



Cloud-Assisted Cooperative Localization for Vehicle Platoons: A Turbo Approach

An Liu , Senior Member, IEEE, Lixiang Lian , Student Member, IEEE, Vincent Lau, Fellow, IEEE, Guanying Liu, and Min-Jian Zhao, Member, IEEE

Abstract—Due to the high resolution of angles of arrivals (AoAs) provided by the massive MIMO base station in 5G wireless systems, it is promising to integrate 5G-based localization technology into autonomous driving to improve the accuracy and robustness of vehicle localization. In this paper, we investigate the problem of 5G cloud-assisted cooperative localization for vehicle platoons. The existing 5G-based localization algorithms focused on single-user localization and are not efficient for the localization of vehicle platoon where the positions of the vehicles are highly correlated. To the best of our knowledge, cloud-assisted cooperative localization tailored to vehicle platoons has not been studied before. To address this challenging problem, we first propose a Gamma-Markov-Group-Sparse (GMGS) model to capture the joint distribution of the vehicle positions in a vehicle platoon. Then we formulate the vehicle platoon cooperative localization as a sparse Bayesian inference (SBI) problem. The existing standard SBI algorithms such as variational Bayesian inference (VBI) and approximate message passing (AMP) cannot be applied to our platoon localization problem due to the complicated GMGS prior and the ill-conditioned measurement matrix. As such, we propose a novel turbo vehicle platoon cooperative localization (Turbo-VPCL) algorithm to fully exploit the correlations of the vehicle positions (as captured by the GMGS prior) under the ill-conditioned measurement matrix. Simulation results verify that the proposed Turbo-VPCL can achieve significant gain over the state-of-art SBI algorithms.

Index Terms—5G, Cloud-assisted cooperative localization, Vehicle platoon, Turbo compressive sensing.

I. INTRODUCTION

AUTONOMOUS driving is a key component in future smart city. An important application scenario of autonomous driving is autonomous vehicle platooning, where a group of vehicles form a platoon to exploit the aerodynamics between the vehicles so as to reduce fuel consumption [1]. In order to achieve a stable and effective platoon with high speed, we

need to have highly accurate, real-time and robust location estimates of the vehicles in a platoon. Existing localization technologies based on Global Positioning System (GPS) or local sensors such as radars, cameras and inertial sensors have their own pros and cons. For example, there are situations when GPS signal is weak. The radars have low localization accuracy and cameras may not work well when the visibility is bad. Therefore, it is necessary to integrate more reliable localization technologies to further enhance the robustness of vehicle localization in vehicle platooning [2].

In a vehicle platoon, the positions of vehicles are highly correlated, which can be potentially exploited to develop more reliable cooperative localization technology. 5G systems may employ a cloud radio access network (C-RAN) architecture [3], where the base stations (BSs) forward the received signals to the cloud via high-speed front-haul links. All baseband processings are performed at the cloud. Such a C-RAN architecture facilitates cloud-assisted vehicle platoon cooperative localization, in which the cloud can jointly estimate the positions of all vehicles based on the received signals from multiple BSs, so as to exploit the highly correlated spatial structure of vehicle platoon to achieve more robust and efficient localization. In addition, 5G wireless systems are equipped with massive MIMO which can achieve a high resolution of angles of arrivals (AoAs) [4]. As a result, 5G cloud-assisted cooperative localization can also achieve high-accuracy localization. Integrating 5G cloud-assisted cooperative localization into autonomous driving provides a cost-effective solution for robust and high-accuracy localization of vehicle platoon.

Recently, a number of works have studied 5G-based localization technology. Localization algorithms in [5]–[7] employ multiple BSs/access points (APs), and the object's position is estimated based on AoAs, signal phases or time of arrivals (ToAs) through triangulation (AoA based methods) or trilateration/multilateration (range based methods). The localization algorithms in [8] and [9] employ one AP/BS, and the positioning algorithms are based on the estimation of multiple channel parameters of multi-path environment, such as received signal strength (RSS) and AoA in [8], AoA/angle of departure (AoD) and ToA in [9]. Then the object's position can be recovered through simple geometrical manipulations. All the above works belong to the indirect localization method, i.e., the intermediate parameters such as the AoA/AoD, ToA and RSS are first estimated, and then the user's position is determined by triangulation/multilateration or the geometrical

Manuscript received December 19, 2018; revised June 21, 2019, October 11, 2019, and December 16, 2019; accepted December 17, 2019. Date of publication January 6, 2020; date of current version January 24, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Antti Tolli. This work was supported in part by the Science and Technology Program of Shenzhen, China, under Grant JCYJ20170818113908577, and in part by the National Science Foundation of China under Project 61571383. The work of An Liu was supported by the China Recruitment Program of Global Young Experts. (Corresponding authors: Vincent Lau; Min-Jian Zhao.)

A. Liu, G. Liu, and M.-J. Zhao are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: anliu@zju.edu.cn; btslgy@126.com; mjzhao@zju.edu.cn).

L. Lian is with the Department of ECE, The Hong Kong University of Science and Technology, Hong Kong (e-mail: llianab@connect.ust.hk).

V. Lau is with the HKUST Shenzhen Research Institute, Hong Kong (e-mail: eeknlau@ece.ust.hk).

Digital Object Identifier 10.1109/TSP.2020.2964198

manipulations. In the presence of multipath, the performance of such indirect localization method will be degraded due to the inability to correctly detect and/or estimate the AoA/RSS of the line-of-sight (LOS) path [10]. To tackle this problem, various direct localization methods [11], [12] have been proposed, where the user location is estimated directly from the received signal, without relying on intermediate parameters, such as the AoA/AoD/ToA/RSS. In particular, a direct localization approach is proposed in [10] for 5G massive MIMO systems based on a novel CS framework, which exploits the fact that LOS components received at multiple base stations (BSs) must originate from a common location whereas non-LOS (NLOS) components have arbitrary AoAs. In direct localization, a fusion center needs to collect the received signals from multiple BSs to estimate the user's locations. Such a requirement can be satisfied by the C-RAN architecture in 5G systems [3].

The above works focus on localization algorithm design for general applications. In [2], the feasibility of providing 5G-based localization for autonomous driving has been verified. While the recent studies have shown the potential of integrating 5G-based localization technology into autonomous driving, most of the existing 5G-based localization algorithms focus on the single-user localization problem. In a vehicle platoon, a group of vehicles travel together at approximately the same speed by following a lead vehicle. Therefore, their positions are highly correlated, i.e., the inter-vehicle distance of two neighbor vehicles is usually within a certain range. Moreover, the inter-vehicle distance is a random variable due to randomness in the vehicle speed control. As such, cloud-assisted cooperative localization can utilize such kind of stochastic correlations between vehicle positions to improve position estimation precision. However, there are several technical challenges of applying cloud-assisted cooperative localization to vehicle platoons.

- **Sparse Modeling of Cooperative Localization for Platoons:** High-precision direct localization can be achieved by exploiting the sparse properties of massive MIMO channel in 5G systems. However, the direct localization algorithm in [10] is based on a simple group-sparse model which cannot exploit the correlations between vehicle positions in a vehicle platoon. In traditional Bayesian based methods, such as sparse Bayesian learning (SBL) and variational Bayesian inference (VBI) in [13], [14], a two-layer hierarchical prior is used to promote independent and identically distributed (i.i.d.) or group sparsity. However, such two-layer hierarchical prior is not flexible enough to capture the highly correlated spatial structure of the vehicle platoon. There is no tractable sparse probability model available for cloud-assisted vehicle platoon cooperative localization. To address this challenge, we propose a Gamma-Markov-Group-Sparse (GMGS) model to capture the joint distribution of the vehicle positions in a vehicle platoon. Specifically, a Markov model with gamma distributed transition probability is used to capture the highly correlated spatial structure of vehicle platoon. Conditioned on Markov model for vehicle platoon, a group-sparse model is used to capture the structured sparsity of massive MIMO channel. To the best of our knowledge, this is the

first work that proposes such a three-layer hierarchical probability model for efficient cooperative localization.

- **Sparse Bayesian Inference with Ill-conditioned Measurement Matrix:** The cloud-assisted cooperative localization can be formulated as a sparse Bayesian inference (SBI) problem [15] with a GMGS prior. However, the existing SBI methods, such as SBL and VBI in [13], [14], cannot handle the GMGS prior due to the complicated correlation between vehicle positions. Recently, approximate message passing (AMP) and turbo approach have been combined to handle more complicated priors [16], [17]. However, AMP-based algorithms only work well for i.i.d. or partial orthogonal measurement matrices. Unfortunately, the measurement matrix associated with the vehicle platoon localization problem is ill-conditioned. To address this challenge, we propose a turbo vehicle platoon cooperative localization (Turbo-VPCL) algorithm which can fully exploit the correlations of the vehicle positions (as captured by the GMGS prior) under the ill-conditioned measurement matrix.

The rest of the paper is organized as follows. In Section II, we give the system model for the cloud-assisted vehicle platoon localization. In Section III, we propose the GMGS model to capture the joint distribution of vehicle positions in a vehicle platoon, and formulate the resulting SBI problem. The proposed Turbo-VPCL algorithm is presented in Section IV. The simulation results are given in Section V to verify the advantages of the proposed solution, and the conclusion is given in Section VI.

Notation: $\mathcal{CN}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the probability density function of a complex Gaussian random vector \mathbf{x} with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. $\Gamma(\rho; a, b)$ denotes a Gamma hyperprior with shape parameter a and rate parameter b . \propto denotes equality after scaling. $\text{tr}(\mathbf{X})$ means the trace of matrix \mathbf{X} . $\mathbf{X} = \text{diag}(\mathbf{x})$ means matrix \mathbf{X} is a diagonal matrix with \mathbf{x} on the diagonal. \mathbf{X}^{-1} , \mathbf{X}^T and \mathbf{X}^H denote the inverse, transpose and conjugate transpose of matrix \mathbf{X} , respectively. $x = \mathcal{O}(a)$ for $a > 0$ denotes that $\exists k_1, k_2 > 0$, such that $k_2 \cdot a \leq x \leq k_1 \cdot a$.

II. SYSTEM MODEL

A. Architecture of Cloud-Assisted Localization System

Consider a vehicle platoon consisting of K consecutive vehicles indexed by $k = 1, 2, \dots, K$, as illustrated in Fig. 1. Each vehicle is equipped with a 5G transceiver of one antenna so that it can enjoy both data service and localization service from the 5G wireless network. The vehicle platoon is in the communication range of N massive MIMO BSs, each equipped with a massive array of $M \gg 1$ antennas. The N BSs are connected to a cloud via high-speed front-haul links. Consider a two-dimensional (2D) scenario, where the k -th vehicle is located at $\mathbf{p}_k = [p_k^x, p_k^y] \in \mathbb{R}^2$ in a 2D area \mathcal{R} along a road known a priori,¹ and the center of gravity of the n -th BS arrays are located at a known position $\tilde{\mathbf{p}}_n = [\tilde{p}_n^x, \tilde{p}_n^y] \in \mathbb{R}^2$.

¹Only the road map must be known such that the grid can be determined. However, there is no need to know the entire map. Since the road map is usually static and contains much less information, it is easy to obtain the road map in practice.

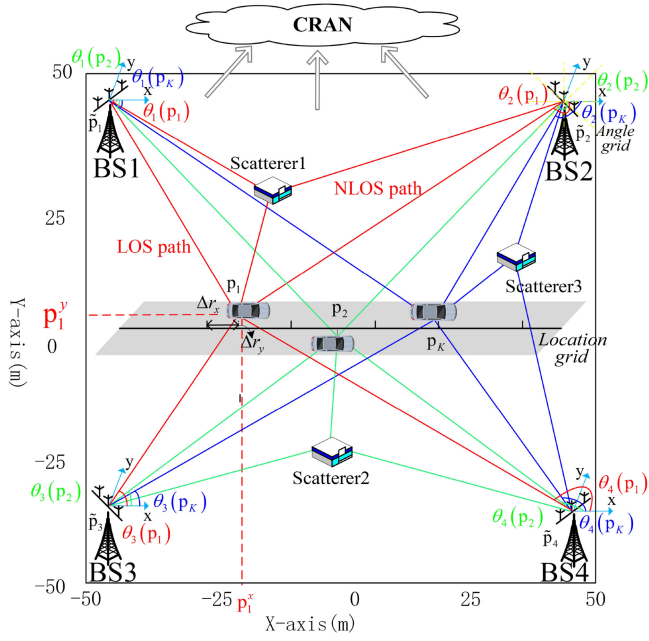


Fig. 1. An illustration of system model and position grid.

At each time slot, all the vehicles simultaneously transmit T pilot symbols $s_k(t)$, $t = 1, \dots, T$, $k = 1, \dots, K$ for localization, where $s_k(t)$ denote the t -th pilot symbol transmitted from vehicle k . The received pilot signals at BS n are given by

$$\mathbf{z}_n(t) = \sum_{k=1}^K \tilde{\mathbf{h}}_{n,k} s_k(t) + \mathbf{w}_n(t), t = 1, \dots, T, \quad (1)$$

where $\tilde{\mathbf{h}}_{n,k}$ is the channel between vehicle k and BS n , $\mathbf{w}_n(t)$ is the AWGN vector with i.i.d. entries of zero mean and variance σ^2 . The received pilot signals at the N BSs are passed to the cloud for cooperative localization of all vehicles based on the wireless channel model and statistical model of vehicle locations in a platoon, as will be detailed in the next two subsections. Then the estimated vehicle locations at the cloud are passed to the vehicles via 5G downlink transmission.

We assume that the cloud can build the mapping between the pilot sequences $s_k(t)$'s and the geometrical order of the vehicles. In 5G C-RAN, each user (vehicle) is assigned with a unique ID and a pilot sequence. Meanwhile, the vehicles in the platoon can easily determine their geometrical order by communicating with each other. Then each vehicle can send its geometrical order to the cloud via the uplink of the C-RAN. Using this information, the cloud can associate each unique user ID with both its geometrical order and its pilot sequence.

B. Wireless Channel Model

We consider flat fading² massive MIMO channel with limited scattering. For a large number of antennas, the spatial resolution

²The proposed algorithm can be extended to frequency-selective fading channels by sending the training signals on one subband, and estimate the vehicles' locations based on the received measurements at this specific subband from all the BSs.

of the angular basis increases and hence, under limited scattering environment (which is usually the case for a massive MIMO BS elevated at a high place), the channel will be sparse under the angular basis [18]. Assume there are $L_{n,k} + 1$ paths for the channel between vehicle k and BS n (one LOS path and $L_{n,k} \ll M$ NLOS paths), as illustrated in Fig. 1. Let $\theta_n(\mathbf{p}_k)$ and $\theta_{n,k}^l$, $l = 1, \dots, L_{n,k}$ denote the AoAs of the LOS path and the l -th NLOS path between vehicle k and BS n , respectively. Note that the LOS AoA $\theta_n(\mathbf{p}_k)$ is related to the position of vehicle k through [10]

$$\theta_n(\mathbf{p}_k) = \arctan \left(\frac{p_k^y - \tilde{p}_n^y}{p_k^x - \tilde{p}_n^x} \right) + \pi \cdot \mathbb{I}(p_k^x < \tilde{p}_n^x),$$

where the angle is computed with respect to the x-axis and anticlockwise, and $\mathbb{I}(E)$ is one if the logical expression E is true. The channel vector between vehicle k and BS n is modeled as [10]

$$\tilde{\mathbf{h}}_{n,k} = h_{n,k} \mathbf{a}_n(\theta_n(\mathbf{p}_k)) + \sum_{l=1}^{L_{n,k}} h_{n,k}^l \mathbf{a}_n(\theta_{n,k}^l), \quad (2)$$

where $\mathbf{a}_n(\theta) \in \mathbb{C}^M$ is the array response vector at BS n for a path with AoA θ and it depends on the array configuration and the antenna pattern, $h_{n,k}$ and $h_{n,k}^l$ are the angular domain channel coefficients associated with the LOS path and l -th NLOS path between vehicle k and BS n , respectively.

C. Statistical Model for Locations of the Vehicle Platoon

We assume that the inter-vehicle distances $\{d_1, d_2, \dots, d_{K-1}\}$ over the vehicle platoon have i.i.d. distributions, where d_k denotes the distance between vehicle k and vehicle $k+1$. Furthermore, the distribution for the inter-vehicle distances is the gamma distribution [1]:

$$p(d_k) \propto \frac{(d_k - d)^{\alpha-1}}{\lambda^\alpha \Gamma(\alpha)} \exp \left(-\frac{d_k - d}{\lambda} \right), d_k \geq d, \quad (3)$$

where d is the minimum allowable inter-vehicle distance, $\alpha \geq 1$ is the common shape parameter, and λ is the scale parameter of the gamma distribution. In practice, the parameters α and λ can be estimated using the collected data via the expectation-maximization (EM) method [1]. This is a common assumption in the literature and has been verified through empirical studies [1].

III. SPARSE BAYESIAN INFERENCE FORMULATION FOR CLOUD-ASSISTED VEHICLE PLATOON COOPERATIVE LOCALIZATION

In this section, we first provide a sparse representation for massive MIMO channels. Then, we propose a GMGS prior to capture the joint distribution of the vehicle positions in a vehicle platoon. Finally, based on the sparse representation, we formulate the cloud-assisted vehicle platoon cooperative localization as an SBI problem with GMGS prior.

A. Sparse Representation of Massive MIMO Channels in the Vehicle Platoon System

Although it is possible to directly estimate the locations $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K\}$ based on the signal model in (2) and (1) using

the maximum a posterior probability (MAP) methods, the resulting problem is non-convex and the MAP algorithm can easily get stuck in a local optimum with a bad localization performance. To overcome this problem, we consider a grid-based solution [10] to exploit the sparsity of massive MIMO for improved localization performance. Specifically, we introduce a uniform grid of Q positions along the center of the road

$$\mathcal{P} = \{\mathbf{r}_1, \dots, \mathbf{r}_Q\} \subset \mathcal{R},$$

as shown in Fig. 1,³ and a uniform grid of \widetilde{M} AoAs over $[0, 2\pi)$

$$\mathcal{A} = \{\vartheta_1, \dots, \vartheta_{\widetilde{M}}\} \subset [0, 2\pi).$$

In practice, the true positions/AoAs usually do not lie exactly on the grid point in \mathcal{P}/\mathcal{A} . In this case, there will be mismatches between the true positions/AoAs and the nearest grid point in \mathcal{P}/\mathcal{A} , as shown in Fig. 1. To overcome this issue, we introduce an off-grid basis for the sparse representation. Specifically, let $q_k = \text{argmin}_q \|\mathbf{p}_k - \mathbf{r}_q\|$ denote the index of the position grid point nearest to the position of vehicle k . We introduce a position offset vector $\Delta\mathbf{r} = [\Delta\mathbf{r}_1; \dots; \Delta\mathbf{r}_Q]$ where $\Delta\mathbf{r}_{q_k} = \mathbf{p}_k - \mathbf{r}_{q_k}$, $k = 1, \dots, K$ and $\Delta\mathbf{r}_q = 0, \forall q \notin \{q_1, \dots, q_K\}$. Similarly, let $m_{n,k}^l = \text{argmin}_m \|\theta_{n,k}^l - \vartheta_m\|$ denote the index of the AoA grid point nearest to the AoA of the l -th NLOS path between vehicle k and BS n . For each channel between vehicle k and BS n , we introduce an AoA offset vector $\Delta\vartheta_{n,k} = [\Delta\vartheta_{n,k}^1, \dots, \Delta\vartheta_{n,k}^{\widetilde{M}}]^T$ such that $\Delta\vartheta_{n,k}^l = \theta_{n,k}^l - \vartheta_{m_{n,k}^l}$, $l = 1, \dots, L_{n,k}$ and $\Delta\vartheta_{n,k}^m = 0, \forall m \notin \{m_{n,k}^1, \dots, m_{n,k}^{L_{n,k}}\}$.

With the above definitions of position/AoA offsets, we can define the off-grid basis for the LOS paths at BS n (i.e., LOS array response matrix at BS n with position offset $\Delta\mathbf{r}$) as

$$\mathbf{A}_n^L(\Delta\mathbf{r}) = [\mathbf{a}_n(\theta_n(\mathbf{r}_1 + \Delta\mathbf{r}_1)), \dots, \mathbf{a}_n(\theta_n(\mathbf{r}_Q + \Delta\mathbf{r}_Q))],$$

and the off-grid basis for the NLOS paths between vehicle k and BS n (i.e., the NLOS array response matrix between vehicle k and BS n with AoA offset $\Delta\vartheta_{n,k}$) as

$$\mathbf{A}_n^{\text{NL}}(\Delta\vartheta_{n,k}) = [\mathbf{a}_n(\vartheta_1 + \Delta\vartheta_{n,k}^1), \dots, \mathbf{a}_n(\vartheta_{\widetilde{M}} + \Delta\vartheta_{n,k}^{\widetilde{M}})].$$

Then the channel vector $\widetilde{\mathbf{h}}_{n,k}$ between vehicle k and BS n in (2) has a sparse representation given by

$$\widetilde{\mathbf{h}}_{n,k} = \mathbf{A}_n^L(\Delta\mathbf{r}) \mathbf{x}_{n,k} + \mathbf{A}_n^{\text{NL}}(\Delta\vartheta_{n,k}) \mathbf{y}_{n,k}, \quad (4)$$

where $\mathbf{x}_{n,k} \in \mathbb{C}^Q$ and $\mathbf{y}_{n,k} \in \mathbb{C}^{\widetilde{M}}$ are called the sparse LOS and NLOS channel vectors between vehicle k and BS n , respectively. To be more specific, the q -th entry of $\mathbf{x}_{n,k}$, denoted by $x_{n,k,q}$, represents the complex gain of a LOS path (associated with vehicle k) from the q -th off-grid location $\mathbf{r}_q + \Delta\mathbf{r}_q$ to BS n . The m -th entry of $\mathbf{y}_{n,k}$, denoted by $y_{n,k,m}$, represents the complex gain of a NLOS path from vehicle k to BS n with AoA $\vartheta_m + \Delta\vartheta_{n,k}^m$. Each $\mathbf{x}_{n,k}$ only has one non-zero element corresponding to the true position of vehicle k , and the index of

³Considering the fact that the road is usually not too wide, it is reasonable to consider a one-dimensional location grid along the center of the road, which helps to reduce the total number of grid points and thus the complexity of the algorithm compared to a 2D grid.

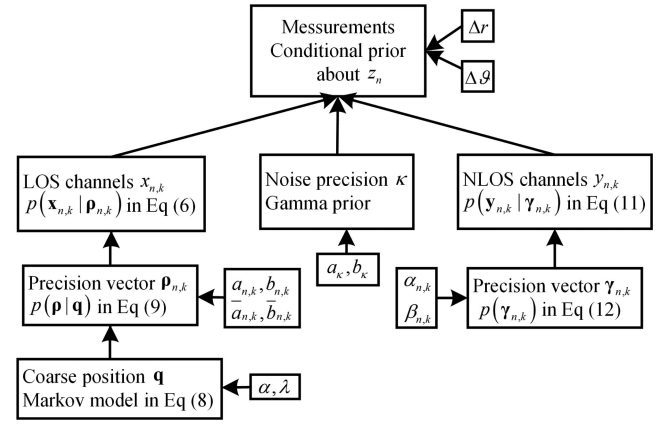


Fig. 2. Prior model for vehicle platoon cooperative localization.

the non-zero element of LOS channels $\mathbf{x}_{1,k}, \dots, \mathbf{x}_{N,k}$ between vehicle k and different BSs are the same, as given by q_k . In other words, the LOS channel vectors $\mathbf{x}_{1,k}, \dots, \mathbf{x}_{N,k}$ can be grouped into Q blocks with block size N , where the q -th block is constituted by $[x_{1,k,q}, \dots, x_{N,k,q}]$ and only the q_k -th block is non-zero. Similarly, each $\mathbf{y}_{n,k}$ only has $L_{n,k} \ll M$ non-zero elements corresponding to the AoAs of the $L_{n,k}$ NLOS paths.

In the above sparse representation, the coarse position \mathbf{r}_{q_k} of vehicle k is determined by the index q_k of the non-zero element of $\mathbf{x}_{n,k}, \forall n$, and the mismatch between the coarse position \mathbf{r}_{q_k} and the true position \mathbf{p}_k is given by the corresponding position offset $\Delta\mathbf{r}_{q_k}$, as illustrated in Fig. 1 for vehicle 1. Therefore, the vehicle's exact position is jointly determined by the coarse position \mathbf{r}_{q_k} and the corresponding position offset $\Delta\mathbf{r}_{q_k}$. In Section IV, we will discuss how to jointly recover vehicles' coarse positions and refine the offset parameters to learn the vehicle's position. Before that, we will firstly introduce a probability model to jointly capture the spatial correlation between vehicles' positions q_k and the group sparsity of LOS channels $\mathbf{x}_{n,k}, \forall n$.

B. Gamma-Markov-Group-Sparse Model for Vehicle Platoon Cooperative Localization

The sparse channel representation in (4) lacks a probability model for $\mathbf{x}_{n,k}$ (which depends on the joint distribution of the positions of the vehicles in the platoon). Such a probability model provides the foundation for exploiting the specific sparse channel structures induced by the correlation between the vehicle positions, which is paramount to the robust and high-accuracy vehicle platoon localization based on 5G-massive MIMO. Besides the spatial correlations among vehicle positions, in cooperative localization, the LOS channels between vehicle $k, \forall k \in \{1, \dots, K\}$ and different BSs all originate from the same vehicle position, which induces a group sparsity structure of the LOS channels. To capture such a complicated sparse structure associated with the grid-based cooperative localization approach, we propose a GMGS probabilistic model, which enables a closed-form update solution for the localization problem, as detailed in Section IV.

Fig. 2 illustrates the high level architecture of the GMGS model for the sparse LOS channel vectors $\mathbf{x}_{n,k}, \forall n, k$. The GMGS prior is a three-layer hierarchical prior. The first layer of random variables is $\mathbf{q} = [q_1, \dots, q_K]^T$, where q_k represents the (coarse) position of vehicle k along the center of the road. The second layer of random variables is $\boldsymbol{\rho} = [\boldsymbol{\rho}_1; \dots; \boldsymbol{\rho}_N]$, where $\boldsymbol{\rho}_n = [\rho_{n,1}; \dots; \rho_{n,K}]$, $\rho_{n,k} = [\rho_{n,k,1}, \dots, \rho_{n,k,Q}]^T$ and $\rho_{n,k,q}$ denotes the precision of $x_{n,k,q}$ (i.e., $1/\rho_{n,k,q}$ denotes the variance of $x_{n,k,q}$). The third layer of random variables is $\mathbf{x} = \{\mathbf{x}_{n,k}, \forall n, k\}$. Then the GMGS prior distribution (joint distribution of $\mathbf{x}, \boldsymbol{\rho}, \mathbf{q}$) is given by

$$p(\mathbf{x}, \boldsymbol{\rho}, \mathbf{q}) = \underbrace{p(\mathbf{q})}_{\text{Coarse position}} \underbrace{p(\boldsymbol{\rho}|\mathbf{q})}_{\text{LOS Channel precision}} \underbrace{p(\mathbf{x}|\boldsymbol{\rho})}_{\text{LOS channel}}. \quad (5)$$

To allow the flexibility to model local characteristics of the signal, a nonstationary Gaussian prior distribution with a distinct precision $\rho_{n,k,q}$ for each element $x_{n,k,q}$ of \mathbf{x} is considered, i.e.,

$$p(\mathbf{x}|\boldsymbol{\rho}) = \prod_{n,k} \mathcal{CN}(\mathbf{x}_{n,k}; \mathbf{0}, \text{diag}(\boldsymbol{\rho}_{n,k})^{-1}), \quad (6)$$

where $\boldsymbol{\rho}_{n,k} = [\rho_{n,k,1}, \rho_{n,k,2}, \dots, \rho_{n,k,Q}]^T$. The coarse position vector \mathbf{q} forms a Markov model and the conditional probability $p(\boldsymbol{\rho}|\mathbf{q})$ of the LOS channel precision $\boldsymbol{\rho}$ forms a group-sparse model, as detailed in Subsection III-B1 and III-B2, respectively. For completeness, the conventional two-layer hierarchical prior for the sparse NLOS channel vectors $\mathbf{y}_{n,k}$'s will also be introduced in Subsection III-B3. Moreover, in practice, the noise precision $\kappa = \sigma^{-2}$ is usually unknown and we model it as a Gamma hyperprior $p(\kappa) = \Gamma(\kappa; a_\kappa, b_\kappa)$, where we set $a_\kappa, b_\kappa \rightarrow 0$ as in [15] so as to obtain a broad hyperprior. The priors for NLOS channel vectors and noise precision are also shown in Fig. 2.

1) *Markov Model of Coarse Position \mathbf{q}* : The Markov model of the coarse position vector \mathbf{q} is derived from the gamma distributed statistical model for inter-vehicle distances in a vehicle platoon, as given in (3). Since $\Delta q_k = q_{k+1} - q_k$ represents the (coarse) distance between vehicle k and $k+1$ along the center of road, $\Delta q_k \in \mathbb{Z}_{++}$ can be modeled using the following sampled gamma distribution:

$$p(\Delta q_k) \propto \frac{(\Delta q_k - d)^{\alpha-1}}{\lambda^\alpha \Gamma(\alpha)} \exp\left(-\frac{\Delta q_k - d}{\lambda}\right), \quad \Delta q_k \geq d. \quad (7)$$

Based on (7), the joint distribution of the coarse position vector \mathbf{q} can be modeled using a Markov chain as

$$p(\mathbf{q}) = p(q_1) \prod_{k=1}^{K-1} p(q_{k+1}|q_k), \quad (8)$$

where the transition probability $p(q_{k+1}|q_k)$ is modeled as a sampled gamma distribution:

$$p(q_{k+1}|q_k) \propto \frac{(\Delta q_k - d)^{\alpha-1}}{\lambda^\alpha \Gamma(\alpha)} \exp\left(-\frac{\Delta q_k - d}{\lambda}\right),$$

for $q_{k+1} \in \{q_k + d, q_k + d + 1, \dots, Q - (K - k - 1)d\}$. In practice, the prior distribution $p(q_1)$ can be obtained via the other localization technologies (such as GNSS or cameras), or

the previous soft estimation of the position from the proposed localization algorithm.

2) *Group-Sparse Model for LOS Channel Precision $\boldsymbol{\rho}$* : The precision vector $\boldsymbol{\rho}$ in (6) is further constrained by imposing a Gamma prior distribution

$$p(\boldsymbol{\rho}|\mathbf{q}) = \prod_{k=1}^K \prod_{n=1}^N \prod_{q=1}^Q p(\rho_{n,k,q}|q_k), \quad (9)$$

where

$$p(\rho_{n,k,q}|q_k) = \Gamma(\rho_{n,k,q}; a_{n,k}, b_{n,k})^{\mathbb{I}(q=q_k)} \Gamma(\rho_{n,k,q}; \bar{a}_{n,k}, \bar{b}_{n,k})^{\mathbb{I}(q \neq q_k)},$$

$\Gamma(\rho; a, b)$ is a Gamma hyperprior with shape parameter a and rate parameter b ; $a_{n,k}, b_{n,k}$ are the shape and rate parameters of the channel precision $\rho_{n,k,q}$, conditioned on the event when the q -th grid location \mathbf{r}_q is close to vehicle k 's position (i.e., $q = q_k$); $\bar{a}_{n,k}, \bar{b}_{n,k}$ are the shape and rate parameters of the channel precision $\rho_{n,k,q}$, conditioned on the opposite event. The Gamma prior is selected because it is conjugate to the Gaussian, and hence the associated Bayesian inference can be performed in closed form [13], [19].

The prior model in (6) and (9) can be used to promote the group sparsity of the LOS channels, as explained below. When $q = q_k$, there is an active LOS path (associated with vehicle k) from the q -th off-grid location $\mathbf{r}_q + \Delta \mathbf{r}_q$ to BS n . In this case, the shape and rate parameters $a_{n,k}, b_{n,k}$ should be chosen such that $\frac{a_{n,k}}{b_{n,k}} = \mathbb{E}[\rho_{n,k,q}] = \mathcal{O}(\frac{1}{g_{n,k}})$, where $g_{n,k}$ is the path gain (variance) of the active LOS path between vehicle k and BS n , so that the corresponding entries will have large probability to be deviated from zero. On the other hand, when $q \neq q_k$, there is no active LOS path (associated with vehicle k) from the q -th off-grid location $\mathbf{r}_q + \Delta \mathbf{r}_q$ to BS n . In this case, the shape and rate parameters $\bar{a}_{n,k}, \bar{b}_{n,k}$ should be chosen such that $\frac{\bar{a}_{n,k}}{\bar{b}_{n,k}} = \mathbb{E}[\rho_{n,k,q}] \gg 1$, so that the complex channel gain $x_{n,k,q}$ associated with the non-active LOS path is close to zero. In the simulation, we find that $\frac{\bar{a}_{n,k}}{\bar{b}_{n,k}} \geq 1000$ can achieve a desirable performance. In summary, the conditional priors in (9) assign smaller precisions to the block $[x_{1,k,q}, \dots, x_{N,k,q}]$, $q = q_k$, to make the block elements deviated from zero, and assign larger precisions to the blocks $[x_{1,k,q}, \dots, x_{N,k,q}]$, $\forall q \neq q_k$, to make the block elements concentrate on zero. Therefore, $\mathbf{x}_{k,n}$, $n = 1, \dots, N$ form a group-sparse model. Note that the MAP estimate of $x_{n,k,q}$ not only depends on the prior distribution, but also the received pilot signals (measurements). The proposed prior distribution can capture the first-order sparsity structure of the LOS/NLOS channel vectors, and combining such priors with the signal measurements allows the accurate estimation of the vehicle positions, as shown by the simulation results.

Remark 1: In practice, the channel coefficients tend to have zero mean due to the aggregation of a large number of independent subpaths with random phases. Therefore, if we do not have prior knowledge about the channel coefficients, it is reasonable to set the mean of the Gaussian prior distribution of the channel coefficient to be zero, as in (6). In some cases, the

C-RAN has some prior knowledge, e.g., it knows that a non-zero channel coefficient $x_{n,k,q}$ is around the value $\hat{h}_{n,k,q}$ but with some uncertainty. In this case, we can model the LOS channel vector \mathbf{x} using a nonstationary Gaussian prior with a distinct mean $\xi_{n,k,q}$ and distinct precision $\rho_{n,k,q}$ for each element $x_{n,k,q}$ of \mathbf{x} , where the prior distribution of the mean vector ξ can be modeled as $p(\xi|\mathbf{q}) = \prod_{k,n,q} p(\xi_{n,k,q}|q_k)$ with

$$p(\xi_{n,k,q}|q_k) = \mathcal{CN}(\xi_{n,k,q}; 0, \epsilon^2)^{\mathbb{I}(q \neq q_k)} \mathcal{CN}(\xi_{n,k,q}; \hat{h}_{n,k,q}, \epsilon^2)^{\mathbb{I}(q = q_k)},$$

where $\epsilon > 0$ is a small number. In other words, when $x_{n,k,q}$ is non-zero ($q = q_k$), its mean is approximately equal to $\hat{h}_{n,k,q}$, otherwise ($q \neq q_k$), its mean is approximately equal to zero. The precision vector ρ is modeled as a Gamma distribution as in (9). The proposed algorithm can also be extended to such a case.

3) *Two-Layer Hierarchical Prior for NLOS Channel*: The sparse NLOS channel vectors $\mathbf{y}_{n,k}$'s are modeled using the following two-layer hierarchical prior, which has been widely adopted in sparse VBI literature. The first layer of random variables is $\gamma = [\gamma_1; \dots; \gamma_N]$, where $\gamma_n = [\gamma_{n,1}; \dots; \gamma_{n,K}]$, $\gamma_{n,k} = [\gamma_{n,k,1}, \dots, \gamma_{n,k,\tilde{M}}]^T$ and $\gamma_{n,k,m}$ denotes the precision of $y_{n,k,m}$ (i.e., $1/\gamma_{n,k,m}$ denotes the variance of $y_{n,k,m}$). The second layer of random variables is $\mathbf{y} = \{\mathbf{y}_{n,k}, \forall n, k\}$. Then the two-layer hierarchical prior for NLOS channel (joint distribution of \mathbf{y}, γ) is given by

$$p(\mathbf{y}, \gamma) = \underbrace{p(\gamma)}_{\text{NLOS channel precision}} \underbrace{p(\mathbf{y}|\gamma)}_{\text{NLOS channel}}, \quad (10)$$

where the conditional probability $p(\mathbf{y}|\gamma)$ has a product form $p(\mathbf{y}|\gamma) = \prod_{k,n} p(\mathbf{y}_{n,k}|\gamma_{n,k})$ and each $p(\mathbf{y}_{n,k}|\gamma_{n,k})$ is modeled as a Gaussian prior distribution

$$p(\mathbf{y}_{n,k}|\gamma_{n,k}) = \mathcal{CN}(\mathbf{y}_{n,k}; \mathbf{0}, \text{diag}(\gamma_{n,k})^{-1}). \quad (11)$$

For $\gamma_{n,k,m}, \forall k, n, m$, we model it as independent Gamma distributions, i.e., $p(\gamma) = \prod_{k=1}^K \prod_{n=1}^N p(\gamma_{n,k})$ with

$$p(\gamma_{n,k}) = \prod_{m=1}^{\tilde{M}} \Gamma(\gamma_{n,k,m}; \alpha_{n,k}, \beta_{n,k}), \quad (12)$$

where we set $\alpha_{n,k}, \beta_{n,k}$ to be a small number. This two-stage hierarchical prior gives the marginal distribution

$$p(\mathbf{y}_{n,k}) \propto \prod_{m=1}^{\tilde{M}} \left(\beta_{n,k} + |y_{n,k,m}|^2 \right)^{-(\alpha_{n,k} + 3/2)}, \quad (13)$$

which promotes sparsity due to the distribution's heavy tails and sharp peak at zero with small $\alpha_{n,k}$ and $\beta_{n,k}$ [20].

C. Sparse Bayesian Inference Formulation with GMGS Prior

Using the sparse channel representation in (4), the received signal in (1) can be rewritten as a CS model with unknown parameters in the measurement matrix [21], [22] as

$$\mathbf{z}_n = \mathbf{A}_n(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n) \begin{bmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{bmatrix} + \mathbf{w}_n, \forall n, \quad (14)$$

where $\mathbf{x}_n = [\mathbf{x}_{n,1}; \dots; \mathbf{x}_{n,K}]$, $\mathbf{y}_n = [\mathbf{y}_{n,1}; \dots; \mathbf{y}_{n,K}]$, $\mathbf{w}_n = [\mathbf{w}_n(1); \dots; \mathbf{w}_n(T)]$, $\mathbf{z}_n = [\mathbf{z}_n(1); \dots; \mathbf{z}_n(T)]$, $\Delta \boldsymbol{\vartheta}_n = [\Delta \vartheta_{n,1}; \dots; \Delta \vartheta_{n,K}]$ and the measurement matrix $\mathbf{A}_n(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n)$ is given by

$$\mathbf{A}_n(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n) = \begin{bmatrix} \mathbf{A}_n^L(1) & \mathbf{A}_n^{\text{NL}}(1) \\ \vdots & \vdots \\ \mathbf{A}_n^L(T) & \mathbf{A}_n^{\text{NL}}(T) \end{bmatrix} \in \mathbb{C}^{MT \times (Q + \tilde{M})K}, \quad (15)$$

where $\mathbf{A}_n^L(t) = [s_1(t)\mathbf{A}_n^L(\Delta \mathbf{r}) \dots s_K(t)\mathbf{A}_n^L(\Delta \mathbf{r})] \in \mathbb{C}^{M \times QK}$ and $\mathbf{A}_n^{\text{NL}}(t) = [s_1(t)\mathbf{A}_n^{\text{NL}}(\Delta \boldsymbol{\vartheta}_{n,1}) \dots s_K(t)\mathbf{A}_n^{\text{NL}}(\Delta \boldsymbol{\vartheta}_{n,K})] \in \mathbb{C}^{M \times \tilde{M}K}$.

Our primary goal is to estimate the position parameter \mathbf{q} and position offset $\Delta \mathbf{r}$ given the observations $\mathbf{z} = [\mathbf{z}_1; \dots; \mathbf{z}_N]$ in model (14). In particular, for given offsets $\Delta \mathbf{r}$ and $\Delta \boldsymbol{\vartheta} = [\Delta \boldsymbol{\vartheta}_1; \dots; \Delta \boldsymbol{\vartheta}_N]$, we are interested in computing the conditional marginal posteriors $p(q_k|\mathbf{z}, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}), \forall k$ (i.e., perform Bayesian inference for $q_k, \forall k$), where

$$\begin{aligned} p(q_k|\mathbf{z}, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}) &\propto \int \mathbf{x}, \boldsymbol{\rho}, \mathbf{y}, \gamma, \mathbf{q}_{-k}, \kappa p(\mathbf{z}, \mathbf{x}, \boldsymbol{\rho}, \mathbf{q}, \mathbf{y}, \gamma, \kappa | \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}) d_{-q_k} \\ &= \int \mathbf{x}, \boldsymbol{\rho}, \mathbf{y}, \gamma, \mathbf{q}_{-k}, \kappa p(\mathbf{x}, \boldsymbol{\rho}, \mathbf{q}) p(\mathbf{y}, \gamma) p(\kappa) \\ &\quad \times \prod_{n=1}^N p(\mathbf{z}_n | \mathbf{x}_n, \mathbf{y}_n, \kappa, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n) d_{-q_k}, \end{aligned} \quad (16)$$

$$\begin{aligned} p(\mathbf{z}_n | \mathbf{x}_n, \mathbf{y}_n, \kappa, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n) &= \mathcal{CN}\left(\mathbf{z}_n; \mathbf{A}_n(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n) \begin{bmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{bmatrix}, \kappa^{-1} \mathbf{I}\right). \end{aligned}$$

We use $\int \mathbf{x}, \boldsymbol{\rho}, \mathbf{y}, \gamma, \mathbf{q}_{-k}, \kappa (\cdot) d_{-q_k}$ to denote integration over $\mathbf{x}, \boldsymbol{\rho}, \mathbf{y}, \gamma, \kappa$ and $q_{k'}', \forall k' \neq k$. On the other hand, the position offset $\Delta \mathbf{r}$ and AoA offset $\Delta \boldsymbol{\vartheta}$ are obtained by maximizing the likelihood function as follows

$$\begin{aligned} [\Delta \mathbf{r}^*, \Delta \boldsymbol{\vartheta}^*] &= \underset{\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}}{\text{argmax}} \ln p(\mathbf{z} | \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}) \\ &= \underset{\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}}{\text{argmax}} \ln \int_{\mathbf{v}} p(\mathbf{z}, \mathbf{v} | \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}) d\mathbf{v}, \end{aligned} \quad (17)$$

where $\mathbf{v} = \{\mathbf{x}, \boldsymbol{\rho}, \mathbf{q}, \mathbf{y}, \gamma, \kappa\}$. Once we obtain the EM estimate of $\Delta \mathbf{r}^*$ and $\Delta \boldsymbol{\vartheta}^*$, and the associated conditional marginal posteriors $p(q_k|\mathbf{z}, \Delta \mathbf{r}^*, \Delta \boldsymbol{\vartheta}^*), \forall k$, we can obtain the MAP estimation of q_k as $q_k^* = \underset{q_k}{\text{argmax}} p(q_k|\mathbf{z}, \Delta \mathbf{r}^*, \Delta \boldsymbol{\vartheta}^*)$. Finally, the MAP estimation of the position of vehicle k is given by $\mathbf{r}_{q_k^*} + \Delta \mathbf{r}_{q_k^*}^*$.

It is intractable to calculate the exact posterior in (16) because the associated factor graph has loops. In the next section, we propose a Turbo-VPCL algorithm which approximately calculates the marginal posteriors $p(q_k|\mathbf{z}, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}), \forall k$ by combining the message passing and VBI approaches via the turbo framework, and uses the EM method to find an approximate solution for (17).

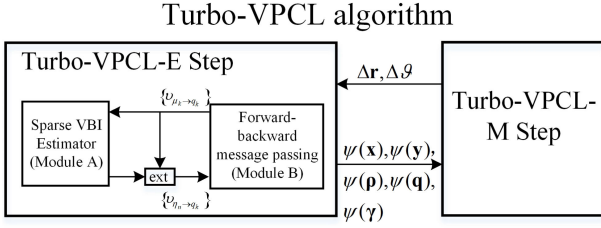


Fig. 3. Block diagram of the Turbo-VPCL algorithm.

IV. TURBO-VPCL ALGORITHM

In the proposed Turbo-VPCL algorithm, the BSs first send the received signals $z_n, \forall n$ to the cloud. Then the cloud runs the Turbo-VPCL algorithm to jointly estimate the positions of the K vehicles based on the received signals $z_n, \forall n$ exploiting the correlation among the vehicle positions to enhance the localization performance. Such a requirement can be satisfied by the C-RAN architecture in 5G systems [3] with small signaling overhead due to the high-bandwidth, high-data rate and low-latency front-haul links between the BSs and cloud.

A. Outline of the Turbo-VPCL Algorithm

The basic idea of Turbo-VPCL is to simultaneously maximize the marginal log-likelihood with respect to the hyperparameters $\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}$ in (17) and approximate the marginal posterior $p(q_k | z, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$ in (16) with a tractable variational distribution. In summary, the Turbo-VPCL algorithm performs iterations between the following two major steps until convergence (as illustrated in Fig. 3).

- **Turbo-VPCL-E Step:** For given $\Delta \mathbf{r}$ and $\Delta \boldsymbol{\vartheta}$, calculate the approximate marginal posterior of $p(q_k | z, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$, denoted as $\psi(q_k)$, by combining the VBI and message passing via the turbo framework, as elaborated in Section IV-C and Section IV-D. We can also obtain an approximate posterior of $p(v | z, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$, denoted as $\psi(v)$.
- **Turbo-VPCL-M Step:** Given $p(v | z, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}) \approx \psi(v)$, construct a surrogate function (lower bound) for the objective function $\ln p(z | \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$, then maximize the surrogate function with respect to $\Delta \mathbf{r}$ and $\Delta \boldsymbol{\vartheta}$, as elaborated in Section IV-B.

In the following, we first elaborate the Turbo-VPCL-M Step, which is an extension of the EM method. Note that the surrogate function in Turbo-VPCL-M Step requires the calculation of the posterior $p(v | z, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$, and we will elaborate how to approximately calculate the posterior $p(v | z, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$ in the Turbo-VPCL-E Step.

B. Turbo-VPCL-M Step

It is difficult to directly maximize the log-likelihood function because there is no closed-form expression due to the multi-dimensional integration over v . To address this challenge, we use the EM method to find an approximate solution of (17) by maximizing a sequence of surrogate functions of $\ln p(z | \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$ with respect to $\Delta \mathbf{r}$ and $\Delta \boldsymbol{\vartheta}$. Specifically, at each iteration i , the

EM method chooses the following surrogate function:

$$F(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i) = \int p(v | z, \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i) \ln \left(\frac{p(v, z | \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})}{p(v | z, \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)} \right) dv, \quad (18)$$

at the current iterate $\Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i$, which satisfies the following properties:

$$F(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i) \leq \ln p(z | \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}), \forall \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}, \quad (19)$$

$$F(\Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i) = \ln p(z | \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i), \quad (20)$$

$$\frac{\partial F(\Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)}{\partial \Delta \mathbf{r}} = \frac{\partial \ln p(z | \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)}{\partial \Delta \mathbf{r}}, \quad (21)$$

$$\frac{\partial F(\Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)}{\partial \Delta \boldsymbol{\vartheta}} = \frac{\partial \ln p(z | \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)}{\partial \Delta \boldsymbol{\vartheta}}. \quad (22)$$

Then in the Turbo-VPCL-M Step of the i -th iteration, we update $\Delta \mathbf{r}$ and $\Delta \boldsymbol{\vartheta}$ as

$$[\Delta \mathbf{r}^{i+1}, \Delta \boldsymbol{\vartheta}^{i+1}] = \underset{\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}}{\operatorname{argmax}} F(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i). \quad (23)$$

However, the maximization problem in (23) is non-convex, and it is difficult to find the global optimal solution of (23). In this case, we can simply perform the following gradient update:

$$\Delta \mathbf{r}^{i+1} = \Delta \mathbf{r}^i + \tau_r^i \frac{\partial F(\Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)}{\partial \Delta \mathbf{r}}, \quad (24)$$

$$\Delta \boldsymbol{\vartheta}^{i+1} = \Delta \boldsymbol{\vartheta}^i + \tau_{\boldsymbol{\vartheta}}^i \frac{\partial F(\Delta \mathbf{r}^{i+1}, \Delta \boldsymbol{\vartheta}^i; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)}{\partial \Delta \boldsymbol{\vartheta}}, \quad (25)$$

where τ_r^i and $\tau_{\boldsymbol{\vartheta}}^i$ are the step sizes determined by the Armijo rule [14]. The detailed expressions of the gradients in (24) and (25) are given in Appendix A.

Based on the convergence of the EM algorithm, we have the following theorem for the convergence of the Turbo-VPCL algorithm.

Theorem 2 (Convergence of the Turbo-VPCL Algorithm): Suppose the surrogate function $F(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)$ satisfies (19)–(22). If at each iteration, we do inexact (gradient) update as in (24) and (25) for the offset vectors $\Delta \mathbf{r}$ and $\Delta \boldsymbol{\vartheta}$, the iterates generated by the Turbo-VPCL algorithm converge to a stationary point of Problem (17).

The proof can be found in Appendix B.

Therefore, if we can calculate the exact posterior $p(v | z, \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)$ for given $\Delta \mathbf{r}^i$ and $\Delta \boldsymbol{\vartheta}^i$, we can construct the surrogate function in (18) and the corresponding Turbo-VPCL algorithm converges to a stationary point of (17). Unfortunately, in our case, the exact posterior $p(v | z, \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)$ is intractable due to the loops in the factor graph. Thus, in the Turbo-VPCL-E Step, we propose to combine the message passing and VBI approaches via the turbo framework to find an alternative probability density function $\psi(v)$ to approximate the posterior $p(v | z, \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)$. The approximate posterior $\psi(v)$

TABLE I
FACTORS, DISTRIBUTIONS AND FUNCTIONAL FORMS IN FIG. 4

Factor	Distribution	Functional form
$g_{n,t}(\mathbf{x}_n, \mathbf{y}_n)$	$p(\mathbf{z}_n(t) \mathbf{x}_n, \mathbf{y}_n, \kappa, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n)$	$\mathcal{CN}(\mathbf{z}_n(t); [\mathbf{A}_n^L(t) \quad \mathbf{A}_n^{NL}(t)] \begin{bmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{bmatrix}, \kappa^{-1} \mathbf{I})$
$f_{n,k,q}(\mathbf{x}_{n,k,q}, \rho_{n,k,q})$	$p(\mathbf{x}_{n,k,q} \rho_{n,k,q})$	$\mathcal{CN}(\mathbf{x}_{n,k,q}; 0, \rho_{n,k,q}^{-1})$
$\phi_{n,k,m}(\mathbf{y}_{n,k,m}, \gamma_{n,k,m})$	$p(\mathbf{y}_{n,k,m} \gamma_{n,k,m})$	$\mathcal{CN}(\mathbf{y}_{n,k,m}; 0, \gamma_{n,k,m}^{-1})$
$\eta_k(\rho_{n,k,q} q_k)$	$p(\rho_{n,k,q} q_k)$	$\Gamma(\rho_{n,k,q}; a_{n,k}, b_{n,k})^{\mathbb{I}(q=q_k)} \Gamma(\rho_{n,k,q}; \bar{a}_{n,k}, \bar{b}_{n,k})^{\mathbb{I}(q \neq q_k)}$
$\mu(\mathbf{q})$	$p(\mathbf{q})$	$p(q_1) \prod_{k=1}^{K-1} p(q_{k+1} q_k)$

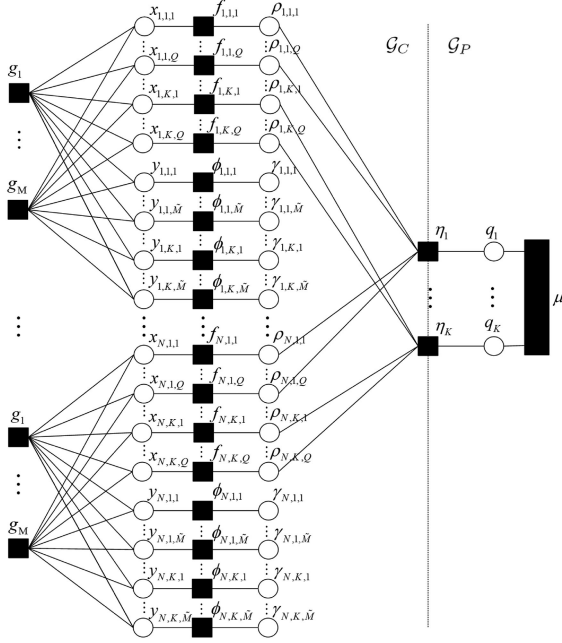


Fig. 4. Factor graph of the joint distribution in (16).

has a factorized form as

$$\psi(\mathbf{v}) = \psi(\mathbf{x}) \psi(\boldsymbol{\rho}) \psi(\mathbf{y}) \psi(\boldsymbol{\gamma}) \prod_k \psi(q_k), \quad (26)$$

where $\psi(\mathbf{x}) \approx p(\mathbf{x} | \mathbf{z}, \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)$, $\psi(\boldsymbol{\rho}) \approx p(\boldsymbol{\rho} | \mathbf{z}, \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)$, e.t.c. Based on the posterior approximation $\psi(\mathbf{v})$, we can construct a tractable surrogate function as

$$\hat{F}(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i) = \int \psi(\mathbf{v}) \ln \left(\frac{p(\mathbf{v}, \mathbf{z} | \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})}{\psi(\mathbf{v})} \right) d\mathbf{v},$$

which is expected to approximately satisfy (19)–(22). Therefore, after the convergence of the Turbo-VPCL algorithm with the tractable surrogate function, we not only obtain an approximate stationary solution $\Delta \hat{\mathbf{r}}, \Delta \hat{\boldsymbol{\vartheta}}$ of (17), but also the associated (approximate) conditional marginal posteriors $p(q_k | \mathbf{z}, \Delta \hat{\mathbf{r}}, \Delta \hat{\boldsymbol{\vartheta}}) \approx \psi(q_k; \Delta \hat{\mathbf{r}}, \Delta \hat{\boldsymbol{\vartheta}}), \forall k$.

C. Turbo-VPCL-E Step

The Turbo-VPCL-E step is based on a novel turbo framework which combines the message passing and VBI approaches. The factor graph of the joint distribution $p(\mathbf{z}, \mathbf{x}, \boldsymbol{\rho}, \mathbf{q}, \mathbf{y}, \boldsymbol{\gamma}, \kappa | \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$, denoted by \mathcal{G} , is illustrated in

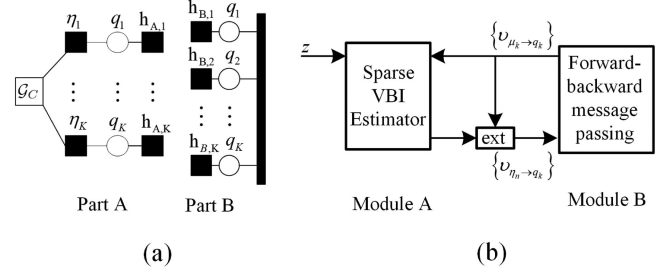


Fig. 5. (a) Decoupling of partitioned factor graph from Fig. 4 into Part A and Part B. (b) Modules of the Turbo-VPCL-E step and message flows between different modules.

Fig. 4, where the function expression of each factor node is listed in Table I. Because the factor graph \mathcal{G} contains many loops, exact inference is known to be NP-hard [23]. To facilitate the practical implementation, we now partition our factor graph into two sub-graphs: the channel subgraph \mathcal{G}_C and the position subgraph \mathcal{G}_P . The messages $\{\nu_{\eta_k \rightarrow q_k}(\cdot)\}$ form the outputs of \mathcal{G}_C and the inputs to the \mathcal{G}_P , while the messages $\{\nu_{\mu \rightarrow q_k}(\cdot)\}$ form the outputs of the \mathcal{G}_P and the inputs to the \mathcal{G}_C . From this, we can interpret the approximate message passing over \mathcal{G} as iterating between two modules, one of which performs inference on the subgraph \mathcal{G}_C that models the structure in the observation, and the other of which performs inference on the subgraph \mathcal{G}_P that models the spatial correlation between vehicle positions. To avoid self reinforcement, the two modules pass extrinsic information.

In the following, we refer to inference on the subgraph \mathcal{G}_C of Fig. 4 as Module A (Sparse VBI Estimator) and inference on the subgraph \mathcal{G}_P of Fig. 4 as Module B (Forward-Backward Message Passing). We now formally decouple these subtasks and represent each of them using a separate factor graph, as in Fig. 5(a). For this, we define two additional turbo-iteration constraint functions,

$$h_{A,k}(q_k) \triangleq \nu_{\mu \rightarrow q_k}(q_k), k = 1, \dots, K,$$

$$h_{B,k}(q_k) \triangleq \nu_{\eta_k \rightarrow q_k}(q_k), k = 1, \dots, K,$$

as shown in Fig. 5(a). For each turbo iteration, and output message $\nu_{\eta_k \rightarrow q_k}(q_k)$ can be calculated as

$$\nu_{\eta_k \rightarrow q_k}(q_k) \propto \psi(q_k) / h_{A,k}(q_k), \forall q_k, \quad (27)$$

which equals the (approximate) likelihood of q_k , since $h_{A,k}(q_k)$ acts as Module A's prior on q_k . As such, the output messages from Module A are extrinsic. Similar reasoning applies to

Module B's output as well. The two modules are executed iteratively until convergence. In the following, we first outline Module A and Module B. The details of Module A are presented in Subsection IV-D.

1) *Module A*: As previously discussed, the Module A accepts independent but non-identical prior probabilities on the position index q_k passed from Module B. We use $\pi_{k,q}$ to abbreviate $h_{A,k}(q_k = q)$, the prior probability of $q_k = q$ assumed by Module A. Thus, Module A performs the sparse VBI to approximate the conditional marginal posteriors $\psi(\mathbf{x})$, $\psi(\boldsymbol{\rho})$, $\psi(\mathbf{y})$, $\psi(\boldsymbol{\gamma})$, $\psi(\boldsymbol{\kappa})$ and $\psi(q_k)$ based on the following joint prior distribution:

$$\begin{aligned} & \hat{p}(\mathbf{x}, \boldsymbol{\rho}, \mathbf{q}, \mathbf{y}, \boldsymbol{\gamma}, \boldsymbol{\kappa}) \\ &= \hat{p}(\mathbf{q}) p(\boldsymbol{\rho}|\mathbf{q}) p(\mathbf{x}|\boldsymbol{\rho}) p(\boldsymbol{\gamma}) p(\mathbf{y}|\boldsymbol{\gamma}) p(\boldsymbol{\kappa}), \end{aligned} \quad (28)$$

where $\hat{p}(\mathbf{q})$ is given by

$$\hat{p}(\mathbf{q}) = \prod_k \left(\sum_{q=1}^Q \pi_{k,q} \delta(q_k - q) \right). \quad (29)$$

Note that the only difference between the prior in (28) and the original prior is that the Markov chain $p(\mathbf{q})$ is replaced by a prior $\hat{p}(\mathbf{q})$ with independent entries. The corresponding posterior distributions $\psi(\mathbf{x})$ and $\psi(\boldsymbol{\gamma})$ of \mathbf{x} and \mathbf{y} obtained by the sparse VBI will be given by (33) and (41), respectively, and the posterior distribution $\psi(q_k)$ of q_k will be given by (39). After that, the messages $\{\nu_{\eta_k \rightarrow q_k}(q_k)\}$ from Module A to B can be calculated based on (27).

2) *Module B*: Based on the messages $\{\nu_{\eta_k \rightarrow q_k}(\cdot)\}$ from Module A, Module B further exploits the correlation among the vehicle positions to improve the estimation performance by performing the forward-backward message passing algorithm over the Markov chain $p(\mathbf{q})$, as detailed in Appendix C. To be more specific, in Part B, the factor nodes $h_{B,k}$ incorporate the prior information from Module A, and the factor node μ incorporates the Markov group sparsity of the vehicle locations. After that, the messages $\{\nu_{\mu \rightarrow q_k}(\cdot)\}$ from Module B to Module A can be calculated according to the sum-product rule as given in (53).

Remark 3: When the measurement matrix is i.i.d. Gaussian or partial orthogonal, Turbo-AMP [16] or Turbo-CS [17] algorithm can be used to achieve approximate message passing over dense graphs. However, in our problem, the measurement matrix $\mathbf{A}_n(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n)$ in (15) is ill-conditioned and the performance of Turbo-AMP or Turbo-CS is very poor. To overcome this challenge, we combine the VBI [13] and message passing approaches [24] via the turbo framework to design the Turbo-VPCL-E step that can achieve approximate message passing over \mathcal{G} with a good performance.

Remark 4: As shown in [13], [25], the posterior approximations obtained by the VBI algorithm can be quite accurate for a general measurement matrix including the ill-conditioned matrix in the cooperative localization problem. Furthermore, it has been shown in [26] that two soft-input/output modules can help each other to efficiently learn the posteriors of the sparsity pattern and the posteriors of other unknown variables. As such, by a non-trivial combination of the VBI and message passing approaches based on the well-recognized turbo framework, the

proposed Turbo-VPCL algorithm can successfully exploit the sparsity structure in vehicle locations to obtain quite accurate posterior approximations. Indeed, the proposed algorithm is shown in the simulations to have significant gain over the various baselines. This also implies that the proposed algorithm can accurately approximate the posteriors of the unknown variables and provide quite accurate position estimations for the vehicle platoon.

D. Sparse VBI Estimator (Module A)

1) *Outline of Sparse VBI*: For convenience, we use \mathbf{v}^l to denote an individual variable in \mathbf{v} , such as $\mathbf{x}, \boldsymbol{\rho}, \mathbf{q}, \mathbf{y}, \boldsymbol{\gamma}, \boldsymbol{\kappa}$. Let $\mathcal{H} = \{l | \forall \mathbf{v}^l \in \mathbf{v}\}$. The approximate conditional marginal posterior could be calculated by minimizing the KLD between $\hat{p}(\mathbf{v}|\mathbf{z}, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$ and $\psi(\mathbf{v})$ subject to a factorized form constraint on $\psi(\mathbf{v})$ as in (26)

$$\begin{aligned} \mathcal{A}_{\text{VBI}} : \quad & \psi^*(\mathbf{v}) = \arg \min_{\psi(\mathbf{v})} \int \psi(\mathbf{v}) \ln \left(\frac{\psi(\mathbf{v})}{\hat{p}(\mathbf{v}|\mathbf{z}, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})} \right) d\mathbf{v} \\ & \text{s.t. (26), } \int \psi(\mathbf{v}^l) d\mathbf{v}^l = 1, \forall l \in \mathcal{H} \end{aligned} \quad (30)$$

where $\hat{p}(\mathbf{v}|\mathbf{z}, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$ is the posterior distribution of \mathbf{v} with the prior $\hat{p}(\mathbf{v})$ in (28). Problem \mathcal{A}_{VBI} is non-convex and thus we aim at finding a stationary solution (denoted by $\psi^*(\mathbf{v})$) of \mathcal{A}_{VBI} , as defined below.

Definition 5 (Stationary Solution): $\psi^*(\mathbf{v}) = \prod_{l \in \mathcal{H}} \psi^*(\mathbf{v}^l)$ is called a stationary solution of Problem \mathcal{A}_{VBI} if it satisfies all the constraints in \mathcal{A}_{VBI} and $\forall l \in \mathcal{H}$,

$$\begin{aligned} & \psi^*(\mathbf{v}^l) \\ &= \arg \min_{\psi(\mathbf{v}^l)} \int \prod_{j \neq l} \psi^*(\mathbf{v}^j) \psi(\mathbf{v}^l) \ln \frac{\prod_{j \neq l} \psi^*(\mathbf{v}^j) \psi(\mathbf{v}^l)}{\hat{p}(\mathbf{v}|\mathbf{z}, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})}. \end{aligned}$$

By finding a stationary solution $\psi^*(\mathbf{v})$ of \mathcal{A}_{VBI} , we could obtain the approximate posterior $\psi^*(\mathbf{v}^l) \approx \hat{p}(\mathbf{v}^l|\mathbf{z}, \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$, $\forall l \in \mathcal{H}$.

A stationary solution of \mathcal{A}_{VBI} can be obtained via alternately optimizing each individual density $\psi(\mathbf{v}^l)$, $l \in \mathcal{H}$, as will be proved by Lemma 6. Specifically, for given $\psi(\mathbf{v}^j)$, $\forall j \neq l$, the optimal $\psi(\mathbf{v}^l)$ that minimizes the KLD in \mathcal{A}_{VBI} is given by [13]

$$\psi(\mathbf{v}^l) \propto \exp \left(\langle \ln \hat{p}(\mathbf{v}, \mathbf{z} | \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}) \rangle_{\prod_{j \neq l} \psi(\mathbf{v}^j)} \right), \quad (31)$$

where $\langle f(x) \rangle_{\psi(x)} = \int f(x) \psi(x) dx$, $\hat{p}(\mathbf{v}, \mathbf{z} | \Delta \mathbf{r}, \Delta \boldsymbol{\vartheta})$ is the joint distribution of \mathbf{v} and \mathbf{z} with the prior $\hat{p}(\mathbf{v})$ in (28). Based on (31), the update equations of all variables are given in the subsequent subsections. The detailed derivation can be found in Appendix D. Note that the operator $\langle \cdot \rangle_{\psi(\mathbf{v}^l)}$ is equivalent to $\langle \cdot \rangle_{\psi(\mathbf{v}^l)}$ and the expectation $\langle f(\mathbf{v}^l) \rangle_{\psi(\mathbf{v}^l)}$ w.r.t. its own approximate posterior is simplified as $\langle f(\mathbf{v}) \rangle$.

2) *Initialization of Sparse VBI*: In order to trigger the a alternating optimization (AO) algorithm, we use the following initialization for the distribution functions $\psi(\mathbf{x})$, $\psi(\mathbf{y})$, $\psi(\boldsymbol{\rho})$, $\psi(\mathbf{q})$, $\psi(\boldsymbol{\gamma})$ in the first outer iteration. In the rest outer iterations, we initialize $\psi(\mathbf{x})$, $\psi(\mathbf{y})$, $\psi(\boldsymbol{\rho})$, $\psi(\mathbf{q})$, $\psi(\boldsymbol{\gamma})$

to the (approximate) posterior calculated from Module A (sparse VBI estimator) in the previous outer iteration.

- Initialize $\psi(\mathbf{q}) = \hat{p}(\mathbf{q}) = \prod_k \hat{p}(q_k)$ with $\hat{p}(q_k) = \sum_{q=1}^Q \pi_{k,q} \delta(q - q_k)$.
- Initialize a gamma distribution for ρ :

$$\psi(\rho) = \prod_{n,k,q} \Gamma(\rho_{n,k,q}; \tilde{a}_{n,k,q}, \tilde{b}_{n,k,q}),$$

where $\tilde{a}_{n,k,q} = \pi_{k,q} a_{n,k} + (1 - \pi_{k,q}) \bar{a}_{n,k}$, $\tilde{b}_{n,k,q} = \pi_{k,q} b_{n,k} + (1 - \pi_{k,q}) \bar{b}_{n,k}$.

- Initialize a gamma distribution for γ :

$$\psi(\gamma) = \prod_{n,k,m} \Gamma(\gamma_{n,k,m}; \tilde{\alpha}_{n,k,m}, \tilde{\beta}_{n,k,m}),$$

where $\tilde{\alpha}_{n,k,m} = \alpha_{n,k}$, $\tilde{\beta}_{n,k,m} = \beta_{n,k}$.

- Initialize a Gaussian distribution for \mathbf{x} and \mathbf{y} :

$$\psi(\mathbf{x}) = \prod_n \mathcal{CN}(\mathbf{x}_n; \boldsymbol{\mu}_n^x, \boldsymbol{\Sigma}_n^x),$$

$$\psi(\mathbf{y}) = \prod_n \mathcal{CN}(\mathbf{y}_n; \boldsymbol{\mu}_n^y, \boldsymbol{\Sigma}_n^y),$$

where $\boldsymbol{\Sigma}_n^x$ and $\boldsymbol{\Sigma}_n^y$ are extracted from the diagonal of $\boldsymbol{\Sigma}_n = (\text{diag}(\langle \boldsymbol{\rho}_n; \boldsymbol{\gamma}_n \rangle) + \mathbf{A}_n(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n)^H \mathbf{A}_n(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n))^{-1}$, $[\boldsymbol{\mu}_n^x; \boldsymbol{\mu}_n^y] = \boldsymbol{\Sigma}_n \mathbf{A}_n(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n)^H \mathbf{z}_n$.

3) *Update for κ* : From (31), the update for $\psi(\kappa)$ only depends on $\psi(\mathbf{x})$ and $\psi(\mathbf{y})$. For given $\psi(\mathbf{x})$ and $\psi(\mathbf{y})$, $\psi(\kappa)$ can be derived as

$$\psi(\kappa) = \Gamma(\kappa; \tilde{a}_\kappa, \tilde{b}_\kappa), \quad (32)$$

where $\tilde{a}_\kappa = a_\kappa + NMT$, $\tilde{b}_\kappa = b_\kappa + \sum_{n=1}^N \langle \|\mathbf{z}_n - \mathbf{A}_n(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n) [\begin{smallmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{smallmatrix}] \|^2 \rangle_{\mathbf{x}_n, \mathbf{y}_n}$.

4) *Update for \mathbf{x}* : From (31), the update for $\psi(\mathbf{x})$ only depends on $\psi(\rho)$, $\psi(\kappa)$ and $\psi(\mathbf{y})$. For given $\psi(\rho)$, $\psi(\kappa)$ and $\psi(\mathbf{y})$, $\psi(\mathbf{x})$ can be derived as

$$\psi(\mathbf{x}) = \prod_n \mathcal{CN}(\mathbf{x}_n; \boldsymbol{\mu}_n^x, \boldsymbol{\Sigma}_n^x). \quad (33)$$

$\boldsymbol{\mu}_n^x$ and $\boldsymbol{\Sigma}_n^x$ can be calculated through

$$\boldsymbol{\Sigma}_n^x = \left(\text{diag}(\langle \boldsymbol{\rho}_n \rangle) + \langle \kappa \rangle \left(\bar{\mathbf{A}}_n^L(\Delta \mathbf{r}) \right)^H \bar{\mathbf{A}}_n^L(\Delta \mathbf{r}) \right)^{-1}, \quad (34)$$

$$\boldsymbol{\mu}_n^x = \langle \kappa \rangle \boldsymbol{\Sigma}_n^x \left(\bar{\mathbf{A}}_n^L(\Delta \mathbf{r}) \right)^H \left(\mathbf{z}_n - \bar{\mathbf{A}}_n^{NL}(\Delta \boldsymbol{\vartheta}_n) \langle \mathbf{y}_n \rangle \right). \quad (35)$$

where $\bar{\mathbf{A}}_n^L(\Delta \mathbf{r}) = [\mathbf{A}_n^L(1); \dots, \mathbf{A}_n^L(T)] \in \mathbb{C}^{MT \times QK}$ and $\bar{\mathbf{A}}_n^{NL}(\Delta \boldsymbol{\vartheta}_n) = [\mathbf{A}_n^{NL}(1); \dots, \mathbf{A}_n^{NL}(T)] \in \mathbb{C}^{MT \times \tilde{M}K}$.

5) *Update for ρ* : From (31), for given $\psi(\mathbf{q})$ and $\psi(\mathbf{x})$, $\psi(\rho)$ can be derived as

$$\psi(\rho) = \prod_{k=1}^K \prod_{n=1}^N \prod_{q=1}^Q \Gamma(\rho_{n,k,q}; \tilde{a}_{n,k,q}, \tilde{b}_{n,k,q}), \quad (36)$$

where $\tilde{a}_{n,k,q}$, $\tilde{b}_{n,k,q}$ are given by:

$$\tilde{a}_{n,k,q} = \langle 1(q = q_k) \rangle a_{n,k} + \langle 1(q \neq q_k) \rangle \bar{a}_{n,k} + 1, \quad (37)$$

$$\tilde{b}_{n,k,q} = \langle 1(q = q_k) \rangle b_{n,k} + \langle 1(q \neq q_k) \rangle \bar{b}_{n,k} + \langle |x_{n,k,q}|^2 \rangle. \quad (38)$$

6) *Update for \mathbf{q}* : From (31), the update for $\psi(\mathbf{q})$ only depends on $\psi(\rho)$. For given $\psi(\rho)$, $\psi(\mathbf{q})$ can be derived as

$$\psi(\mathbf{q}) = \prod_{k=1}^K \sum_{q=1}^Q \tilde{\pi}_{k,q} \delta(q_k - q), \quad (39)$$

where $\tilde{\pi}_{k,q}$ is given by

$$\begin{aligned} \tilde{\pi}_{k,q} = & \frac{1}{C} \pi_{k,q} \prod_{n=1}^N e^{(a_{n,k}-1) \langle \ln \rho_{n,k,q} \rangle - b_{n,k} \langle \rho_{n,k,q} \rangle} \\ & \times \prod_{q' \neq q} \prod_{n=1}^N e^{(\bar{a}_{n,k}-1) \langle \ln \rho_{n,k,q'} \rangle - \bar{b}_{n,k} \langle \rho_{n,k,q'} \rangle} \end{aligned} \quad (40)$$

and C is the normalization constant, given by $C = \sum_q \pi_{k,q} \prod_{n=1}^N e^{(a_{n,k}-1) \langle \ln \rho_{n,k,q} \rangle - b_{n,k} \langle \rho_{n,k,q} \rangle} \times \prod_{q' \neq q} \prod_{n=1}^N e^{(\bar{a}_{n,k}-1) \langle \ln \rho_{n,k,q'} \rangle - \bar{b}_{n,k} \langle \rho_{n,k,q'} \rangle}$.

7) *Update for \mathbf{y}* : From (31), the update for $\psi(\mathbf{y})$ only depends on $\psi(\gamma)$, $\psi(\kappa)$ and $\psi(\mathbf{x})$. For given $\psi(\gamma)$, $\psi(\kappa)$ and $\psi(\mathbf{x})$, $\psi(\mathbf{y})$ can be derived as

$$\psi(\mathbf{y}) = \prod_n \mathcal{CN}(\mathbf{y}_n; \boldsymbol{\mu}_n^y, \boldsymbol{\Sigma}_n^y). \quad (41)$$

$\boldsymbol{\mu}_n^y$ and $\boldsymbol{\Sigma}_n^y$ can be calculated through

$$\boldsymbol{\Sigma}_n^y = \left(\text{diag}(\langle \boldsymbol{\gamma}_n \rangle) + \langle \kappa \rangle \left(\bar{\mathbf{A}}_n^{NL}(\Delta \boldsymbol{\vartheta}_n) \right)^H \bar{\mathbf{A}}_n^{NL}(\Delta \boldsymbol{\vartheta}_n) \right)^{-1}, \quad (42)$$

$$\boldsymbol{\mu}_n^y = \langle \kappa \rangle \boldsymbol{\Sigma}_n^y \left(\bar{\mathbf{A}}_n^{NL}(\Delta \boldsymbol{\vartheta}_n) \right)^H \left(\mathbf{z}_n - \bar{\mathbf{A}}_n^L(\Delta \mathbf{r}) \langle \mathbf{x}_n \rangle \right). \quad (43)$$

8) *Update for γ* : From (31), the update for $\psi(\gamma)$ only depends on $\psi(\mathbf{y})$. For given $\psi(\mathbf{y})$, $\psi(\gamma)$ can be derived as

$$\psi(\gamma) = \prod_{k=1}^K \prod_{n=1}^N \prod_{m=1}^{\tilde{M}} \Gamma(\gamma_{n,k,m}; \tilde{\alpha}_{n,k,m}, \tilde{\beta}_{n,k,m}), \quad (44)$$

where $\tilde{\alpha}_{n,k,m} = \alpha_{n,k} + 1$, $\tilde{\beta}_{n,k,m} = \beta_{n,k} + \langle |y_{n,k,m}|^2 \rangle$.

The involved expectations are given as follows for $\forall n \in \{1, \dots, N\}$, $k \in \{1, \dots, K\}$, $q \in \{1, \dots, Q\}$, $m \in \{1, \dots, \tilde{M}\}$:

$$\begin{aligned} & \left\langle \left\| \mathbf{z}_n - \mathbf{A}_n(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}_n) \begin{bmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{bmatrix} \right\|^2 \right\rangle_{\mathbf{x}_n, \mathbf{y}_n} \\ &= \left\| \mathbf{z}_n - \bar{\mathbf{A}}_n^L(\Delta \mathbf{r}) \boldsymbol{\mu}_n^x - \bar{\mathbf{A}}_n^{NL}(\Delta \boldsymbol{\vartheta}_n) \boldsymbol{\mu}_n^y \right\|^2 \\ &+ \text{tr} \left(\bar{\mathbf{A}}_n^L(\Delta \mathbf{r}) \boldsymbol{\Sigma}_n^x \left(\bar{\mathbf{A}}_n^L(\Delta \mathbf{r}) \right)^H \right) \end{aligned}$$

TABLE II
COMPLEXITY ORDER AND CPU TIME FOR DIFFERENT SCHEMES

Schemes	Complexity order	CPU time
proposed scheme	$\mathcal{O}(Q^2 K^2 NMT)$	6.7s
VBI scheme	$\mathcal{O}(Q^2 K^2 NMT)$	7s
DiSoul scheme	$\mathcal{O}((Q + \tilde{M})^3 K^3 N^3)$	5.6s

$$+ \text{tr} \left(\bar{\mathbf{A}}_n^{NL} (\Delta \boldsymbol{\vartheta}_n) \Sigma_n^y \left(\bar{\mathbf{A}}_n^{NL} (\Delta \boldsymbol{\vartheta}_n) \right)^H \right), \quad (45)$$

$$\langle \kappa \rangle = \frac{\tilde{a}_\kappa}{\tilde{b}_\kappa}, \langle \gamma_{n,k,m} \rangle = \frac{\tilde{\alpha}_{n,k,m}}{\tilde{\beta}_{n,k,m}}, \langle \rho_{n,k,q} \rangle = \frac{\tilde{a}_{n,k,q}}{\tilde{b}_{n,k,q}}, \quad (46)$$

$$\langle \ln \rho_{n,k,q} \rangle = \text{Digamma}(\tilde{a}_{n,k,q}) - \ln(\tilde{b}_{n,k,q}), \quad (47)$$

$$\langle 1(q = q_k) \rangle = \tilde{\pi}_{k,q}, \langle 1(q \neq q_k) \rangle = 1 - \tilde{\pi}_{k,q}, \quad (48)$$

$$\langle \mathbf{x}_n \rangle = \boldsymbol{\mu}_n^x, \langle |x_{n,k,q}|^2 \rangle = |\mu_{n,k,q}^x|^2 + \Sigma_{n,k,q}^x, \quad (49)$$

$$\langle \mathbf{y}_n \rangle = \boldsymbol{\mu}_n^y, \langle |y_{n,k,m}|^2 \rangle = |\mu_{n,k,m}^y|^2 + \Sigma_{n,k,m}^y, \quad (50)$$

where $\text{Digamma}(x) = \frac{d}{dx} \ln(\Gamma(x))$ is the digamma function, defined as the logarithmic derivative of the gamma function, $\mu_{n,k,q}^x$ is the $((k-1)Q + q)$ -th element of $\boldsymbol{\mu}_n^x$, $\mu_{n,k,m}^y$ is the $((k-1)\tilde{M} + m)$ -th element of $\boldsymbol{\mu}_n^y$, $\Sigma_{n,k,q}^x$ is the $((k-1)Q + q)$ -th diagonal element of Σ_n^x , and $\Sigma_{n,k,m}^y$ is the $((k-1)\tilde{M} + m)$ -th diagonal element of matrix Σ_n^y .

9) *Convergence of Sparse VBI*: We have the following convergence theorem for the sparse VBI.

Lemma 6 (Convergence of Sparse VBI): Every limiting point $\psi^*(\mathbf{v}) = \prod_{l \in \mathcal{H}} \psi^*(\mathbf{v}^l)$ generated by the sparse VBI using (32), (33), (36), (39), (41) and (44) with the initialization in Section IV-D2 is a stationary solution of Problem \mathcal{A}_{VBI} .

The proof can be found in Appendix E.

Finally, the overall Turbo-VPCL algorithm is summarized in Algorithm 1.

E. Computational Complexity

We discuss the computational complexity of proposed algorithm, which is dominated by the updates of $\psi(\mathbf{x})$. The total number of divisions to update $\psi(\mathbf{x})$ is $QKN + N$; the total number of multiplications to update $\psi(\mathbf{x})$ is $2QKN + 2NTMQK + 2NT^2M^2QK + 2K^2Q^2NMT + MNTK\tilde{M}$; the complexity order induced by the matrix inversion (after applying the matrix inversion lemma to (34) to reduce the complexity) is $\mathcal{O}(NM^3)$. Then the total computational complexity order of the proposed algorithm is $\mathcal{O}(Q^2K^2NMT)$ per iteration (assume that $Q > \tilde{M}$). In Table II, we summarize the per iteration complexity orders of the proposed scheme and other schemes. The computational complexity of the second-order cone program (SOCP) used in DiSoul [10] can be found in [27]. As seen from Table II, the complexity order of the proposed scheme is similar to the VBI scheme and lower than the DiSoul scheme. Note that the CPU time in Table II is computed on a regular desktop computer with a MATLAB implementation. The low-latency requirements

Algorithm 1: Turbo-VPCL Algorithm.

Input: \mathbf{z} , measurement matrix $\mathbf{A}_n(\mathbf{0}, \mathbf{0})$, $\forall n$, hyperparameters: $a_{n,k} = b_{n,k} = 1$, $\bar{a}_{n,k} = 1$, $\bar{b}_{n,k} = 0.001$, $\forall n, k$, $\alpha_{n,k} = \beta_{n,k} = a_\kappa = b_\kappa = 0.001$, the parameters for the Markov model of the coarse position \mathbf{q} : α, λ, d .

Output: position estimate: $\hat{\mathbf{p}}_1, \dots, \hat{\mathbf{p}}_K$

- 1: Initialize $\Delta \mathbf{r} = \mathbf{0}$, $\Delta \boldsymbol{\vartheta} = \mathbf{0}$, and the message $\nu_{\mu \rightarrow q_k}(q_k = q) \triangleq \pi_{k,q} = \frac{1}{Q}$, $\forall k, q$.
- 2: **while** not converge **do**
- 3: **Turbo-VPCL-E Step**:
- 4: **%Module A: Sparse VBI Estimator**
- 5: Initialize the distribution functions $\psi(\mathbf{x})$, $\psi(\mathbf{y})$, $\psi(\boldsymbol{\rho})$, $\psi(\mathbf{q})$ and $\psi(\boldsymbol{\gamma})$, and the related expectations using (45)–(50).
- 6: **while** not converge **do**
- 7: Update $\psi(\kappa)$ using (32) and the related expectations using (46).
- 8: Update $\psi(\mathbf{x})$ using (33) and the related expectations using (45) and (49).
- 9: Update $\psi(\boldsymbol{\rho})$ using (36) and the related expectations using (46) and (47).
- 10: Update $\psi(\mathbf{q})$ using (39) and the related expectations using (48).
- 11: Update $\psi(\mathbf{y})$ using (41) and the related expectations using (45) and (50).
- 12: Update $\psi(\boldsymbol{\gamma})$ using (44) and the related expectations using (46).
- 13: **end while**
- 14: Calculate the extrinsic information of q_k based on (27), send $\nu_{\eta_k \rightarrow q_k}(q_k)$ to Module B.
- 15: **%Module B**:
- 16: Perform forward-backward message passing over subgraph \mathcal{G}_P , send $\nu_{\mu \rightarrow q_k}(q_k)$ to Module A.
- 17: **Turbo-VPCL-M Step**:
- 18: Construct the surrogate function F in (18) using the approximate posterior output of module A, i.e., $\psi(\mathbf{v})$.
- 19: Update $\Delta \mathbf{r}$ and $\Delta \boldsymbol{\vartheta}$ using (24) and (25).
- 20: **end while**
- 21: $\Delta \mathbf{r}^* = \Delta \mathbf{r}$, $\Delta \boldsymbol{\vartheta}^* = \Delta \boldsymbol{\vartheta}$, $p(q_k | \mathbf{z}, \Delta \mathbf{r}^*, \Delta \boldsymbol{\vartheta}^*) \approx \psi(q_k)$, then we have $q_k^* = \arg \max p(q_k | \mathbf{z}, \Delta \mathbf{r}^*, \Delta \boldsymbol{\vartheta}^*)$ and $\hat{\mathbf{p}}_k = \mathbf{r}_{q_k^*} + \Delta \mathbf{r}_{q_k^*}^*$, $\forall k$.

of the considered cloud-assisted vehicle positioning scenarios can be satisfied by the C-RAN architecture in 5G and future systems [3] due to the powerful and parallel computing power of the cloud.

V. SIMULATION RESULTS AND DISCUSSIONS

In this section, we verify the cooperative localization performance of the proposed algorithm and compare it with the following existing methods:

- **Baseline 1 (VBI)** [13]: This is the variational Bayesian inference method in [13]. The VBI algorithm assumes

group sparse LOS channel prior and i.i.d sparse NLOS channel prior for every vehicle.

- **Baseline 2** (DiSouL) [10]: This is the direct localization method recently proposed in [10]. The LOS and NLOS channels are recovered through the mixed- $l_{2,1}$ norm minimization to exploit the group sparsity of LOS channels, and the grid is refined around the estimated location and angles adaptively.
- **Baseline 3** (Direct MAP): We directly apply the MAP estimator to the parameters \mathbf{h}^L , \mathbf{p} , \mathbf{h}^{NL} , $\boldsymbol{\theta}^{NL}$ without introducing the location and AoA grid, where $\mathbf{z} \triangleq \{\mathbf{z}_n(t)\}_{t=1, n=1}^{T, N}$, $[\mathbf{h}^L, \mathbf{h}^{NL}, \boldsymbol{\theta}^{NL}] \triangleq \{h_{n,k}, h_{n,k}^l, \theta_{n,k}^l\}_{n=1, k=1, l=1}^{N, K, L_{n,k}}$, and $\mathbf{p} \triangleq \{\mathbf{p}_k\}_{k=1}^K$. Since the MAP problem is non-convex, we consider two versions of baseline 3. The first one finds a stationary point of the MAP problem starting from an initial point obtained from the true vehicles' positions and channels (gene-aided MAP). In this way, the solution found by the gradient descent method is expected to be very close to the optimal MAP solution. The second one first finds 20 stationary points starting from 20 random initial points, respectively, and then chooses the best stationary point as the final estimation (MAP with 20 initial points).

Note that baseline 1 and baseline 2 cannot exploit the correlations between vehicle positions in a vehicle platoon since they only focus on single vehicle localization. Baseline 3 can exploit the gamma distributed inter-vehicle distances in the vehicle platoon.

In the simulations, the platoon consist of 4 vehicles positioned on a highway, as shown in Fig. 1. The vehicle positions are generated as follows. First, we place the first vehicle on a random point in the center of the road. Then we place the other vehicles one after another along the center of the road with the inter-vehicle distance generated from the gamma distribution in (7) with parameters $\alpha = 2$, $\lambda = 0.3$ so that the average distance between vehicles is 8 meters. Finally, we add a small random position offset to each vehicle. The BSs are at coordinates [50 m, 50 m], [50 m, -50 m], [-50 m, 50 m] and [-50 m, -50 m]. The signals transmitted from different vehicles propagate through different scatterers. The carrier frequency is 7 GHz and the wavelength $\lambda = 43$ mm. Each BS is equipped with a massive uniform linear array (ULA) with $\frac{\lambda}{2}$ inter antenna spacing. We adopt the standard cellular channel parameters in 3GPP [28].

For baseline 1 and baseline 2, we consider a sequential pilot transmission scheme where the vehicles take turns to transmit $K = 4$ pilots to the BSs to estimate their locations. Since vehicles transmit pilots at orthogonal resource blocks, there is no interference between different vehicles. Specifically, assume the vehicle k is the k -th one to transmit pilot s_k . Then $\forall t = 1, \dots, K$, we have $s_k(t) = s_k$, if $t = k$; otherwise $s_k(t) = 0$. For baseline 3, we consider a simultaneous pilot transmission scheme where all vehicles simultaneously transmit K randomly generated pilots to all BSs. In this case, we have $s_k(t) \neq 0, \forall k = 1, \dots, K, \forall t = 1, \dots, T$, and $T = K$. For the proposed scheme, we consider both sequential pilot transmission and simultaneous pilot transmission. In the sequential pilot transmission, we set

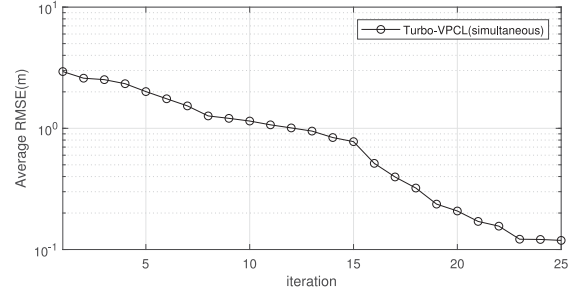


Fig. 6. The convergence behavior of the proposed algorithm.

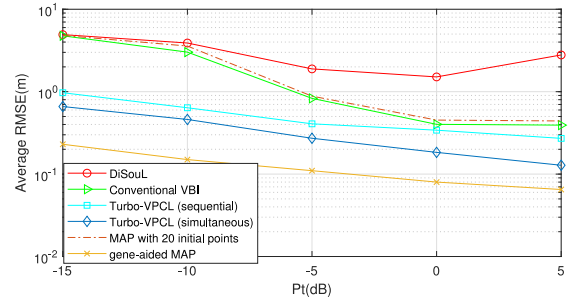


Fig. 7. Average RMSE performance of different algorithms versus transmit power with $L_{n,k} = 4$ and $M = 64$.

$|s_k|^2 = P_T, \forall k$, and in the simultaneous transmission scheme, we set $|s_k(t)|^2 = P_T/K, \forall k, t$, which both correspond to an average transmit power of P_T . We use the average/worst root-mean-square-error (RMSE) as performance metric. The average RMSE is defined as $\frac{1}{JK} \sum_{j=1}^J \sum_{k=1}^K \|\hat{\mathbf{p}}_k^j - \mathbf{p}_k^j\|$ and the worst RMSE is defined as $\frac{1}{J} \sum_{j=1}^J \max_k \|\hat{\mathbf{p}}_k^j - \mathbf{p}_k^j\|$, where \mathbf{p}_k^j and $\hat{\mathbf{p}}_k^j$ denote the true position and position estimation in the j -th time slot and $J = 500$ is the total number of time slots in the simulations.

A. Convergence Behavior

In Fig. 6, we show the convergence behavior of the proposed algorithm. The proposed algorithm can converge within about 20 iterations. This further implies that the posterior approximation in proposed algorithm is accurate enough so that it has little effect on the convergence performance of the whole algorithm.

B. RMSE Versus Transmit Power

The average and worst RMSE performance of different algorithms versus transmit power P_T are shown in Fig. 7 and Fig. 8. It can be seen that the RMSE performance of all schemes improves with transmit power, except for DiSouL, whose RMSE first decreases with P_T and then increases with P_T . DiSouL [10] becomes worse for higher transmit powers because as P_T increases, the *effective noise power* (i.e., the mismatch between the measurements and the signal model) caused by the position/angle offset is enlarged, but the constraint bound ϵ (which is supposed to reflect the mismatch between the measurements and the signal model) in [10, (17b)] only depends on the AWGN power and is not adjusted according

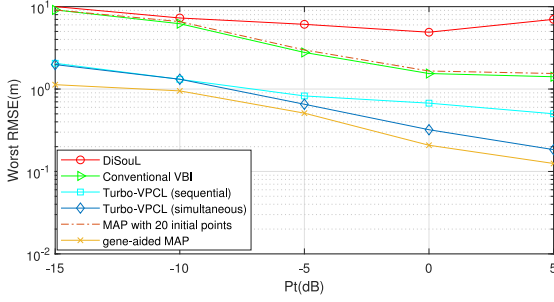


Fig. 8. Worst RMSE performance of different algorithms versus transmit power with $L_{n,k} = 4$ and $M = 64$.

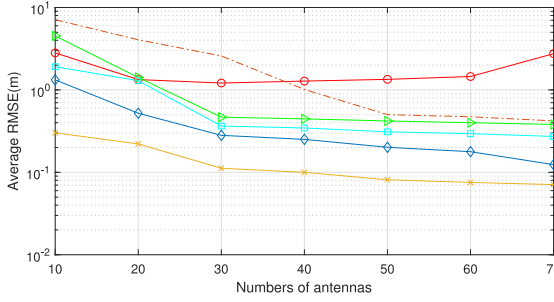


Fig. 9. Average RMSE of different algorithms versus the number of antennas M with $L_{n,k} = 4$ and $P_T = 0$ dB.

to the increased effective noise power. On the other hand, the other VBI-based methods can automatically learn the effective noise power and thus is less sensitive to the position/angle offset. Clearly, the proposed Turbo-VPCL performs significantly better than all the baselines (except for gene-aided MAP) under both sequential and simultaneous pilot transmission due to exploiting the correlations between vehicle positions in a vehicle platoon. With simultaneous pilot transmission, the position of each vehicle is estimated from more measurements, and thus a better performance can be achieved compared to the sequential pilot transmission. Moreover, the proposed grid-based solution can approach the near-optimal performance of the gene-aided MAP for large transmit power by exploiting the channel sparsity and spatial correlations within vehicle platoon.

C. RMSE Versus Antenna Numbers

Fig. 9 and Fig. 10 illustrates the average/worst RMSE performance of different algorithms versus the number of antennas M . As M increases, the angular resolution is improved, which could potentially improve the localization accuracy, as shown in Fig. 9 and Fig. 10 for the VBI-based methods. However, the RMSE of DiSouL first decreases and then increases with the number of antennas for the same reason as described above, that is the effective noise power caused by the position/angle offset increases with the number of antennas, but DiSouL does not have the inherit learning mechanism as the VBI-based methods to adjust its constraint bound ϵ accordingly. From Fig. 9 and Fig. 10, it can be seen that the proposed Turbo-VPCL outperforms all other baselines (except for gene-aided MAP) especially for massive

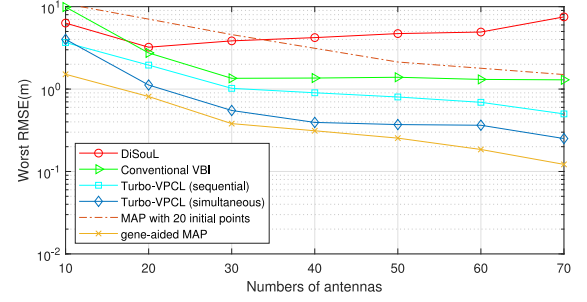


Fig. 10. Worst RMSE of different algorithms versus the number of antennas M with $L_{n,k} = 4$ and $P_T = 0$ dB.

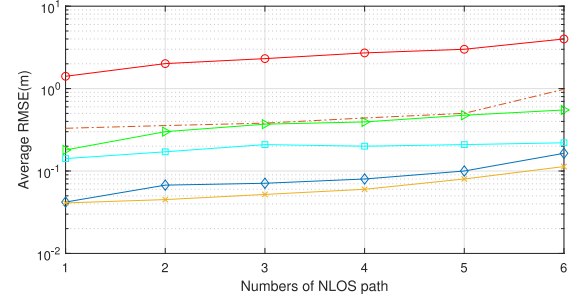


Fig. 11. Average RMSE of different algorithms versus the numbers of NLOS paths $L_{n,k}$ with $P_T = 5$ dB and $M = 64$.

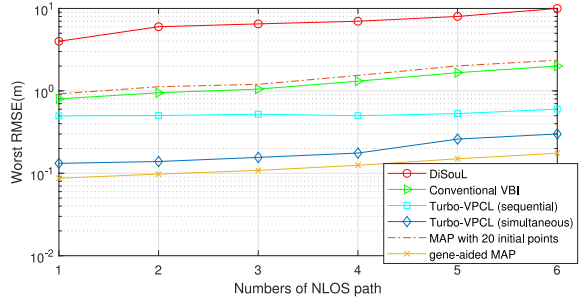


Fig. 12. Worst RMSE of different algorithms versus the numbers of NLOS paths $L_{n,k}$ with $P_T = 5$ dB and $M = 64$.

antenna arrays (large M). Compared to the gene-aided MAP, the proposed algorithm can achieve near-optimal performance when antenna number is large.

D. RMSE Versus the Numbers of NLOS Path

In Fig. 11 and Fig. 12, we plot the RMSE performance versus the number of NLOS paths. As the number of NLOS path increases, it becomes more difficult to identify the LOS path (which contains useful information for localization) from the NLOS paths, resulting in the decrease of localization accuracy. Again, the proposed Turbo-VPCL performs better than the baselines (except for gene-aided MAP) and the performance gap between gene-aided MAP is small. Therefore, Turbo-VPCL can also work well even when there are more NLOS paths.

VI. CONCLUSION

We propose a novel cloud-assisted cooperative localization method for accurate and robust localization of vehicle platoons.

To capture the sophisticated correlations of the vehicle positions in the platoon and the structured sparsity of massive MIMO channel, we propose a new GMGS prior model and formulate the cooperative vehicle localization as an SBI problem. Then we propose a Turbo-VPCL algorithm which approximately calculates the posteriors of the coarse positions and sparse massive MIMO channels by combining the message passing and VBI approaches via the turbo framework, and uses the EM method to refine the position estimation. In Turbo-VPCL, the ill-conditioned measurement model and structured sparsity (as captured by the GMGS prior) in the cooperative vehicle localization SBI problem are handled using the VBI approach and message passing approach, respectively. The proposed Turbo-VPCL algorithm is shown in the simulations to achieve significant gains over baseline algorithms.

APPENDIX

A. Gradient in (24)–(25)

Let $\Delta \mathbf{r}_q = [\Delta r_q^x, \Delta r_q^y]$ and $\mathbf{r}_q = [r_q^x, r_q^y]$, where $\Delta r_q^x, \Delta r_q^y$ (r_q^x, r_q^y) are the x- and y- coordinates of $\Delta \mathbf{r}_q$ (\mathbf{r}_q) respectively. Then the partial gradient $\frac{\partial F(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)}{\partial \Delta r_q^x}$ can be calculated as follows:

$$\begin{aligned} & \frac{\partial F(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)}{\partial \Delta r_q^x} \\ &= -c_0 \sum_{n=1}^N \sum_{t=1}^T 2\text{Re} \left(\mathbf{a}'_n(\theta_{n,q})^H (\mathbf{a}_n(\theta_{n,q}) c_1 + \mathbf{c}_2) \right) c_{n,q}^x, \end{aligned}$$

where $c_0 = \tilde{\alpha}_\kappa / \tilde{\beta}_\kappa$, $\mathbf{a}_n(\theta_{n,q}) = \mathbf{a}_n(\theta_n(\mathbf{r}_q + \Delta \mathbf{r}_q))$, $\mathbf{a}'_n(\theta_{n,q}) = \frac{\partial \mathbf{a}_n(\theta)}{\partial \theta} \big|_{\theta=\theta_n(\mathbf{r}_q + \Delta \mathbf{r}_q)}$. c_1 , \mathbf{c}_2 and $c_{n,q}^x$ are given by

$$\begin{aligned} c_1 &= \left| \sum_{k=1}^K s_k(t) \mu_{n,k,q}^x \right|^2 + \sum_{k=1, j=1}^K s_k(t) s_j(t)^* \Sigma_{n,(k,j),(q,q)}^x, \\ \mathbf{c}_2 &= \sum_{q' \neq q} \sum_{k=1, j=1}^K s_k(t) s_j(t)^* \Sigma_{n,(k,j),(q',q)}^x \mathbf{a}_n(\theta_{n,q'}) \\ &\quad - \left(\sum_{k=1}^K s_k(t) \mu_{n,k,q}^x \right)^* \mathbf{z}_{n,-q}(t), \\ c_{n,q}^x &= - (r_q^y + \Delta r_q^y - \tilde{p}_n^y) / \|\mathbf{r}_q + \Delta \mathbf{r}_q - \tilde{\mathbf{p}}_n\|^2. \end{aligned} \quad (51)$$

In (51), $\mathbf{z}_{n,-q}(t)$ is given by

$$\begin{aligned} \mathbf{z}_{n,-q}(t) &= \mathbf{z}_n(t) - \mathbf{A}_n^{NL}(t) \boldsymbol{\mu}_n^y \\ &\quad - \sum_k s_k(t) \sum_{q' \neq q} \mathbf{a}_n(\theta_{n,q'}) \mu_{n,k,q'}^x. \end{aligned}$$

$\mu_{n,k,q}^x$ is the $((k-1)Q + q)$ -th element of $\boldsymbol{\mu}_n^x$, $\Sigma_{n,(k,j),(q',q)}^x$ is the (q',q) -th element of matrix $\Sigma_{n,(k,j)}^x \in \mathbb{C}^{Q \times Q}$, which is the (k,j) -th block matrix of $\Sigma_n^x \in \mathbb{C}^{KQ \times KQ}$, $(x)^*$ means the conjugate of x , and $\tilde{\alpha}_\kappa, \tilde{\beta}_\kappa, \boldsymbol{\mu}_n^x, \Sigma_n^x$ are given by (32)–(33). The

expression for the partial gradient $\frac{\partial F(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)}{\partial \Delta r_q^y}$ is similar but with $c_{n,q}^x$ replaced by $c_{n,q}^y = (r_q^x + \Delta r_q^x - \tilde{p}_n^x) / \|\mathbf{r}_q + \Delta \mathbf{r}_q - \tilde{\mathbf{p}}_n\|^2$.

The gradient $\frac{\partial F(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)}{\partial \Delta \vartheta_{n,k}^m}$ can be calculated as follows:

$$\begin{aligned} & \frac{\partial F(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)}{\partial \Delta \vartheta_{n,k}^m} \\ &= -c_0 \sum_{t=1}^T 2\text{Re} \left(\mathbf{a}'_n(\vartheta_{n,k,m})^H (\mathbf{a}_n(\vartheta_{n,k,m}) c_3 + \mathbf{c}_4) \right), \end{aligned}$$

where $\mathbf{a}_n(\vartheta_{n,k,m}) = \mathbf{a}_n(\vartheta_m + \Delta \vartheta_{n,k}^m)$, $\mathbf{a}'_n(\vartheta_{n,k,m}) = \frac{\partial \mathbf{a}_n(\theta)}{\partial \theta} \big|_{\theta=\vartheta_m + \Delta \vartheta_{n,k}^m}$. c_3 and \mathbf{c}_4 are given by

$$\begin{aligned} c_3 &= \left| s_k(t) \mu_{n,k,m}^y \right|^2 + |s_k(t)|^2 \Sigma_{n,(k,k),(m,m)}^y, \\ \mathbf{c}_4 &= \sum_{m' \neq m} |s_k(t)|^2 \Sigma_{n,(k,k),(m',m)}^y \mathbf{a}_n(\vartheta_{n,k,m'}) \\ &\quad + \sum_{j \neq k} s_j(t) s_k(t)^* \mathbf{A}_n^{NL}(\Delta \boldsymbol{\vartheta}_{n,j}) \Sigma_{n,(j,k)}^y[:, m] \\ &\quad - \left(s_k(t) \mu_{n,k,m}^y \right)^* \mathbf{z}_{n,-k,m}(t). \end{aligned} \quad (52)$$

$\mathbf{z}_{n,-k,m}(t)$ in (52) is given by

$$\begin{aligned} \mathbf{z}_{n,-k,m}(t) &= \mathbf{z}_n(t) - \mathbf{A}_n^L(t) \boldsymbol{\mu}_n^x - \sum_{j \neq k} s_j(t) \mathbf{A}_n^{NL}(\Delta \boldsymbol{\vartheta}_{n,j}) \\ &\quad \times \boldsymbol{\mu}_{n,j}^y - s_k(t) \sum_{m' \neq m} \mathbf{a}_n(\vartheta_{n,k,m'}) \mu_{n,k,m'}^y. \end{aligned}$$

$\mu_{n,k,m}^y$ is the $((k-1)\tilde{M} + m)$ -th element of $\boldsymbol{\mu}_n^y$, $\Sigma_{n,(j,k),(m',m)}^y$ is the (m',m) -th element of matrix $\Sigma_{n,(j,k)}^y \in \mathbb{C}^{\tilde{M} \times \tilde{M}}$, which is the (j,k) -th block matrix of $\Sigma_n^y \in \mathbb{C}^{K\tilde{M} \times K\tilde{M}}$, $\Sigma_{n,(j,k)}^y[:, m]$ is the m -th column of matrix $\Sigma_{n,(j,k)}^y$, $\boldsymbol{\mu}_n^y$ and Σ_n^y are given by (41).

B. Proof of Theorem 2

Theorem 2 is the convergence of the Generalized EM (GEM) algorithm, which relaxes the requirement of maximization in (23) to inexact maximization in (24) and (25), which simply increase $F(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)$ with respect to $\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}$ using a gradient update, so that $F(\Delta \mathbf{r}^{i+1}, \Delta \boldsymbol{\vartheta}^{i+1}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i) \geq F(\Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)$. This approach, to simply increase and not necessarily maximize $F(\Delta \mathbf{r}, \Delta \boldsymbol{\vartheta}; \Delta \mathbf{r}^i, \Delta \boldsymbol{\vartheta}^i)$ is known as the generalized EM algorithm. The convergence of the generalized EM algorithm to the stationary point can be found in detail in [29].

C. Message Passing Over \mathcal{G}_P

Given the input $\nu_{\eta_k \rightarrow q_k}(q_k) = \sum_{q=1}^Q \pi_{k,q}^{in} \delta(q_k - q)$, where $\pi_{k,q}^{in} = \nu_{\eta_k \rightarrow q_k}(q_k = q)$, for $2 \leq k \leq K$ and $1 \leq q \leq Q$, we can

calculate

$$\zeta_{k,q}^f = \begin{cases} \frac{\sum_{s=1+(k-2)d}^{q-d} \frac{\pi_{k-1,s}^{in} \xi_{k-1,s}^f}{\sum_{i=1}^Q \pi_{k-1,i}^{in} \xi_{k-1,i}^f}}{\lambda^{\frac{(q-s-d)^{\alpha-1}}{\alpha \Gamma(\alpha)}}} \exp\left(-\frac{q-s-d}{\lambda}\right), & q \in \mathcal{X} \\ 0, & \text{otherwise} \end{cases}$$

where $\mathcal{X} = \{q : 1 + (k-1)d \leq q \leq Q - (K-k)d\}$. Therefore, $\xi_{k,q}^f = \frac{\zeta_{k,q}^f}{\sum_{i=1}^Q \xi_{k,i}^f}$. If $k=1$, $\xi_1^f = [\xi_{1,1}^f, \dots, \xi_{1,Q}^f]$ can be obtained via the other localization technologies.

For $1 \leq k \leq K-1$ and $1 \leq q \leq Q$, we can calculate

$$\zeta_{k,q}^b = \begin{cases} \frac{\sum_{s=q+d}^{Q-(K-k-1)d} \frac{\pi_{k+1,s}^{in} \xi_{k+1,s}^b}{\sum_{i=1}^Q \pi_{k+1,i}^{in} \xi_{k+1,i}^b}}{\lambda^{\frac{(s-q-d)^{\alpha-1}}{\alpha \Gamma(\alpha)}}} \exp\left(-\frac{s-q-d}{\lambda}\right), & q \in \mathcal{X} \\ 0, & \text{otherwise} \end{cases}$$

and $\xi_{k,q}^b = \frac{\zeta_{k,q}^b}{\sum_{i=1}^Q \xi_{k,i}^b}$. If $k=K$, ξ_K^b is set to be

$$\xi_{K,q}^b = \begin{cases} \frac{1}{Q-(K-1)d}, & 1 + (K-1)d \leq q \leq Q \\ 0, & \text{otherwise} \end{cases}$$

Therefore, for $1 \leq k \leq K$ and $1 \leq q \leq Q$, we can get the output messages $\nu_{\mu \rightarrow q_k}$ as follows:

$$\nu_{\mu \rightarrow q_k}(q_k = q) = \frac{\zeta_{k,q}^f \xi_{k,q}^b}{\sum_{i=1}^Q \zeta_{k,i}^f \xi_{k,i}^b}. \quad (53)$$

D. Derivation of (32)–(44)

From (31), $\psi(\kappa)$ in (32) can be obtained as

$$\begin{aligned} \ln \psi(\kappa) &\propto \langle \ln p(\mathbf{z}|\mathbf{x}, \mathbf{y}, \kappa, \Delta \mathbf{r}, \Delta \mathbf{v}) \rangle_{\mathbf{x}, \mathbf{y}} + \ln p(\kappa) \\ &\propto -\kappa \sum_{n=1}^N \left\langle \left\| \mathbf{z}_n - \mathbf{A}_n(\Delta \mathbf{r}, \Delta \mathbf{v}_n) \begin{bmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{bmatrix} \right\|^2 \right\rangle_{\mathbf{x}_n, \mathbf{y}_n} \\ &\quad + NMT \ln \kappa + (a_\kappa - 1) \ln \kappa - b_\kappa \kappa \\ &\propto (\tilde{a}_\kappa - 1) \ln \kappa - \tilde{b}_\kappa \kappa. \end{aligned}$$

$\psi(\mathbf{x})$ in (33) can be obtained as

$$\begin{aligned} \ln \psi(\mathbf{x}_n) &\propto \langle \ln p(\mathbf{z}_n|\mathbf{x}_n, \mathbf{y}_n, \kappa, \Delta \mathbf{r}, \Delta \mathbf{v}_n) \rangle_{\mathbf{y}_n, \kappa} + \langle \ln p(\mathbf{x}_n|\rho_n) \rangle_{\rho_n} \\ &\propto -\langle \kappa \rangle \left\langle \left\| \mathbf{z}_n - \bar{\mathbf{A}}_n^L(\Delta \mathbf{r}) \mathbf{x}_n - \bar{\mathbf{A}}_n^{NL}(\Delta \mathbf{v}_n) \mathbf{y}_n \right\|^2 \right\rangle_{\mathbf{y}_n} \\ &\quad - \mathbf{x}_n^H \text{diag}(\langle \rho_n \rangle) \mathbf{x}_n \\ &\propto -(\mathbf{x}_n - \mu_n^x)^H (\Sigma_n^x)^{-1} (\mathbf{x}_n - \mu_n^x). \end{aligned}$$

$\psi(\mathbf{y})$ in (41) can be obtained similarly. $\psi(\rho)$ in (36) can be obtained as

$$\begin{aligned} \ln \psi(\rho) &\propto \langle \ln p(\mathbf{x}|\rho) \rangle_{\mathbf{x}} + \langle \ln p(\rho|q) \rangle_q \\ &\propto \sum_{k,n,q} (\tilde{a}_{n,k,q} - 1) \ln \rho_{n,k,q} - \tilde{b}_{n,k,q} \rho_{n,k,q}. \end{aligned}$$

$\psi(q)$ in (39) can be obtained as

$$\begin{aligned} \ln \psi(q) &\propto \langle \ln p(\rho|q) \rangle_{\rho} + \ln \hat{p}(q) \\ &\propto \sum_k \sum_q 1(q = q_k) \sum_n \left(\ln \frac{b_{n,k}^{a_{n,k}}}{\Gamma(a_{n,k})} + (a_{n,k} - 1) \langle \ln \rho_{n,k,q} \rangle \right. \\ &\quad \left. - b_{n,k} \langle \rho_{n,k,q} \rangle \right) + 1(q \neq q_k) \sum_n \left(\ln \frac{\bar{b}_{n,k}^{\bar{a}_{n,k}}}{\Gamma(\bar{a}_{n,k})} + (\bar{a}_{n,k} - 1) \right. \\ &\quad \left. \times \langle \ln \rho_{n,k,q} \rangle - \bar{b}_{n,k} \langle \rho_{n,k,q} \rangle \right) + \sum_k \ln \sum_q \pi_{k,q} \delta(q_k - q) \\ &\propto \sum_k \ln \sum_q \tilde{\pi}_{k,q} \delta(q_k - q). \end{aligned}$$

$\psi(\gamma)$ in (44) can be obtained as

$$\begin{aligned} \ln \psi(\gamma) &\propto \langle \ln p(\mathbf{y}|\gamma) \rangle_{\mathbf{y}} + \ln p(\gamma) \\ &\propto \sum_{n,k,m} \ln \gamma_{n,k,m} - \gamma_{n,k,m} \langle |y_{n,k,m}|^2 \rangle \\ &\quad + (\alpha_{n,k} - 1) \ln \gamma_{n,k,m} - \beta_{n,k} \gamma_{n,k,m} \\ &\propto \sum_{n,k,m} (\tilde{\alpha}_{n,k,m} - 1) \ln \gamma_{n,k,m} - \tilde{\beta}_{n,k,m} \gamma_{n,k,m}. \end{aligned}$$

E. Proof of Lemma 6

In our problem, the factorized form $\psi(\mathbf{v}) = \psi(\mathbf{x})\psi(\rho)\psi(\mathbf{y})\psi(\gamma)\prod_k \psi(q_k)$ in (26) can be considered as some parameterized functions, e.g., a gamma distribution function with parameters $\tilde{\alpha}_{n,k,m}$ and $\tilde{\beta}_{n,k,m}$ for $\psi(\gamma)$, a Gaussian distribution function with parameters μ_n^x and Σ_n^x for $\psi(\mathbf{x})$ and so on. As a result, the optimization problem \mathcal{A}_{VBI} which is optimization over function spaces can be converted into a conventional parameter optimization problem. The sparse VBI algorithm adopts an alternating optimization algorithm to find a stationary solution. The alternating optimization algorithm can be viewed as a special case of the block MM algorithm. The sequence of the objective function in (30) generated by the alternating updating is non-increasing. Together with the fact that the objective function (30) has a lower bound, the sequence converges to a limit. Let the surrogate function be chosen as the objective function itself. Since for any variable in problem \mathcal{A}_{VBI} , we have a unique solution, given by (31), according to Theorem 2-b in [30], the sparse VBI algorithm converges to a stationary solution of Problem \mathcal{A}_{VBI} .

REFERENCES

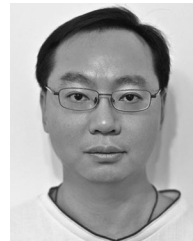
- [1] B. Li, "Stochastic modeling for vehicle platoons (I): Dynamic grouping behavior and online platoon recognition," *Transp. Res. Part B Methodological*, vol. 95, 2016.
- [2] J. A. del Peral-Rosado, J. A. Lopez-Salcedo, S. Kim, and G. Seco-Granados, "Feasibility study of 5G-based localization for assisted driving," in *Proc. Int. Conf. Localization GNSS (ICL-GNSS)*, Jun. 2016, pp. 1–6.
- [3] "C-RAN the road towards green RAN," China Mobile Research Institute, Beijing, China, Tech. Rep., Oct. 2011.
- [4] E. Larsson, O. Edfors, F. Tufvesson, and T. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

- [5] T. Wei and X. Zhang, "mTrack: High-precision passive tracking using millimeter wave radios," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 117–129.
- [6] X. Cui, T. A. Gulliver, J. Li, and H. Zhang, "Vehicle positioning using 5G millimeter-wave systems," *IEEE Access*, vol. 4, pp. 6964–6973, 2016.
- [7] J. Palacios, P. Casari, and J. Widmer, "Jade: Zero-knowledge device localization and environment mapping for millimeter wave systems," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2017, pp. 1–9.
- [8] Z. Lin, T. Lv, and P. T. Mathiopoulos, "3-D indoor positioning for millimeter-wave massive MIMO systems," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2472–2486, Jun. 2018.
- [9] A. Shahmansoori, G. E. Garcia, G. Destino, G. Seco-Granados, and H. Wymeersch, "Position and orientation estimation through millimeter-wave MIMO in 5G systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1822–1835, Mar. 2018.
- [10] 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.
- [11] J. C. Chen, R. E. Hudson, and K. Yao, "Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field," *IEEE Trans. Signal Process.*, vol. 50, no. 8, pp. 1843–1854, Aug. 2002.
- [12] A. Amar and A. J. Weiss, "Direct position determination of multiple radio signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2004, vol. 2, pp. ii–81.
- [13] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for bayesian inference," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 131–146, Nov. 2008.
- [14] J. Dai, A. Liu, and V. K. N. Lau, "FDD massive MIMO channel estimation with arbitrary 2D-array geometry," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2584–2599, May 2018.
- [15] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.
- [16] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a markov-tree prior," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3439–3448, Jul. 2012.
- [17] L. Chen, A. Liu, and X. Yuan, "Structured turbo compressed sensing for massive MIMO channel estimation using a markov prior," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4635–4639, May 2018.
- [18] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge University Press, 2005.
- [19] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Jun. 2001.
- [20] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.
- [21] X. Rao and V. Lau, "Compressive sensing with prior support quality information and application to massive MIMO channel estimation with temporal correlation," *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4914–4924, Sep. 2015.
- [22] A. Liu, V. K. N. Lau, and W. Dai, "Exploiting burst-sparsity in massive MIMO with partial channel support information," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7820–7830, Nov. 2016.
- [23] G. F. Cooper, "The computational complexity of probabilistic inference using bayesian belief networks," *Artif. Intell.*, vol. 42, no. 2-3, pp. 393–405, 1990.
- [24] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Info. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [25] C. W. Fox and S. J. Roberts, "A tutorial on variational Bayesian inference," *Artif. Intell. Rev.*, vol. 38, no. 2, pp. 85–95, 2012.
- [26] P. Schniter, "Turbo reconstruction of structured sparse signals," in *Proc. 44th Annu. Conf. Inf. Sci. Syst.*, Mar. 2010, pp. 1–6.
- [27] A. Rakotomamonjy, "Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms," *Signal Process.*, vol. 91, pp. 1505–1526, 2011.
- [28] K. Haneda *et al.*, "5G 3GPP-like channel models for outdoor urban microcellular and macrocellular environments," in *Proc. IEEE 83rd Veh. Technol. Conf. (Spring)*, Nanjing, China, May 2016, pp. 1–7.
- [29] C. J. Wu *et al.*, "On the convergence properties of the EM algorithm," *Ann. Statist.*, vol. 11, no. 1, pp. 95–103, 1983.
- [30] M. Razaviyayn, "Successive convex approximation: Analysis and applications," Ph.D. dissertation, University of Minnesota, 2014.



An Liu (Senior Member, IEEE) received the Ph.D. and B.S. degrees in electrical engineering from Peking University, Beijing, China, in 2011 and 2004 respectively. From 2008 to 2010, he was a Visiting Scholar with the Department of Electronics and Communication Engineering, University of Colorado at Boulder. He was a Postdoctoral Research Fellow in 2011–2013, Visiting Assistant Professor in 2014, and Research Assistant Professor in 2015–2017, with the Department of Electronics and Communication Engineering, Hong Kong University of Science and Technology. He is currently a Distinguished Research Fellow with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include wireless communications, stochastic optimization, and compressive sensing.

Lixiang Lian (Student Member, IEEE) photograph and biography not available at the time of publication.



Vincent K. N. Lau (Fellow, IEEE) received the B.Eng. degree (with distinction 1st hon.) from the University of Hong Kong, Hong Kong, in 1992, and the Ph.D. degree from Cambridge University, Cambridge, U.K., in 1997. He joined Bell Labs from 1997 to 2004 and the Department of Electronics and Communication Engineering, Hong Kong University of Science and Technology (HKUST), in 2004. He is currently a Chair Professor and the Founding Director of Huawei-HKUST Joint Innovation Lab, HKUST. His current research interests include robust and delay-optimal cross layer optimization for MIMO/OFDM wireless systems, interference mitigation techniques for wireless networks, massive MIMO, M2M and network control systems.

Guanying Liu photograph and biography not available at the time of publication.



Minjian Zhao (Member, IEEE) received the M.Sc. and Ph.D. degrees in communication and information systems from Zhejiang University, Hangzhou, China, in 2000 and 2003, respectively. He was a Visiting Scholar with the University of York, U.K., in 2010. He is currently a Professor and the Deputy Director with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include modulation theory, channel estimation and equalization, MIMO, signal processing for wireless communications, anti-jamming technology for wireless transmission and networking, communication SOC chip design, etc.