In reality, all the integrals are definite over appropriate sets.

Remark 1: The predictive distribution $p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l)$ *in (17) is non-Gaussian and cannot be obtained in closed-form because the integral on the right hand side is intractable.*

Proof: See **Appendix A**. □

As a consequence of **Remark 1**, we can only obtain an approximation to the predictive distribution $p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l)$, using either numerical or analytical approximation procedures. While the numerical approximation is possible (c.f. [6] for an example based on Monte-Carlo sampling), we take an analytical approximation approach here because it allows us to obtain analytical insights on how the resulting mean and variance compare against those obtained from the conventional GP method.

The GaGP method analytically approximates the true predictive distribution $p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l)$ in (17) as a Gaussian distribution with the same first and second order moments, as follows:

$$p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l) \approx \mathcal{N}([\widehat{\mathbf{x}}]_l;[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l,[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}),$$

where

$$[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l = \mathbb{E}_{[\widehat{\mathbf{x}}]_l}([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l)$$

and

$$[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll} = \mathbb{E}_{[\widehat{\mathbf{x}}]_l}(([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l)^2) - ([\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l)^2. \quad (18)$$

In (18), $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ and $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$ are the estimated mean and variance of the test $x-$coordinate $[\widehat{\mathbf{x}}]_l$.

Previous works on time-series forecasting [29], [30] and system identification [31] have derived closed-form expressions for the GaGP mean and variance when the GP covariance function $\phi(.,.)$ is modeled as a squared exponential. However, the expressions in [29]–[31] do not extend in a straightforward manner to the weighted-sum covariance model (7) used in our work. We make the necessary extensions here and provide closed-form expressions for the GaGP mean and variance, $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ and $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$, in **Lemma 1** below.

Lemma 1: When the weighted-sum covariance model given in (7) is employed, the GaGP method yields the following closed-form expressions for the predicted mean $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ *and variance* $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$ *of the* $x-$*coordinate* $[\widehat{\mathbf{x}}]_l$ *of the test user* l *with the RSS vector* $\widehat{\mathbf{p}}_l$*:*

$$[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l = \sum_{i=1}^{\widetilde{L}} \alpha'[\boldsymbol{\psi}]_i N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B}+\widehat{\boldsymbol{\Sigma}}_l) + \gamma[\boldsymbol{\psi}]_i\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i,$$

and

$$[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll} = \sigma_{er}^2 + \alpha + \gamma\widehat{\mathbf{p}}_l^T\widehat{\mathbf{p}}_l + \gamma\,\mathrm{Tr}(\widehat{\boldsymbol{\Sigma}}_l) - \sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}}[\boldsymbol{\xi}]_{ij} \times\Big\{(\alpha')^2 N(\widetilde{\mathbf{p}}_i;\widetilde{\mathbf{p}}_j,2\mathbf{B})N(\widehat{\mathbf{p}}_l;\frac{\widetilde{\mathbf{p}}_i+\widetilde{\mathbf{p}}_j}{2},\frac{\mathbf{B}}{2}+\widehat{\boldsymbol{\Sigma}}_l) + [\boldsymbol{\Upsilon}]_j^T\widetilde{\mathbf{p}}_i + (\widetilde{\mathbf{p}}_j)^T[\boldsymbol{\Upsilon}]_i + \gamma^2\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l + \gamma^2\,\mathrm{Tr}(\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\boldsymbol{\Sigma}}_l)\Big\} - ([\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l)^2, \quad (19)$$

respectively, where the vector $\boldsymbol{\psi}\in\mathbb{R}^{\widetilde{L}}$ *and the matrices* $\boldsymbol{\Upsilon}\in\mathbb{R}^{M\times\widetilde{L}}$ *and* $\boldsymbol{\xi}\in\mathbb{R}^{\widetilde{L}\times\widetilde{L}}$ *are defined such that*

$$\boldsymbol{\psi} = \widetilde{\boldsymbol{\Phi}}^{-1}\widetilde{\mathbf{x}},$$
$$[\boldsymbol{\Upsilon}]_i = \alpha'\gamma N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B}+\widehat{\boldsymbol{\Sigma}}_l) \times(\widehat{\boldsymbol{\Sigma}}_l\mathbf{B}^{-1}+\mathbf{I})^{-1}\widehat{\boldsymbol{\Sigma}}_l(\mathbf{B}^{-1}\widetilde{\mathbf{p}}_i+\widehat{\boldsymbol{\Sigma}}_l^{-1}\widehat{\mathbf{p}}_l),$$

and

$$[\boldsymbol{\xi}]_{ij} = [\widetilde{\boldsymbol{\Phi}}^{-1}]_{ij} - [\boldsymbol{\psi}]_i[\boldsymbol{\psi}]_j, \quad \forall i,j=1,\ldots,\widetilde{L}. \quad (20)$$

Proof: See **Appendix B**. □

Besides obtaining closed-form expressions for the predicted mean and variance of the test users' $x-$coordinates, we are also able to derive few analytical insights from these expressions, as summarized in **Remark 2** below.

Remark 2: Although not obvious from an initial observation, $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ *and* $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$ *obtained from (19) are similar in structure as the* $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l$ *and* $[\widehat{\mathbf{C}}_x^{(\mathrm{CGP})}]_{ll}$ *terms obtained from (14), but with several multiplicative and additive correction factors. The GaGP method introduces these correction factors so as to account for the stochastic nature of the test RSS* $\widehat{\mathbf{p}}_l$*. In addition, if the test RSS vectors are noise free, i.e., if* $\widehat{\boldsymbol{\Sigma}}_l = \mathbf{0}$*, we can verify that* $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ *and* $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$ *are exactly the same as* $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l$ *and* $[\widehat{\boldsymbol{C}}_x^{(\mathrm{CGP})}]_{ll}$*, respectively.*

Proof: See **Appendix C** and **Appendix D**. □

Observe from (17)-(19) that, unlike the conventional GP method, the GaGP method accounts for the noisy nature of the test RSS. GaGP handles the noise-free terms in the test RSS vectors as latent variables and integrates them out using the statistical properties of the shadowing noise in the test RSS. This allows the GaGP method to provide realistic 2σ error-bars on the estimated locations. Despite this improvement in the 2σ error-bars, we find through simulations in Section VII that the RMSE performance of the GaGP method is similar to that of the conventional GP method. Improvements in the RMSE may be possible if, in addition to learning from the noise present in test RSS, we reduce the amount of noise present. We make use of this idea to develop the RecGaGP method in the next section.

V. LOCATION PREDICTION WITH RECONSTRUCTION-CUM-GAUSSIAN APPROXIMATION GP (RecGaGP)

We now propose RecGaGP - a reconstruction-cum-Gaussian approximation GP method which (i) reconstructs the noisy test RSS vectors from a low-dimensional principal subspace of the noise-free training RSS, and (ii) applies the Gaussian approximation procedure followed in GaGP to the reconstructed test RSS vectors for estimating the test user locations. While the reconstruction step reduces the amount of noise in the test RSS vectors to lower the RMSE, the Gaussian approximation procedure learns from statistical properties of

the residual noise to derive realistic 2σ error-bars on the estimated locations.

We know from [27] that the noise-free uplink RSS in a distributed massive MIMO system spans a low-dimensional principal subspace. Since our training matrix $\widetilde{\mathbf{P}}$ comprises of noise-free uplink RSS vectors, we then know that $\widetilde{\mathbf{P}}$ spans a low-dimensional principal subspace. Keeping this property in mind, we can reconstruct the noisy test RSS vectors from a subspace spanned by the first M_0 ($M_0 \leq M$) principal components (PCs) of $\widetilde{\mathbf{P}}$ as follows [39], [40]:

$$\widehat{\mathbf{P}}^{(\text{rec})} = \widehat{\mathbf{P}}\mathbf{V}^{[M_0]}(\mathbf{V}^{[M_0]})^T, \tag{21}$$

where $\widehat{\mathbf{P}}^{(\text{rec})}$ is the reconstructed test RSS matrix, $\widehat{\mathbf{P}}$ is the original test RSS matrix, and $\mathbf{V}^{[M_0]}$ is a matrix whose columns are the first M_0 right singular vectors of $\widetilde{\mathbf{P}}$. Since the reconstruction step in (21) is a linear algebraic operation, we can derive the statistical properties of the noise present in $\widehat{\mathbf{P}}^{(rec)}$ from that of the original test RSS matrix $\widehat{\mathbf{P}}$ as described below.

Observe from (21) that any reconstructed test RSS vector $\widehat{\mathbf{p}}_l^{rec}$ in $\widehat{\mathbf{P}}^{(rec)}$ can be expressed in terms of its original counterpart $\widehat{\mathbf{p}}_l$ in $\widehat{\mathbf{P}}$ as

$$\begin{aligned}\widehat{\mathbf{p}}_l^{(rec)} &= (\widehat{\mathbf{p}}_l^T\mathbf{V}^{[M_0]}(\mathbf{V}^{[M_0]})^T)^T \quad \forall l = 1,\ldots\widehat{L},\\ &= (\widehat{\mathbf{p}}_l^{*T}\mathbf{V}^{[M_0]}(\mathbf{V}^{[M_0]})^T)^T\\ &\quad + (\widehat{\mathbf{z}}_l^T\mathbf{V}^{[M_0]}(\mathbf{V}^{[M_0]})^T)^T, \text{ (from (15))}\\ &= \widehat{\mathbf{p}}_l^{(rec)*} + \widehat{\mathbf{z}}_l^{(rec)},\end{aligned} \tag{22}$$

where we have defined $\widehat{\mathbf{p}}_l^{(rec)*} = (\widehat{\mathbf{p}}_l^{*T}\mathbf{V}^{[M_0]}(\mathbf{V}^{[M_0]})^T)^T$ as the noise-free component and $\widehat{\mathbf{z}}_l^{(rec)} = (\widehat{\mathbf{z}}_l^T\mathbf{V}^{[M_0]}(\mathbf{V}^{[M_0]})^T)^T$ as the residual noise in $\widehat{\mathbf{p}}_l^{(rec)}$ respectively. Statistical properties of $\widehat{\mathbf{z}}_l^{(rec)}$ are given by **Lemma 2** below.

Lemma 2: The residual noise $\widehat{\mathbf{z}}_l^{(rec)}$ in the reconstructed RSS vector $\widehat{\mathbf{p}}_l^{(rec)}$ is Gaussian distributed with mean zero and covariance $\widehat{\mathbf{\Sigma}}_l^{(rec)}$, whose elements are given by

$$\begin{aligned}[\widehat{\mathbf{\Sigma}}_l^{(rec)}]_{ij} = &\sum_{m=1}^{M}[\widehat{\mathbf{\Sigma}}_l]_{mm}(\sum_{m'=1}^{M_0}[\mathbf{V}^{[M_0]}]_{mm'}[(\mathbf{V}^{[M_0]})^T]_{m'i})\\ &\times(\sum_{m'=1}^{M_0}[\mathbf{V}^{[M_0]}]_{mm'}[(\mathbf{V}^{[M_0]})^T]_{m'j})\\ &\forall i,j = 1,\ldots,\widehat{L}.\end{aligned} \tag{23}$$

Consequently, the noise-free component $\widehat{\mathbf{p}}_l^{(rec)*}$ in $\widehat{\mathbf{p}}_l^{(rec)}$ is conditionally Gaussian and is distributed as $(\widehat{\mathbf{p}}_l^{(rec)*}|\widehat{\mathbf{p}}_l^{(rec)}, \widehat{\mathbf{\Sigma}}_l) \sim \mathcal{N}(\widehat{\mathbf{p}}_l^{(rec)}, \widehat{\mathbf{\Sigma}}_l^{(rec)})$.

Proof: See **Appendix E**. □

Lemma 2 gives us the probability distribution of the residual noise present in the reconstructed test RSS vectors, along with the conditional distribution of the noise-free components in the reconstructed test RSS. In order to estimate the test user locations, we can use this statistical knowledge to apply the Gaussian approximation procedure followed in GaGP (c.f. (15), (18)) to the reconstructed test RSS vectors. Doing so gives us closed-form expressions for the predicted mean and variance of the test users' $x-$coordinates, as stated in **Theorem 1** below.

Theorem 1: Using the weighted-sum covariance model in (8), the RecGaGP method provides the following closed-form expressions for the predicted mean $[\widehat{\boldsymbol{\mu}}_x^{(\text{RGP})}]_l$ and variance $[\widehat{\boldsymbol{C}}_x^{(\text{RGP})}]_{ll}$ of the $x-$coordinate $[\widehat{\mathbf{x}}]_l$ of the test user n whose reconstructed RSS vector is $\widehat{\mathbf{p}}_l^{(rec)}$:

$$\begin{aligned}[\widehat{\boldsymbol{\mu}}_x^{(\text{RGP})}]_l = &\sum_{i=1}^{\widetilde{L}}\alpha'[\boldsymbol{\psi}]_i N(\widehat{\mathbf{p}}_l^{(rec)};\widetilde{\mathbf{p}}_i, \mathbf{B}+\widehat{\mathbf{\Sigma}}_l^{(rec)})\\ &+\gamma[\boldsymbol{\psi}]_i(\widehat{\mathbf{p}}_l^{(rec)})^T\widetilde{\mathbf{p}}_i,\end{aligned}$$

and

$$\begin{aligned}&[\widehat{\boldsymbol{C}}_x^{(\text{RGP})}]_{ll}\\ &= \sigma_{er}^2 + (\alpha + \gamma(\widehat{\mathbf{p}}_l^{(rec)})^T\widehat{\mathbf{p}}_l^{(rec)} + \gamma\,\text{Tr}(\widehat{\mathbf{\Sigma}}_l^{(rec)}))\\ &\quad - \sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}}[\boldsymbol{\xi}]_{ij}\Big\{(\alpha')^2\,N(\widetilde{\mathbf{p}}_i;\widetilde{\mathbf{p}}_j, 2\mathbf{B})N(\widehat{\mathbf{p}}_l^{(rec)};\frac{\widetilde{\mathbf{p}}_i+\widetilde{\mathbf{p}}_j}{2},\frac{\mathbf{B}}{2}\\ &\quad + \widehat{\mathbf{\Sigma}}_l^{(rec)}) + [\mathbf{\Upsilon}^{(rec)}]_j^T\widetilde{\mathbf{p}}_i + (\widetilde{\mathbf{p}}_j)^T[\mathbf{\Upsilon}^{(rec)}]_i + \gamma^2(\widehat{\mathbf{p}}_l^{(rec)})^T\widetilde{\mathbf{p}}_i\\ &\quad \times(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l^{(rec)} + \gamma^2\,\text{Tr}(\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{\Sigma}}_l^{(rec)})\Big\} - ([\widehat{\boldsymbol{\mu}}_x^{(\text{RGP})}]_l)^2,\end{aligned} \tag{24}$$

respectively, where the matrix $\mathbf{\Upsilon}^{(\text{rec})} \in \mathbb{R}^{M\times\widetilde{L}}$ is defined such that, $\forall i = 1,\ldots,\widetilde{L}$

$$\begin{aligned}[\mathbf{\Upsilon}^{(rec)}]_i = &\alpha'\gamma N(\widehat{\mathbf{p}}_l^{(rec)};\widetilde{\mathbf{p}}_i, \mathbf{B}+\widehat{\mathbf{\Sigma}}_l^{(rec)})\\ &\times(\mathbf{B}^{-1} + (\widehat{\mathbf{\Sigma}}_l^{(rec)})^{-1})^{-1}(\mathbf{B}^{-1}\widetilde{\mathbf{p}}_i\\ &+ (\widehat{\mathbf{\Sigma}}_l^{(rec)})^{-1}\widehat{\mathbf{p}}_l^{(rec)}).\end{aligned} \tag{25}$$

The vector $\boldsymbol{\psi}$ and the matrix $\boldsymbol{\xi}$ are the same as defined in (20).

Proof: (Sketch) We apply the Gaussian approximation procedure followed in (17) and (18) to predict the test user l's $x-$coordinate $[\widehat{\mathbf{x}}]_l$ from its reconstructed RSS vector $\widehat{\mathbf{p}}_l^{(rec)}$, as summarized here. We treat the noise-free component $\widehat{\mathbf{p}}_l^{(rec)*}$ in $\widehat{\mathbf{p}}_l^{(rec)}$ as a hidden variable and use (14) to obtain expressions for the mean and variance of $[\widehat{\mathbf{x}}]_l$ in terms of $\widehat{\mathbf{p}}_l^{(rec)*}$. The hidden variable $\widehat{\mathbf{p}}_l^{(rec)*}$ is then integrated out, using the conditional distribution $(\widehat{\mathbf{p}}_l^{(rec)*}|\widehat{\mathbf{p}}_l^{(rec)}, \widehat{\mathbf{\Sigma}}_l) \sim \mathcal{N}(\widehat{\mathbf{p}}_l^{(rec)}, \widehat{\mathbf{\Sigma}}_l^{(rec)})$ from **Lemma 2**, by following the steps (17) and (18), so as to obtain the closed-form expressions for $[\widehat{\boldsymbol{\mu}}_x^{(\text{RGP})}]_l$ and $[\widehat{\boldsymbol{C}}_x^{(\text{RGP})}]_{ll}$ in terms of $\widehat{\mathbf{p}}_l^{(rec)}$, as given by (24)-(25). Notice that $[\widehat{\boldsymbol{\mu}}_x^{(\text{RGP})}]_l$ and $[\widehat{\boldsymbol{C}}_x^{(\text{RGP})}]_{ll}$ in (24)-(25) are the same as given by GaGP in (19), but with $\widehat{\mathbf{p}}_l$ and $\widehat{\mathbf{\Sigma}}_l$ replaced by $\widehat{\mathbf{p}}_l^{(rec)}$ and $\widehat{\mathbf{\Sigma}}_l^{(rec)}$, respectively. This is because the RecGaGP method applies the Gaussian approximation technique (c.f. (17) and (18)) to the reconstructed test RSS $\widehat{\mathbf{p}}_l^{(rec)}$, while the GaGP method does so to the original test RSS $\widehat{\mathbf{p}}_l$, for location prediction. □

Location estimates obtained from the RecGaGP method, as given by **Theorem 1**, yield lower RMSE values than the

conventional GP and GaGP methods because the reconstruction step reduces the amount of noise present in the test RSS vectors. Also, the RecGaGP method provides realistic 2σ error-bars on the estimated locations because, similar to the GaGP method, the RecGaGP method learns from the noise present in the input test RSS vectors. Simulation studies in Section VII confirm this superior prediction performance of the RecGaGP method.

VI. COMPLEXITY ANALYSIS

We now investigate the computational cost associated with the presented GP methods. **A**ppendix H presents an overview of (i) how some frequently-occurring matrix operations in the presented GP methods can be implemented in a numerically stable manner, and (ii) the computational complexity of such numerically stable implementations. Below, we make use of the overview in **A**ppendix H to present a detailed analysis of the complexity of each GP method under study.

Before analyzing the complexity of the prediction phase of each GP method, we make the **Remark 3** below about predicting multiple test user locations simultaneously.

Remark 3: A common attribute of the proposed GP methods is that they allow us to predict multiple test user locations simultaneously, without affecting the localization accuracy for any given test user. Also, for cost savings, we can precompute certain terms and reuse them when calculating the location estimates in parallel.

Proof: See Appendix F. □

A. CONVENTIONAL GP METHOD

Observe from (14) that the predictive mean $\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}$ requires computation of (i) $\widetilde{\boldsymbol{\Phi}}^{-1}\widetilde{\mathbf{x}}$, which needs $\mathcal{O}(\widetilde{L}^3)$ operations to obtain the Cholesky factor of $\widetilde{\boldsymbol{\Phi}}$ and $\mathcal{O}(\widetilde{L}^2)$ operations to obtain the product $\widetilde{\boldsymbol{\Phi}}^{-1}\widetilde{\mathbf{x}}$, and (ii) $\widetilde{L}$ evaluations of the covariance function $\phi(\widehat{\mathbf{p}}_l, \widetilde{\mathbf{p}}_i)$ for each test user, amounting to $\mathcal{O}(\widehat{L}\widetilde{L}M)$ operations for the $\widehat{L}$ test users. In total, since we generally have $\widehat{L} << \widetilde{L}$ and $M << \widetilde{L}$, calculating $\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}$ incurs a complexity of $\mathcal{O}(\widetilde{L}^3)$.

To calculate the predictive variance $[\widehat{\mathbf{C}}_x^{(\mathrm{CGP})}]_{ll}$, $\forall l = 1, \ldots, \widehat{L}$, observe from (14) that we need to calculate $\sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}}\phi(\widehat{\mathbf{p}}_l, \widetilde{\mathbf{p}}_i)[(\widetilde{\boldsymbol{\Phi}})^{-1}]_{ij}\phi(\widetilde{\mathbf{p}}_j, \widehat{\mathbf{p}}_l)$. The $\phi(\widehat{\mathbf{p}}_l, \widetilde{\mathbf{p}}_i)$ terms are obtained in $\mathcal{O}(\widehat{L}M)$ operations. Since the matrix $\widetilde{\boldsymbol{\Phi}}$ and its Cholesky factor are already known from the $\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}$ calculation, the product $[(\widetilde{\boldsymbol{\Phi}})^{-1}]_{ij}\phi(\widetilde{\mathbf{p}}_j, \widehat{\mathbf{p}}_l)$ requires $\mathcal{O}(\widetilde{L}\widehat{L}M)$ operations to obtain the $\phi(\widetilde{\mathbf{p}}_j, \widehat{\mathbf{p}}_l)$ terms and $\mathcal{O}(\widetilde{L}^2\widehat{L})$ to obtain the product. Once these terms are available, evaluating the sum $\sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}}\phi(\widehat{\mathbf{p}}_l, \widetilde{\mathbf{p}}_i)[(\widetilde{\boldsymbol{\Phi}})^{-1}]_{ij}\phi(\widetilde{\mathbf{p}}_j, \widehat{\mathbf{p}}_l)$ requires another $\mathcal{O}(\widetilde{L}\widehat{L})$ operations. In total, since we generally have $\widehat{L} << \widetilde{L}$ and $M << \widetilde{L}$, calculating the $[\widehat{\mathbf{C}}_x^{(\mathrm{CGP})}]_{ll}$ for all the $\widehat{L}$ test users incurs complexity of $\mathcal{O}(\widetilde{L}^2\widehat{L})$.

B. GaGP METHOD

Observe from (19) that the calculation of $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ for the $\widehat{L}$ users requires one computation of $\boldsymbol{\psi} = \widetilde{\boldsymbol{\Phi}}^{-1}\widetilde{\mathbf{x}}$, $\widehat{L}\widetilde{L}$ computations of $\alpha' N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B} + \widehat{\boldsymbol{\Sigma}}_l)$ and $\widehat{L}\widetilde{L}$ computations of $\gamma\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i$. Calculating $\widetilde{\boldsymbol{\Phi}}^{-1}\widetilde{\mathbf{x}}$ requires $\mathcal{O}(\widetilde{L}^3)$ operations. The $\widehat{L}\widetilde{L}$ computations of $\alpha' N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B} + \widehat{\boldsymbol{\Sigma}}_l)$ require a total of $\mathcal{O}(\widehat{L}\widetilde{L}M)$ operations if the $\{\widehat{\boldsymbol{\Sigma}}_l\}$ are diagonal matrices and a total of $\mathcal{O}(\widehat{L}M^3 + \widehat{L}\widetilde{L}M^2)$ operations otherwise. The $\widehat{L}\widetilde{L}$ computations of $\gamma\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i$ require a total of $\mathcal{O}(\widehat{L}\widetilde{L}M)$ operations. Therefore, considering non-diagonal $\{\widehat{\boldsymbol{\Sigma}}_l\}$, $\widehat{L} << \widetilde{L}$, $M << \widetilde{L}$, and $\widehat{L}M^2 < \widetilde{L}^2$, the calculation of $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ for the $\widehat{L}$ users incurs a total of $\mathcal{O}(\widetilde{L}^3)$ complexity.

Next, observe from (19) that the calculation of $[\widehat{\mathbf{C}}_x^{(\mathrm{GaGP})}]_{ll}$ for the $\widehat{L}$ test users requires $\widehat{L}$ computations of $\mathrm{Tr}(\boldsymbol{\Sigma}_l)$ and $\widehat{\mathbf{p}}_l^T\widehat{\mathbf{p}}_l$, one computation of $\boldsymbol{\xi}$, $\widehat{L}^2$ computations of $N(\widetilde{\mathbf{p}}_i; \widetilde{\mathbf{p}}_j, 2\mathbf{B})$, $\widehat{L}\widetilde{L}^2$ computations of $N(\widehat{\mathbf{p}}_l; \frac{\widetilde{\mathbf{p}}_i+\widetilde{\mathbf{p}}_j}{2}, \frac{\mathbf{B}}{2} + \widehat{\boldsymbol{\Sigma}}_l)$, $\widehat{L}\widetilde{L}$ computations of $\{[\boldsymbol{\Upsilon}]_j\}$, $\widehat{L}\widetilde{L}^2$ computations of $[\boldsymbol{\Upsilon}]_j^T\widetilde{\mathbf{p}}_i$, $\widehat{L}\widetilde{L}$ computations of $\gamma\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i$, $\widehat{L}\widetilde{L}^2$ computations of $\mathrm{Tr}(\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\boldsymbol{\Sigma}}_l)$, and $\widehat{L}$ computations of the $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ values. The $\widehat{L}$ computations of $\mathrm{Tr}(\boldsymbol{\Sigma}_l)$ and $\widehat{\mathbf{p}}_l^T\widehat{\mathbf{p}}_l$ require $\mathcal{O}(\widehat{L}M)$ operations. The computation of $\boldsymbol{\xi}$ requires $\mathcal{O}(\widetilde{L}^3)$ operations for the Cholesky decomposition of $\widetilde{\boldsymbol{\Phi}}$, $\mathcal{O}(\widetilde{L}^2)$ operations to obtain $\boldsymbol{\psi}$, and another $\mathcal{O}(\widetilde{L}^2)$ operations to obtain the products $[\boldsymbol{\psi}]_i[\boldsymbol{\psi}]_j$. The $\widetilde{L}^2$ computations of $N(\widetilde{\mathbf{p}}_i; \widetilde{\mathbf{p}}_j, 2\mathbf{B})$ require $\mathcal{O}(\widetilde{L}^2 M)$ operations. If the matrices $\{\widehat{\boldsymbol{\Sigma}}_l\}$ are diagonal, the $\widehat{L}\widetilde{L}^2$ computations of $N(\widehat{\mathbf{p}}_l; \frac{\widetilde{\mathbf{p}}_i+\widetilde{\mathbf{p}}_j}{2}, \frac{\mathbf{B}}{2} + \widehat{\boldsymbol{\Sigma}}_l)$ require $\mathcal{O}(\widehat{L}M^3)$ operations to obtain the Cholesky factors of $\{\frac{\mathbf{B}}{2} + \widehat{\boldsymbol{\Sigma}}_l\}$ and another $\mathcal{O}(\widehat{L}\widetilde{L}^2 M^2)$ operations to obtain the Gaussian terms. Otherwise, the $\widehat{L}\widetilde{L}^2$ computations of $N(\widehat{\mathbf{p}}_l; \frac{\widetilde{\mathbf{p}}_i+\widetilde{\mathbf{p}}_j}{2}, \frac{\mathbf{B}}{2} + \widehat{\boldsymbol{\Sigma}}_l)$ would require $\mathcal{O}(\widehat{L}\widetilde{L}^2M)$ operations. When computing $\{[\boldsymbol{\Upsilon}]_j\}$, we know that the Gaussian terms (c.f. (20)) are already available from the $\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}$ calculations (c.f. (19)). Therefore, if the $\{\widehat{\boldsymbol{\Sigma}}_l\}$ are non-diagonal matrices, the $\widehat{L}\widetilde{L}$ computations of $\{[\boldsymbol{\Upsilon}]_j\}$ would require $\mathcal{O}(\widehat{L}M^3)$ operations for Cholesky decomposition of $\widehat{\boldsymbol{\Sigma}}_l$ and $\widehat{\boldsymbol{\Sigma}}_l\mathbf{B} + \mathbf{I}$ and another $\mathcal{O}(\widehat{L}\widetilde{L}M^2)$ operations for the matrix-vector products in $[\boldsymbol{\Upsilon}]_i$ (c.f. (20)). If the $\{\widehat{\boldsymbol{\Sigma}}_l\}$ are diagonal matrices, the $\widehat{L}\widetilde{L}$ computations of $[\boldsymbol{\Upsilon}]_j$ require a total of $\mathcal{O}(\widetilde{L}\widehat{L}M)$ operations. When all the $[\boldsymbol{\Upsilon}]_j$ are available, the $\widehat{L}\widetilde{L}^2$ computations of $[\boldsymbol{\Upsilon}]_j^T\widetilde{\mathbf{p}}_i$ require $\mathcal{O}(\widetilde{L}^2 M)$ operations. The $\widehat{L}\widetilde{L}$ computations of $\gamma\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i$ require $\mathcal{O}(\widehat{L}\widetilde{L}M)$ operations. The $\widehat{L}\widetilde{L}^2$ computations of $\mathrm{Tr}(\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\boldsymbol{\Sigma}}_l)$ require $\mathcal{O}(\widehat{L}\widetilde{L}^2 M^2)$ operations if $\{\widehat{\boldsymbol{\Sigma}}_l\}$ are non-diagonal and $\mathcal{O}(\widehat{L}\widetilde{L}^2 M)$ otherwise. Lastly, from our earlier discussion, calculation of $\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}$ requires $\mathcal{O}(\widetilde{L}^3)$ operations. In total, considering non-diagonal $\{\widehat{\boldsymbol{\Sigma}}_l\}$, $\widehat{L} << \widetilde{L}$, and $M << \widetilde{L}$, calculation of $\widehat{\mathbf{C}}_x^{(\mathrm{GaGP})}$ incurs complexity of $\mathcal{O}(\widetilde{L}^3 + \widetilde{L}^2\widehat{L}M^2)$.

C. RecGaGP METHOD

The computational cost of RecGaGP method can be obtained as the sum cost of the reconstruction step in (21) and the GaGP method. The reconstruction step (21) incurs complexity of $\mathcal{O}(\widetilde{L}^3)$ because we require (i) $\mathcal{O}(\widetilde{L}^3)$ operations to obtain the right singular matrix $\mathbf{V}$ via singular value decomposition, (ii) $\mathcal{O}(M^2 M_0)$ operations to obtain the product $\mathbf{V}^{[M_0]}(\mathbf{V}^{[M_0]})^T$, (iii) $\mathcal{O}(\widehat{L}M^2)$ operations to obtain the product $\widehat{\mathbf{P}}\mathbf{V}^{[M_0]}(\mathbf{V}^{[M_0]})^T$, and another $\mathcal{O}(\widehat{L}M)$ operations to obtain the residual noise covariances $\widehat{\boldsymbol{\Sigma}}_l^{(\mathrm{rec})}$ (c.f. **Lemma 2**).

Consequently, similar to the GaGP method, calculation of $\widehat{\boldsymbol{\mu}}_x^{(\mathrm{RGP})}$ and $\widehat{\mathbf{C}}_x^{(\mathrm{RGP})}$ in the RecGaGP method incurs complexity of $\mathcal{O}(\widetilde{L}^3)$ and $\mathcal{O}(\widetilde{L}^3+\widetilde{L}^2\widehat{L}M^2)$, respectively.

Remark 4: We may note from the discussion in Section VI that the location estimates from the GaGP and RecGaGP methods can be obtained in $\mathcal{O}(\widetilde{L}^3)$ computations, which is also the case with the conventional GP method. Since we have in general that the number of training locations is much higher than the number of RRHs, i.e., $\widetilde{L} >> M$, we know that the complexity would still remain $\mathcal{O}(\widetilde{L}^3)$ when we increase M. Also note that the 2σ error-bars from the GaGP and RecGaGP methods can be obtained in $\mathcal{O}(\widetilde{L}^3+\widetilde{L}^2\widehat{L}M^2)$ operations, i.e., the complexity only increases quadratically with M. These observations reveal that the proposed GP methods are indeed suitable for operation in the massive MIMO regime, as long as the number of RRHs used for localization is not excessively large. This is also confirmed in Section VII, where numerical examples demonstrate that we do not really need an excessively large number of RRHs for user positioning because the RMSE performance decreases initially and saturates beyond a certain point when M is increased.

VII. NUMERICAL EXAMPLES AND DISCUSSIONS

We now present numerical examples to evaluate the performance of the GaGP and RecGaGP methods in estimating the test user locations. We study the estimation performance under different shadowing variance σ_z^2, number of remote radio heads M, number of principal components of the training RSS M_0, and the number of training points $\widetilde{L}$.

A. PARAMETERS AND SETUP

We consider a simulation setup in which there are $M = 30$ RRHs and $\widetilde{L} = 1024$ training locations, both placed uniformly in a service area of 200m $\times$ 200m. We wish to estimate the x and y coordinates of $\widehat{L} = 10$ test users which are uniformly distributed across the setup area. We assume that both the training and test user locations have a measurement error variance σ_{er}^2 of 1dB. A noise-free training RSS matrix $\widetilde{\mathbf{P}}$ is generated from (4) by setting $\sigma_z^2 = 0$ and other parameters as listed in Table 3.

The entries of Table 3 are chosen as follows. The path-loss parameters l_0, d_0, and η are chosen as per the 3GPP Urban Micro model [42]. The user transmit power is chosen as per LTE standards to be 21dBm [44]. Total noise power in the system comprises of the receiver noise figure, which we set at 2.2dB, and the thermal noise power, which we set at -109.7dB (corresponding to 15 LTE resource blocks of size 180kHz allocated on the uplink). Since we extract the per-user RSS values during the channel estimation phase, we should take the minimum required signal-to-noise ratio (SNR) for channel estimation into account. In practice, the minimum required SNR is determined from the acceptable level of the normalized mean squared error of the channel estimates [45]. For our simulations, we set the minimum required SNR to 1dB. The receiver sensitivity, computed as the sum of the minimum required SNR and the noise power in the system, is the minimum detection threshold for the receiver.

TABLE 3. Parameters for simulation studies.

System Parameters	Value
path loss parameters (3GPP UMi [42])	$d_0 = 10$m, $b_0 = -47.5$dB, $\eta = \begin{cases} 0 \text{ if } d_{mk} < 10\text{m}, \\ 2 \text{ if } 10\text{m} \le d_{mk} \le 45\text{m}, \\ 6.7 \text{ if otherwise.} \end{cases}$
UE transmit power	21dBm (125mW)
Receiver noise figure	2.2dB
Thermal noise[4]	-109.7dB
Noise power	-107.5 dBm
Minimum SNR for channel estimation	1 dB
BS receiver sensitivity	-106.5 dBm

During the training phase, the free parameter vector $\boldsymbol{\theta}$ in the GP model is learned by solving the maximum-likelihood problem in (11) through conjugate gradient (CG) method [41]. We run multiple instances of the CG method with randomly chosen starting points, so as to avoid convergence to a bad local optimum. The convergence properties of the CG method are not discussed here because they are well-known [41]. The same vector $\bar{\boldsymbol{\theta}}$ is used to evaluate the performance of both the GaGP and the RecGaGP methods because both the methods share the same training procedure.

B. BASELINE SCHEMES

Our first baseline is the conventional GP (CGP) method, which provides location estimates and their 2σ error-bars by naively treating the noisy test RSS vectors as noise-free (c.f. (14)). As the second baseline, we consider the NaGP method proposed in [6]. The NaGP method is similar to the GaGP method in that it provides realistic 2σ error-bars on the estimated locations, but unlike GaGP, the NaGP method takes a numerical approach to approximate the true predictive distribution as Gaussian with the same first and second order moments. As the third baseline, we consider the RecGP method proposed in [27]. Similar to RecGaGP, the RecGP method reconstructs the test RSS from a low-dimensional principal subspace of the noise-free training RSS, as done in (21). However, unlike the RecGaGP method, the RecGP method naively treats the reconstructed test RSS vectors as noise-free (as done in CGP) to estimate the test user locations.

C. PERFORMANCE METRICS AND BOUNDS

Location prediction performance is measured in terms of two metrics, namely, (i) the root-mean-squared estimation error (RMSE) and (ii) the log-predictive density (LPD).

[4]When temperature is 290K and 15 LTE resource blocks of size 180 kHz are allocated on the uplink.

Mathematically, the RMSE and LPD are defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{l=1}^{\widehat{L}} ([\widehat{\mathbf{x}}]_l - [\widehat{\boldsymbol{\mu}}_x^{(.)}]_l)^2 + ([\widehat{\mathbf{y}}]_l - [\widehat{\boldsymbol{\mu}}_y^{(.)}]_l)^2}{\widehat{L}}},$$

and

$$\begin{aligned}\text{LPD} &= \frac{1}{\widehat{L}}(\log(p(\widehat{\mathbf{x}}|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{P}})) + \log(p(\widehat{\mathbf{y}}|\widetilde{\mathbf{y}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{P}}))), \\ &= -\log(2\pi) - \frac{1}{2\widehat{L}} \sum_{l=1}^{\widehat{L}} \Bigg\{ \log([\widehat{\mathbf{C}}_x^{(.)}]_{ll}) + \log([\widehat{\mathbf{C}}_y^{(.)}]_{ll}) \\ &\quad + \frac{([\widehat{\mathbf{x}}]_l - [\widehat{\boldsymbol{\mu}}_x^{(.)}]_l)^2}{[\widehat{\mathbf{C}}_x^{(.)}]_{ll}} + \frac{([\widehat{\mathbf{y}}]_l - [\widehat{\boldsymbol{\mu}}_y^{(.)}]_l)^2}{[\widehat{\mathbf{C}}_y^{(.)}]_{ll}} \Bigg\}, \end{aligned} \tag{26}$$

where $[\widehat{\mathbf{x}}]_l$ and $[\widehat{\mathbf{y}}]_l$ denote the x and y coordinates of the test user l, $[\widehat{\boldsymbol{\mu}}_x^{(.)}]_l$ and $[\widehat{\boldsymbol{\mu}}_y^{(.)}]_l$ are the estimates of $[\widehat{\mathbf{x}}]_l$ and $[\widehat{\mathbf{y}}]_l$ given by the chosen GP method (for example, if we choose the GaGP method, $[\widehat{\boldsymbol{\mu}}_x^{(.)}]_l = [\widehat{\boldsymbol{\mu}}_x^{(\text{GaGP})}]_l$ and $[\widehat{\boldsymbol{\mu}}_y^{(.)}]_l = [\widehat{\boldsymbol{\mu}}_y^{(\text{GaGP})}]_l$), and $[\widehat{\mathbf{C}}_x^{(.)}]_{ll}$ and $[\widehat{\mathbf{C}}_y^{(.)}]_{ll}$ are the variances associated with the estimates $[\widehat{\boldsymbol{\mu}}_x^{(.)}]_l$ and $[\widehat{\boldsymbol{\mu}}_y^{(.)}]_l$, respectively. Note from (26) that the RMSE only takes the location estimates $[\widehat{\boldsymbol{\mu}}_x^{(.)}]_l$ and $[\widehat{\boldsymbol{\mu}}_y^{(.)}]_l$ into account. The uncertainties $[\widehat{\mathbf{C}}_x^{(.)}]_{ll}$ and $[\widehat{\mathbf{C}}_y^{(.)}]_{ll}$ around these estimates are ignored. On the other hand, LPD utilizes the entire predictive distribution - it penalizes overconfident estimates by allocating larger weights to the estimation errors $([\widehat{\mathbf{x}}]_l - [\widehat{\boldsymbol{\mu}}_x^{(.)}]_l)$ and $([\widehat{\mathbf{y}}]_l - [\widehat{\boldsymbol{\mu}}_y^{(.)}]_l)$ when the corresponding uncertainties $[\widehat{\mathbf{C}}_x^{(.)}]_{ll}$ and $[\widehat{\mathbf{C}}_y^{(.)}]_{ll}$ are small. We say that the prediction performance is better when the RMSE values are lower and/or the LPD values are higher.

As a performance bound on the RMSE performance of the two GP methods under study, we utilize the following Bayesian Cramer Rao lower bound (BCRLB) [28]:

$$\text{BCRLB}^{(\text{RMSE})} = \sqrt{\frac{1}{\widehat{L}} \text{Tr}(\widehat{\mathbf{C}}_x^{(.)} + \widehat{\mathbf{C}}_y^{(.)})}, \tag{27}$$

where $[\widehat{\mathbf{C}}_x^{(.)}]_{ll}$ and $[\widehat{\mathbf{C}}_y^{(.)}]_{ll}$ are the variances associated with the x and y coordinate estimates provided by the chosen GP method and $\widehat{L}$ is the number of test users.

We generate 200 Monte-Carlo test RSS matrices each for shadowing variance σ_z^2 ranging from 1dB to 5dB, using (4) with relevant system parameters as listed in Table 3. During simulations, any instantaneous test RSS value that is lower than the receiver sensitivity is replaced with the noise power in the system. The RMSE and LPD values averaged over the Monte-Carlo realizations are reported. For the NaGP method [6], we set the number of Monte-Carlo samples to 10. For the RecGaGP and RecGP methods, the number of PCs M_0 of the noise-free training RSS matrix $\widetilde{\mathbf{P}}$ is chosen as the M_0 which most frequently gives the lowest RMSE among the Monte-Carlo datasets.

D. RMSE PERFORMANCE

In Fig. 2, we plot the average RMSE achieved by the two GP methods under study, for shadowing variance $\sigma_z^2 = 1, \ldots, 5$dB and for $M = 10, 30$. For comparison, we also plot the RMSE performance of the three baseline schemes, namely the CGP, NaGP, and the RecGP methods. Firstly, we observe that the RMSE of all five GP methods increases with the noise level in the test RSS. This is expected because we train the GP models with noise-free RSS data - all the five GP methods, therefore, tend to project the noise present in the test RSS onto the output location coordinate space. Secondly, we observe that the CGP, NaGP, and GaGP methods provide higher RMSE values than the RecGP and RecGaGP methods. This is because the first three methods utilize the original test RSS vectors for location estimation, whereas the latter two utilize the reconstructed test RSS vectors for the same. The reconstruction procedure reduces the shadowing noise levels in the test RSS, hence the lower RMSE levels for RecGP and RecGaGP. Thirdly, we note that the NaGP and GaGP methods do not provide much improvement in the RMSE over CGP. This is because of an inherent bias introduced by the integration procedure in (17), as is also confirmed in prior works on approximate inference GP methods for time-series analysis [29], [30]. The RecGaGP and RecGP methods achieve similar RMSE performance for the same reason as

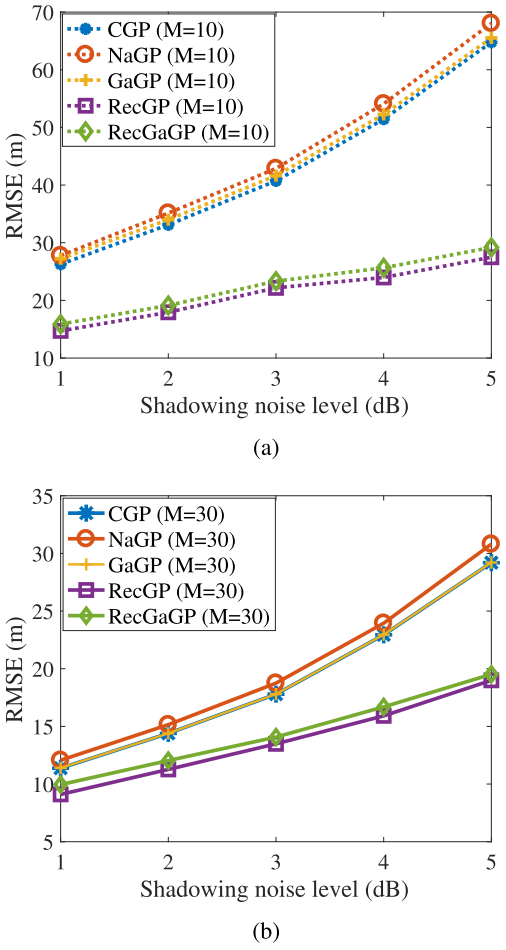


FIGURE 2. Plots of the RMSE performance of the proposed GP methods and the three baseline GP methods for different shadowing noise levels, when $M = 10, 30$. (a) RMSE vs σ_z^2 for $M = 10$. (b) RMSE vs σ_z^2 for $M = 30$.

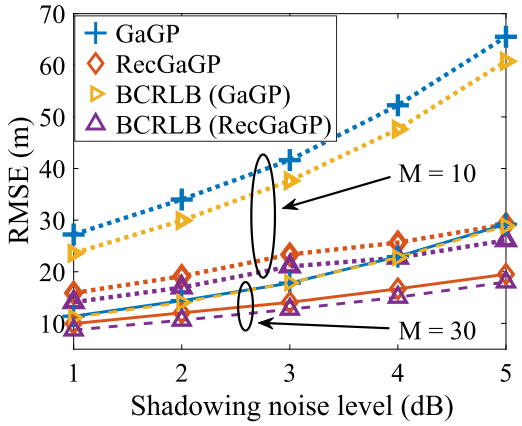


FIGURE 3. RMSE performance of the GaGP and RecGaGP methods, along with their BCRLBs, for different shadowing noise levels in the test RSS vectors.

the GaGP and CGP methods achieving similar RMSE performance. Lastly, when the number of RRHs is increased from $M = 10$ to $M = 30$, we observe significant improvements in the RMSE performance of all the five GP methods. This demonstrates the advantage of employing massive MIMO over conventional MIMO for positioning users with their uplink RSS data.

In Fig. 3, we plot the Bayesian Cramer-Rao lower bounds (BCRLBs) on the RMSE performance of the two GP methods under study, for shadowing noise level $\sigma_z^2 = 1, \ldots, 5$dB and for the number of RRHs $M = 10, 30$. We observe that the achieved RMSE performances are very close to the theoretical BCRLBs for both $M = 10$ and $M = 30$. We also note that the BCRLBs are tighter for larger M. This is expected because with larger M, there is a smaller chance of errors introduced by the receiver sensitivity threshold, i.e., a smaller fraction of the total number of RRHs would experience test RSS values that are below the receiver sensitivity level. We also note that the BCRLBs are looser for the RecGaGP method because of the small amount of information loss in the test RSS from the reconstruction procedure (21).

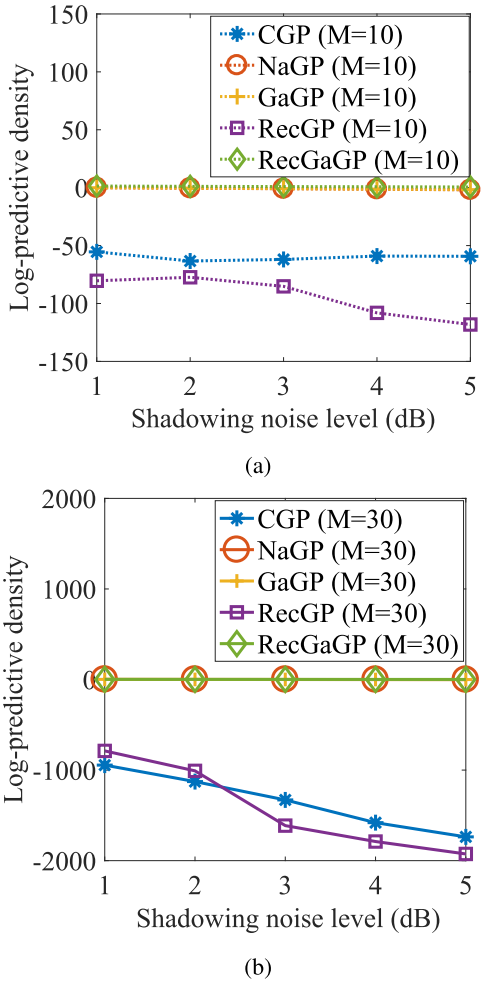


FIGURE 4. Plots of the average LPD performance of the five GP methods under study for different shadowing noise levels. The CGP and RecGP methods achieve very low LPD values because they provide unrealistically small 2σ error-bars over the location estimates (c.f. Fig. 5), with less than 30% (for $M = 10$) and 5% (for $M = 30$) of the true user locations within the 2σ confidence range of the estimated locations (c.f. Fig. 6). The GaGP, NaGP, and RecGaGP methods achieve much higher LPD values because they provide more realistic 2σ error-bars on the location estimates (c.f. Fig. 5), with more than 90% of the true user locations within the 2σ confidence range of the estimated locations (c.f. Fig. 6). (a) LPD vs σ_z^2 performance for $M = 10$. (b) LPD vs σ_z^2 performance for $M = 30$.

E. LPD PERFORMANCE

In Fig. 4, we plot the LPD performance of the five GP methods under study, for shadowing variance $\sigma_z^2 = 1, \ldots, 5$dB and for $M = 10, 30$. Fig. 5 plots the average 2σ error-bars $2\sqrt{[\widehat{\mathbf{C}}_x^{(.)}]_{ll}}$ and $2\sqrt{[\widehat{\mathbf{C}}_y^{(.)}]_{ll}}$ on the estimated x and y coordinates of the test users. Fig. 6 plots the average fraction of true test user locations that are within the 2σ confidence range $([\widehat{\boldsymbol{\mu}}_x^{(.)}]_l \pm 2\sqrt{[\widehat{\mathbf{C}}_x^{(.)}]_{ll}}, [\widehat{\boldsymbol{\mu}}_y^{(.)}]_l \pm 2\sqrt{[\widehat{\mathbf{C}}_y^{(.)}]_{ll}})$ of the estimated locations. We observe that the CGP method achieves very low LPD values because it provides unrealistically small $[\widehat{\mathbf{C}}_x^{(.)}]_{ll}$ and $[\widehat{\mathbf{C}}_y^{(.)}]_{ll}$ values (c.f. Fig. 5), with less than 30% (when $M = 10$) and 5% (when $M = 30$) of the true user locations falling inside the 2σ confidence range of the estimated locations (c.f. Fig. 6). Note from (26) that the LPD metric penalizes such overconfident estimates by allocating bigger weights to the estimation error. The RecGP method also provides very low LPD values for the same reason as CGP because RecGP applies conventional GP principles to the reconstructed test RSS for location prediction [27]. The GaGP, NaGP and RecGaGP methods achieve much higher LPD values than the CGP and RecGP methods because they provide realistic $[\widehat{\mathbf{C}}_x]_{ll}$ and $[\widehat{\mathbf{C}}_y]_{ll}$ values (c.f. Fig. 5), with more than 90% of the true user locations falling inside the 2σ confidence range of estimated locations (c.f. Fig. 6).

Taking both the RMSE and LPD performance into perspective, we observe that the RecGaGP method consistently achieves the best prediction performance. While the superior RMSE performance is because the RecGaGP method reduces noise in the test RSS vectors through the reconstruction

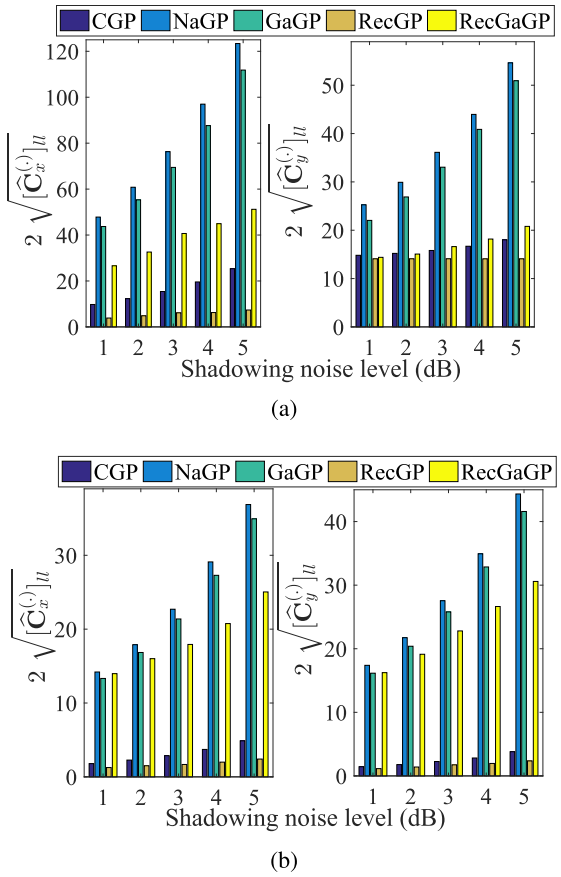


FIGURE 5. Plots of the average 2σ error-bars on the estimated locations, as provided by the five GP methods under study. The CGP and RecGP methods provide unrealistically small 2σ error-bars, with less than 30% (for $M = 10$) and 5% (for $M = 30$) of the true user locations within the 2σ confidence range of the predicted locations (c.f. Fig. 6). The GaGP, NaGP, and RecGaGP methods provide more realistic 2σ error-bars on the location estimates, with more than 90% of the true user locations within the 2σ confidence range of the estimated locations (c.f. Fig. 6).
(a) Average 2σ error-bars on the estimated x and y coordinates of the test user locations when $M = 10$. (b) Average 2σ error-bars on the estimated x and y coordinates of the test user locations when $M = 30$.

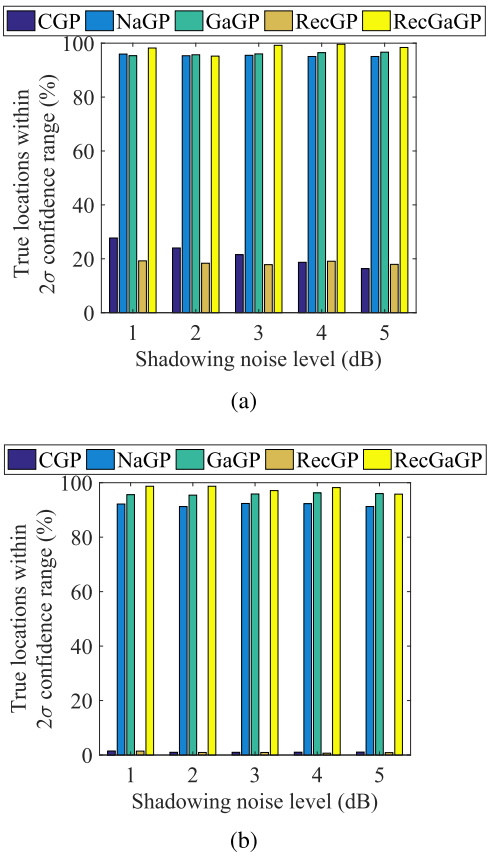


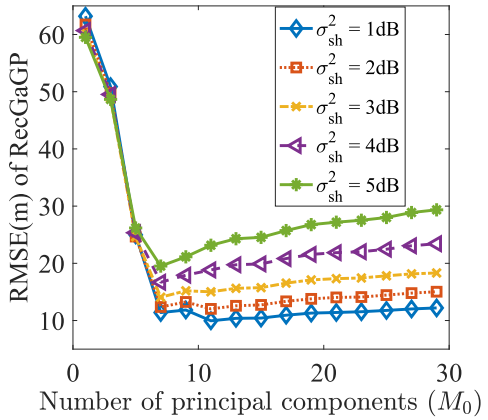
FIGURE 6. Plots of the average number of test users within the 2σ confidence range ($[\widehat{\mu}_x^{(\cdot)}]_l \pm 2\sqrt{[\widehat{C}_x^{(\cdot)}]_{ll}}, [\widehat{\mu}_y^{(\cdot)}]_l \pm 2\sqrt{[\widehat{C}_y^{(\cdot)}]_{ll}}$) of the location estimates, as given by the five GP methods under study. For the CGP and RecGP methods, less than 30% (when $M = 10$) and 5% (when $M = 30$) of the true user locations fall within the 2σ confidence range of the location estimates. For the GaGP and RecGaGP methods, more than 90% (for both $M = 10, 30$) of the true user locations fall within the 2σ confidence range of the estimated locations. (a) True locations (%) within the 2σ confidence range of the location estimates when $M = 10$. (b) True locations (%) within the 2σ confidence range of the location estimates when $M = 30$.

procedure, the superior LPD performance is because RecGaGP learns from statistical properties of the residual noise present in the reconstructed RSS to provide realistic 2σ error-bars on the estimated locations.

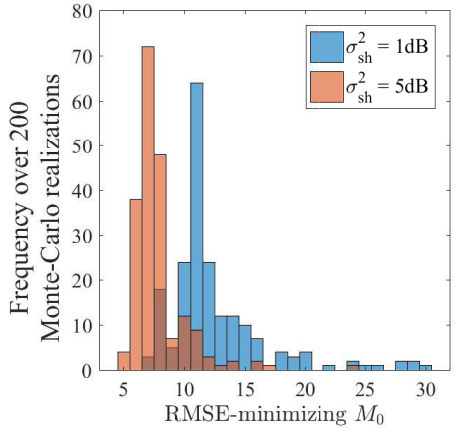
F. CHOOSING THE NUMBER OF PRINCIPAL COMPONENTS

Observe from (21) that increasing the number of principal components (M_0) increases the amount of information retained in the test RSS upon reconstruction. Also, observe from (23) in **Lemma 2** that increasing M_0 increases the amount of residual noise present in the reconstructed test RSS. Consequently, we expect that the RMSE performance of the RecGaGP (and RecGP) method would vary with M_0. In Fig. 7a, we plot the average RMSE values obtained from RecGaGP when $M = 30$, for M_0 ranging from 1 to 30. For low M_0, we observe very high RMSE values because most of the information contained in the test RSS $\widehat{\mathbf{P}}$ is lost in the reconstruction step (21). When M_0 is increased, the RMSE values decrease initially because of the increase in the information retained in the test RSS upon reconstruction. This trend ceases at a certain M_0 and the RMSE values attain a minimum level, followed by a gradual increase with M_0. This is because the noise-free training RSS spans a low-dimensional principal subspace and any further increase in M_0 would not increase the amount of information retained in the test RSS but would increase the amount of residual noise. The increase with M_0 is more prominent for higher noise levels because the amount of residual noise $\widehat{\mathbf{\Sigma}}_l^{(rec)}$ in the reconstructed test RSS increases with the amount of noise $\widehat{\mathbf{\Sigma}}_l$ in the original test RSS (c.f. (23)). Lastly, we observe that the RMSE-minimizing M_0 is different for different noise levels. Choosing an appropriate M_0 is, therefore, an important decision which we make as follows.

We plot histograms of the RMSE-minimizing M_0 over 200 Monte-Carlo realizations of the test RSS matrices, as shown in Fig. 7b for $\sigma_z^2 = 1$dB and 5dB. We observe that the



(a)



(b)

FIGURE 7. Average RMSE performance of the RecGaGP method against the number of principal components M_0 of the training RSS and the histogram of RMSE-minimizing M_0 values for $\sigma_z^2 = 1$dB and $\sigma_z^2 = 5$dB. In Fig. 7a, we observe that the RMSE decreases with M_0 initially, attains a minimum, followed by a gradual increase. In Fig. 7b, we observe that the histograms have one to three peak values. (a) Average RMSE vs. M_0 in RecGaGP. (b) Histogram of the RMSE-minimizing M_0 over 200 Monte-Carlo realizations of the test RSS dataset.

histograms have one to three peak values. We, therefore, resort to a heuristic method to choose an appropriate M_0: for a given noise level, we select the top three most-frequent RMSE-minimizing M_0 values from the histogram and choose the one which gives the best average RMSE and LPD performance as the appropriate number of principal components. When reporting the average RMSE and LPD performance of RecGaGP in Fig. 2-4, we follow this heuristic approach and choose M_0 to be 11, 11, 7, 7 and 7 for noise levels $\sigma_z^2 = 1, 2, 3, 4$, and 5dB, respectively. For a fair comparison, the same set of M_0 values are chosen for the RecGP method as well.

G. IMPACT OF THE NUMBER OF RRHs M ON THE RMSE PERFORMANCE

In Fig. 8, we plot the RMSE performance of the GaGP and RecGaGP methods when the number of RRHs M is varied from 10 to 100. We observe that the RMSE performance

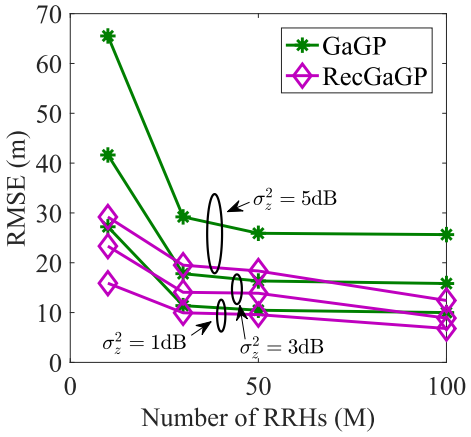


FIGURE 8. RMSE performance of the GaGP and RecGaGP methods when the number of RRHs M is varied.

decreases initially, followed by saturation beyond a certain M. Similar behavior is observed for different shadowing noise levels in the test RSS. Consequently, while we have noticed from Fig. 2 that moving from conventional MIMO to massive MIMO is beneficial from the user positioning point of view, we also note from Fig. 8 that the benefit becomes incremental after a certain value of M. Therefore, when operating with an excessively large number of RRHs (for spectral and/or energy efficiency gains [3], [43]), we would only experience minor losses in the RMSE performance if we choose a subset of the total number of RRHs for user positioning.

H. IMPACT OF THE NUMBER OF TRAINING LOCATIONS $\widetilde{L}$ ON THE RMSE PERFORMANCE

In the training phase, a general rule of thumb is that we train the GP model with as much data as possible so as to allow the GP model to learn all hidden features in the relationship between the input RSS space and the output location coordinate space. However, as may be noted from Section VI, the complexities of both GaGP and RecGaGP methods increase in the cubic order with the number of training locations $\widetilde{L}$. Therefore, it is important for us to choose $\widetilde{L}$ that is sufficient for learning the free parameter vector $\boldsymbol{\theta}$ and is also not excessively large. For insights on choosing $\widetilde{L}$, in Fig. 9, we plot the RMSE performance of the GaGP and RecGaGP methods when $\widetilde{L}$ is varied from 100 to 1600. Similar to the case with increasing M, we observe that the RMSE performance decreases initially, followed by saturation beyond a certain $\widetilde{L}$. Same is the case for different levels of shadowing noise in the test RSS. To choose the number of training locations $\widetilde{L}$, we therefore recommend observing the saturation regions in the RMSE vs. $\widetilde{L}$ plots. For example, from Fig. 9, we note that $\widetilde{L} = 1000$ is sufficient for the numerical example under study. As a side note, we emphasize that when building the training RSS matrix $\widetilde{\mathbf{P}}$, we should choose training user locations that are spread across the service area. Doing so would allow the proposed GP methods to capture

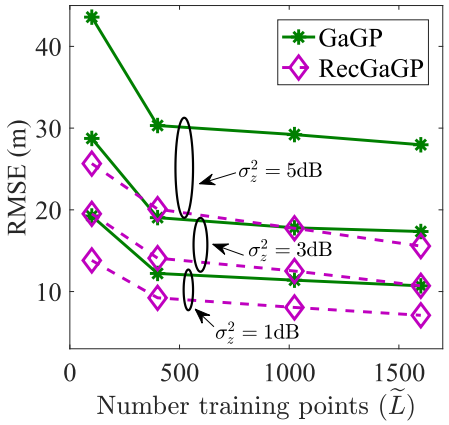


FIGURE 9. RMSE performance of the GaGP and RecGaGP methods when the number of training user locations $\widetilde{L}$ is varied.

the relationship between the RSS vectors and the location coordinate vectors in a more effective manner than when the training locations span a small portion of the service area.

VIII. CONCLUSION

We have proposed a supervised machine learning approach based on Gaussian process regression (GP) for localizing users in a distributed massive multiple-input multiple-output (MIMO) system. Our focus has been on the scenario where noise-free RSS is available for training a GP model but only noisy RSS of the test user is available for estimating its location. First, we have applied the Gaussian approximation GP (GaGP) method and made the necessary extensions to suit the localization problem under study. The GaGP method provides similar root-mean-squared prediction error (RMSE) as the conventional GP method, but with more realistic 2σ error-bars on the estimated locations. Second, we have proposed RecGaGP, a GP method which not only achieves lower RMSE than the GaGP and the conventional GP methods, but also provides realistic 2σ error-bars on the estimated locations. While the lower RMSE values are achieved through a reconstruction procedure which performs noise reduction in the test RSS, the realistic 2σ error-bars are obtained by learning from the statistical properties of the residual noise present in the reconstructed test RSS. For the two GP methods, we have derived closed-form expressions for the location estimates and the associated 2σ error-bars, in terms of the training user locations, training RSS, and the test RSS data.

Numerical examples have validated the prediction performance of the proposed GP methods in terms of two metrics: (i) the RMSE, which measures the prediction accuracy and (ii) the log-predictive density (LPD), which weighs prediction accuracy against the uncertainty in predictions to penalize overconfident estimates. We observe that (i) the GaGP method performs better than the conventional GP in terms of the LPD, (ii) the RecGaGP method performs better than the conventional GP in terms of both the RMSE and the LPD, (iii) the RMSE performances of both the GaGP and RecGaGP methods are very close to the theoretical Bayesian Cramer-Rao lower bounds, and (iv) when the number of BS antennas or the number of training points is increased, the RMSE performances of both the GP methods decrease initially, followed by saturation beyond a certain point.

The presented study opens up several exciting research directions for future work. First, practical experimentations need to be conducted to investigate the robustness of the proposed GP methods. This would be an important step in analyzing the impact of realistic aspects such as colored noise and hardware impairments. Second, multi-output GP methods, which account for the correlation between x and y coordinates of the mobile users, need to be designed. Since the x and y coordinates generally bear some correlation with each other, we expect that the localization performance would improve if this correlation is captured by the machine learning model. This work can also be extended to account for time-variations in the test RSS data and for thresholding errors introduced by the receiver sensitivity.

APPENDIX

A. PROOF OF REMARK 1

The predictive distribution $p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l)$ in (17) is non-Gaussian and cannot be obtained in a closed-form because the integral in (17) is analytically intractable. This is in turn because, as explained below, the first term $p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l^*)$ inside the integral in (17) is a complicated non-linear function of the Gaussian random vector $\widehat{\mathbf{p}}_l^*$, over which we integrate.

Observe from (14) that $p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l^*)$ is given by

$$p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l^*) = \mathcal{N}([\widehat{\mathbf{x}}]_l; \sum_{i=1}^{\widetilde{L}} \phi(\widehat{\mathbf{p}}_l^*, \widetilde{\mathbf{p}}_i)[\widetilde{\mathbf{\Phi}}^{-1}\widetilde{\mathbf{x}}]_i,$$
$$\phi(\widehat{\mathbf{p}}_l^*, \widehat{\mathbf{p}}_l^*) - \sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}} \phi(\widehat{\mathbf{p}}_l^*, \widetilde{\mathbf{p}}_i)[(\widetilde{\mathbf{\Phi}})^{-1}]_{ij}\phi(\widetilde{\mathbf{p}}_j, \widehat{\mathbf{p}}_l^*)), \quad (28)$$

where the training covariance matrix $\widetilde{\mathbf{\Phi}}$ is defined as in (12). From (28), we note that $p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l^*)$ is a complicated non-linear function of $\widehat{\mathbf{p}}_l^*$ for two reasons: (i) any Gaussian distribution is non-linear in its mean and covariance, and (ii) the mean and covariance of the Gaussian distribution in the R.H.S of (28) are both non-linear functions of $\widehat{\mathbf{p}}_l^*$ (c.f. (7)). This not only makes the integral in (17) analytically intractable, but also renders the predictive distribution $p([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l)$ in (17) as non-Gaussian.

B. PROOF OF LEMMA 1

Closed-form expression for $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ can be obtained as follows:

$$[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$$
$$\overset{(a)}{=} \mathbb{E}_{[\widehat{\mathbf{x}}]_l}([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l)$$
$$\overset{(b)}{=} \mathbb{E}_{\widehat{\mathbf{p}}_l^*}(\mathbb{E}_{[\widehat{\mathbf{x}}]_l}([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}}, \widetilde{\mathbf{P}}, \widehat{\mathbf{p}}_l^*))$$

$$
\begin{aligned}
&\overset{(c)}{=} \mathbb{E}_{\widehat{\mathbf{p}}_l^*}\Big(\sum_{i=1}^{\widetilde{L}}[\boldsymbol{\psi}]_i\phi(\widehat{\mathbf{p}}_l^*,\widetilde{\mathbf{p}}_i)\Big)\\
&\overset{(d)}{=} \sum_{i=1}^{\widetilde{L}}[\boldsymbol{\psi}]_i\int(\alpha' N(\widehat{\mathbf{p}}_l^*;\widetilde{\mathbf{p}}_i,\mathbf{B})+\gamma\widehat{\mathbf{p}}_l^{*T}\widetilde{\mathbf{p}}_i+\sigma_e^2\delta_{li})\\
&\quad\times\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)\,d\widehat{\mathbf{p}}_l^*\\
&\overset{(e)}{=} \sum_{i=1}^{\widetilde{L}}[\boldsymbol{\psi}]_i\Big\{\alpha' N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B}+\widehat{\boldsymbol{\Sigma}}_l)\int(\mathcal{N}(\widehat{\mathbf{p}}_l^*;(\mathbf{B}^{-1}+\widehat{\boldsymbol{\Sigma}}_l^{-1})^{-1}\\
&\quad\times(\mathbf{B}^{-1}\widetilde{\mathbf{p}}_i+\widehat{\boldsymbol{\Sigma}}_l^{-1}\widehat{\mathbf{p}}_l),(\mathbf{B}^{-1}+\widehat{\boldsymbol{\Sigma}}_l^{-1})^{-1}))d\widehat{\mathbf{p}}_l^*+\gamma\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i\Big\}\\
&= \sum_{i=1}^{\widetilde{L}}\alpha'[\boldsymbol{\psi}]_i N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B}+\widehat{\boldsymbol{\Sigma}}_l)+\gamma[\boldsymbol{\psi}]_i\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i, \qquad (29)
\end{aligned}
$$

where (a) is obtained from (18), (b) from the law of iterated expectations (c) by substituting $\mathbb{E}_{[\widehat{\mathbf{x}}]_l}([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l^*) = \sum_{i=1}^{\widetilde{L}}\phi(\widehat{\mathbf{p}}_l^*,\widetilde{\mathbf{p}}_i)[\boldsymbol{\psi}]_i$ from (14), (d) by substituting covariance model from (8), and (e) by substituting product of Gaussian terms from (52).

Similarly, we can derive a closed-form expression for $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$ as follows:

$$
\begin{aligned}
&[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}\\
&\overset{(a)}{=} \mathbb{E}_{[\widehat{\mathbf{x}}]_l}(([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l)^2)-([\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l)^2\\
&\overset{(b)}{=} \mathbb{E}_{\widehat{\mathbf{p}}_l^*}(\mathbb{E}_{[\widehat{\mathbf{x}}]_l}(([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l^*)^2))-([\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l)^2\\
&\overset{(c)}{=} \mathbb{E}_{\widehat{\mathbf{p}}_l^*}([\widehat{\boldsymbol{C}}_x^{(\mathrm{CGP})}]_{ll}+(\mathbb{E}_{[\widehat{\mathbf{x}}]_l}([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l^*))^2)-([\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l)^2\\
&\overset{(d)}{=} \mathbb{E}_{\widehat{\mathbf{p}}_l^*}\Big\{\phi(\widehat{\mathbf{p}}_l^*,\widehat{\mathbf{p}}_l^*)-\sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}}[\widetilde{\boldsymbol{\Phi}}^{-1}]_{ij}\phi(\widehat{\mathbf{p}}_l^*,\widetilde{\mathbf{p}}_i)\phi(\widetilde{\mathbf{p}}_j,\widehat{\mathbf{p}}_l^*)\\
&\quad+(\mathbb{E}_{[\widehat{\mathbf{x}}]_l}([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l^*))^2\Big\}-([\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l)^2\\
&\overset{(e)}{=} \mathbb{E}_{\widehat{\mathbf{p}}_l^*}(\phi(\widehat{\mathbf{p}}_l^*,\widehat{\mathbf{p}}_l^*))-([\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l)^2\\
&\quad-\sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}}([\widetilde{\boldsymbol{\Phi}}^{-1}]_{ij}-[\boldsymbol{\psi}]_i[\boldsymbol{\psi}]_j)\mathbb{E}_{\widehat{\mathbf{p}}_l^*}(\phi(\widehat{\mathbf{p}}_l^*,\widetilde{\mathbf{p}}_i)\phi(\widetilde{\mathbf{p}}_j,\widehat{\mathbf{p}}_l^*))
\end{aligned}
\qquad (30)
$$

where (a) is obtained from (18), (b) from the law of iterated expectations, (c) from definition of covariance in (53), (d) by substituting $[\widehat{\boldsymbol{C}}_x^{(\mathrm{CGP})}]_{ll}$ from (14), and (e) by substituting $\mathbb{E}_{[\widehat{\mathbf{x}}]_l}([\widehat{\mathbf{x}}]_l|\widetilde{\mathbf{x}},\widetilde{\mathbf{P}},\widehat{\mathbf{p}}_l^*) = \sum_{i=1}^{\widetilde{L}}\phi(\widehat{\mathbf{p}}_l^*,\widetilde{\mathbf{p}}_i)[\boldsymbol{\psi}]_i$ from (14). To proceed further, we require closed-form expressions for $\mathbb{E}_{\widehat{\mathbf{p}}_l^*}(\phi(\widehat{\mathbf{p}}_l^*,\widehat{\mathbf{p}}_l^*))$ and $\mathbb{E}_{\widehat{\mathbf{p}}_l^*}(\phi(\widehat{\mathbf{p}}_l^*,\widetilde{\mathbf{p}}_i)\phi(\widetilde{\mathbf{p}}_j,\widehat{\mathbf{p}}_l^*))$, which we derive as follows:

$$
\begin{aligned}
\mathbb{E}_{\widehat{\mathbf{p}}_l^*}(\phi(\widehat{\mathbf{p}}_l^*,\widehat{\mathbf{p}}_l^*)) &\overset{(a)}{=} \mathbb{E}_{\widehat{\mathbf{p}}_l^*}(\alpha' N(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l^*,\mathbf{B})+\gamma\widehat{\mathbf{p}}_l^{*T}\widehat{\mathbf{p}}_l^*+\sigma_e^2\delta_{ll})\\
&= \sigma_e^2+\int(\alpha+\gamma\widehat{\mathbf{p}}_l^{*T}\widehat{\mathbf{p}}_l^*)\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)\,d\widehat{\mathbf{p}}_l^*\\
&\overset{(b)}{=} \sigma_e^2+\alpha+\gamma\widehat{\mathbf{p}}_l^T\widehat{\mathbf{p}}_l+\gamma\,\mathrm{Tr}(\widehat{\boldsymbol{\Sigma}}_l), \qquad (31)
\end{aligned}
$$

where (a) is obtained by substituting covariance model from (8) and (b) by applying the integral of quadratic from (54). Next, we have

$$
\begin{aligned}
&\mathbb{E}_{\widehat{\mathbf{p}}_l^*}(\phi(\widehat{\mathbf{p}}_l^*,\widetilde{\mathbf{p}}_i)\phi(\widetilde{\mathbf{p}}_j,\widehat{\mathbf{p}}_l^*))\\
&= \int(\alpha' N(\widehat{\mathbf{p}}_l^*;\widetilde{\mathbf{p}}_i,\mathbf{B})+\gamma\widehat{\mathbf{p}}_l^{*T}\widetilde{\mathbf{p}}_i+\sigma_e^2\delta_{li})\\
&\quad\times(\alpha N(\widehat{\mathbf{p}}_l^*;\widetilde{\mathbf{p}}_j,\mathbf{B})+\gamma(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l^*+\sigma_e^2\delta_{jl})\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)d\widehat{\mathbf{p}}_l^*\\
&= \mathcal{I}_1+\mathcal{I}_2+\mathcal{I}_3+\mathcal{I}_4, \qquad (32)
\end{aligned}
$$

where, for notational convenience, we have defined

$$
\begin{aligned}
\mathcal{I}_1 &= (\alpha')^2\int N(\widehat{\mathbf{p}}_l^*;\widetilde{\mathbf{p}}_i,\mathbf{B})N(\widehat{\mathbf{p}}_l^*;\widetilde{\mathbf{p}}_j,\mathbf{B})\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)d\widehat{\mathbf{p}}_l^*\\
\mathcal{I}_2 &= \alpha'\gamma\int\widehat{\mathbf{p}}_l^{*T}\widetilde{\mathbf{p}}_i N(\widehat{\mathbf{p}}_l^*;\widetilde{\mathbf{p}}_j,\mathbf{B})\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)d\widehat{\mathbf{p}}_l^*,\\
\mathcal{I}_3 &= \alpha'\gamma\int N(\widehat{\mathbf{p}}_l^*;\widetilde{\mathbf{p}}_i,\mathbf{B})(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l^*\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)d\widehat{\mathbf{p}}_l^*,\\
\mathcal{I}_4 &= \gamma^2\int\widehat{\mathbf{p}}_l^{*T}\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l^*\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)d\widehat{\mathbf{p}}_l^* \qquad (33)
\end{aligned}
$$

Closed-form expressions for $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$, and $\mathcal{I}_4$ can be obtained as follows:

$$
\begin{aligned}
\mathcal{I}_1 &= (\alpha')^2\int N(\widehat{\mathbf{p}}_l^*;\widetilde{\mathbf{p}}_i,\mathbf{B})N(\widehat{\mathbf{p}}_l^*;\widetilde{\mathbf{p}}_j,\mathbf{B})\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)d\widehat{\mathbf{p}}_l^*\\
&\overset{(a)}{=} (\alpha')^2\,N(\widetilde{\mathbf{p}}_i;\widetilde{\mathbf{p}}_j,2\mathbf{B})\int N(\widehat{\mathbf{p}}_l^*;\frac{\widetilde{\mathbf{p}}_i+\widetilde{\mathbf{p}}_j}{2},\frac{\mathbf{B}}{2})\\
&\quad\times\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)d\widehat{\mathbf{p}}_l^*\\
&\overset{(b)}{=} (\alpha')^2\,N(\widetilde{\mathbf{p}}_i;\widetilde{\mathbf{p}}_j,2\mathbf{B})N(\widehat{\mathbf{p}}_l;\frac{\widetilde{\mathbf{p}}_i+\widetilde{\mathbf{p}}_j}{2},\frac{\mathbf{B}}{2}+\widehat{\boldsymbol{\Sigma}}_l),\\
\mathcal{I}_2 &= \int\alpha'\gamma\widehat{\mathbf{p}}_l^{*T}\widetilde{\mathbf{p}}_i N(\widehat{\mathbf{p}}_l^*;\widetilde{\mathbf{p}}_j,\mathbf{B})\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)d\widehat{\mathbf{p}}_l^*\\
&\overset{(c)}{=} \alpha'\gamma\int\widehat{\mathbf{p}}_l^{*T}\widetilde{\mathbf{p}}_i N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_j,\mathbf{B}+\widehat{\boldsymbol{\Sigma}}_l)\\
&\quad\times\mathcal{N}(\widehat{\mathbf{p}}_l^*;(\mathbf{B}^{-1}+\widehat{\boldsymbol{\Sigma}}_l^{-1})^{-1}(\mathbf{B}^{-1}\widetilde{\mathbf{p}}_j+\widehat{\boldsymbol{\Sigma}}_l^{-1}\widehat{\mathbf{p}}_l),\\
&\quad\times(\mathbf{B}^{-1}+\widehat{\boldsymbol{\Sigma}}_l^{-1})^{-1})d\widehat{\mathbf{p}}_l^*\\
&= \alpha'\gamma N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_j,\mathbf{B}+\widehat{\boldsymbol{\Sigma}}_l)\{(\mathbf{B}^{-1}+\widehat{\boldsymbol{\Sigma}}_l^{-1})^{-1}\\
&\quad\times(\mathbf{B}^{-1}\widetilde{\mathbf{p}}_j+\widehat{\boldsymbol{\Sigma}}_l^{-1}\widehat{\mathbf{p}}_l)\}^T\widetilde{\mathbf{p}}_i,\\
\mathcal{I}_3 &= \int\alpha'\gamma N(\widehat{\mathbf{p}}_l^*;\widetilde{\mathbf{p}}_i,\mathbf{B})(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l^*\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)d\widehat{\mathbf{p}}_l^*\\
&\overset{(d)}{=} \alpha'\gamma(\widetilde{\mathbf{p}}_j)^T\int\widehat{\mathbf{p}}_l^* N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B}+\widehat{\boldsymbol{\Sigma}}_l)\\
&\quad\times\mathcal{N}(\widehat{\mathbf{p}}_l^*;(\mathbf{B}^{-1}+\widehat{\boldsymbol{\Sigma}}_l^{-1})^{-1}(\mathbf{B}^{-1}\widetilde{\mathbf{p}}_i+\widehat{\boldsymbol{\Sigma}}_l^{-1}\widehat{\mathbf{p}}_l),\\
&\quad\times(\mathbf{B}^{-1}+\widehat{\boldsymbol{\Sigma}}_l^{-1})^{-1})d\widehat{\mathbf{p}}_l^*\\
&= \alpha'\gamma N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B}+\widehat{\boldsymbol{\Sigma}}_l)(\widetilde{\mathbf{p}}_j)^T(\mathbf{B}^{-1}+\widehat{\boldsymbol{\Sigma}}_l^{-1})^{-1}\\
&\quad\times(\mathbf{B}^{-1}\widetilde{\mathbf{p}}_i+\widehat{\boldsymbol{\Sigma}}_l^{-1}\widehat{\mathbf{p}}_l),\\
\mathcal{I}_4 &= \int\gamma^2\widehat{\mathbf{p}}_l^{*T}\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l^*\mathcal{N}(\widehat{\mathbf{p}}_l^*;\widehat{\mathbf{p}}_l,\widehat{\boldsymbol{\Sigma}}_l)d\widehat{\mathbf{p}}_l^*\\
&\overset{(e)}{=} \gamma^2\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l+\gamma^2\,\mathrm{Tr}(\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\boldsymbol{\Sigma}}_l), \qquad (34)
\end{aligned}
$$

where (a)-(d) are obtained by applying product of Gaussian terms from (52) in each step and (e) by applying the integral of quadratic from (54).

Substituting (34) into (32) gives us a closed-form expression for $\mathbb{E}_{\widehat{\mathbf{p}}_l^*}(\phi(\widehat{\mathbf{p}}_l^*, \widetilde{\mathbf{p}}_i)\phi(\widetilde{\mathbf{p}}_j, \widehat{\mathbf{p}}_l^*))$. We then substitute (31) and (32) into (30) and define matrices $\boldsymbol{\Upsilon} \in \mathbb{R}^{M\times\widetilde{L}}$ and $\boldsymbol{\xi} \in \mathbb{R}^{\widetilde{L}\times\widetilde{L}}$ such that $[\boldsymbol{\Upsilon}]_i = \alpha'\gamma N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B} + \widehat{\boldsymbol{\Sigma}}_l)((\mathbf{B}^{-1} + \widehat{\boldsymbol{\Sigma}}_l^{-1})^{-1}(\mathbf{B}^{-1}\widetilde{\mathbf{p}}_i + \widehat{\boldsymbol{\Sigma}}_l^{-1}\widehat{\mathbf{p}}_l))$, and $[\boldsymbol{\xi}]_{ij} = [\widetilde{\boldsymbol{\Phi}}^{-1}]_{ij} - [\boldsymbol{\psi}]_i[\boldsymbol{\psi}]_j$, $\forall i, j = 1, \ldots, \widetilde{L}$, to obtain the expression for $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$ as given by **Theorem 1**.

C. RELATION BETWEEN $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ AND $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l$

From (14), we know that $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l$ is given by

$$[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l = \sum_{i=1}^{\widetilde{L}} \alpha'[\boldsymbol{\psi}]_i N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B}) + \gamma[\boldsymbol{\psi}]_i \widehat{\mathbf{p}}_l^T \widetilde{\mathbf{p}}_i. \quad (35)$$

In order to express the GaGP mean $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$, given by (19), in a similar form as (35), let us first express the Gaussian term $N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B} + \widehat{\boldsymbol{\Sigma}}_l)$ in $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ (c.f. (19)) in terms of $N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B})$ as follows:

$$\begin{aligned} &N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B} + \widehat{\boldsymbol{\Sigma}}_l) \\ &= \frac{\exp(-\frac{1}{2}(\widehat{\mathbf{p}}_l - \widetilde{\mathbf{p}}_i)^T(\mathbf{B} + \widehat{\boldsymbol{\Sigma}}_l)^{-1}(\widehat{\mathbf{p}}_l - \widetilde{\mathbf{p}}_i))}{(2\pi)^{M/2}|\mathbf{B} + \widehat{\boldsymbol{\Sigma}}_l|^{1/2}} \\ &\overset{(a)}{=} \frac{\exp(-\frac{1}{2}(\widehat{\mathbf{p}}_l - \widetilde{\mathbf{p}}_i)^T(\mathbf{B}^{-1} + (\mathbf{B} + \widehat{\boldsymbol{\Sigma}}_l)^{-1} - \mathbf{B}^{-1})(\widehat{\mathbf{p}}_l - \widetilde{\mathbf{p}}_i))}{(2\pi)^{M/2}|\mathbf{B} + \widehat{\boldsymbol{\Sigma}}_l|^{1/2}} \\ &\overset{(b)}{=} N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B})\lambda_i, \end{aligned} \quad (36)$$

where (a) is obtained by adding and subtracting $\mathbf{B}^{-1}$ and (b) by introducing a new variable λ_i, defined as,

$$\lambda_i = \frac{\exp(-\frac{1}{2}(\widehat{\mathbf{p}}_l - \widetilde{\mathbf{p}}_i)^T((\mathbf{B} + \widehat{\boldsymbol{\Sigma}}_l)^{-1} - \mathbf{B}^{-1})(\widehat{\mathbf{p}}_l - \widetilde{\mathbf{p}}_i))}{|\mathbf{I} + \mathbf{B}^{-1}\widehat{\boldsymbol{\Sigma}}_l|^{1/2}}, \quad (37)$$

for notational convenience. Substituting (36) into the expression for $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ in (19) gives us

$$[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l = \sum_{i=1}^{\widetilde{L}} \alpha'[\boldsymbol{\psi}]_i N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B})\lambda_i + \gamma[\boldsymbol{\psi}]_i \widehat{\mathbf{p}}_l^T \widetilde{\mathbf{p}}_i. \quad (38)$$

Observe from (38) and (35) that $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ is similar in structure as $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l$, but with multiplicative correction factors $\{\lambda_i\}$ introduced by the GaGP method to the Gaussian terms in (35). The correction factors account for the stochastic nature of $\widehat{\mathbf{p}}_l$. Lastly, we can verify that when the test RSS is noise-free, i.e., when $\widehat{\boldsymbol{\Sigma}}_l = \mathbf{0}$, we get $\lambda_i = 1$. Consequently, the expressions for $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ and $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l$ in (38) and (35) respectively, turn out to be the exactly the same when the test RSS is noise-free.

D. RELATIONSHIP BETWEEN $[\widehat{\mathbf{C}}_x^{(\mathrm{GaGP})}]_{ll}$ AND $[\widehat{\mathbf{C}}_x^{(\mathrm{CGP})}]_{ll}$

From (14), we know that $[\widehat{\boldsymbol{C}}_x^{(\mathrm{CGP})}]_{ll}$ is given by (39) (see the bottom of this page), where (a) is obtained by substituting the covariance model from (8) into $[\widehat{\boldsymbol{C}}_x^{(\mathrm{CGP})}]_{ll}$ in (14) and (b) by applying the product of Gaussian terms from (52). In order to express the GaGP variance $[\widehat{\mathbf{C}}_x^{(\mathrm{GaGP})}]_{ll}$ given by (19) in a similar form as (39), we firstly simplify the Gaussian term $N(\widehat{\mathbf{p}}_l; \frac{\widetilde{\mathbf{p}}_i+\widetilde{\mathbf{p}}_j}{2}, \frac{\mathbf{B}}{2} + \widehat{\boldsymbol{\Sigma}}_l)$ in $[\widehat{\mathbf{C}}_x^{(\mathrm{GaGP})}]_{ll}$ (c.f. (19)) by following the same procedure as in (36) to obtain

$$N(\widehat{\mathbf{p}}_l; \frac{\widetilde{\mathbf{p}}_i + \widetilde{\mathbf{p}}_j}{2}, \frac{\mathbf{B}}{2} + \widehat{\boldsymbol{\Sigma}}_l) = N(\widehat{\mathbf{p}}_l; \frac{\widetilde{\mathbf{p}}_i + \widetilde{\mathbf{p}}_j}{2}, \frac{\mathbf{B}}{2})\lambda_{ij},$$

where,

$$\lambda_{ij} = \frac{\exp(\frac{-1}{2}(\widehat{\mathbf{p}}_l - \frac{\widetilde{\mathbf{p}}_i+\widetilde{\mathbf{p}}_j}{2})^T((\frac{\mathbf{B}}{2} + \widehat{\boldsymbol{\Sigma}}_l)^{-1} - (\frac{\mathbf{B}}{2})^{-1})(\widehat{\mathbf{p}}_l - \frac{\widetilde{\mathbf{p}}_i+\widetilde{\mathbf{p}}_j}{2}))}{|\mathbf{I} + (\frac{\mathbf{B}}{2})^{-1}\widehat{\boldsymbol{\Sigma}}_l|^{1/2}}. \quad (41)$$

$$\begin{aligned} [\widehat{\boldsymbol{C}}_x^{(\mathrm{CGP})}]_{ll} &\overset{(a)}{=} \alpha + \gamma\widehat{\mathbf{p}}_l^T\widehat{\mathbf{p}}_l + \sigma_{er}^2 - \sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}}[(\widetilde{\boldsymbol{\Phi}})^{-1}]_{ij}\Big\{(\alpha')^2 N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B})N(\widetilde{\mathbf{p}}_j; \widehat{\mathbf{p}}_l, \mathbf{B}) \\ &\quad + \alpha'\gamma\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i N(\widetilde{\mathbf{p}}_j; \widehat{\mathbf{p}}_l, \mathbf{B}) + \alpha'\gamma(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B}) + \gamma^2\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l\Big\} \\ &\overset{(b)}{=} \alpha + \gamma\widehat{\mathbf{p}}_l^T\widehat{\mathbf{p}}_l + \sigma_{er}^2 - \sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}}(\alpha')^2[(\widetilde{\boldsymbol{\Phi}})^{-1}]_{ij}\Big\{N(\widetilde{\mathbf{p}}_i; \widetilde{\mathbf{p}}_j, 2\mathbf{B})N(\widehat{\mathbf{p}}_l; \frac{\widetilde{\mathbf{p}}_i + \widetilde{\mathbf{p}}_j}{2}, \frac{\mathbf{B}}{2})\Big\} \\ &\quad + \alpha'\gamma[(\widetilde{\boldsymbol{\Phi}})^{-1}]_{ij}\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i\Big\{N(\widetilde{\mathbf{p}}_j; \widehat{\mathbf{p}}_l, \mathbf{B})\Big\} + \alpha'\gamma[(\widetilde{\boldsymbol{\Phi}})^{-1}]_{ij}(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l\Big\{N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B})\Big\} + \gamma^2\Big\{\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l\Big\}, \end{aligned} \quad (39)$$

$$\begin{aligned} &[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll} \\ &= \alpha + \gamma\widehat{\mathbf{p}}_l^T\widehat{\mathbf{p}}_l + \sigma_{er}^2 + \gamma\,\mathrm{Tr}(\widehat{\boldsymbol{\Sigma}}_l) - \sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}}(\alpha')^2([\boldsymbol{\xi}]_{ij}\lambda_{ij} + [\boldsymbol{\psi}]_i[\boldsymbol{\psi}]_j\lambda_i\lambda_j)\Big\{N(\widetilde{\mathbf{p}}_i; \widetilde{\mathbf{p}}_j, 2\mathbf{B})N(\widehat{\mathbf{p}}_l; \frac{\widetilde{\mathbf{p}}_i + \widetilde{\mathbf{p}}_j}{2}, \frac{\mathbf{B}}{2})\Big\} \\ &\quad + \alpha'\gamma([\boldsymbol{\xi}]_{ij}\lambda_j(\boldsymbol{\nu}_j + \Gamma\widehat{\mathbf{p}}_l))^T\widetilde{\mathbf{p}}_i + [\boldsymbol{\psi}]_i[\boldsymbol{\psi}]_j\lambda_j\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i)\Big\{N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_j, \mathbf{B})\Big\} + \alpha'\gamma([\boldsymbol{\xi}]_{ij}\lambda_i(\widetilde{\mathbf{p}}_j)^T(\boldsymbol{\nu}_i + \Gamma\widehat{\mathbf{p}}_l) + [\boldsymbol{\psi}]_i[\boldsymbol{\psi}]_j\lambda_i(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l) \\ &\quad \times \Big\{N(\widehat{\mathbf{p}}_l; \widetilde{\mathbf{p}}_i, \mathbf{B})\Big\} + \gamma^2([\boldsymbol{\xi}]_{ij} + [\boldsymbol{\psi}]_i[\boldsymbol{\psi}]_j)\Big\{\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l\Big\} + [\boldsymbol{\xi}]_{ij}\gamma^2\,\mathrm{Tr}(\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\boldsymbol{\Sigma}}_l). \end{aligned} \quad (40)$$

In (41), λ_{ij} is a multiplicative correction factor introduced by GaGP. Let us also simplify the expressions for $([\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l)^2$ in $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$ (c.f. (19)), as follows:

$$\begin{aligned}
&([\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l)^2 \\
&\overset{(a)}{=} \sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}}(\alpha'[\boldsymbol{\psi}]_i N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B})\lambda_i + \gamma[\boldsymbol{\psi}]_i\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i) \\
&\quad\times(\alpha'[\boldsymbol{\psi}]_j N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_j,\mathbf{B})\lambda_j + \gamma[\boldsymbol{\psi}]_j(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l) \\
&\overset{(b)}{=} \sum_{i=1}^{\widetilde{L}}\sum_{j=1}^{\widetilde{L}}[\boldsymbol{\psi}]_i[\boldsymbol{\psi}]_j\Big\{(\alpha')^2\lambda_i\lambda_j N(\widetilde{\mathbf{p}}_i;\widetilde{\mathbf{p}}_j,2\mathbf{B}) \\
&\quad\times N(\widehat{\mathbf{p}}_l;\frac{\widetilde{\mathbf{p}}_i+\widetilde{\mathbf{p}}_j}{2},\frac{\mathbf{B}}{2}) + \alpha'\gamma\lambda_i N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B})(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l \\
&\quad+\alpha'\gamma\lambda_j N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_j,\mathbf{B})\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i + \gamma^2\widehat{\mathbf{p}}_l^T\widetilde{\mathbf{p}}_i(\widetilde{\mathbf{p}}_j)^T\widehat{\mathbf{p}}_l\Big\}, \qquad (42)
\end{aligned}$$

where (a) is obtained by substituting (38) and (b) by applying the product of Gaussian terms from (52). Next, let us simplify the expression for $[\boldsymbol{\Upsilon}]_i$ in $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$ (c.f. (19)) as follows

$$\begin{aligned}
[\boldsymbol{\Upsilon}]_i &= \alpha'\gamma N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B}+\widehat{\boldsymbol{\Sigma}}_l)((\mathbf{B}^{-1}+\widehat{\boldsymbol{\Sigma}}_l^{-1})^{-1} \\
&\quad\times(\mathbf{B}^{-1}\widetilde{\mathbf{p}}_i + \widehat{\boldsymbol{\Sigma}}_l^{-1}\widehat{\mathbf{p}}_l)) \\
&= \alpha'\gamma N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B}+\widehat{\boldsymbol{\Sigma}}_l)((\widehat{\boldsymbol{\Sigma}}_l\mathbf{B}^{-1}+\mathbf{I})^{-1}\widehat{\boldsymbol{\Sigma}}_l \\
&\quad\times\widehat{\boldsymbol{\Sigma}}_l^{-1}(\widehat{\boldsymbol{\Sigma}}_l\mathbf{B}^{-1}\widetilde{\mathbf{p}}_i+\widehat{\mathbf{p}}_l)) \\
&\overset{(a)}{=} \alpha'\gamma\lambda_i N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B})((\widehat{\boldsymbol{\Sigma}}_l\mathbf{B}^{-1}+\mathbf{I})^{-1}(\widehat{\boldsymbol{\Sigma}}_l\mathbf{B}^{-1}\widetilde{\mathbf{p}}_i+\widehat{\mathbf{p}}_l)) \\
&\overset{(b)}{=} \alpha'\gamma\lambda_i N(\widehat{\mathbf{p}}_l;\widetilde{\mathbf{p}}_i,\mathbf{B})(\boldsymbol{\nu}_i+\Gamma\widehat{\mathbf{p}}_l), \qquad (43)
\end{aligned}$$

where (a) is obtained by substituting (36) and (b) by introducing the terms $\boldsymbol{\nu}_i = (\widehat{\boldsymbol{\Sigma}}_l\mathbf{B}^{-1}+\mathbf{I})^{-1}\widehat{\boldsymbol{\Sigma}}_l\mathbf{B}^{-1}\widetilde{\mathbf{p}}_i$ and $\Gamma = (\widehat{\boldsymbol{\Sigma}}_l\mathbf{B}^{-1}+\mathbf{I})^{-1}$ for notational convenience. Substituting (41)-(43) into $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$ in (19) gives us (40), as shown at the bottom of the previous page.

Observe from (39) and (40) that the $[\widehat{\mathbf{C}}_x^{(\mathrm{CGP})}]_{ll}$ and $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$ expressions have a similar structure, with the GaGP introducing several additive and multiplicative correction factors to the Gaussian and inner product terms in $[\widehat{\mathbf{C}}_x^{(\mathrm{CGP})}]_{ll}$. When $\widehat{\boldsymbol{\Sigma}}_l = \mathbf{0}$, i.e., when the test RSS is noise-free, we can verify that $\lambda_i = 1, \lambda_{ij} = 1, \boldsymbol{\nu}_i = \mathbf{0}$, and $\Gamma = \mathbf{I}$, $\forall\, i,j,= 1,\ldots,\widetilde{L}$. Substituting these values, along with $[\boldsymbol{\xi}]_{ij} = [\widetilde{\boldsymbol{\Phi}}^{-1}]_{ij} - [\boldsymbol{\psi}]_i[\boldsymbol{\psi}]_j$ into (40), gives us an expression for $[\widehat{\boldsymbol{C}}_x^{(\mathrm{GaGP})}]_{ll}$ which is exactly the same as $[\widehat{\mathbf{C}}_x^{(\mathrm{CGP})}]_{ll}$ given in (39).

E. PROOF OF LEMMA 2

By definition, since we know $\widehat{\mathbf{z}}_l^{(rec)} = (\widehat{\mathbf{z}}_l^T\mathbf{V}^{[M_0]}(\mathbf{V}^{[M_0]})^T)^T$, we can obtain elements of $\widehat{\mathbf{z}}_l^{(rec)}$ as

$$[\widehat{\mathbf{z}}_l^{(rec)}]_m = \sum_{m'=1}^{M}\sum_{m''=1}^{M_0}[\widehat{\mathbf{z}}_l]_{m'}[\mathbf{V}^{[M_0]}]_{m'm''}[(\mathbf{V}^{[M_0]})^T]_{m''m}, \quad \forall m = 1,\ldots,M. \qquad (44)$$

We conclude from (44) and (15) that $\widehat{\mathbf{z}}_l^{(rec)}$ is also Gaussian distributed because each element $[\widehat{\mathbf{z}}_l^{(rec)}]_m$ in $\widehat{\mathbf{z}}_l^{(rec)}$ is a weighted sum of MM_0 Gaussian random variables. Also, we can obtain the elements of the mean of $\widehat{\mathbf{z}}_l^{(rec)}$ as

$$\begin{aligned}
&\mathbb{E}([\widehat{\mathbf{z}}_l^{(rec)}]_m) \qquad (45)\\
&= \mathbb{E}(\sum_{m'=1}^{M}\sum_{m''=1}^{M_0}[\widehat{\mathbf{z}}_l]_{m'}[\mathbf{V}^{[M_0]}]_{m'm''}[(\mathbf{V}^{[M_0]})^T]_{m''m}), \\
&= \sum_{m'=1}^{M}\sum_{m''=1}^{M_0}\mathbb{E}([\widehat{\mathbf{z}}_l]_{m'})[\mathbf{V}^{[M_0]}]_{m'm''}[(\mathbf{V}^{[M_0]})^T]_{m''m} \\
&= 0, \quad \forall m = 1,\ldots,M, \ (\text{since } \mathbb{E}([\widehat{\mathbf{z}}_l]) = \mathbf{0}). \qquad (46)
\end{aligned}$$

Similarly, if $\widehat{\boldsymbol{\Sigma}}_l^{(rec)}$ is the covariance matrix of $\widehat{\mathbf{z}}_l^{(rec)}$, elements of $\widehat{\boldsymbol{\Sigma}}_l^{(rec)}$ can be obtained as, $\forall i,j = 1,\ldots,\widehat{L}$,

$$\begin{aligned}
[\widehat{\boldsymbol{\Sigma}}_l^{(rec)}]_{ij} &= [\mathbb{E}(\widehat{\mathbf{z}}_l^{(rec)}\widehat{\mathbf{z}}_l^{(rec)T}) - \mathbb{E}(\widehat{\mathbf{z}}_l^{(rec)})\mathbb{E}(\widehat{\mathbf{z}}_l^{(rec)T})]_{ij}, \\
&= [\mathbb{E}(\widehat{\mathbf{z}}_l^{(rec)}\widehat{\mathbf{z}}_l^{(rec)T})]_{ij} \ (\text{since } \mathbb{E}(\widehat{\mathbf{z}}_l^{(rec)}) = \mathbf{0}) \\
&\overset{(a)}{=} \mathbb{E}\Big\{\sum_{m=1}^{M}\sum_{m''=1}^{M_0}[\widehat{\mathbf{z}}_l]_m[\mathbf{V}^{[M_0]}]_{mm''}[(\mathbf{V}^{[M_0]})^T]_{m''i} \\
&\quad\times\sum_{m'=1}^{M}\sum_{m''=1}^{M_0}[\widehat{\mathbf{z}}_l]_{m'}[\mathbf{V}^{[M_0]}]_{m'm''}[(\mathbf{V}^{[M_0]})^T]_{m''j}\Big\} \\
&\overset{(b)}{=} \sum_{m=1}^{M}\mathbb{E}([\widehat{\mathbf{z}}_l]_m^2)\sum_{m''=1}^{M_0}[\mathbf{V}^{[M_0]}]_{mm''}[(\mathbf{V}^{[M_0]})^T]_{m''i} \\
&\quad\times\sum_{m''=1}^{M_0}[\mathbf{V}^{[M_0]}]_{m''}[(\mathbf{V}^{[M_0]})^T]_{m''j}) \\
&= \sum_{m=1}^{M}[\widehat{\boldsymbol{\Sigma}}_l]_{mm}\sum_{m''=1}^{M_0}[\mathbf{V}^{[M_0]}]_{mm''}[(\mathbf{V}^{[M_0]})^T]_{m''i} \qquad (47)\\
&\quad\times\sum_{m''=1}^{M_0}[\mathbf{V}^{[M_0]}]_{mm''}[(\mathbf{V}^{[M_0]})^T]_{m''j}, \qquad (48)
\end{aligned}$$

where (a) is obtained by substituting (44) and (b) follows from the mutual independence assumption on the elements of $\widehat{\mathbf{z}}_l$. Combining (45) and (47) gives us the probability distribution $\widehat{\mathbf{z}}_l^{(rec)} \sim \mathcal{N}(\mathbf{0}, \widehat{\boldsymbol{\Sigma}}_l^{(rec)})$, with elements of $\widehat{\boldsymbol{\Sigma}}_l^{(rec)}$ as given by **Lemma 2**. In addition, since we also know from (22) that $\widehat{\mathbf{p}}_l^{(rec)} = \widehat{\mathbf{p}}_l^{(rec)*} + \widehat{\mathbf{z}}_l^{(rec)}$, we know that $\widehat{\mathbf{p}}_l^{(rec)*}$ is conditionally distributed as

$$(\widehat{\mathbf{p}}_l^{(rec)*}|\widehat{\mathbf{p}}_l^{(rec)}, \widehat{\boldsymbol{\Sigma}}_l) \sim \mathcal{N}(\widehat{\mathbf{p}}_l^{(rec)}, \widehat{\boldsymbol{\Sigma}}_l^{(rec)}). \qquad (49)$$

This completes the proof.

F. PROOF OF REMARK 3

The GaGP method allows us to parallelize the computation of test user $x-$cordinate estimates $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ and their 2σ error-bars $\pm 2\sqrt{[\widehat{\mathbf{C}}_x^{(\mathrm{GaGP})}]_{ll}}$ for the $\widehat{L}$ test users. As may be observed from (19)-(20), this is because the computation of $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ and $[\widehat{\mathbf{C}}_x^{(\mathrm{GaGP})}]_{ll}$ for any user l does not rely on the computation of $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_{l'}$ and $[\widehat{\mathbf{C}}_x^{(\mathrm{GaGP})}]_{l'l'}$ of any other user l'.

Also, when the RecGaGP method is employed, we note from (21) and (23) that we can compute the reconstructed test RSS vectors $\{\widehat{\mathbf{p}}_l^{(rec)}\}$ and the residual noise covariances $\{\widehat{\mathbf{\Sigma}}_l^{(rec)}\}$ for the $\widehat{L}$ test users in parallel. Once the $\widehat{\mathbf{p}}_l^{(rec)}$ and $\widehat{\mathbf{\Sigma}}_l^{(rec)}$ values are available, we can also parallelize the computation of the $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{RGP})}]_l$ and $[\widehat{\mathbf{C}}_x^{(\mathrm{RGP})}]_{ll}$ values for the $\widehat{L}$ test users. As may be observed from (24)-(25), this is because the computation of $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{RGP})}]_l$ and $[\widehat{\mathbf{C}}_x^{(\mathrm{RGP})}]_{ll}$ does not rely on the computation of $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{RGP})}]_{l'}$ and $[\widehat{\mathbf{C}}_x^{(\mathrm{RGP})}]_{l'l'}$, $\forall l \neq l'$.

When parallelizing the computation of the location estimates (and their 2σ error-bars), we can also reduce the cost of each GP method by pre-computing few terms which are reusable across the $\widehat{L}$ users. For example, we can pre-compute $\widetilde{\mathbf{\Phi}}^{-1}$ and $\widetilde{\mathbf{\Phi}}^{-1}\widetilde{\mathbf{x}}$ when using (14) to compute the $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{CGP})}]_l$ and $[\widehat{\mathbf{C}}_x^{(\mathrm{CGP})}]_{ll}$ in the conventional GP method. Similarly, we can pre-compute the terms α', $\boldsymbol{\psi}$, $\boldsymbol{\xi}$, and $N(\widetilde{\mathbf{p}}_i; \widetilde{\mathbf{p}}_j, 2\mathbf{B})$ when using (19)-(20) to calculate the $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{GaGP})}]_l$ and $[\widehat{\mathbf{C}}_x^{(\mathrm{GaGP})}]_{ll}$ in GaGP and when using (24)-(25) to calculate the $[\widehat{\boldsymbol{\mu}}_x^{(\mathrm{RGP})}]_l$ and $[\widehat{\mathbf{C}}_x^{(\mathrm{RGP})}]_{ll}$ in RecGaGP.

G. MATHEMATICAL FORMULAE

(1) [Conditioning a joint Gaussian distribution [37] (pg. 200)] If $\mathbf{a}$ is a $W \times 1$ Gaussian random vector with $\mathbf{a} \sim \mathcal{N}(\mathbf{u}, \mathbf{A})$ and the random variables in $\mathbf{a}$ are partitioned into two sets $\mathbf{a}_\zeta = [[\mathbf{a}]_1\,[\mathbf{a}]_2, \ldots [\mathbf{a}]_W]^T \in \mathbb{R}^w$ and $\mathbf{a}_{\zeta'} = [[\mathbf{a}]_{w+1}\,[\mathbf{a}]_{w+2}, \ldots [\mathbf{a}]_W]^T \in \mathbb{R}^{W-w}$ such that

$$\begin{bmatrix}\mathbf{a}_\zeta \\ \mathbf{a}_{\zeta'}\end{bmatrix} \sim \mathcal{N}\left[\begin{pmatrix}\mathbf{u}_\zeta \\ \mathbf{u}_{\zeta'}\end{pmatrix}, \begin{pmatrix}\mathbf{A}_{\zeta\zeta} & \mathbf{A}_{\zeta\zeta'} \\ \mathbf{A}_{\zeta\zeta'}^T & \mathbf{A}_{\zeta'\zeta'}\end{pmatrix}\right], \tag{50}$$

then $\mathbf{a}_\zeta|\mathbf{a}_{\zeta'}$ and $\mathbf{a}_{\zeta'}|\mathbf{a}_\zeta$ are also Gaussian such that

$$\begin{aligned}\mathbf{a}_\zeta|\mathbf{a}_{\zeta'} &\sim \mathcal{N}(\mathbf{u}_\zeta + \mathbf{A}_{\zeta\zeta'}\mathbf{A}_{\zeta'\zeta'}^{-1}(\mathbf{a}_{\zeta'} - \mathbf{u}_{\zeta'}), \mathbf{A}_{\zeta\zeta} \\ &\quad - \mathbf{A}_{\zeta\zeta'}\mathbf{A}_{\zeta'\zeta'}^{-1}\mathbf{A}_{\zeta\zeta'}^T), \\ \mathbf{a}_{\zeta'}|\mathbf{a}_\zeta &\sim \mathcal{N}(\mathbf{u}_{\zeta'} + \mathbf{A}_{\zeta\zeta'}^T\mathbf{A}_{\zeta\zeta}^{-1}(\mathbf{a}_\zeta - \mathbf{u}_\zeta), \mathbf{A}_{\zeta'\zeta'} \\ &\quad - \mathbf{A}_{\zeta\zeta'}^T\mathbf{A}_{\zeta\zeta}^{-1}\mathbf{A}_{\zeta\zeta'}).\end{aligned} \tag{51}$$

(2) [Product of Gaussian expressions] Let us consider three deterministic $W-$dimensional vectors $\mathbf{a}$, $\mathbf{u}$ and $\mathbf{u}_0$, and two $W \times W$ positive definite matrices $\mathbf{A}$ and $\mathbf{A}_0$. The product of Gaussian expressions $N(\mathbf{a}; \mathbf{u}, \mathbf{A})$ and $N(\mathbf{a}; \mathbf{u}_0, \mathbf{A}_0)$ is then given by

$$\begin{aligned}&N(\mathbf{a}; \mathbf{u}, \mathbf{A})N(\mathbf{a}; \mathbf{u}_0, \mathbf{A}_0) \\ &\qquad = N(\mathbf{u}; \mathbf{u}_0, \mathbf{A} + \mathbf{A}_0)N(\mathbf{a}; \mathbf{u}_1, \mathbf{A}_1),\end{aligned}$$

where

$$\mathbf{A}_1 = (\mathbf{A}^{-1} + \mathbf{A}_0^{-1})^{-1} \quad \text{and } \mathbf{u}_1 = \mathbf{A}_1(\mathbf{A}^{-1}\mathbf{u} + \mathbf{A}_0^{-1}\mathbf{u}_0). \tag{52}$$

(3) [Covariance of a random vector] The covariance matrix $\mathbf{A}$ of an $W-$dimensional vector $\mathbf{a}$ has elements given by

$$[\mathbf{A}]_{ww} = \mathbb{E}_{[\mathbf{a}]_w}(([\mathbf{a}]_w)^2) - (\mathbb{E}_{[\mathbf{a}]_w}([\mathbf{a}]_w))^2, \quad \forall w = 1, \ldots, W. \tag{53}$$

(4) [Integral of a quadratic expression with respect to a Gaussian random vector] If $\mathbf{u}_0$ is a deterministic $W \times 1$ vector, $\mathbf{A}_0$ is a deterministic positive definite matrix of size $W \times W$, and $\mathbf{a}$ is a $W \times 1$ Gaussian random vector with mean $\mathbf{u}$ and covariance $\mathbf{A}$, i.e., $\mathbf{a} \sim \mathcal{N}(\mathbf{u}, \mathbf{A})$, then

$$\begin{aligned}&\int (\mathbf{a} - \mathbf{u}_0)^T\mathbf{A}_0^{-1}(\mathbf{a} - \mathbf{u}_0)\mathcal{N}(\mathbf{a}; \mathbf{u}, \mathbf{A})\,d\mathbf{a} \\ &\qquad = (\mathbf{u}_0 - \mathbf{u})^T\mathbf{A}_0^{-1}(\mathbf{u}_0 - \mathbf{u}) + \mathrm{Tr}(\mathbf{A}_0^{-1}\mathbf{A}).\end{aligned} \tag{54}$$

(5) [Derivative of the log-determinant and inverse of a matrix] If $\mathbf{a}$ is a vector of unknown variables, $\mathbf{A}$ is a positive definite matrix whose entries are functions of $\mathbf{a}$, and $\nabla_\mathbf{a}(\mathbf{A})$ is the matrix of element-wise derivatives of $\mathbf{A}$ with respect to $\mathbf{a}$, then

$$\nabla_\mathbf{a}(\log|\mathbf{A}|) = \mathrm{Tr}(\mathbf{A}^{-1}\nabla_\mathbf{a}(\mathbf{A})), \tag{55}$$

and

$$\nabla_\mathbf{a}(\mathbf{A}^{-1}) = -\mathbf{A}^{-1}\nabla_\mathbf{a}(\mathbf{A})\mathbf{A}^{-1}. \tag{56}$$

H. NUMERICALLY STABLE IMPLEMENTATION OF MATRIX OPERATIONS IN THE STUDIED GP METHODS AND THE ASSOCIATED COST OF COMPUTATION

Let $\mathbf{A}$ be a matrix of size $W \times W$, $\boldsymbol{\chi}$ be the Cholesky factor of $\mathbf{A}$ such that $\mathbf{A} = \boldsymbol{\chi}\boldsymbol{\chi}^T$, $\mathbf{u}$ be a vector of size $W \times 1$, and $\nabla_a\mathbf{A}$ be the matrix of element-wise derivatives of $\mathbf{A}$ w.r.t a.

(i) $[\mathbf{A}^{-1}\mathbf{u}]$: The matrix-vector multiplication $\mathbf{A}^{-1}\mathbf{u}$ is stably calculated as $\boldsymbol{\chi}^{-T}(\boldsymbol{\chi}^{-1}\mathbf{u})$. Cholesky decomposition of $\mathbf{A}$ requires $\mathcal{O}(W^3)$ operations and the product $\boldsymbol{\chi}^{-T}(\boldsymbol{\chi}^{-1}\mathbf{u})$ requires $\mathcal{O}(W^2)$ operations when computed via forward and backward substitution.

(ii) $[\mathrm{Tr}(\mathbf{A}^{-1}\nabla_a\mathbf{A})]$: The matrix product $\mathrm{Tr}(\mathbf{A}^{-1}\nabla_a\mathbf{A})$ can be stably implemented as $\mathrm{Tr}(\boldsymbol{\chi}^{-T}(\boldsymbol{\chi}^{-1}\nabla_a\mathbf{A}))$. The Cholesky factor $\boldsymbol{\chi}$ of $\mathbf{A}$ can be obtained in $\mathcal{O}(W^3)$ operations. After obtaining $\boldsymbol{\chi}$, the calculation of $\mathrm{Tr}(\boldsymbol{\chi}^{-T}(\boldsymbol{\chi}^{-1}\nabla_a\mathbf{A}))$ requires another $\mathcal{O}(W^3)$ operations.

(iii) $[\log(|\mathbf{A}|)]$: The term $\log(|\mathbf{A}|)$ can be stably calculated as $2\sum_{w=1}^{W}\log([\boldsymbol{\chi}]_{ww})$. This requires $\mathcal{O}(W^3)$ operations to obtain $\boldsymbol{\chi}$ and $\mathcal{O}(W)$ operations to calculate the sum.

REFERENCES

[1] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*. New York, NY, USA: Wiley, 2001.

[2] P. Keikhosrokiani, N. Mustaffa, N. Zakaria, and M. Sarwar, "Wireless positioning techniques and location-based services: a literature review," *Multimedia and Ubiquitous Engineering* (Lecture Notes in Electrical Engineering), vol. 240. Rotterdam, The Netherlands: Springer-Verlag, 2013, pp. 785–797.

[3] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[4] A. Zanella, "Best practice in RSS measurements and ranging," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2662–2686, 4th Quart., 2016.

[5] V. Savic and E. Larsson, "Fingerprinting-based positioning in distributed massive MIMO systems," in *Proc. IEEE 82nd Veh. Tech. Conf. (VTC Fall)*, Sep. 2015, pp. 1–5.

[6] K. N. R. S. V. Prasad, E. Hossain, and V. K. Bhargava, "A numerical approximation method for RSS-based user positioning in distributed massive MIMO," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2017.

[7] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–2553, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128606003227

[8] T. Van Nguyen, Y. Jeong, H. Shin, and M. Z. Win, "Machine learning for wideband localization," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1357–1380, Jul. 2015.

[9] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Identification and mitigation of non-line-of-sight conditions using received signal strength," in *Proc. IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2013, pp. 667–674.

[10] K. Yu and Y. J. Guo, "Statistical NLOS identification based on AOA, TOA, and signal strength," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 274–286, Jan. 2009.

[11] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Netw.*, vol. 10, no. 6, pp. 701–710, 2004.

[12] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1996–2018, 4th Quart., 2014.

[13] D. A. Tran and T. Nguyen, "Localization in wireless sensor networks based on support vector machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 7, pp. 981–994, Jul. 2008.

[14] R. Battiti, M. Brunato, and A. Villani, "Statistical learning theory for location fingerprinting in wireless LANs," Dept. Inf. Telecomun., Univ. Trento, Trento, Italy, Tech. Rep. DIT-02-0086, 2002.

[15] X. Wang, L. Gao, and S. Mao, "BiLoc: Bi-modal deep learning for indoor localization with commodity 5GHz WiFi," *IEEE Access*, vol. 5, pp. 4209–4220, 2017.

[16] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[17] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, and A. Criminisi, "Measuring neural n et robustness with constraints," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2016, pp. 2613–2621.

[18] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, "GPPS: A Gaussian process positioning system for cellular networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2003, pp. 579–586.

[19] B. Ferris, D. Haehnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," *Roboti., Sci. Syst.*, vol. 2, pp. 303–310, Aug. 2006.

[20] M. Aravecchia and S. Messelodi, "Gaussian process for rss-based localisation," in *Proc. IEEE 10th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Nov. 2014, pp. 654–659.

[21] S. Kumar, R. M. Hegde, and N. Trigoni, "Gaussian process regression for fingerprinting based localization," *Ad Hoc Netw.*, vol. 51, pp. 1–10, Nov. 2016.

[22] N. Garcia, H. Wymeersch, E. G. Larsson, A. M. Haimovich, and M. Coulon, "Direct localization for massive MIMO," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2475–2487, May 2017.

[23] S. A. Shaikh and A. M. Tonello, "Localization based on angle of arrival in EM lens-focusing massive MIMO," in *Proc. IEEE Int. Conf. Consum. Electron.-Berlin (ICCE-Berlin)*, Sep. 2016, pp. 124–128.

[24] A. Hu, T. Lv, H. Gao, Z. Zhang, and S. Yang, "An ESPRIT-based approach for 2-D localization of incoherently distributed sources in massive MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 996–1011, Oct. 2014.

[25] A. Guerra, F. Guidi, and D. Dardari, "Position and orientation error bound for wideband massive antenna arrays," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 853–858.

[26] A. Shahmansoori, G. E. Garcia, G. Destino, G. Seco-Granados, and H. Wymeersch, "5G position and orientation estimation through millimeter wave MIMO," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–6.

[27] K. N. R. S. V. Prasad, E. Hossain, and V. K. Bhargava, "Low-dimensionality of noise-free RSS and its application in distributed massive MIMO," *IEEE Wireless Commun. Lett.*, to be published, doi: 10.1109/LWC.2017.2787764.

[28] K. N. R. S. V. Prasad, E. Hossain, and V. K. Bhargava, "Machine learning methods for user positioning with uplink RSS in distributed massive MIMO," *IEEE Trans. Wireless Commun.*, to be published. [Online]. Available: https://arxiv.org/abs/1801.06619

[29] A. Girard *et al.*, "Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2003, pp. 545–552.

[30] J. Q. Candela, A. Girard, J. Larsen, and C. E. Rasmussen, "Propagation of uncertainty in Bayesian kernel models—Application to multiple-step ahead forecasting," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2003, pp. II-701-1–II-701-4.

[31] H. Bijl, T. B. Schon, J.-W. Wingerden, and M. Verhaegen, "System identification through online sparse Gaussian process regression with input noise," *IFAC J. Syst. Control*, vol. 2, pp. 1–11, Dec. 2016.

[32] L. S. Muppirisetty, T. Svensson, and H. Wymeersch, "Spatial wireless channel prediction under location uncertainty," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1031–1044, Feb. 2016.

[33] M. Frohle, L. S. Muppirisetty, and H. Wymeersch, "Channel gain prediction for multi-agent networks in the presence of location uncertainty," in *Proc. IEEE Int. Conf Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 3911–3915.

[34] M. Frohle, T. Charalambous, I. Nevat, and H. Wymeersch, "Channel prediction with location uncertainty for ad-hoc networks," *IEEE Trans. Signal Inf. Process. Netw.*, to be published, doi: 10.1109/TSIPN.2017.2705425.

[35] M. Malmirchegini and Y. Mostofi, "On the spatial predictability of communication channels," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 964–978, Mar. 2012.

[36] A. McHutchon and C. E. Rasmussen, "Gaussian process training with input noise," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, Dec. 2011, pp. 1341–1349.

[37] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[38] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[39] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philos. Mag.*, vol. 2, no. 6, pp. 559–572, 1901.

[40] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.

[41] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2006.

[42] *Further Advancements for E-UTRA Physical Layer Aspects (Release 9)* document TS 36.814, 3GPP, Mar. 2010.

[43] K. N. R. S. V. Prasad, E. Hossain, and V. K. Bhargava, "Energy efficiency in massive MIMO-based 5G networks: opportunities and challenges," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 86–94, Jun. 2017.

[44] J. Salo *et al.*, "Practical introduction to LTE radio planning," Eur. Commun. Engg., Tekniikantie, Finland, White Paper, Nov. 2010.

[45] D. Katselis, E. Kofidis, A. Rontogiannis, and S. Theodoridis, "Preamble-based channel estimation for CP-OFDM and OFDM/OQAM systems: A comparative study," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2911–2916, May 2010.

K. N. R. SURYA VARA PRASAD received the B.Tech. degree in electrical engineering from IIT Bhubaneswar, India, in 2012, and the M.A.Sc. degree in electrical and computer engineering from The University of British Columbia, Vancouver, in 2016, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. His current research focus is on machine learning and its applications in wireless telecommunication networks. At IEEE COMSNETS 2014, he was a recipient of the Best Demo & Exhibits Award.

EKRAM HOSSAIN (F'15) received the Ph.D. degree in electrical engineering from the University of Victoria, Canada, in 2001. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Canada. His current research interests include design, analysis, and optimization of wireless/mobile communication networks, cognitive radio systems, and network economics. He has authored/edited several books in these areas. He is a Member (Class of 2016) with the College of the Royal Society of Canada. He is also a member of the IEEE Press Editorial Board. He was a recipient of several research awards including the 2017 IEEE Communications Society Best Survey Paper Award, the IEEE Vehicular Technology Conference (VTC-Fall 2016) Best Student Paper Award as a co-author, the IEEE Communications Society Transmission, Access, and Optical Systems Technical Committee's Best Paper Award in IEEE Globecom 2015, the University of Manitoba Merit Award in 2010, 2014, and 2015 (for Research and Scholarly Activities), the 2011 IEEE Communications Society Fred Ellersick Prize Paper Award, and the IEEE Wireless Communications and Networking Conference 2012 Best Paper Award. He was also a recipient of the 2017 IEEE Communications Society Technical Committee on Green Communications & Computing Distinguished Technical Achievement Recognition Award for outstanding technical leadership and achievement in green wireless communications and networking. He was elevated to an IEEE Fellow for spectrum management and resource allocation in cognitive and cellular radio networks. He serves as an Editor for the IEEE Wireless Communications. Previously, he served as an Editor for the IEEE Transactions on Mobile Computing from 2007 to 2012, an Area Editor for the IEEE Transactions on Wireless Communications in the area of resource management and multiple access from 2009 to 2011, an Editor for the IEEE Journal on Selected Areas in Communications-Cognitive Radio Series from 2011 to 2014, and the Editor-in-Chief for the IEEE Communications Surveys and Tutorials from 2012 to 2016. Also, he was listed as a Clarivate Analytics Highly Cited Researcher in computer science in 2017. He is an elected Member of the Board of Governors of the IEEE Communications Society from 2018 to 2020. He was a Distinguished Lecturer of the IEEE Communications Society from 2012 to 2015. He is currently a Distinguished Lecturer of the IEEE Vehicular Technology Society. He is also a registered Professional Engineer in the province of Manitoba, Canada.

VIJAY K. BHARGAVA (S'70–M'74–SM'82–F'92–LF'13) was born in Beawar, India, in 1948. He came to Canada in 1966 and received the B.A.Sc., M.A.Sc., and Ph.D. degrees from Queen's University, Kingston, in 1970, 1972, and 1974, respectively. He is currently a Professor with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, where he served as the Department Head from 2003 to 2008. Previously, he was with the Indian Institute of Science from 1974 to 1975, the University of Waterloo in 1976, Concordia University from 1976 to 1984, and the University of Victoria from 1984 to 2003. He has held visiting appointments at the Ecole Polytechnique de Montreal, the NTT Research Laboratory, the Tokyo Institute of Technology, the University of Indonesia, The Hong Kong University of Science and Technology, Tohoku University, and Friedrich Alexander University, Germany. He is currently a Honorary Professor with the University of Electronic Science and Technology of China, Chengdu, and a Gandhi Distinguished Professor with IIT Bombay. He is currently with the Institute for Scientific Information Highly Cited list. He served as the Founder and President of Binary Communications Inc., from 1983 to 2000. He has co-authored /co-editor of seven books the latest of which is *Wireless-Powered Communication Networks* (Cambridge University Press, 2016). He is a Fellow of The Royal Society of Canada, The Canadian Academy of Engineering and the Engineering Institute of Canada. He is a Foreign Fellow of the National Academy of Engineering, India, and has served as a Distinguished Visiting Fellow of the Royal Academy of Engineering, U.K. He was a recipient of the awards for his teaching, research, and service to the IEEE. He also a recipient of the latest awards is the Killam Prize in Engineering awarded by the Canada Council for the Arts and the Humboldt Research Prize awarded by the Alexander von Humboldt Foundation of Germany. A long-time Volunteer of the IEEE, he has served as a Director of Region 7, in 1992 and 1993, a Vice President of Regional Activities Board-RAB (now MGA), from 1994 to 1995, a President of the Information Theory Society in 2000, a President of the IEEE Communications Society, in 2012 and 2013, and he is currently the Director for Division III, in 2018and 2019. He has served as an Editor of the IEEE Transactions on Communications and as the Editor-in-Chief of the IEEE Transactions on Wireless Communications.

SHANKHANAAD MALLICK received the B.Sc. and M.Sc. degrees in electrical and electronic engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2006 and 2008, respectively, and the Ph.D. degree from The University of British Columbia (UBC), Vancouver, Canada, in 2015. From 2015 to 2017, he was a Post-Doctoral Fellow with the Information Theory and Systems Laboratory, UBC. His current research interests include signal processing, and design and optimization of wireless communication networks.