Bayesian Over-The-Air Computation

Yulin Shao, *Member, IEEE*, Deniz Gündüz, *Senior Member, IEEE*, Soung Chang Liew, *Fellow, IEEE*

***Abstract*—Over-the-air computation (OAC) is an efficient solution to a class of uplink data aggregation tasks in the multiple-access channel (MAC), wherein the common receiver of the MAC, dubbed the fusion center, aims to reconstruct a function of the data distributed at edge devices rather than the individual data themselves. Traditional OAC relies on the maximum likelihood (ML) estimation at the fusion center to recover the arithmetic sum of the transmitted signals from different devices. ML estimation, however, is very much susceptible to noise. In particular, in the misaligned OAC where there are channel misalignments among transmitted signals, ML estimation suffers from severe error propagation and noise enhancement. To address these challenges, this paper puts forth a Bayesian approach for OAC, leveraging two pieces of statistical information transmitted from each edge device. Three OAC systems are studied: the aligned OAC with perfectly-aligned signals; the synchronous OAC with misaligned channel gains among the received signals; and the asynchronous OAC with both channel-gain and time misalignments. Taking the statistics transmitted from the devices as prior information, we devise linear minimum mean squared error (LMMSE) estimators and a sum-product maximum a posteriori (SP-MAP) estimator for the three OAC systems. Numerical and simulation results verify that, 1) For the aligned OAC and the synchronous OAC, our LMMSE estimator significantly outperforms the ML estimator. In the low-EsN0 regime, the LMMSE estimator reduces the MSE by at least 6 dB; in the high-EsN0 regime, the LMMSE estimator lowers the error floors by 86.4%. 2) For the asynchronous OAC, our LMMSE and SP-MAP estimators are on an equal footing in terms of the MSE performance, and are significantly better than the ML estimator. On the other hand, in terms of the computational complexity, the SP-MAP estimator is much more efficient than the LMMSE estimator.**

***Index Terms*—Over-the-air computation, analog communication, Bayesian estimation, misalignment, linear minimum mean square error, sum product algorithm.**

I. Introduction

In multiple-access channels (MACs), the fusion center (i.e., the common receiver of the MAC) is often interested in some functions of the data distributed across the edge devices, but not their individual data [1]–[15]. In distributed sensing networks, for example, the fusion center is only interested in functions of the sensor readings such as mean humidity and maximum temperature [3], [6]; in federated learning systems, the fusion center is only interested in the arithmetic mean of the local updates transmitted from the edge devices but not their individual updates [1], [16].

Y. Shao was with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong. He is now with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: y.shao@imperial.ac.uk). D. Gündüz is with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: d.gunduz@imperial.ac.uk). S. C. Liew is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong (e-mail: soung@ie.cuhk.edu.hk).

Aiming for function computations, the distributed data aggregation in MAC can be realized in a digital fashion via traditional multiple-access technologies [17], [18] (e.g., TDMA, CDMA, OFDMA). Specifically, the messages to be transmitted from the edge devices are first digitized and then transmitted in orthogonal links to the fusion center. The desired function values can be computed after decoding all individual messages from the edge devices. However, such a separate communication-and-computation approach is highly suboptimal in that the individual messages are undesired at the fusion center – transmitting them causes excessive bandwidth and latency overhead.

Over-the-air computation (OAC) is an alternative technique to realize efficient distributed data aggregation in MACs [1], [3], [5]–[15], [19], [20]. Compared with the digital approach, OAC is a joint computation-and-communication scheme exploiting the fact that the MAC is inherently a superposition of signals. The underpinnings of OAC are pre-processing, channel precoding, and post-processing [3], [5]. As shown in Fig. 1, each device first pre-processes the transmitted symbols by a pre-processing function, and then precodes the pre-processed symbols by the inversion of the uplink channel. The precoded symbols are transmitted to the fusion center in an analog fashion (amplitude modulation). In particular, different devices transmit simultaneously in the same communication link and their signals overlap at the receiver. The fusion center then post-processes the overlapped signal to reconstruct the desired function values.

The pre-processing and post-processing functions are chosen so that the desired function values are directly produced after post-processing. For example, to compute the geometric mean at the fusion center, we can choose the pre-processing function to be a logarithm function and the post-processing function to be an exponential function [3]. In general, the functions that are computable via OAC are functions that can be broken into a post-processed summation of multiple pre-processed functions. This class of functions is named *nomographic functions* [5]. On the other hand, the purpose of channel precoding is to compensate for the channel impairments so that the fading MAC degenerates to a Gaussian MAC. As a result, when the transmitted signals arrive at the fusion center simultaneously, the signals overlapped over-the-air naturally produce the arithmetic sum of the pre-processed signals.

In practice, however, accurate channel-gain precoding and strict synchronization among devices are very expensive to achieve [1], [2], [8], especially with low-cost Internet-of-Things (IoT) devices. With either channel-gain mismatches or time asynchronies, the signals from different devices are misaligned at the fusion center, which we refer to as the misaligned OAC [8].

Prior works on the aligned OAC [1]–[3], [5], [7], [9], [10], [12], [13], [21] or the misaligned OAC [8] rely exclusively on the maximum likelihood (ML) estimator to recover the arithmetic sum of the transmitted signals from different devices. ML estimation, however, is very much susceptible to noise. Our companion paper [8] showed that, in the misaligned OAC, the arithmetic-sum estimation boils down to multi-user estimation and the estimation space is infinitely large considering the continuous nature of OAC. As a result, ML estimation suffers from severe error propagation and noise enhancement with even mild channel-gain or time misalignment.

This paper puts forth a Bayesian approach for OAC to address the problems faced by ML estimation. Specifically, we let each edge device transmit two pieces of statistical information (i.e., the first and second sample moments) of the transmitted data and leverage these statistical characteristics as a kind of prior information to construct Bayesian estimators at the receiver to estimate the arithmetic-sum of the transmitted signals. Three OAC systems are considered: 1) The aligned OAC where the transmitted signals are perfectly aligned at the fusion center with neither channel-gain misalignment nor time misalignment. 2) The synchronous OAC where there is only channel-gain misalignment but no time misalignment. The aligned OAC is a special case of the synchronous OAC when the channel-gain precoding is perfect and there are no channel-gain mismatches in the overlapped signal. 3) The asynchronous OAC where there are both channel-gain and time misalignments. The synchronous OAC is a special case of the asynchronous OAC when the calibrations of transmission timings at the edge devices are accurate and there are no asynchronies among the overlapped signals.

The main contributions of this paper are as follows:

1) For the aligned OAC and the synchronous OAC, we devise a linear minimum mean square error (LMMSE) estimator using the two pieces of prior information transmitted from the edge devices. The MSE performances of both ML and LMMSE estimators are derived. Numerical and simulation results verify that i) for the aligned OAC, the use of prior information (i.e., the LMMSE estimator) brings at least 6 dB gains over the ML estimator in the low-EsN0 regime; ii) for the synchronous OAC, in addition to the EsN0 gains, the error floor in the high-EsN0 regime is lowered by a large margin with the LMMSE estimator. When there is mild phase misalignment, for example, the error floor is lowered by 86.4%.
2) For the asynchronous OAC, we make use of a whitened matched-filtering and sampling scheme to produce oversampled, but independent samples, whereby an ML estimator, an LMMSE estimator, and a sum-product maximum *a posteriori* (SP-MAP) estimator are devised. In particular, our SP-MAP estimator exploits both the prior information transmitted from the edge devices and the sparsity of the sample structure, and is verified to be the most preferred estimator in the asynchronous OAC in terms of both the MSE performance and the computational complexity. Compared with the ML estimator, the SP-MAP estimator addresses the problems of error propagation and noise enhancement and attains significantly lower MSE performances under various degrees of time and phase misalignments. Compared with the LMMSE estimator, the MSE performance of the SP-MAP estimator is on the same footing. Yet, the SP-MAP estimator reduces the computational complexity of the LMMSE estimator from $\Omega(L^2 \log L)$ to $\Omega(L)$ for a packet of length L.

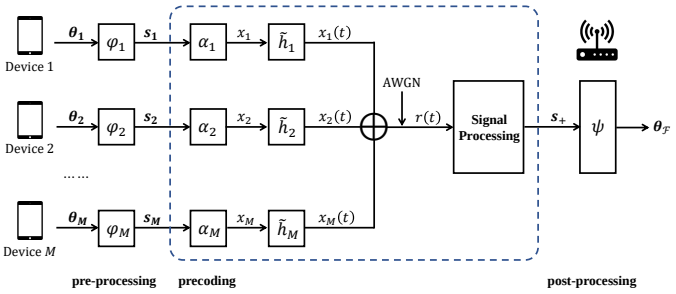

Figure 1. Distributed data aggregation in the MAC via OAC.

Notations – We use boldface lowercase letters to denote column vectors (e.g., $\boldsymbol{\theta}$, $\boldsymbol{s}$) and boldface uppercase letters to denote matrices (e.g., $\boldsymbol{V}$, $\boldsymbol{D}$). For a vector or matrix, $(\cdot)^\top$ denotes the transpose, $(\cdot)^*$ denotes the complex conjugate, $(\cdot)^H$ denotes the conjugate transpose, and $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse. $\mathbb{R}$ and $\mathbb{C}$ stand for the sets of real and complex values, respectively. $(\cdot)^{\mathfrak{r}}$ and $(\cdot)^{\mathfrak{i}}$ stand for the real and imaginary components of a complex symbol or vector, respectively. The imaginary unit is represented by j. $\mathcal{C}$ and $\mathcal{CN}$ stand for the real and complex Gaussian distribution, respectively. The cardinality of a set $\mathcal{V}$ is denoted by $|\mathcal{V}|$. The sign function is denoted by $\text{sgn}(\cdot)$.

II. System Model

We consider a MAC where M edge devices communicate with a fusion center, as shown in Fig. 1. The message of the m-th device is a vector of L complex values $\boldsymbol{\theta_m} \in \mathbb{C}^L$. The desired message of the fusion center, denoted by $\boldsymbol{\theta}_{\mathcal{F}} \in \mathbb{C}^L$, is a function of $\{\boldsymbol{\theta_m} : m = 1, 2, ..., M\}$ and each element of $\boldsymbol{\theta}_{\mathcal{F}}$ is $\theta_{\mathcal{F}}[\ell] = \mathcal{F}(\theta_1[\ell], \theta_2[\ell], ..., \theta_M[\ell])$. In particular, $\mathcal{F}$ can be written in a nomographic form as

$$\mathcal{F}(\theta_1[\ell], \theta_2[\ell], ..., \theta_M[\ell]) = \psi\left(\sum_{m=1}^{M} \varphi_m(\theta_m[\ell])\right), \tag{1}$$

where $\{\varphi_m,\ m = 1, 2, ..., M\}$ are the pre-processing functions and ψ is a post-processing function. In other words, the nomographic form is a post-processed summation of multiple pre-processed functions. It has been shown in [22] that in general any function can be written into a nomographic form.

A. Distributed data aggregation via OAC

With OAC, the distributed data aggregation in the MAC works in the following manner. First, each of the M devices pre-processes the message $\boldsymbol{\theta_m}$ by a preprocessing function φ_m and obtains

$$\boldsymbol{s_m} = \varphi_m(\boldsymbol{\theta_m}).$$

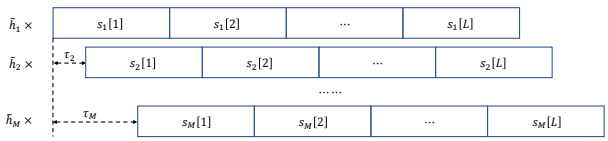

Figure 2. The transmitted signals from different devices overlap at the fusion center with channel-gain and time misalignments.

The pre-processed message $\boldsymbol{s_m}$ is then precoded by a channel-precoding factor α_m (in the ideal case, α_m is the channel inversion).

After pulse shaping, the time-domain signal to be transmitted by the m-th device is given by[1]

$$x_m(t) = \alpha_m \sum_{\ell=1}^{L} s_m[\ell] p(t - \ell T), \tag{2}$$

where $p(t) = 1/2\,[\text{sgn}(t+T) - \text{sgn}(t)]$ is a rectangular pulse of duration T. Each of the M edge devices then carefully calibrates the transmission timing, based on its distance from the fusion center and its moving speed, so that the signals from different devices arrive at the fusion center simultaneously.

In practice, however, both the channel-gain pre-compensation and the transmission-timing calibration can be imperfect due to the non-ideal hardware and inaccurate channel-gain/delay estimation. After passing through the fading MAC, the signals $x_m(t)$, $\forall m$, overlap at the fusion center with relative time offsets. The received signal $r(t)$ can be written as

$$r(t) = \sum_{m=1}^{M} \tilde{h}_m x_m(t - \tau_m) + z(t), \tag{3}$$

where $\tilde{h}_m$ is the time-domain complex channel gain. We consider flat and slow fading channels, and hence, $\tilde{h}_m$ is a constant over one transmission. $z(t)$ is the zero-mean baseband complex AWGN, the double-sided power spectral densities of which is N_0. Without loss of generality, we sort the M devices so that the devices with smaller indexes arrive at the receiver earlier. The delay of the first device is set to $\tau_1 = 0$, and the relative delay of the i-th device with respect to the first device is denoted by τ_m. We assume the time offsets τ_m, $\forall m$, are less than the symbol duration T, as shown in Fig. 2. In the ideal case where the transmission-timing calibrations are prefect, the relative delays among signals $\tau_m = 0$, $\forall m$.

Substituting (2) into (3) gives us

$$\begin{aligned} r(t) &= \sum_{m=1}^{M} \tilde{h}_m \alpha_m \sum_{\ell=1}^{L} s_m[\ell] p(t - \tau_m - \ell T) + z(t) \\ &= \sum_{\ell=1}^{L} \sum_{m=1}^{M} h_m s_m[\ell] p(t - \tau_m - \ell T) + z(t), \end{aligned} \tag{4}$$

where $h_m = \tilde{h}_m \alpha_m$ is the residual channel gain. To summarize, there can be two kinds of misalignments among the signals transmitted from different devices: channel-gain misalignment $\{h_m : m = 1, 2, ..., M\}$ caused by inaccurate channel-gain compensation at the transmitter and time misalignment $\{\tau_m : m = 1, 2, ..., M\}$ caused by imperfect calibration of the transmission timing.

As per (1), the objective of the fusion center is to estimate

$$\boldsymbol{\theta}_{\mathcal{F}} = \psi\left(\sum_{m=1}^{M} \varphi_m(\boldsymbol{\theta_m})\right) \triangleq \psi(\boldsymbol{s}_+),$$

where each element of sequence $\boldsymbol{s}_+$ is defined as

$$s_+[\ell] = \sum_{m=1}^{M} \varphi_m(\theta_m[\ell]) = \sum_{m=1}^{M} s_m[\ell].$$

For general pre-processing and post-processing functions, we shall focus exclusively on the estimation of $\boldsymbol{s}_+$ from the received signal $r(t)$. In particular, our goal is to minimize the mean squared error (MSE) between the true $\boldsymbol{s}_+$ and the estimated $\hat{\boldsymbol{s}}_+$:

$$\text{MSE}(\boldsymbol{s}_+, \hat{\boldsymbol{s}}_+) = \frac{1}{L} \sum_{\ell=1}^{L} \left| \hat{s}_+[\ell] - \sum_{m=1}^{M} s_m[\ell] \right|^2.$$

Remark. *It is worth noting that, for a specific pair of pre-processing and post-processing functions φ_m and ψ, minimizing the MSE between $\boldsymbol{s}_+$ and $\hat{\boldsymbol{s}}_+$ does not necessarily minimize the MSE between $\boldsymbol{\theta}_{\mathcal{F}}$ and the estimated $\hat{\boldsymbol{\theta}}_{\mathcal{F}}$*

$$\textit{MSE}(\boldsymbol{\theta}_{\mathcal{F}}, \hat{\boldsymbol{\theta}}_{\mathcal{F}}) = \frac{1}{L} \sum_{\ell=1}^{L} \left| \hat{\theta}_{\mathcal{F}}[\ell] - \psi\left(\sum_{m=1}^{M} \varphi_m(\theta_m[\ell])\right) \right|^2.$$

If the fusion center aims to compute the arithmetic mean of $\{\boldsymbol{\theta_m} : m = 1, 2, .., M\}$, for example, we choose $\varphi_m(x) = x$, $\psi(x) = x/M$, and

$$\textit{MSE}(\boldsymbol{\theta}_{\mathcal{F}}, \hat{\boldsymbol{\theta}}_{\mathcal{F}}) = \frac{1}{M^2} \textit{MSE}(\boldsymbol{s}_+, \hat{\boldsymbol{s}}_+).$$

Therefore, minimizing MSE$(\boldsymbol{\theta}_{\mathcal{F}}, \hat{\boldsymbol{\theta}}_{\mathcal{F}})$ is equivalent to minimizing MSE$(\boldsymbol{s}_+, \hat{\boldsymbol{s}}_+)$ in this case.

On the other hand, if the fusion center aims to compute the geometric mean of $\{\boldsymbol{\theta_m} : m = 1, 2, .., M\}$, we choose $\varphi_m(x) = \ln x$, $\psi(x) = \exp(x/M)$, and

$$\textit{MSE}(\boldsymbol{\theta}_{\mathcal{F}}, \hat{\boldsymbol{\theta}}_{\mathcal{F}}) = \frac{1}{L} \sum_{\ell=1}^{L} \left| \exp\left(\frac{\hat{s}_+[\ell]}{M}\right) - \exp\left(\frac{\sum_{m=1}^{M} s_m[\ell]}{M}\right) \right|^2.$$

In this case, MSE$(\boldsymbol{\theta}_{\mathcal{F}}, \hat{\boldsymbol{\theta}}_{\mathcal{F}}) \approx \frac{1}{M^2}MSE(\boldsymbol{s}_+, \hat{\boldsymbol{s}}_+)$ only when both $\frac{1}{M}\hat{s}_+[\ell] \to 0$ and $\frac{1}{M}\sum_{m=1}^{M} s_m[\ell] \to 0$.

In this paper, we consider general pre-processing and post-processing functions and focus on the MSE between $\hat{\boldsymbol{s}}_+$ and $\hat{\boldsymbol{s}}_+$) (i.e., the part inside the dashed box in Fig. 1).

B. The Aligned OAC

Most prior works on OAC considered only the perfectly aligned case where there is neither channel-gain misalignment nor time misalignment, which we refer to as the *aligned OAC*. The received signal in the aligned OAC is given by

$$r(t) = \sum_{\ell=1}^{L} \sum_{m=1}^{M} s_m[\ell] p(t - \ell T) + z(t). \tag{5}$$

[1]In this paper, we formulate the channel-misaligned OAC considering the time-domain realization of OAC. OAC can also be realized in the frequency domain via OFDM. Interested readers may refer to Section VI of our companion paper [8] for a more detailed discussion.

Matched filtering $r(t)$ by the same rectangular pulse $p(t)$ and sampling at $t = iT$, $i = 1, 2, ..., L$ give us

$$r[i] = \frac{1}{T}\int_{(i-1)T}^{iT} r(t)dt = \sum_{m=1}^{M} s_m[i] + z[i] = s_+[i] + z[i], \quad (6)$$

where the noise sequence $z[i]$ in the samples is independent and identically distributed (i.i.d.), $z[i] \sim \mathcal{CN}(0, \frac{N_0}{T})$. As can be seen, the target signal $s_+[i]$ appears explicitly on the RHS of (6). A simple estimator can then be used to estimate the sequence $\boldsymbol{s}_+$.

Definition 1 (ML estimation for the aligned OAC)**.** *Given a sequence of samples $\boldsymbol{r} \in \mathcal{C}^L$ in* (6)*, an ML estimator estimates the target sequence $\boldsymbol{s}_+ \in \mathcal{C}^L$ symbol-by-symbol by*

$$\hat{s}_+[i] = r[i]. \quad (7)$$

Eq. (7) is an ML estimator in that the sample $r[i]$ in (6) is conditional Gaussian – given $s_+[i]$, the likelihood function

$$f(r[i] \mid s_+[i]) \sim \mathcal{CN}\left(s_+[i], \frac{N_0}{T}\right).$$

Therefore, the ML estimate of $s_+[i]$ for a given observation $r[i]$ is

$$\hat{s}_+[i] = \arg\max_{s_+[i]} \Pr\left(r[i] \mid s_+[i]\right) = r[i].$$

Accurate precoding and transmission-timing calibration at the transmitters admit a very simple sample structure at the receiver since the target signal $\boldsymbol{s}_+$ is explicitly presented in the samples. In this case, the fading MAC degenerates to a Gaussian MAC and the M devices can be abstracted as a single device transmitting the summation of $\boldsymbol{s_m}$ directly to the fusion center. In practice, however, both the channel-gain compensation and the calibration of transmission timing can be inaccurate. With either channel-gain or time misalignment, clean samples as in (6) with $\boldsymbol{s}_+$ explicitly presented are no longer available. The design of optimal estimators is thus more challenging.

In the context of federated edge learning, our companion paper [8] studied the estimation problem for the misaligned OAC and devised estimators that are ML optimal. In this paper, we further extend our study in [8] considering a Bayesian approach. Unlike ML estimation where the transmitted symbols $\boldsymbol{s}_m$ are treated as constants, this paper treats $\boldsymbol{s}_m$ as random variables and exploits the statistical characteristic of $\boldsymbol{s}_m$ as a kind of prior information to perform Bayesian estimation at the receiver. In so doing, significant gains can be achieved for both aligned and misaligned OAC.

III. Synchronous OAC

This section focuses on the synchronous OAC where there is channel-gain misalignment but no time misalignment in the received signal. That is, we assume the calibrations of transmission timing at the edge devices are satisfactory and the relative time offsets of different signals at the fusion center are negligible.

After matched filtering and sampling, the samples we obtained can be written as

$$r[i] = \sum_{m=1}^{M} h_m s_m[i] + z[i], \quad (8)$$

where h_m is the residual channel gain due to inaccurate channel-gain precoding. From each sample $r[i]$, our goal is to estimate $s_+[i] = \sum_{m=1}^{M} s_m[i]$.

Eq. (8) is an underdetermined equation since we have one equation for M unknowns. To estimate $s_+[i]$, the only viable estimator in the literature is the ML estimator given in Definition 1 – the raw sample $r[i]$ is the best prediction about $s_+[i]$ [8]. In this section, however, we show that a much more powerful estimator can be devised by letting each edge device transmit two pieces of prior information to the receiver, whereby the MSE of the estimated $\hat{\boldsymbol{s}}_+$ can be significantly reduced.

A. The ML Estimator

To start with, let us first analyze the MSE performance of the ML estimator to estimate $\boldsymbol{s}_+$ from (8).

Proposition 1 (MSE of the ML estimator in synchronous OAC)**.** *Given the samples in* (8)*, the MSE performance of the ML estimator is*

$$MSE_{ML} = (\boldsymbol{h} - \boldsymbol{1})^H \boldsymbol{V}(\boldsymbol{h} - \boldsymbol{1}) + \frac{N_0}{T}, \quad (9)$$

where $\boldsymbol{h} = [h_1, h_2, \cdots, h_M]^\top$,

$$\boldsymbol{V} = \begin{bmatrix} \hat{\mathbb{V}}_1 & \hat{\mathbb{E}}_1^*\hat{\mathbb{E}}_2 & \cdots & \hat{\mathbb{E}}_1^*\hat{\mathbb{E}}_M \\ \hat{\mathbb{E}}_2^*\hat{\mathbb{E}}_1 & \hat{\mathbb{V}}_2 & \cdots & \hat{\mathbb{E}}_2^*\hat{\mathbb{E}}_M \\ \cdots & \cdots & \cdots & \cdots \\ \hat{\mathbb{E}}_M^*\hat{\mathbb{E}}_1 & \hat{\mathbb{E}}_M^*\hat{\mathbb{E}}_2 & \cdots & \hat{\mathbb{V}}_M \end{bmatrix}, \quad (10)$$

and

$$\hat{\mathbb{E}}_m = \frac{1}{L}\sum_{i=1}^{L} s_m[i], \quad \hat{\mathbb{V}}_m = \frac{1}{L}\sum_{i=1}^{L} |s_m[i]|^2, \quad (11)$$

are the first sample moment and the second sample moment of the symbols transmitted by the m-th device in one transmission.

Proof. As per Definition 1, we have $\hat{s}_+[i] = r[i]$ with the ML estimator. The MSE of $\hat{\boldsymbol{s}}_+$ is then

$$\begin{aligned} \text{MSE}_{\text{ML}} &= \frac{1}{L}\sum_{i=1}^{L} |\hat{s}_+[i] - s_+[i]|^2 \\ &= \frac{1}{L}\sum_{i=1}^{L} \left|\sum_{m=1}^{M} (h_m - 1)s_m[i] + z[i]\right|^2 \\ &\overset{(a)}{=} \frac{1}{L}\sum_{i=1}^{L} \left|\sum_{m=1}^{M} (h_m - 1)s_m[i]\right|^2 + \frac{1}{L}\sum_{i=1}^{L} |z[i]|^2 \\ &= \frac{1}{L}\sum_{i=1}^{L} \left|\sum_{m=1}^{M} (h_m - 1)s_m[i]\right|^2 + \frac{N_0}{T}, \end{aligned}$$

where (a) follows because $s_m[i]$ is independent of the noise term $z[i]$; the last equality follows since the noise terms are i.i.d. for different i.

Defining $\boldsymbol{h} = [h_1, h_2, ..., h_M]^\top$ and $\boldsymbol{s[i]} = [s_1[i], s_2[i], ..., s_M[i]]^\top$, MSE_{ML} can be written in a more compact form as

$$\begin{aligned}\text{MSE}_{\text{ML}} &= \frac{1}{L}\sum_{i=1}^{L}\left|(\boldsymbol{h}-\boldsymbol{1})^\top \boldsymbol{s[i]}\right|^2 + \frac{N_0}{T} \\ &= (\boldsymbol{h}-\boldsymbol{1})^H \frac{1}{L}\sum_{i=1}^{L}\boldsymbol{s^*[i]}\boldsymbol{s}^\top\boldsymbol{[i]}(\boldsymbol{h}-\boldsymbol{1}) + \frac{N_0}{T} \\ &\triangleq (\boldsymbol{h}-\boldsymbol{1})^H \boldsymbol{V}(\boldsymbol{h}-\boldsymbol{1}) + \frac{N_0}{T},\end{aligned}$$

where the matrix $\boldsymbol{V}$ is given by

$$\begin{aligned}\boldsymbol{V} &= \frac{1}{L}\sum_{i=1}^{L}\boldsymbol{s^*[i]}\boldsymbol{s}^\top\boldsymbol{[i]} \\ &= \frac{1}{L}\sum_{i=1}^{L}\begin{bmatrix} |s_1[i]|^2 & s_1^*[i]s_2[i] & \cdots & s_1^*[i]s_M[i] \\ s_2^*[i]s_1[i] & |s_2[i]|^2 & \cdots & s_2^*[i]s_M[i] \\ \cdots & \cdots & \cdots & \cdots \\ s_M^*[i]s_1[i] & s_M^*[i]s_2[i] & \cdots & |s_M[i]|^2 \end{bmatrix}. \end{aligned} \quad (12)$$

If we define the first and second sample moments of the symbols transmitted by the m-th device as in (11), $\boldsymbol{V}$ can be written in the form of (10). ■

Corollary 2 (MSE of the ML estimator in the aligned OAC)**.** *Consider the aligned OAC where there is neither channel-gain nor time misalignment. Given the samples in* (6)*, the MSE of the ML estimator is*

$$\textit{MSE}_{\textit{ML}} = \frac{N_0}{T}, \quad (13)$$

Proof. Eq. (13) follows directly from (9) by setting $\boldsymbol{h} = \boldsymbol{1}$. ■

To summarize, the MSE of the ML estimator is simply the noise variance in the aligned OAC. When there is channel-gain misalignment (i.e., the synchronous OAC), the additional MSE introduced by the channel-gain misalignment is $(\boldsymbol{h}-\boldsymbol{1})^H\boldsymbol{V}(\boldsymbol{h}-\boldsymbol{1})$.

B. The LMMSE Estimator

According to (9), the MSE performance of the ML estimator can be poor when either $|\boldsymbol{h}-\boldsymbol{1}|^2$ is large or the noise variance is large. To improve the estimation performance, this subsection devises a linear minimum MSE (LMMSE) estimator. We will show that, by letting each device transmit two pieces of prior information, MSE can be significantly reduced with the LMMSE estimator.

Definition 2 (LMMSE estimation)**.** *In each transmission, we let each device transmit the first sample moment $\hat{\mathbb{E}}_m$ and the second sample moment $\hat{\mathbb{V}}_m$ to the fusion center reliably (in a digital way, with channel coding and automatic repeat request, for example) before the data transmission.*

At the fusion center, given a sequence of received samples $\boldsymbol{r} \in \mathcal{C}^L$ in (8)*, an LMMSE estimator estimates the sequence $\boldsymbol{s}_+ \in \mathcal{C}^L$ symbol-by-symbol by*

$$\hat{s}_+[i] = \frac{\boldsymbol{h}^H\boldsymbol{D}\boldsymbol{1}}{\boldsymbol{h}^H\boldsymbol{D}\boldsymbol{h}+\frac{N_0}{T}}r[i] + \left(\boldsymbol{1} - \frac{\boldsymbol{h}^H\boldsymbol{D}\boldsymbol{1}}{\boldsymbol{h}^H\boldsymbol{D}\boldsymbol{h}+\frac{N_0}{T}}\boldsymbol{h}\right)^\top \hat{\boldsymbol{\mu}}, \quad (14)$$

where the vector $\hat{\boldsymbol{\mu}}$ and the matrix $\boldsymbol{D}$ can be constructed from the first and second sample moments transmitted from the edge devices, giving

$$\hat{\boldsymbol{\mu}} = \left[\hat{\mathbb{E}}_1, \hat{\mathbb{E}}_2, \cdots, \hat{\mathbb{E}}_M\right]^\top, \quad (15)$$

$$\boldsymbol{D} = \textit{diag}\left(\hat{\mathbb{D}}_1, \hat{\mathbb{D}}_2, \cdots, \hat{\mathbb{D}}_M\right), \quad (16)$$

and $\hat{\mathbb{D}}_m$ is defined to be the sample variance of the symbols transmitted by the m-th device in one transmission, giving

$$\hat{\mathbb{D}}_m = \frac{1}{L}\sum_{i=1}^{L}|s_m[i] - \hat{\mathbb{E}}_m|^2 = \hat{\mathbb{V}}_m - |\hat{\mathbb{E}}_m|^2. \quad (17)$$

Next, we prove that (14) is indeed an LMMSE estimator and analyze its MSE.

Lemma 3 (positive definiteness of $\boldsymbol{D}$ and $\boldsymbol{V}$)**.** *Matrix $\boldsymbol{D}$ is positive definite; matrix $\boldsymbol{V}$ is Hermitian positive definite.*

Proof. For the diagonal matrix $\boldsymbol{D}$, its element $\hat{\mathbb{D}}_m$ is the sample variance of the symbols transmitted by a device. This suggests that $\hat{\mathbb{D}}_m > 0$, $\forall m$. As a result, $\boldsymbol{D}$ is positive definite.

For the complex matrix $\boldsymbol{V}$, it is obvious from (12) that $\boldsymbol{V}$ is Hermitian since $\boldsymbol{V}^H = \boldsymbol{V}$. For any complex vector $\boldsymbol{x} = [x_1, x_2, ..., x_M] \in \mathcal{C}^M$, $\boldsymbol{x} \neq \boldsymbol{0}$, we have

$$\begin{aligned}&\boldsymbol{x}^H\boldsymbol{V}\boldsymbol{x} = \\ &[x_1^*, x_2^*, ..., x_M^*]\frac{1}{L}\sum_{i=1}^{L}\begin{bmatrix} |s_1[i]|^2 & s_1^*[i]s_2[i] & \cdots & s_1^*[i]s_M[i] \\ s_2^*[i]s_1[i] & |s_2[i]|^2 & \cdots & s_2^*[i]s_M[i] \\ \cdots & \cdots & \cdots & \cdots \\ s_M^*[i]s_1[i] & s_M^*[i]s_2[i] & \cdots & |s_M[i]|^2 \end{bmatrix}\begin{bmatrix} x_1^* \\ x_2^* \\ \cdots \\ x_M^* \end{bmatrix} \\ &= \frac{1}{L}\sum_{i=1}^{L}\left|\sum_{m=1}^{M}x_m s_m[i]\right|^2 > 0.\end{aligned}$$

Thus, $\boldsymbol{V}$ is Hermitian positive definite. ■

Theorem 4 (MSE of the LMMSE estimator)**.** *The estimator in* (14) *is an LMMSE estimator that minimizes the MSE*

$$\textit{MSE} = \frac{1}{L}\sum_{i=1}^{L}\left|\lambda r[i] + c - s_+[i]\right|^2, \quad (18)$$

for any linear estimator with λ, $c \in \mathcal{C}$. The MSE of the LMMSE estimator is given by

$$\textit{MSE}_{\textit{LMMSE}} = \boldsymbol{1}^\top\boldsymbol{D}\boldsymbol{1} - \frac{\left|\boldsymbol{h}^H\boldsymbol{D}\boldsymbol{1}\right|^2}{\boldsymbol{h}^H\boldsymbol{D}\boldsymbol{h}+\frac{N_0}{T}}, \quad (19)$$

and we have $\textit{MSE}_{\textit{LMMSE}} \leq \textit{MSE}_{\textit{ML}}$.

Proof. See Appendix A. ■

Corollary 5 (LMMSE estimator for the aligned OAC)**.** *Consider the aligned OAC where there is neither channel-gain nor time misalignment. Given the samples in* (6)*, an LMMSE estimator estimates the sequence $\boldsymbol{s}_+ \in \mathcal{C}^L$ symbol-by-symbol by*

$$\hat{s}_+[i] = \frac{\boldsymbol{1}^\top\boldsymbol{D}\boldsymbol{1}}{\boldsymbol{1}^\top\boldsymbol{D}\boldsymbol{1}+\frac{N_0}{T}}r[i] + \frac{\frac{N_0}{T}}{\boldsymbol{1}^\top\boldsymbol{D}\boldsymbol{1}+\frac{N_0}{T}}\boldsymbol{1}^\top\hat{\boldsymbol{\mu}}, \quad (20)$$

The MSE of the LMMSE estimator in the aligned OAC is

$$\textit{MSE}_{\textit{LMMSE}} = \frac{\frac{N_0}{T}\boldsymbol{1}^\top\boldsymbol{D}\boldsymbol{1}}{\boldsymbol{1}^\top\boldsymbol{D}\boldsymbol{1}+\frac{N_0}{T}}. \quad (21)$$

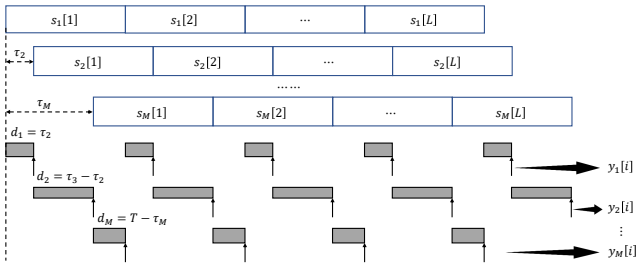

Figure 3. Matched filtering the received signal by a bank of M filters of lengths $d_k = \tau_{k+1} - \tau_k$.

Compared with the aligned estimator, we have $MSE_{LMMSE} < MSE_{ML}$ where MSE_{ML} is given in (13).

Proof. Eq. (20) and (21) follow directly from (14) and (19) by setting $\boldsymbol{h} = \mathbf{1}$. Further, we have

$$\frac{\text{MSE}_{\text{ML}}}{\text{MSE}_{\text{LMMSE}}} = \frac{\mathbf{1}^\top \boldsymbol{D}\mathbf{1} + \frac{N_0}{T}}{\mathbf{1}^\top \boldsymbol{D}\mathbf{1}} > 1, \tag{22}$$

and hence $\text{MSE}_{\text{LMMSE}} < \text{MSE}_{\text{ML}}$. ■

To summarize, the ML estimator is susceptible to both channel-gain misalignment $\boldsymbol{h}$ and noise. The LMMSE estimator, on the other hand, can alleviate the estimation errors caused by both the channel-gain misalignment and noise, thanks to the prior information transmitted from the edge devices. The MSE gains are not analytically straightforward from (9) and (19) for the general channel-gain misalignment $\boldsymbol{h}$. Thus, let us consider the aligned OAC wherein $\boldsymbol{h} = \mathbf{1}$. It can be seen from (22) that the MSE gains of the LMMSE estimator over the ML estimator are 3 dB when $\mathbf{1}^\top \boldsymbol{D}\mathbf{1} = \frac{N_0}{T}$. Further, when $\mathbf{1}^\top \boldsymbol{D}\mathbf{1} \leq \frac{N_0}{T}$, the gains can be significant.

IV. Asynchronous OAC

This section considers the asynchronous OAC where there are both channel-gain and time misalignments in the received signal. Let us first reproduce the received signal $r(t)$ in (4) below.

$$r(t) = \sum_{\ell=1}^{L}\sum_{m=1}^{M} h_m s_m[\ell] p(t - \tau_m - \ell T) + z(t), \tag{23}$$

where $s_m[\ell]$ is the ℓ-th complex number transmitted from the m-th device; $h_m = |h_m| e^{j\phi_m}$ is the residual channel gain; τ_m is the time offset of the m-th device relative to the first device.

A. Whitened Matched Filtering and Sampling

As shown in Fig. 2, the symbols from different devices in the asynchronous OAC are misaligned in time. To obtain a whitened discrete model from the received signal, we employ a bank of M matched filters of different lengths to collect power judiciously from $r(t)$.

The matched filtering and sampling process are illustrated in Fig. 3. Specifically, the M matched filters $\{p'_k(t) : k = 1, 2, ..., M\}$ are defined as

$$p'_k(t) = \frac{1}{2}\big[\text{sgn}(t+T) - \text{sgn}(t+T-d_k)\big], \tag{24}$$

where the length of the k-th matched filter is $d_k = \tau_{k+1} - \tau_k$, $k = 1, 2, ..., M$. For completeness, we define $\tau_{M+1} = T$.

The signal filtered by the k-th matched filter is given by

$$\begin{aligned} y_k(t) &= \frac{1}{d_k}\int_{-\infty}^{\infty} r(\zeta) p'_k(t-\zeta)\, d\zeta \\ &= \frac{1}{d_k}\int_{-\infty}^{\infty}\left(\sum_{\ell=1}^{L}\sum_{m=1}^{M} h_m s_m[\ell] p(\zeta - \tau_m - \ell T) + z(\zeta)\right) p'_k(t-\zeta) d\zeta, \end{aligned}$$

and we sample $y_k(t)$ at $(i-1)T + \tau_{k+1} : i = 1, 2, ..., L, L+1$. The samples we get are

$$\begin{aligned} y_k[i] \;=\; y_k(t = (i-1)T + \tau_{k+1}) &= \frac{1}{d_k}\int_{(i-1)T+\tau_k}^{(i-1)T+\tau_{k+1}} \\ &\sum_{m=1}^{M} h_m s_m[i - \mathbb{1}_{m>k}]\, d\zeta + \frac{1}{d_k}\int_{(i-1)T+\tau_k}^{(i-1)T+\tau_{k+1}} z(\zeta)\, d\zeta \\ &= \sum_{m=1}^{M} h_m s_m[i - \mathbb{1}_{m>k}] + z_k[i], \end{aligned} \tag{25}$$

where we have defined $s_m[0] = s_m[L+1] = 0$, $\forall m$, for completeness.

Eq. (25) is very informative, let us take a closer look.

1) Each sample in (25) is related to M complex symbols, each of which comes from a different device.
2) It can be verified that $\mathbb{E}\left[z_k[i] z_{k'}[i']\right] = \frac{N_0 T}{d_k}\delta\big((i-i')(k-k')\big)$. This means that the noise sequence $\boldsymbol{z_k}$ is white: $z_k[i] \sim \mathcal{CN}(0, N_0 T/d_k)$ and is independent for different k and i.
3) The symbols are aligned in time within the integral interval of the M-th matched filter. Specifically, let $k = M$, we have

$$y_M[i] = \sum_{m=1}^{M} h_m s_m[i] + z_M[i], \tag{26}$$

where $z_M[i] \sim \mathcal{CN}(0, N_0 T/d_M)$. Unlike the outputs of other matched filters, the sampling outputs of the M-th matched filter form a synchronous OAC with the noise variance being amplified by T/d_M times.

The third observation suggests that the ML and LMMSE estimators designed for the synchronous OAC can also be used in the asynchronous case, utilizing the outputs of the M-th matched filter only (for the purpose of differentiation, we add a prefix "p-" before the ML and LMMSE estimators since only partial samples are used here).

Corollary 6 (MSEs of the p-ML and p-LMMSE estimators in asynchronous OAC)**.** *In asynchronous OAC, given the output of the M-th matched filter in* (26)*, the MSEs of the ML and LMMSE estimators are*

$$MSE_{p\text{-}ML} \;=\; (\boldsymbol{h} - \mathbf{1})^H \boldsymbol{V}(\boldsymbol{h} - \mathbf{1}) + \frac{N_0 T}{d_M}, \tag{27}$$

$$MSE_{p\text{-}LMMSE} \;=\; \mathbf{1}^\top \boldsymbol{D}\mathbf{1} - \frac{\left|\boldsymbol{h}^H \boldsymbol{D}\mathbf{1}\right|^2}{\boldsymbol{h}^H \boldsymbol{D}\boldsymbol{h} + \frac{N_0 T}{d_M}}. \tag{28}$$

Proof. Corollary 6 follows from Proposition 1 and Theorem 4 by substituting d_M/T for T since only the sampling outputs of the M-th matched filter are used. ■

From (27) and (28), it is clear that the MSEs of the p-ML and p-LMMSE estimators hinge on the maximum time offset τ_M as it determines the duration of the M-th matched filter $d_M = T - \tau_M$. Take the p-LMMSE estimator for example. In the synchronous OAC, we have $\tau_M = 0$, and hence, $d_M = T$. In the asynchronous OAC, on the other hand, $\text{MSE}_{\text{p-LMMSE}}$ increases with the increase of τ_M. To the extent that $\tau_M \to T$ (hence $d_M \to 0$), $\text{MSE}_{\text{A-LMMSE}} \to \mathbf{1}^\top \boldsymbol{D} \mathbf{1}$. In the next section, we shall devise more powerful estimators that makes use of the samples from all matched filters $\boldsymbol{y} = \{\boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M}\}$.

B. ML estimation and LMMSE estimation

To start with, let us rewrite all the samples $\boldsymbol{y} = \{\boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M}\}$ given by (25) into a more compact form as

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{s} + \boldsymbol{z}, \tag{29}$$

where $\boldsymbol{y}$, $\boldsymbol{s}$, and $\boldsymbol{z}$ are defined as

$$\begin{aligned}
\boldsymbol{y} &= \Big[y_1[1], y_2[1], ..., y_M[1], y_1[2], y_2[2], ..., y_M[2], ..., \\
&\quad y_1[L], y_2[L], ..., y_M[L], y_1[L+1], y_2[L+1], ..., y_{M-1}[L+1]\Big]^\top, \\
\boldsymbol{s} &= \Big[s_1[1], s_2[1], ..., s_M[1], s_1[2], s_2[2], ..., s_M[2], ..., \\
&\quad s_1[L], s_2[L], ..., s_M[L]\Big]^\top, \\
\boldsymbol{z} &= \Big[z_1[1], z_2[1], ..., z_M[1], z_1[2], z_2[2], ..., z_M[2], ..., \\
&\quad z_1[L], z_2[L], ..., z_M[L], z_1[L+1], z_2[L+1], ..., z_{M-1}[L+1]\Big]^\top,
\end{aligned}$$

and the $M(L+1)-1$ by ML coefficient matrix $\boldsymbol{G}$ is

$$\boldsymbol{G} = \begin{bmatrix}
h_1 & & & & & & & & \\
h_1 & h_2 & & & & & & & \\
... & h_2 & ... & & & & & & \\
h_1 & ... & ... & h_M & & & & & \\
 & h_2 & ... & h_M & h_1 & & & & \\
 & & ... & ... & h_1 & h_2 & & & \\
 & & & h_M & ... & h_2 & ... & & \\
 & & & & h_1 & ... & ... & h_M & \\
 & & & & & h_2 & ... & h_M & ... \\
 & & & & & & ... & ... & ... \\
 & & & & & & & h_M & ... \\
 & & & & & & & & ...
\end{bmatrix}.$$

Eq. (29) is in the form of a classic ISI-channel model in digital communications with two main difference:

1) The sequence of transmitted symbols $\boldsymbol{s}$ are continuous complex values instead of discrete constellations. In digital communications, the discrete constellation is a kind of prior information to the receiver whereby the detection space is naturally narrowed down to the possible constellation points only. In OAC, however, we do not have such prior information due to the continuous nature of the transmitted signal. The estimation space is thus infinitely large.
2) Our aim is not to estimate the transmitted symbols $\boldsymbol{s}$, but a linear transformation of $\boldsymbol{s}$, giving

$$\boldsymbol{s}_+ = \boldsymbol{F}\boldsymbol{s}, \tag{30}$$

where L by ML dimensional matrix $\boldsymbol{F}$ is given by

$$\boldsymbol{F} = \begin{bmatrix}
\mathbf{1}_{1\times M} & & & \\
 & \mathbf{1}_{1\times M} & & \\
 & & ... & \\
 & & & \mathbf{1}_{1\times M}
\end{bmatrix},$$

in which $\mathbf{1}_{1\times M}$ represents a $1 \times M$ all-ones matrix.

To estimate $\boldsymbol{s}_+$ in (30), a viable estimator that utilizes all the samples $\boldsymbol{y}$ is the ML estimator.

Definition 3 (ML estimation for the misaligned OAC)**.** *Given a sequence of samples $\boldsymbol{y} \in \mathcal{C}^{M(L+1)}$ in* (29)*, the ML estimate of sequence $\boldsymbol{s}_+ \in \mathcal{C}^L$ is*

$$\hat{\boldsymbol{s}}_+^{ml} = \boldsymbol{F}(\boldsymbol{G}^{\boldsymbol{H}}\boldsymbol{\Sigma}_{\boldsymbol{z}}^{-\mathbf{1}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\boldsymbol{H}}\boldsymbol{\Sigma}_{\boldsymbol{z}}^{-\mathbf{1}}\boldsymbol{y}, \tag{31}$$

where $\boldsymbol{\Sigma_z}$ denotes the covariance matrix of the noise sequence $\boldsymbol{z}$. In particular, $\boldsymbol{\Sigma_z}$ is a diagonal matrix since $\boldsymbol{z}$ is white.

The MSE of the ML estimator can be derived as

$$\begin{aligned}
\text{MSE} &= \frac{1}{L}\mathbb{E}\big[(\hat{\boldsymbol{s}}_+^{\text{ml}} - \boldsymbol{s}_+)^H(\hat{\boldsymbol{s}}_+^{\text{ml}} - \boldsymbol{s}_+)\big] = \frac{1}{L}\text{Tr}\big\{\boldsymbol{F}(\boldsymbol{G}^{\boldsymbol{H}}\boldsymbol{\Sigma}_{\boldsymbol{z}}^{-\mathbf{1}}\boldsymbol{G})^{-1} \\
&\qquad \boldsymbol{G}^{\boldsymbol{H}}\boldsymbol{\Sigma}_{\boldsymbol{z}}^{-\mathbf{1}}\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^H]\boldsymbol{\Sigma}_{\boldsymbol{z}}^{-H}\boldsymbol{G}(\boldsymbol{G}^H\boldsymbol{\Sigma}_{\boldsymbol{z}}^{-\mathbf{1}}\boldsymbol{G})^{-H}\boldsymbol{F}^H\big\} \\
&= \frac{1}{L}\text{Tr}\big\{\boldsymbol{F}(\boldsymbol{G}^H\boldsymbol{\Sigma}_{\boldsymbol{z}}^{-\mathbf{1}}\boldsymbol{G})^{-H}\boldsymbol{F}^H\big\} \\
&= \frac{1}{L}\text{Tr}\big\{\boldsymbol{F}(\boldsymbol{G}^H\boldsymbol{\Sigma}_{\boldsymbol{z}}^{-\mathbf{1}}\boldsymbol{G})^{-1}\boldsymbol{F}^\top\big\}.
\end{aligned}$$

Our companion paper [8] reveals that, in the misaligned OAC, ML estimation is very much susceptible to noise due to the infinitely large estimation space. As a result, error propagation and noise enhancement are severe with the ML estimator.

Unlike ML estimation which treats the transmitted sequence $\boldsymbol{s}$ as a constant sequence, this paper treats $\boldsymbol{s}$ as a random sequence and puts forth a Bayesian approach to address the problems faced by ML estimation. We will show that the estimation performance can be significantly improved by making good use of two pieces of prior information (i.e., the first and second sample moments) transmitted from each edge device.

Definition 4 (LMMSE estimation for the misaligned OAC)**.** *Given a sequence of samples $\boldsymbol{y} \in \mathcal{C}^{M(L+1)}$ in* (29)*, an LMMSE estimator estimates the sequence $\boldsymbol{s}_+ \in \mathcal{C}^L$ by*

$$\hat{\boldsymbol{s}}_+^{LMMSE} = \boldsymbol{F}\tilde{\boldsymbol{D}}\boldsymbol{G}^H(\boldsymbol{G}\tilde{\boldsymbol{D}}\boldsymbol{G}^H + \boldsymbol{\Sigma_z})^{-1}(\boldsymbol{y} - \boldsymbol{G}\tilde{\boldsymbol{\mu}}) + \boldsymbol{F}\tilde{\boldsymbol{\mu}}, \tag{32}$$

where the vector $\tilde{\boldsymbol{\mu}} = \mathbb{E}[\boldsymbol{s}]$ and matrix $\tilde{\boldsymbol{D}} = \mathbb{E}[\boldsymbol{s}\boldsymbol{s}^H] - \mathbb{E}[\boldsymbol{s}]\mathbb{E}^H[\boldsymbol{s}]$ can be constructed by the first and second sample moments transmitted from the devices, giving

$$\begin{aligned}
\tilde{\boldsymbol{\mu}} &= \Big[\hat{\mathbb{E}}_1, \hat{\mathbb{E}}_2, ..., \hat{\mathbb{E}}_M, \hat{\mathbb{E}}_1, \hat{\mathbb{E}}_2, ..., \hat{\mathbb{E}}_M, ..., \hat{\mathbb{E}}_1, \hat{\mathbb{E}}_2, ..., \hat{\mathbb{E}}_M\Big]^\top, \\
\tilde{\boldsymbol{D}} &= diag\Big(\hat{\mathbb{D}}_1, ..., \hat{\mathbb{D}}_M, \hat{\mathbb{D}}_1, ..., \hat{\mathbb{D}}_M, ..., \hat{\mathbb{D}}_1, ..., \hat{\mathbb{D}}_M\Big).
\end{aligned}$$

Proposition 7 (MSE of the LMMSE estimator)**.** *The estimator in* (32) *is an LMMSE estimator for the misaligned OAC. The MSE of the estimator is*

$$MSE_{LMMSE} = \frac{1}{L}Tr\Big[(\boldsymbol{A}\boldsymbol{G} - \boldsymbol{F})\tilde{\boldsymbol{D}}(\boldsymbol{A}\boldsymbol{G} - \boldsymbol{F})^H + \boldsymbol{A}\boldsymbol{\Sigma_z}\boldsymbol{A}^H\Big], \tag{33}$$

where the matrix $\boldsymbol{A} = \boldsymbol{F}\tilde{\boldsymbol{D}}\boldsymbol{G}^H(\boldsymbol{G}\tilde{\boldsymbol{D}}\boldsymbol{G}^H + \boldsymbol{\Sigma_z})^{-1}$.

Proof. We first prove that the estimator in (32) is an LMMSE estimator. Given the signal model in (29), a linear estimator estimates $\boldsymbol{s}_+$ by $\hat{\boldsymbol{s}}_+ = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{c}$. The MSE of the linear estimate $\hat{\boldsymbol{s}}_+$ is then given by

$$\text{MSE} = \frac{1}{L}\mathbb{E}\left[(\boldsymbol{A}\boldsymbol{y} + \boldsymbol{c} - \boldsymbol{s}_+)^H(\boldsymbol{A}\boldsymbol{y} + \boldsymbol{c} - \boldsymbol{s}_+)\right]. \tag{34}$$

The matrix $\boldsymbol{A}$ and vector $\boldsymbol{c}$ that yield the minimum MSE can then be obtained by setting $\partial\text{MSE}/\partial\boldsymbol{A}=0$ and $\partial\text{MSE}/\partial\boldsymbol{c}=0$. Thus, we have

$$\begin{aligned}
&\frac{\partial\text{MSE}}{\partial\boldsymbol{A}}=\mathbb{E}\frac{\partial\text{Tr}[(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{c}-\boldsymbol{s}_+)(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{c}-\boldsymbol{s}_+)^H]}{\partial\boldsymbol{A}}\\
&=\boldsymbol{A}\mathbb{E}[\boldsymbol{y}\boldsymbol{y}^{\boldsymbol{H}}]+\boldsymbol{c}\mathbb{E}^H[\boldsymbol{y}]-\mathbb{E}[\boldsymbol{s}_+\boldsymbol{y}^{\boldsymbol{H}}]\\
&=\boldsymbol{A}\boldsymbol{G}\mathbb{E}[\boldsymbol{s}\boldsymbol{s}^{\boldsymbol{H}}]\boldsymbol{G}^{\boldsymbol{H}}+\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{z}}+\boldsymbol{c}\mathbb{E}^{\boldsymbol{H}}[\boldsymbol{s}]\boldsymbol{G}^{\boldsymbol{H}}-\boldsymbol{F}\mathbb{E}[\boldsymbol{s}\boldsymbol{s}^{\boldsymbol{H}}]\boldsymbol{G}^{\boldsymbol{H}}=0,\\
&\frac{\partial\text{MSE}}{\partial\boldsymbol{c}}=\boldsymbol{A}\mathbb{E}[\boldsymbol{y}]+\boldsymbol{c}-\mathbb{E}[\boldsymbol{s}_+]=\boldsymbol{A}\boldsymbol{G}\mathbb{E}[\boldsymbol{s}]+\boldsymbol{c}-\boldsymbol{F}\mathbb{E}[\boldsymbol{s}]=0.
\end{aligned}$$

Let $\tilde{\boldsymbol{\mu}}=\mathbb{E}[\boldsymbol{s}]$ and $\tilde{\boldsymbol{D}}=\mathbb{E}[\boldsymbol{s}\boldsymbol{s}^{\boldsymbol{H}}]-\mathbb{E}[\boldsymbol{s}]\mathbb{E}^H[\boldsymbol{s}]$, we have

$$\begin{aligned}
\boldsymbol{A}&=\boldsymbol{F}\tilde{\boldsymbol{D}}\boldsymbol{G}^{\boldsymbol{H}}(\boldsymbol{G}\tilde{\boldsymbol{D}}\boldsymbol{G}^{\boldsymbol{H}}+\boldsymbol{\Sigma}_{\boldsymbol{z}})^{-1},\\
\boldsymbol{c}&=\boldsymbol{F}\tilde{\boldsymbol{\mu}}-(\boldsymbol{F}\tilde{\boldsymbol{D}}\boldsymbol{G}^{\boldsymbol{H}})(\boldsymbol{G}\tilde{\boldsymbol{D}}\boldsymbol{G}^{\boldsymbol{H}}+\boldsymbol{\Sigma}_{\boldsymbol{z}})^{-1}\boldsymbol{G}\tilde{\boldsymbol{\mu}}.
\end{aligned}$$

This gives us the LMMSE estimator in (32). The MSE of the LMMSE estimator can be obtained by substituting $\boldsymbol{A}$ and $\boldsymbol{c}$ into (34), giving

$$\begin{aligned}
&\text{MSE}=\frac{1}{L}\mathbb{E}\left[(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{c}-\boldsymbol{s}_+)^H(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{c}-\boldsymbol{s}_+)\right]\\
&=\frac{1}{L}\mathbb{E}\left\{[(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})\boldsymbol{s}+\boldsymbol{A}\boldsymbol{z}+\boldsymbol{c}]^H[(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})\boldsymbol{s}+\boldsymbol{A}\boldsymbol{z}+\boldsymbol{c}]\right\}\\
&=\frac{1}{L}\mathbb{E}\big[\boldsymbol{s}^H(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})^H(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})\boldsymbol{s}+\boldsymbol{c}^H(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})\boldsymbol{s}+\\
&\quad\boldsymbol{z}^H\boldsymbol{A}^H\boldsymbol{A}\boldsymbol{z}+\boldsymbol{s}^H(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})^H\boldsymbol{c}+\boldsymbol{c}^H\boldsymbol{c}\big]\\
&\overset{(a)}{=}\frac{1}{L}\mathbb{E}\big[\boldsymbol{s}^H(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})^H(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})\boldsymbol{s}+\boldsymbol{z}^H\boldsymbol{A}^H\boldsymbol{A}\boldsymbol{z}\\
&\quad-\frac{1}{L}\tilde{\boldsymbol{\mu}}^H(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})^H(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})\tilde{\boldsymbol{\mu}}\big]\\
&=\frac{1}{L}\text{Tr}\left\{(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})(\mathbb{E}[\boldsymbol{s}\boldsymbol{s}^H]-\tilde{\boldsymbol{\mu}}\tilde{\boldsymbol{\mu}}^H)(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})^H+\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{z}}\boldsymbol{A}^H\right\}\\
&=\frac{1}{L}\text{Tr}\left[(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})\tilde{\boldsymbol{D}}(\boldsymbol{A}\boldsymbol{G}-\boldsymbol{F})^H+\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{z}}\boldsymbol{A}^H\right].
\end{aligned}$$

where (a) follows by substituting $\boldsymbol{c}=(\boldsymbol{F}-\boldsymbol{A}\boldsymbol{G})\tilde{\boldsymbol{\mu}}$. ■

The performances of the ML and LMMSE estimators are compared numerically in Section V. As will be shown, the LMMSE estimator outperforms the ML estimator significantly in terms of MSE. Intuitively, the LMMSE estimator leverages the sample mean and sample variance transmitted from each edge device as a kind of prior information, thereby reducing the estimation space by much.

Yet, another problem with the ML and LMMSE estimators is the high computational complexity since the dimensionalities of the matrices in (31) and (32) grow linearly with the number of devices M and the packet length L. As a result, (31) and (32) are computationally expensive for large M and L. Take the LMMSE estimator for example. Let us measure the computational complexity of (32) by the matrix inversion $(\boldsymbol{G}\tilde{\boldsymbol{D}}\boldsymbol{G}^H+\boldsymbol{\Sigma}_{\boldsymbol{z}})^{-1}$, which is the most computationally demanding part of (32). To invert an n by n matrix, the best proven lower bound of the computational complexity is $\Omega(n^2\log n)$ [23]. Thus, the complexity of (32) is $\Omega(M^2L^2\log(ML))$. In OAC systems, the packet length L is often much larger than the number of devices M. Let us fix M as a constant, the computational complexity of (32) is then $\Omega(L^2\log L)$.

Our companion paper [8] reduced the computational complexity of the ML estimator by exploiting the sparsity of the coefficient matrix and putting forth a sum-product ML (SP-ML) estimator. The same idea applies here. In the next subsection, we shall devise a sum-product maximum a posterior (SP-MAP) estimator. The MSE performance of the SP-MAP estimator is on an equal footing with the LMMSE estimator, but its computational complexity is only $\Omega(L)$.

C. MAP Estimation

To tackle the high-complexity of the LMMSE estimation and devise a practical estimator for the misaligned OAC, this subsection resorts to the MAP estimation and puts forth an SP-MAP estimator for the misaligned OAC.

Definition 5 (MAP estimation for the misaligned OAC)**.** *Given the white samples $\{\boldsymbol{y_1},\boldsymbol{y_2},...,\boldsymbol{y_M}\}$ in* (25)*, an MAP estimator estimates the target sequence $\boldsymbol{s}_+\in\mathcal{C}^L$ symbol-by-symbol by*

$$\hat{s}_+[i]=\arg\max_{s_+[i]}\Pr\left(s_+[i]=\sum_{m=1}^{M}s_m[i]|\boldsymbol{y_1},\boldsymbol{y_2},...,\boldsymbol{y_M}\right), \quad (35)$$

where

$$\begin{aligned}
&\Pr\left(s_+[i]=\sum_{m=1}^{M}s_m[i]|\boldsymbol{y_1},\boldsymbol{y_2},...,\boldsymbol{y_M}\right)=\int_{\sum_{m=1}^{M}s_m[i]=s_+[i]}\\
&f(s_1[i],s_2[i],...,s_M[i]|\boldsymbol{y_1},\boldsymbol{y_2},...,\boldsymbol{y_M})d(s_1[i],s_2[i],...,s_M[i]).
\end{aligned}$$

As can be seen, to obtain the MAP estimate, a first step is to derive the joint posterior probability distribution $f(s_1[i],s_2[i],...,s_M[i]|\boldsymbol{y_1},\boldsymbol{y_2},...,\boldsymbol{y_M})$. To this intent, we shall start from the joint posterior distribution of all transmitted symbols conditioned on the samples observed at the receiver, i.e., $f(\boldsymbol{s_1},\boldsymbol{s_2},...,\boldsymbol{s_M}|\boldsymbol{y_1},\boldsymbol{y_2},...,\boldsymbol{y_M})$. To ease exposition, we name $f(s_1[i],s_2[i],...,s_M[i]|\boldsymbol{y_1},\boldsymbol{y_2},...,\boldsymbol{y_M})$ the marginal posterior distribution and $f(\boldsymbol{s_1},\boldsymbol{s_2},...,\boldsymbol{s_M}|\boldsymbol{y_1},\boldsymbol{y_2},...,\boldsymbol{y_M})$ the global posterior distribution.

The global posterior distribution can be factorized in the following way:

$$\begin{aligned}
&f(\boldsymbol{s_1},\boldsymbol{s_2},...,\boldsymbol{s_M}|\boldsymbol{y_1},\boldsymbol{y_2},...,\boldsymbol{y_M})\\
&\propto f(\boldsymbol{y_1},\boldsymbol{y_2},...,\boldsymbol{y_M}|\boldsymbol{s_1},\boldsymbol{s_2},...,\boldsymbol{s_M})f(\boldsymbol{s_1},\boldsymbol{s_2},...,\boldsymbol{s_M})\\
&\overset{(a)}{\propto}\prod_{k=1}^{M}\prod_{i=1}^{L+1}f(y_k[i]|\boldsymbol{s_1},\boldsymbol{s_2},...,\boldsymbol{s_M})\prod_{m=1}^{M}f(\boldsymbol{s_m})\\
&\overset{(b)}{\propto}\prod_{k=1}^{M}\prod_{i=1}^{L+1}f(y_k[i]|\mathcal{V}(y_k[i]))\prod_{m=1}^{M}\prod_{i=1}^{L}f(s_m[i]),
\end{aligned}\quad (36)$$

where $\propto$ stands for "proportional to". Step (a) follows because i) given the transmitted sequence $\boldsymbol{s_1}$, $\boldsymbol{s_2}$, ..., $\boldsymbol{s_M}$, all the samples $\boldsymbol{y_1}$, $\boldsymbol{y_2}$, ..., $\boldsymbol{y_M}$ are independent since the noise sequence is white; ii) the transmitted symbols $\boldsymbol{s_m}$ from different devices are independent. Step (b) follows since

i) Each sample $y_k[i]$ is only related to a set of complex symbols $\mathcal{V}(y_k[i])=\{s_1[i],s_2[i],...,s_k[i],s_{k+1}[i-1],s_{k+2}[i-1],...,s_M[i-1]\}$. We call them the *neighbor symbols* of the sample $y_k[i]$. In particular, the number of non-zero symbols in $\mathcal{V}(y_k[i])$ is

$$|\mathcal{V}(y_k[i])|=\begin{cases}k, & \text{when } i=1;\\ M, & \text{when } 1\le i\le L;\\ M-k, & \text{when } i=L+1.\end{cases} \quad (37)$$

ii) $f(\boldsymbol{s_m})$ is the prior distribution of the transmitted symbols from the m-th device. To construct this information at the receiver, we assume the symbols of each edge device are generated from a Gaussian distribution in an i.i.d. manner. In particular, the Gaussian distribution is parameterized by the first and second sample moments transmitted from each edge device. This is a plausible assumption since there is no randomness once a packet is generated at the transmitter and the receiver can assume they are sampled from an i.i.d. Gaussian with mean and variance being its sample mean and sample variance. We emphasize that the devices have to generate the first and second sample moments (i.e., the prior information to the receiver) for each new packet and transmit it to the receiver. The receiver then estimates different packets using different prior information.

The factorizations in (36) can be depicted by a graphical model [4], [8], [24]–[26], as shown in Fig. 4, where we use a Forney-style factor graph [25] to represent the factorization. Specifically, each edge in the graph corresponds to a variable in (25), e.g., an observation $y_k[i]$ or a noise term $z_k[i]$. The variable $\boldsymbol{W}_{k,i}$ is a high-dimensional variable consisting of all complex symbols in $\mathcal{V}(y_k[i])$, i.e., $\boldsymbol{W}_{k,i} = \mathcal{V}(y_k[i]) = \{s_1[i], s_2[i], ..., s_k[i], s_{k+1}[i-1], s_{k+2}[i-1], ..., s_M[i-1]\}$.

The equality function/constraint "=" in Fig. 4 means that the variables connecting to this function are exactly the same (but may have different posterior distributions). The compatibility function δ, on the other hand, represent the constraint that the values of the common symbols contained in the adjacent variables must be equal. For example, $\boldsymbol{W}_{1,1} = \{s_1[1]\}$, $\boldsymbol{W}_{2,1} = \{s_1[1], s_2[1]\}$, and the common symbol between $\boldsymbol{W}_{1,1}$ and $\boldsymbol{W}_{2,1}$ is $s_1[1]$. Therefore, we have to add a constraint $\delta(\boldsymbol{W}_{1,1}, \boldsymbol{W}_{2,1})$ between $\boldsymbol{W}_{1,1}$ and $\boldsymbol{W}_{2,1}$ to ensure that the values of $s_1[1]$ in $\boldsymbol{W}_{1,1}$ and $\boldsymbol{W}_{2,1}$ are the same. In general, for any two adjacent variables $\boldsymbol{W}$ and $\boldsymbol{W}'$ connecting to the same delta function $\delta(\boldsymbol{W}, \boldsymbol{W}')$, we have

$$\delta(\boldsymbol{W}, \boldsymbol{W}') = \begin{cases} 1, \text{ if the values of all common symbols} \\ \quad \text{between } \boldsymbol{W} \text{ and } \boldsymbol{W}' \text{ are equal;} \\ 0, \text{ otherwise.} \end{cases}$$

Succinctly speaking, function δ is an on-off function ensuring that the messages passed from $\boldsymbol{W}$ to $\boldsymbol{W}'$ and that passed from $\boldsymbol{W}'$ to $\boldsymbol{W}$ satisfy the constraint that the values of the common symbols between $\boldsymbol{W}$ and $\boldsymbol{W}'$ are equal.

Finally, the prior information $f_{\boldsymbol{s_1}, \boldsymbol{s_2}, ..., \boldsymbol{s_M}}$ is an M-dimensional Gaussian, the mean vector and covariance matrix of which are given in (15) and (16), respectively. To avoid unnecessary loops, the prior information is added every M samples, as shown in Fig. 4.

D. The SP-MAP Estimator

The marginal posterior distribution $f(s_1[i], s_2[i], ..., s_M[i] | \boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M})$ is a marginal function of the global posterior distribution $f(\boldsymbol{s_1}, \boldsymbol{s_2}, ..., \boldsymbol{s_M} | \boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M})$. Therefore, it can be derived by a marginalization process operated on Fig. 4, which can be implemented efficiently via the sum-product algorithm.

A caveat here is that the variables on the graph are continuous random variables (since $\boldsymbol{s_m}$ are continuous complex values). Therefore, we have to perform analog message passing by parameterizing the probability density functions (PDF) of the continuous variables and passing the parameters instead of the PDF along the edges. The detailed processes of analog message passing on graphs can be found in Appendix B of our companion paper [8]. Below, we summarize the main results of analog message passing on Fig. 4, based on which an SP-MAP estimator is devised.

Theorem 8 (Conditional Gaussian of the posterior distributions)**.** *Consider the MAP estimator defined in Definition 5. Let $\boldsymbol{s[i]} = (s_1[i], s_2[i], ..., s_M[i])$, we have the following results:*

1) The marginal posterior distribution $f(\boldsymbol{s[i]} | \boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M})$ is an M-dimensional complex Gaussian distribution, giving

$$f(\boldsymbol{s[i]} | \boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M}) \sim \mathcal{N}\left(\boldsymbol{s[i]}, \boldsymbol{\mu_{s[i]}} = \begin{bmatrix} \boldsymbol{\mu}^{\mathtt{r}}_{\boldsymbol{s[i]}} \\ \boldsymbol{\mu}^{\mathtt{i}}_{\boldsymbol{s[i]}} \end{bmatrix}, \boldsymbol{\Sigma_{s[i]}} = \begin{bmatrix} \boldsymbol{\Sigma}^{\mathtt{rr}}_{\boldsymbol{s[i]}} & \boldsymbol{\Sigma}^{\mathtt{ri}}_{\boldsymbol{s[i]}} \\ \boldsymbol{\Sigma}^{\mathtt{ir}}_{\boldsymbol{s[i]}} & \boldsymbol{\Sigma}^{\mathtt{ii}}_{\boldsymbol{s[i]}} \end{bmatrix}\right), \tag{38}$$

where $\boldsymbol{\mu_{s[i]}}$ is a $2M$ by 1 real vector consists of the real and imaginary parts of the mean of $\boldsymbol{s[i]}$, that is, $\boldsymbol{\mu}^{\mathtt{r}}_{\boldsymbol{s[i]}}$ and $\boldsymbol{\mu}^{\mathtt{i}}_{\boldsymbol{s[i]}}$ are the real and imaginary parts of sequence $\mathbb{E}[\boldsymbol{s[i]}]^\top$; the matrix $\boldsymbol{\Sigma_{s[i]}}$ is a $2M$ by $2M$ covariance matrix. The moment parameters $(\boldsymbol{\mu_{s[i]}}, \boldsymbol{\Sigma_{s[i]}})$ can be computed by analog message passing.

2) The posterior distribution $f(s_+[i] = \sum_{m=1}^{M} s_m[i] | \boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M})$ is a complex Gaussian distribution, giving

$$f(s_+[i] | \boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M}) \sim \mathcal{N}\left(s_+[i], \boldsymbol{\mu}_{s_+[i]} = \begin{bmatrix} \mu^{\mathtt{r}}_{s_+[i]} \\ \mu^{\mathtt{i}}_{s_+[i]} \end{bmatrix}, \boldsymbol{\Sigma}_{s_+[i]} = \begin{bmatrix} \Sigma^{\mathtt{rr}}_{s_+[i]} & \Sigma^{\mathtt{ri}}_{s_+[i]} \\ \Sigma^{\mathtt{ir}}_{s_+[i]} & \Sigma^{\mathtt{ii}}_{s_+[i]} \end{bmatrix}\right), \tag{39}$$

where $\mu^{\mathtt{r}}_{s_+[i]} = \mathbf{1}^\top \boldsymbol{\mu}^{\mathtt{r}}_{\boldsymbol{s[i]}}$, $\mu^{\mathtt{i}}_{s_+[i]} = \mathbf{1}^\top \boldsymbol{\mu}^{\mathtt{i}}_{\boldsymbol{s[i]}}$, $\Sigma^{\mathtt{rr}}_{s_+[i]} = \mathbf{1}^\top \boldsymbol{\Sigma}^{\mathtt{rr}}_{\boldsymbol{s[i]}} \mathbf{1}$, $\Sigma^{\mathtt{ri}}_{s_+[i]} = \mathbf{1}^\top \boldsymbol{\Sigma}^{\mathtt{ri}}_{\boldsymbol{s[i]}} \mathbf{1}$, $\Sigma^{\mathtt{ir}}_{s_+[i]} = \mathbf{1}^\top \boldsymbol{\Sigma}^{\mathtt{ir}}_{\boldsymbol{s[i]}} \mathbf{1}$, $\Sigma^{\mathtt{ii}}_{s_+[i]} = \mathbf{1}^\top \boldsymbol{\Sigma}^{\mathtt{ii}}_{\boldsymbol{s[i]}} \mathbf{1}$.

Proof. See Appendix B. ∎

Based on Theorem 8, we refine the MAP estimator defined in Definition 5 as follows.

Definition 6 (SP-MAP estimation)**.** *Given the output of the M matched filters $\{\boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M}\}$ in (25), an SP-MAP estimator first computes the moment parameters of the Gaussian distribution $f(\boldsymbol{s[i]} | \boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M})$ in (38) by a sum-product process. From the mean vector $\boldsymbol{\mu_{s[i]}} = [\boldsymbol{\mu}^{\mathtt{r}}_{\boldsymbol{s[i]}}, \boldsymbol{\mu}^{\mathtt{i}}_{\boldsymbol{s[i]}}]^\top$, the SP-MAP estimator estimates $s_+[i]$ by*

$$\hat{s}_+[i] = \mathbf{1}^\top \boldsymbol{\mu}^{\mathtt{r}}_{\boldsymbol{s[i]}} + j \mathbf{1}^\top \boldsymbol{\mu}^{\mathtt{i}}_{\boldsymbol{s[i]}}. \tag{40}$$

Eq. (40) can be understood in the following way: since $f(s_+[i] | \boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M})$ is Gaussian, its mean vector maximizes the posterior probability. As per the MAP rule in (35), an SP-MAP estimator chooses the mean of $f(s_+[i] | \boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M})$, i.e., (40), as the MAP estimate.

We now compare the complexity of the SP-MAP estimator with that of the LMMSE estimator. With Gaussian message

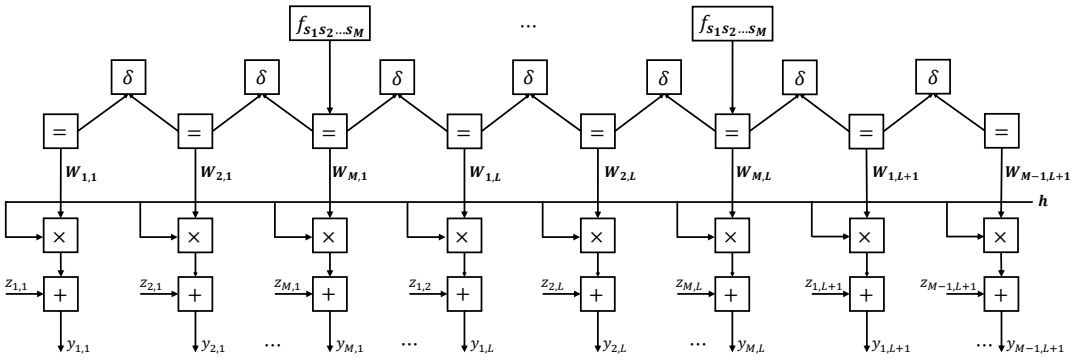

Figure 4. A graphical interpretation of the factorization in (36). To simplify notations, we denote $y_k[i]$ and $z_k[i]$ by $y_{k,i}$ and $z_{k,i}$, respectively. The high-dimensional variable $\boldsymbol{W}_{k,i} = \mathcal{V}(y_k[i]) = \{s_1[i], s_2[i], ..., s_k[i], s_{k+1}[i-1], s_{k+2}[i-1], ..., s_M[i-1]\}$.

passing, the messages passed on the graph are simply the parameters of the Gaussian distribution instead of the continuous Gaussian PDF. Thus, the computations involved in the analog message passing are only 1) the sum of 2M-dimensional vectors/matrices, and 2) 2M-dimensional matrix inversion. The computational complexity of the SP-MAP estimator is then $\Omega(LM^2 \log M)$. If we fix M as a constant, the computational complexity of the SP-MAP estimator is simply $\Omega(L)$. In contrast, the computational complexity of the LMMSE estimator is $\Omega(L^2 \log L)$.

V. Numerical and Simulation Results

This section evaluates the MSE performance of various Bayesian estimators devised in this paper benchmarked against the ML estimator. Specifically, we consider a MAC where $M = 4$ devices communicate with a fusion center via OAC. There can be channel-gain and time misalignments among the signals received by the AP. We shall evaluate the MSEs of different estimators under various degrees of channel-gain misalignment, time misalignment, and EsN0:

1) The residual channel gain of the m-th device is $h_m = |h_m| e^{j\phi_m}$. In the simulations, we set $|h_m| = 1$, $\forall m$ and focus on the impact of the phase misalignment caused by residual phase offsets ϕ_m. Specifically, we assume $\{\phi_m : m = 1, 2, ..., M\}$ are uniformly distributed in $(0, \phi)$ and ϕ is the maximum phase offset. That is, $\phi_m \sim U(0, \phi)$. It is worth noting that ϕ_m can be any distribution in general.
2) Without loss of generality, the symbol duration is set to $T = 1$. Recall that the p-ML and p-LMMSE estimators make use of only the outputs of the M-th matched filter. For these two estimators, the estimation performance hinges on the length of the M-th matched filter $d_M = 1 - \tau_M$. In view of this, we will set the time offsets τ_m, $\forall m$, in the following manner: first, we fix the time offset of the M-th device τ_M (and hence d_M); then, we generate the time offsets of other devices uniformly in $(0, \tau_M)$.
3) EsN0 is defined as the received energy per symbol to noise power spectral density ratio, giving

$$\text{EsN0} = \frac{1}{N_0} \frac{1}{L} \sum_{i=1}^{L} \left| \sum_{m=1}^{M} e^{j\phi_m} s_m[i] \right|^2. \tag{41}$$

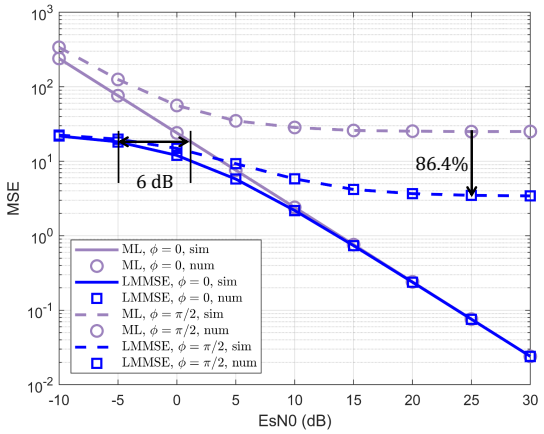

Figure 5. Numerical and simulation results of the ML and LMMSE estimators in synchronous OAC under various degrees of phase misalignments.

4) The transmitted symbols of the four devices $\boldsymbol{s_m}$ are generated uniformly in range $[-6, 0]$, $[-4, 2]$, $[-2, 4]$, $[0, 6]$, respectively.

A. Synchronous OAC

We first consider the synchronous OAC and compare the ML and LMMSE estimators under various degrees of phase misalignments and EsN0. The maximum phase offset ϕ is set to 0 (no phase misalignment), $\pi/2$ (mild),[2] π (moderate), or 2π (severe).

Fig. 5 presents the MSEs of the ML and LMMSE estimators versus EsN0 (in dB), wherein $\phi = 0$ and $\pi/2$. The numerical MSEs are generated by (9) and (19), respectively. In all our simulations, the simulation results match the numerical results very well. To ease presentation, we shall omit the numerical results and focus on the simulation results in the following.

Two main observations from Fig. 5 are as follows:

1) In the aligned OAC ($\phi = 0$), our LMMSE estimator outperforms the ML estimator by much in the low-EsN0

[2]Notice that ϕ is the maximum phase offset and the phase offsets of all devices are uniformly distributed in $[0, \phi]$. If we look at the phase misalignment between any two devices, the average pairwise-phase-misalignment is only $\phi/3$. That is why we classify $\pi/2$ as the mild phase misalignment because the average pairwise-phase-misalignment is only $\pi/6$.

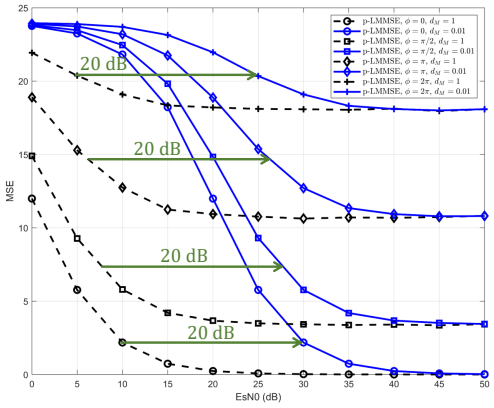

Figure 6. MSEs of the p-LMMSE estimator in asynchronous OAC under various degrees of time and phase misalignments.

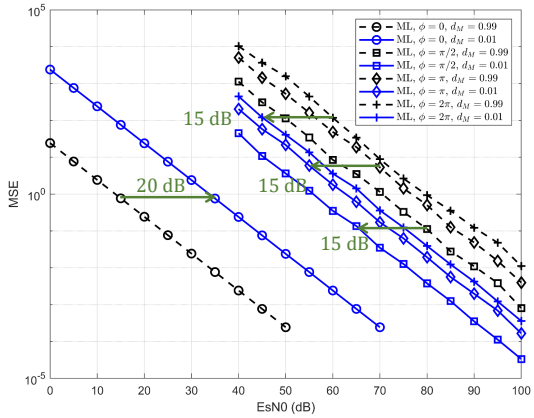

Figure 7. MSEs of the ML estimator in asynchronous OAC under various degrees of time and phase misalignments.

regime. At an EsN0 of -5 dB, the MSE gains are up to 6 dB. In the high-EsN0 regime, the two estimators are equally optimal. This is consistent with our analysis in Section III.

2) When there is phase misalignment, both the ML and LMMSE estimators suffer from error floor in the high-EsN0 regime. In particular, the error floor of the aligned-sample estimator is fairly pronounced even with mild phase offset $\phi = \pi/2$. The LMMSE estimator, on the other hand, lowers the error floor by 86.4%.

If we further increase ϕ to π and 2π, the same results can be observed and the LMMSE estimator consistently outperforms the ML estimator. Overall, we conclude that the prior information is very helpful in the synchronous OAC. With the prior information, the LMMSE estimator is strictly better than the conventional ML estimator which uses no prior information. In the low-EsN0 regime, the MSE gains are up to 6 dB even for the aligned OAC; in the high-EsN0 regime, the LMMSE estimator significantly lowers the error floor by 86.4% when there is mild phase misalignment.

B. Asynchronous OAC

Next, we evaluate the MSEs of the ML and Bayesian estimators designed for the asynchronous OAC. With no prior information, the viable estimators are the p-ML estimator (Corollary 6) and the ML estimator (Definition 3). With prior information, this paper devised a p-LMMSE estimator (Corollary 6), an LMMSE estimator (Definition 4), and an SP-MAP estimator (Definition 6).

1) The p-ML and the p-LMMSE estimators: The p-ML and p-LMMSE estimators utilize only the outputs of the M-th matched filter. Thus, compared with their performance in the synchronous OAC (i.e., Fig. 5), the introduction of time offset simply results in an EsN0 penalty. For example, if the maximum time offset $\tau_M = 0.9$ (hence $d_M = 1 - \tau_M = 0.1$), then we only need to shift the curve of the p-ML/p-LMMSE estimator in Fig. 5 by 10 dB to the right, where 10 dB is calculated from $10\log_{10}(1/d_M)$. An immediate result is that the p-LMMSE estimator is still strictly better than the ML estimator after the right shift.

Fig. 6 presents the MSE performance of the p-LMMSE estimator (the performance of the p-ML estimator is omitted). As predicted, the MSE performance deteriorates when there is either time or phase misalignment – time misalignment introduces a 20 dB EsN0 penalty ($d_M = 0.01$ corresponds to 20 dB) while phase misalignment results in both EsN0 penalty and error floor.

2) The ML estimator: Fig. 7 presents the MSEs of the ML estimator under different time and phase misalignments. As can be seen, when there is no phase misalignment, the ML estimator is susceptible to time offset, the MSE performance deteriorates for 20 dB when we decrease d_M from 0.99 to 0.01. When there is phase misalignment, on the other hand, an interesting observation is that the ML estimator benefits from the time misalignment: the MSEs are improved by 15 dB when $\phi = \pi/2$, π, and 2π.

ML uses the samples from all matched filters, but it utilizes no prior information. Compare Fig. 6 with Fig. 7, it can be seen that the ML estimator is even worse than the p-LMMSE estimator (which uses only partial samples) when there is phase misalignment. For example, to achieve an MSE of 10 when $\phi = \pi/2$ and $\tau_M = 0.01$, the p-LMMSE estimator requires an EsN0 of 24 dB while the ML estimator requires an EsN0 of 45 dB. On the other hand, the advantage of the ML estimator is that it exhibits no error floor in the high-EsN0 regime.

To summarize, a major problem of the ML estimator is that it is very sensitive to noise due to the infinite estimation space. As a result, it suffers from severe error propagation and noise enhancement when there is phase misalignment [8]. This is also validated in Fig. 7: there is a large EsN0 gap between the phase-aligned OAC and the phase-misaligned OAC.

Next, we evaluate the MSEs of our LMMSE and SP-MAP estimators designed for the asynchronous OAC.

3) The SP-MAP and LMMSE estimators: The SP-MAP and LMMSE estimators utilize all the samples, and also, the prior information transmitted from the edge devices. Their MSEs are presented in Fig. 8 and 9, respectively, under various degrees of time and phase misalignments. Considering the intensive computational complexity of the LMMSE estimator (see (32)), Fig. 9 is simulated using a much smaller packet length than

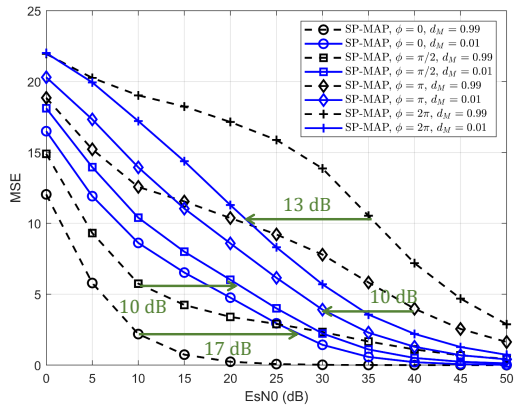

Figure 8. MSEs of the SP-MAP estimator in asynchronous OAC under various degrees of time and phase misalignments.

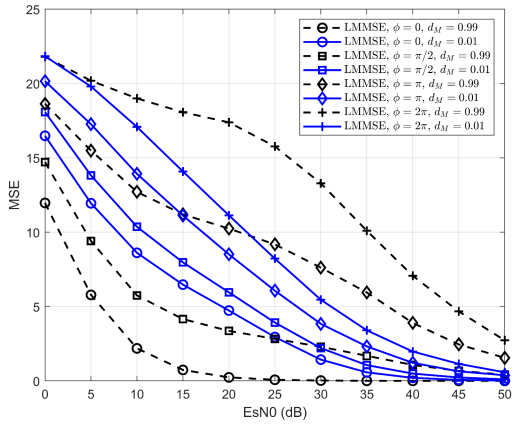

Figure 9. MSEs of the LMMSE estimator in asynchronous OAC under various degrees of time and phase misalignments.

that of Fig. 8. As shown, the estimation performance of the two estimators is on the same footing in terms of MSE. Thus, we shall focus on the SP-MAP estimator in the following.

We have two observations from Fig. 8:

1) When there is no phase misalignment ($\phi = 0$), the EsN0 penalty caused by asynchrony is 17 dB. Compared with Fig. 6 and Fig. 7 where the EsN0 penalty is 20 dB with the p-LMMSE and ML estimators, the SP-MAP estimator compensates the EsN0 penalty introduced by asynchrony by 3 dB.
2) When there is phase misalignments, the SP-MAP estimator also benefits from the time misalignment. Take the $\phi = \pi$ and 2π curves in Fig. 9 for example. When there is time misalignment, the MSEs of the SP-MAP estimator are improved by 10 dB and 13 dB, respectively.

In summary, 1) compared with the p-LMMSE estimator, the SP-MAP estimator utilizes all the sampling output of the matched filters and completely eliminates the error floors faced by the p-LMMSE estimator; 2) compared with the ML estimator, the SP-MAP estimator addresses the error propagation problem by taking advantage of the prior information transmitted from the devices. The MSEs are significantly reduced in all cases; 3) compared with the LMMSE estimator, the SP-MAP estimator attains the same level of MSE performance, but is much more computationally efficient.

VI. Conclusion

OAC is an efficient scheme to speed up the distributed data aggregation in MACs. The main spirit of OAC is joint computation-and-communication by exploiting the superposition property of the MAC that its output is an arithmetic sum of the input signals.

A main ingredient of the estimators devised in this paper is the "prior information" transmitted from the edge devices. In digital communications, the transmitted symbols are discrete constellations and the detection space is naturally narrowed down to all possible constellation points – the discrete constellation itself serves as a kind of prior information to the receiver. In OAC, however, the transmitted symbols are continuous complex values and the estimation space is infinitely large. In this context, additional prior information can be conducive to narrowing down the estimation space and achieving a good estimation performance.

In this paper, we let each device transmit two pieces of statistical information of the transmitted data (i.e., the first and the second sample moments) to the fusion center prior to the data transmission. These two pieces of prior information were shown to be very helpful in the estimation of the arithmetic-sum. In the aligned OAC and the synchronous OAC, the prior information admits an LMMSE estimator at the receiver. Compared with the ML estimator that uses no prior information, the LMMSE estimator reduces the MSE by more than 6 dB in the low-EsN0 regime and lowers the error floor by 86.4% in the high-EsN0 regime. In the asynchronous OAC, an LMMSE estimator and an SP-MAP estimator were devised to tackle both the channel-gain and time misalignments among signals thanks to the two pieces of statistical information. Compared with the ML estimator, the LMMSE and SP-MAP estimators address the error propagation and noise enhancement problems and significantly reduce the MSE under various degrees of phase and time misalignments.

Appendix A

This appendix proves that the estimator in (14) is an LMMSE estimator and derives its MSE.

First, the MSE of a linear estimator (18) is given by

$$\begin{aligned}
\text{MSE} &= \frac{1}{L}\sum_{i=1}^{L}\left|\sum_{m=1}^{M}(\lambda h_m - 1)s_m[i] + c + \lambda z[i]\right|^2 \\
&= \frac{1}{L}\sum_{i=1}^{L}\left|\sum_{m=1}^{M}(\lambda h_m - 1)s_m[i]\right|^2 + c\frac{1}{L}\sum_{i=1}^{L}\left[\sum_{m=1}^{M}(\lambda h_m - 1)s_m[i]\right]^* \\
&\quad + c^*\frac{1}{L}\sum_{i=1}^{L}\left[\sum_{m=1}^{M}(\lambda h_m - 1)s_m[i]\right] + |\lambda|^2\frac{N_0}{T} + |c|^2 \\
&= (\lambda\boldsymbol{h} - \mathbf{1})^H\frac{1}{L}\sum_{i=1}^{L}\boldsymbol{s}^*[\boldsymbol{i}]\boldsymbol{s}[\boldsymbol{i}]^\top(\lambda\boldsymbol{h} - \mathbf{1}) + c(\lambda\boldsymbol{h} - \mathbf{1})^H\frac{1}{L}\sum_{i=1}^{L}\boldsymbol{s}^*[\boldsymbol{i}] \\
&\quad + c^*(\lambda\boldsymbol{h} - \mathbf{1})^\top\frac{1}{L}\sum_{i=1}^{L}\boldsymbol{s}[\boldsymbol{i}] + |\lambda|^2\frac{N_0}{T} + |c|^2 \\
&= (\lambda\boldsymbol{h} - \mathbf{1})^H\boldsymbol{V}(\lambda\boldsymbol{h} - \mathbf{1}) + c(\lambda\boldsymbol{h} - \mathbf{1})^H\hat{\boldsymbol{\mu}}^* + c^*(\lambda\boldsymbol{h} - \mathbf{1})^\top\hat{\boldsymbol{\mu}}
\end{aligned}$$

$$+ |\lambda|^2 \frac{N_0}{T} + |c|^2, \tag{42}$$

where $\boldsymbol{V}$ is defined in (10) and $\hat{\boldsymbol{\mu}}$ is defined in (15). As can be seen, MSE is a quadratic function of both λ and c. The optimal λ and c that minimize the MSE can be obtained by setting the first derivative of MSE (with respect to λ and c, respectively) to 0.

$$\frac{\partial \text{MSE}}{\partial \lambda} = \lambda^* \boldsymbol{h}^H \boldsymbol{V} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} + c^* \boldsymbol{h}^\top \hat{\boldsymbol{\mu}} + \lambda^* \frac{N_0}{T} = 0, \tag{43}$$

$$\frac{\partial \text{MSE}}{\partial c} = (\lambda \boldsymbol{h} - \mathbf{1})^H \hat{\boldsymbol{\mu}}^* + c^* = 0. \tag{44}$$

Substituting (44) into (43) yields

$$\lambda = \frac{\boldsymbol{h}^H \left(\boldsymbol{V} - \hat{\boldsymbol{\mu}}^* \hat{\boldsymbol{\mu}}^\top\right) \mathbf{1}}{\boldsymbol{h}^H \left(\boldsymbol{V} - \hat{\boldsymbol{\mu}}^* \hat{\boldsymbol{\mu}}^\top\right) \boldsymbol{h} + \frac{N_0}{T}}.$$

To simplify λ, let us further define

$$\boldsymbol{D} = \boldsymbol{V} - \hat{\boldsymbol{\mu}}^* \hat{\boldsymbol{\mu}}^\top = \text{diag}\left(\hat{\mathbb{D}}_1, \hat{\mathbb{D}}_2, \cdots, \hat{\mathbb{D}}_M\right),$$

and we finally have

$$\lambda = \frac{\boldsymbol{h}^H \boldsymbol{D} \mathbf{1}}{\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \frac{N_0}{T}}, \quad c = \left(\mathbf{1} - \frac{\boldsymbol{h}^H \boldsymbol{D} \mathbf{1}}{\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \frac{N_0}{T}} \boldsymbol{h}\right)^\top \hat{\boldsymbol{\mu}}.$$

Thus, (14) is the LMMSE estimator that minimizes the MSE in (18). Notice that this is an unbiased estimator since

$$\frac{1}{L} \sum_{i=1}^{L} (\hat{s}_+[i] - s_+[i]) = \lambda \boldsymbol{h}^\top \hat{\boldsymbol{\mu}} + (\mathbf{1} - \lambda \boldsymbol{h})^\top \hat{\boldsymbol{\mu}} - \mathbf{1}^\top \hat{\boldsymbol{\mu}} = 0.$$

Substituting λ and c back into (42) yields

$$\begin{aligned}
\text{MSE} &= (\lambda \boldsymbol{h} - \mathbf{1})^H \boldsymbol{V} (\lambda \boldsymbol{h} - \mathbf{1}) - (\lambda \boldsymbol{h} - \mathbf{1})^\top \hat{\boldsymbol{\mu}} \hat{\boldsymbol{\mu}}^H (\lambda \boldsymbol{h} - \mathbf{1})^* \\
&\quad - (\lambda \boldsymbol{h} - \mathbf{1})^H \hat{\boldsymbol{\mu}}^* \hat{\boldsymbol{\mu}}^\top (\lambda \boldsymbol{h} - \mathbf{1}) + |\lambda|^2 \frac{N_0}{T} + |(\lambda \boldsymbol{h} - \mathbf{1})^\top \hat{\boldsymbol{\mu}}|^2 \\
&= (\lambda \boldsymbol{h} - \mathbf{1})^H \boldsymbol{D} (\lambda \boldsymbol{h} - \mathbf{1}) + |\lambda|^2 \frac{N_0}{T} \\
&= \mathbf{1}^\top \boldsymbol{D} \mathbf{1} - \frac{\left|\boldsymbol{h}^H \boldsymbol{D} \mathbf{1}\right|^2}{\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \frac{N_0}{T}}.
\end{aligned}$$

We next compare $\text{MSE}_{\text{LMMSE}}$ with MSE_{ML}. From (9) and (19), we have

$$\begin{aligned}
&\text{MSE}_{\text{ML}} - \text{MSE}_{\text{LMMSE}} \qquad (45) \\
&= (\boldsymbol{h} - \mathbf{1})^H \boldsymbol{V} (\boldsymbol{h} - \mathbf{1}) + \frac{N_0}{T} - \left(\mathbf{1}^\top \boldsymbol{D} \mathbf{1} - \frac{\left|\boldsymbol{h}^H \boldsymbol{D} \mathbf{1}\right|^2}{\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \frac{N_0}{T}}\right).
\end{aligned}$$

Multiplying both sides of (45) by $\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \frac{N_0}{T}$ and defining

$$\begin{aligned}
q\left(\frac{N_0}{T}\right) &= (\text{MSE}_{\text{ML}} - \text{MSE}_{\text{LMMSE}}) \left(\boldsymbol{h}^H \boldsymbol{V} \boldsymbol{h} + \frac{N_0}{T}\right) \\
&= \left(\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \frac{N_0}{T}\right) \left(\boldsymbol{h}^H \boldsymbol{V} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} - \boldsymbol{h}^H \boldsymbol{V} \mathbf{1} \right. \\
&\quad \left. + \mathbf{1}^\top \boldsymbol{V} \mathbf{1} - \mathbf{1}^\top \boldsymbol{D} \mathbf{1} + \frac{N_0}{T}\right) + |\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h}|^2. \qquad (46)
\end{aligned}$$

Since $\boldsymbol{D}$ is positive definite, we have $\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \frac{N_0}{T} > 0$. To prove $\text{MSE}_{\text{LMMSE}} \leq \text{MSE}_{\text{ML}}$, we only need to prove the minimum value of $q\left(\frac{N_0}{T}\right)$ is no less than 0.

From (46), we know that $q(N_0/T)$ is a quadratic function of N_0/T, the N_0/T that minimizes $q(N_0/T)$ can then be obtained by setting the first derivative of $q(N_0/T)$ (with respect to N_0/T) to 0, giving

$$\begin{aligned}
\frac{\partial q(N_0/T)}{\partial (N_0/T)} &= 2\frac{N_0}{T} + \boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \boldsymbol{h}^H \boldsymbol{V} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} - \boldsymbol{h}^H \boldsymbol{V} \mathbf{1} \\
&\quad + \mathbf{1}^\top \boldsymbol{V} \mathbf{1} - \mathbf{1}^\top \boldsymbol{D} \mathbf{1} = 0.
\end{aligned}$$

Since the noise variance cannot be negative, we have

$$\begin{aligned}
\frac{N_0}{T} = \max\Big(0, \frac{1}{2}\big[&\mathbf{1}^\top \boldsymbol{D} \mathbf{1} - (\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \boldsymbol{h}^H \boldsymbol{V} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} \\
&- \boldsymbol{h}^H \boldsymbol{V} \mathbf{1} + \mathbf{1}^\top \boldsymbol{V} \mathbf{1})\big]\Big).
\end{aligned}$$

1) When $\frac{1}{2}\big[\mathbf{1}^\top \boldsymbol{D} \mathbf{1} - (\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \boldsymbol{h}^H \boldsymbol{V} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} - \boldsymbol{h}^H \boldsymbol{V} \mathbf{1} + \mathbf{1}^\top \boldsymbol{V} \mathbf{1})\big] \leq 0$, we have $N_0/T = 0$ and

$$\begin{aligned}
q\left(\frac{N_0}{T}\right) \geq q(0) &= \boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} (\boldsymbol{h}^H \boldsymbol{V} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} - \boldsymbol{h}^H \boldsymbol{V} \mathbf{1} \\
&\quad + \mathbf{1}^\top \boldsymbol{V} \mathbf{1}) - \boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} \mathbf{1}^\top \boldsymbol{D} \mathbf{1} + |\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h}|^2
\end{aligned}$$

Let us define $\boldsymbol{E} = \boldsymbol{V} - \boldsymbol{D} = \hat{\boldsymbol{\mu}}^* \hat{\boldsymbol{\mu}}^\top$, then

$$\begin{aligned}
q(0) &= \boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} (\boldsymbol{h}^H \boldsymbol{V} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} - \boldsymbol{h}^H \boldsymbol{V} \mathbf{1} + \mathbf{1}^\top \boldsymbol{E} \mathbf{1}) + |\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h}|^2 \\
&= \boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} (\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{D} \boldsymbol{h} - \boldsymbol{h}^H \boldsymbol{D} \mathbf{1}) + \boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} (\boldsymbol{h}^H \boldsymbol{E} \boldsymbol{h} \\
&\quad - \mathbf{1}^\top \boldsymbol{E} \boldsymbol{h} - \boldsymbol{h}^H \boldsymbol{E} \mathbf{1} + \mathbf{1}^\top \boldsymbol{E} \mathbf{1}) + |\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h}|^2 \\
&= (\boldsymbol{h} - \mathbf{1})^H \boldsymbol{D} \boldsymbol{h} \boldsymbol{h}^H \boldsymbol{D} (\boldsymbol{h} - \mathbf{1}) + \boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} (\boldsymbol{h} - \mathbf{1})^H \boldsymbol{E} (\boldsymbol{h} - \mathbf{1}) \\
&= |(\boldsymbol{h} - \mathbf{1})^H \boldsymbol{D} \boldsymbol{h}|^2 + \boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} |(\boldsymbol{h} - \mathbf{1})^\top \hat{\boldsymbol{\mu}}|^2 \geq 0,
\end{aligned}$$

where the last inequality follows because $\boldsymbol{D}$ is positive definite. Therefore,

$$\text{MSE}_{\text{ML}} - \text{MSE}_{\text{LMMSE}} \geq \frac{q(0)}{\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h}} \geq 0.$$

This formula matches our intuition: when the noise variance $N_0/T = 0$ and the channel precoding is perfect, i.e., $\boldsymbol{h} = \mathbf{1}$, we have $q(0) = 0$. The ML and the LMMSE estimators are the same in this case as $\lambda = 1$ and $c = 0$.

2) When $\frac{1}{2}\big[\mathbf{1}^\top \boldsymbol{D} \mathbf{1} - (\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \boldsymbol{h}^H \boldsymbol{V} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} - \boldsymbol{h}^H \boldsymbol{V} \mathbf{1} + \mathbf{1}^\top \boldsymbol{V} \mathbf{1})\big] > 0$, we have

$$\begin{aligned}
q\left(\frac{N_0}{T}\right) &\geq q\Big(\frac{1}{2}\big[\mathbf{1}^\top \boldsymbol{D} \mathbf{1} - (\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \boldsymbol{h}^H \boldsymbol{V} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} \\
&\quad - \boldsymbol{h}^H \boldsymbol{V} \mathbf{1} + \mathbf{1}^\top \boldsymbol{V} \mathbf{1})\big]\Big) \\
&= |\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h}|^2 - \frac{1}{4}\Big(\mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} + \boldsymbol{h}^H \boldsymbol{V} \mathbf{1} - \boldsymbol{h}^H \boldsymbol{E} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{E} \mathbf{1}\Big)^2 \\
&= |\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h}|^2 - \frac{1}{4}\Big[\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h} + \boldsymbol{h}^H \boldsymbol{D} \mathbf{1} - (\boldsymbol{h} - \mathbf{1})^H \boldsymbol{E} (\boldsymbol{h} - \mathbf{1})\Big]^2 \\
&= |\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h}|^2 - \frac{1}{4}\Big[2(\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h})^{\mathfrak{r}} - |(\boldsymbol{h} - \mathbf{1})^\top \hat{\boldsymbol{\mu}}|^2\Big]^2 \\
&\overset{(a)}{\geq} |\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h}|^2 - \big[(\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h})^{\mathfrak{r}}\big]^2 = \big[(\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h})^{\mathfrak{i}}\big]^2 \geq 0
\end{aligned}$$

where (a) follows from $\frac{1}{2}\big[\mathbf{1}^\top \boldsymbol{D} \mathbf{1} - (\boldsymbol{h}^H \boldsymbol{D} \boldsymbol{h} + \boldsymbol{h}^H \boldsymbol{V} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} - \boldsymbol{h}^H \boldsymbol{V} \mathbf{1} + \mathbf{1}^\top \boldsymbol{V} \mathbf{1})\big] > 0$. That is, we have $\mathbf{1}^\top \boldsymbol{V} \boldsymbol{h} + \boldsymbol{h}^H \boldsymbol{V} \mathbf{1} - \boldsymbol{h}^H \boldsymbol{E} \boldsymbol{h} - \mathbf{1}^\top \boldsymbol{E} \mathbf{1} \geq 0$, and hence $2(\mathbf{1}^\top \boldsymbol{D} \boldsymbol{h})^{\mathfrak{r}} - |(\boldsymbol{h} - \mathbf{1})^\top \hat{\boldsymbol{\mu}}|^2 > 0$.

Overall, we have $\text{MSE}_{\text{LMMSE}} \leq \text{MSE}_{\text{ML}}$.

APPENDIX B

This appendix proves Theorem 8. The proof is in general analogous to the proof in Appendix B of our companion paper [8]. To avoid unnecessary duplication, we omit the detailed derivations but point out an important result: *if all inputs to a graph is Gaussian, then all the variables on the graph are multivariate Gaussian random variables.*

In Fig. 4, two inputs to the factor graph are the likelihood function $f(y_{k,i}|\boldsymbol{W}_{k,i})$ and the prior information $f_{\boldsymbol{s_1},\boldsymbol{s_2}....,\boldsymbol{s}_M}$. We next show that both messages are multivariate Gaussian distributions with respect to $\boldsymbol{W_{k,i}}$.

1) The likelihood function $f(y_{k,i}|\boldsymbol{W}_{k,i})$. Let us consider a complex random variable as a pair of real random variables: the real part and the imaginary part. Then, $\boldsymbol{W_{k,i}}$ can be viewed as a $2M$-dimensional real random variable and denoted by

$$\boldsymbol{w_{k,i}} = \left(b_1^{\mathrm{r}}, b_2^{\mathrm{r}}, ..., b_M^{\mathrm{r}}, b_1^{\mathrm{i}}, b_2^{\mathrm{i}}, ..., b_M^{\mathrm{i}}\right).$$

As per (25), we have

$$\begin{aligned} y_{k,i} &= \sum_{k=1}^{M}(h_k^{\mathrm{r}} + jh_k^{\mathrm{i}})(b_k^{\mathrm{r}} + jb_k^{\mathrm{i}}) + z_{k,i}^{\mathrm{r}} + jz_{k,i}^{\mathrm{i}} \\ &= \sum_{k=1}^{M}[(h_k^{\mathrm{r}}b_k^{\mathrm{r}} - h_k^{\mathrm{i}}b_k^{\mathrm{i}} + z_{k,i}^{\mathrm{r}}) + j(h_k^{\mathrm{r}}b_k^{\mathrm{i}} + h_k^{\mathrm{i}}b_k^{\mathrm{r}} + z_{k,i}^{\mathrm{i}})], \end{aligned}$$

where $z_{k,i}^{\mathrm{r}}$, $z_{k,i}^{\mathrm{i}} \sim \mathcal{N}(0, \frac{N_0}{2d_k})$. Thus, the likelihood function

$$\begin{aligned} f(y_{k,i}|\boldsymbol{w_{k,i}}) &\propto \exp\left\{-\frac{d_k}{N_0}[y_{k,i}^{\mathrm{r}} - \sum_k (h_k^{\mathrm{r}}b_k^{\mathrm{r}} - h_k^{\mathrm{i}}b_k^{\mathrm{i}})]^2\right\} \\ &\quad \times \exp\left\{-\frac{d_k}{N_0}[y_{k,i}^{\mathrm{i}} - \sum_k (h_k^{\mathrm{r}}b_k^{\mathrm{i}} + h_k^{\mathrm{i}}b_k^{\mathrm{r}})]^2\right\} \\ &\propto \mathcal{N}\left(\frac{2d_k}{N_0}\begin{bmatrix}\boldsymbol{\beta_1}\\ \boldsymbol{\beta_2}\end{bmatrix}\begin{bmatrix}y_{k,i}^{\mathrm{r}}\\ y_{k,i}^{\mathrm{i}}\end{bmatrix}, \frac{2d_k}{N_0}\begin{bmatrix}\boldsymbol{\beta_1}\boldsymbol{\beta_1}^\top & \boldsymbol{\beta_1}\boldsymbol{\beta_2}^\top \\ \boldsymbol{\beta_2}\boldsymbol{\beta_1}^\top & \boldsymbol{\beta_1}\boldsymbol{\beta_1}^\top\end{bmatrix}\right), \end{aligned}$$

where $\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_1}$ are composed of channel coefficients, giving

$$\boldsymbol{\beta_1} = \begin{bmatrix} h_1^{\mathrm{r}} & h_1^{\mathrm{i}} \\ h_2^{\mathrm{r}} & h_2^{\mathrm{i}} \\ ... & ... \\ h_M^{\mathrm{r}} & h_M^{\mathrm{i}} \end{bmatrix}, \quad \boldsymbol{\beta_2} = \begin{bmatrix} -h_1^{\mathrm{i}} & h_1^{\mathrm{r}} \\ -h_2^{\mathrm{i}} & h_2^{\mathrm{r}} \\ ... & ... \\ -h_M^{\mathrm{i}} & h_M^{\mathrm{r}} \end{bmatrix}.$$

2) The prior information $f_{\boldsymbol{s_1},\boldsymbol{s_2}....,\boldsymbol{s}_M}$ is an M-dimensional complex Gaussian distribution, as stated in IV-C. If we treat it as a $2M$-dimensional real Gaussian distribution, then

$$f_{\boldsymbol{s_1},\boldsymbol{s_2}....,\boldsymbol{s}_M} \propto \mathcal{N}\left(\boldsymbol{\mu}_p, \boldsymbol{\Sigma_p}\right),$$

where the mean and covariance $\boldsymbol{\mu}_p$, $\boldsymbol{\Sigma_p}$ can be constructed from the sample means $\{\hat{\mathbb{E}}_m : m = 1, 2, ..., M\}$ and the sample variances $\{\hat{\mathbb{D}}_m : m = 1, 2, ..., M\}$, giving

$$\begin{aligned} \boldsymbol{\mu_p} &= \left[\hat{\mathbb{E}}_1^{\mathrm{r}}, \hat{\mathbb{E}}_2^{\mathrm{r}}, ..., \hat{\mathbb{E}}_M^{\mathrm{r}}, \hat{\mathbb{E}}_1^{\mathrm{i}}, \hat{\mathbb{E}}_2^{\mathrm{i}}, ..., \hat{\mathbb{E}}_M^{\mathrm{i}}\right]^\top, \\ \boldsymbol{\Sigma_p} &= \frac{1}{2}\mathrm{diag}\left(\hat{\mathbb{D}}_1, \hat{\mathbb{D}}_2, ..., \hat{\mathbb{D}}_M, \hat{\mathbb{D}}_1, \hat{\mathbb{D}}_2, ..., \hat{\mathbb{D}}_M\right). \end{aligned}$$

Since both the likelihood function $f(y_{k,i}|\boldsymbol{W}_{k,i})$ and the prior information $f_{\boldsymbol{s_1},\boldsymbol{s_2}....,\boldsymbol{s}_M}$ are Gaussian, all the messages passed on the graph are (multivariate) Gaussian. The Gaussian PDF can then be parameterized by its mean vector and covariance matrix – passing these parameters is equivalent to passing the continuous PDF on the edges. As a result, the first part of Theorem 8 is valid: $f(\boldsymbol{s[i]}|\boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M})$ is Gaussian; its parameters can be computed via the analog message passing algorithm.

The second part of Theorem 8 follows directly from the first part since $s_+[i]$ is a linear transformation of $\boldsymbol{s[i]}$. If $f(\boldsymbol{s[i]}|\boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M})$ is Gaussian, then $f(s_+[i]|\boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_M})$ is Gaussian after marginalization.

REFERENCES

[1] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, 2020.

[2] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: wireless communication meets machine learning," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19–25, 2020.

[3] G. Zhu, J. Xu, and K. Huang, "Over-the-air computing for wireless data aggregation in massive IoT," *arXiv:2009.02181*, 2020.

[4] Y. Shao, S. C. Liew, and L. Lu, "Asynchronous physical-layer network coding: symbol misalignment estimation and its effect on decoding," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6881–6894, 2017.

[5] M. Goldenbaum, H. Boche, and S. Stańczak, "Harnessing interference for analog function computation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 61, no. 20, pp. 4893–4906, 2013.

[6] M. Goldenbaum and S. Stanczak, "Robust analog function computation via wireless multiple-access channels," *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3863–3877, 2013.

[7] X. Cao, G. Zhu, J. Xu, and K. Huang, "Optimized power control for over-the-air computation in fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7498–7513, 2020.

[8] Y. Shao, D. Gündüz, and S. C. Liew, "Federated learning with misaligned over-the-air computation," *Technical report, available online: https://arxiv.org/abs/2009.13441*, 2021.

[9] O. Abari, H. Rahul, and D. Katabi, "Over-the-air function computation in sensor networks," *arXiv:1612.02307*, 2016.

[10] T. Sery, N. Shlezinger, K. Cohen, and Y. C. Eldar, "Over-the-air federated learning from heterogeneous data," *arXiv:2009.12787*, 2020.

[11] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, 2007.

[12] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, 2020.

[13] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, 2020.

[14] N. Zhang and M. Tao, "Gradient statistics aware power control for over-the-air federated learning in fading channels," *arXiv:2003.02089*, 2020.

[15] M. M. Amiri, T. M. Duman, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Blind federated edge learning," *arXiv cs.IT.2010.10030*, 2020.

[16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AI Statistics*. PMLR, 2017, pp. 1273–1282.

[17] A. Gupta and R. K. Jha, "A survey of 5G network: architecture and emerging technologies," *IEEE access*, vol. 3, pp. 1206–1232, 2015.

[18] Y. Shao and S. C. Liew, "Flexible subcarrier allocation for interleaved frequency division multiple access," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7139–7152, 2020.

[19] G. Zhu and K. Huang, "MIMO over-the-air computation for high-mobility multimodal sensing," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6089–6103, 2018.

[20] Y. Shao, S. C. Liew, and D. Gündüz, "Denoising noisy neural networks: a Bayesian approach with compensation," *available online: https://arxiv.org/abs/2105.10699*, 2021.

[21] H. Xing, O. Simeone, and S. Bi, "Decentralized federated learning via sgd over wireless d2d networks," *arXiv:2002.12507*, 2020.

[22] R. C. Buck, "Approximate complexity and functional representation," *J. Math. Analysis App*, vol. 70, pp. 280–298, 1979.

[23] A. Tveit, "On the complexity of matrix inversion," *Mathematical Note*, p. 1, 2003.

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

[25] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[26] T. Wang, L. Shi, S. Zhang, and H. Wang, "Gaussian mixture message passing for blind known interference cancellation," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4268–4282, 2019.