Underwater Acoustic Channel Estimation via Fast Bayesian Matching Pursuit

Huajian Chen and Chenhao Qi
School of Information Science and Engineering
Southeast University, Nanjing 210096, China
Email: qch@seu.edu.cn

***Abstract*—Underwater acoustic (UWA) channel estimation based on fast Bayesian matching pursuit (FBMP) is investigated in this paper. In UWA multipath channel, the positions of UWA channel taps usually obey the Bernoulli distribution, while the coefficients of UWA channel taps obey the complex-valued Gaussian distribution. Such knowledge of distribution is considered as a prior for UWA channel estimation based on the Bayesian framework. The positions of UWA channel taps are first determined. Then the coefficients of UWA channel taps are estimated. Regarding the complexities for Bayesian estimation, the fast implementation is also studied, where the stop condition for the proposed iterative algorithm is also optimized. Simulation results show that FBMP outperforms orthogonal matching pursuit (OMP). In order to achieve the same bit error rate (BER) of 0.016, FBMP can save around 4dB of signal to noise ratio (SNR) compared to OMP.**

***Index Terms*—Underwater acoustic (UWA) communications, sparse channel estimation, compressed sensing, Bayesian estimation**

I. Introduction

Underwater acoustic (UWA) channel is one of the most adverse medium for communications, since it is fast time-varying with abundant multipath propagation. In particular, UWA channel parameters are different at different sea area, different water depth, different sea temperature and different water salinity. Therefore, it is essentially important to make UWA channel estimation for reliable coherent UWA communications.

It is shown that the impulse response (CIR) of UWA channel is usually sparse, where only a small number of channel coefficients are nonzero [1]. Therefore, by exploiting the sparse property of UWA channel, compressed sensing (CS) technique can be applied for sparse channel estimation. In [2], an algorithm based on the combination of the matching pursuit and the maximum a posteriori probability (MAP) based space-alternating generalized expectation-maximization technique is presented to improve the estimates of UWA channel taps iteratively. In [3], a new UWA channel-estimate-based decision feedback equalizer (DFE) is presented to deal with high platform mobility by exploiting the sparse multipath structure, where the key component of this DFE is a linear-complexity sparse channel estimator to detect and reject impulses based on two noise models. In [4], the selection of nonzero taps for sparse linear equalizer under the criterion of minimum mean square error (MMSE) is investigated, where the joint determination of positions and weights of nonzero taps of sparse equalizer is considered and then formulated as a subset selection problem. In [5], a multi-input multi-output (MIMO) UWA communication detector that uses the improved proportionate normalized least mean squares (IPNLMS) algorithm for iterative channel estimation in turbo equalization is presented, where by exploiting the sparse nature of UWA channels the iterative IPNLMS channel estimator achieves fast convergence and fast tracking by data reuse of training sequence and decision-directed channel reestimation over each turbo iterations. In [6], UWA channel estimation based on sparse recovery algorithms as well as the optimal pilot design is presented. In [7], a sparse recovery algorithm named complex homotopy with a reasonable tradeoff between the sparse recovery performance and the computational complexity is proposed for UWA channel estimation.

Bayesian compressed sensing (BCS) [8], as a special form of CS, has better performance than the traditional CS algorithms in many applications by using the statistics of sparse signal as a priori knowledge.

In this paper, we investigate UWA channel estimation based on fast Bayesian matching pursuit (FBMP) [9]. In UWA multipath channel, the positions of UWA channel taps usually obey the Bernoulli distribution, while the coefficients of UWA channel taps obey the complex-valued Gaussian distribution. Such knowledge of distribution can be considered as a prior for UWA channel estimation based on the Bayesian framework. We first determine the positions of UWA channel taps and then estimate the coefficients of UWA channel taps. Regarding the complexities for Bayesian estimation, we also study the fast implementation of FBMP, where the stop condition for the proposed iterative algorithm is also optimized.

The notations used in this paper are defined as follows. Symbols for matrices (upper case) and vectors (lower case) are in boldface. $(\cdot)^T$, $(\cdot)^H$, diag$\{\cdot\}$, $\boldsymbol{I}_L$, $\mathbb{C}^M$, $\mathbf{0}^M$, $\mathbf{0}^{M\times N}$, E$\{\cdot\}$, det$\{\cdot\}$, $\|\cdot\|_2$, $\mathcal{CN}$ and $\varnothing$ denote the transpose, conjugate transpose (Hermitian), diagonal matrix, identity matrix with dimension L, the set of complex-valued vectors with dimension M, zero vector with dimension M, $M\times N$ zero matrix, expectation, determinant of a matrix, ℓ_2 norm, complex Gaussian distribution and the empty set, respectively.

II. Problem formulation

The UWA channel is characterized by abundant multipath reflection and severe Doppler effect. Therefore, UWA channel

978-1-5386-2062-5/17/$31.00 ©2017 IEEE

can be modeled with a time-varying multipath CIR as

$$h(\tau,t)=\sum_{i=1}^{K}\alpha_i(t)\delta(\tau-\tau_i(t)) \tag{1}$$

where K, $\alpha_i(t)$ and $\tau_i(t)$ are the number of total paths, the i-th path attenuation and the path delay, respectively. In [10], it is assumed that the dominant Doppler shift is caused by the relative movement between the transmitter and the receiver while all paths have the similar Doppler scaling factor $\varepsilon(t)$ so that $\tau_i(t)\approx\tau_i-\varepsilon(t)t$. Let $s(t)=\mathrm{Re}\{x(t)e^{j2\pi f_c t}\}$ denote the transmitted signal in passband, where f_c is the carrier frequency of UWA communications and $x(t)$ is the baseband OFDM signal. We assume the transmission is in the unit of OFDM packet, where each OFDM packet contains one preamble, one postamble and several OFDM symbols. Since the coherence time of UWA channel is usually on the order of seconds while each OFDM packet is no more than hundreds of milliseconds [10], UWA channel parameters can be treated to be constant during the transmission of each OFDM packet. Then the receive passband signal is

$$r(t)=\mathrm{Re}\left\{\sum_{i=1}^{S}\alpha_i x\Big((1+\varepsilon)t-\tau_i\Big)e^{j2\pi f_c\Big((1+\varepsilon)t-\tau_i\Big)}\right\}+n(t) \tag{2}$$

where $n(t)$ is the additive Gaussian noise. It can be observed from (2) that $x(t)$ is scaled in time by $1/(1+\varepsilon)$ and the Doppler shift $e^{j2\pi f_c\varepsilon t}$ is frequency-dependent. Since in UWA communications the bandwidth is comparable with the carrier frequency, the Doppler shift cannot be treated the same for the whole band.

To mitigate the Doppler effect, we employ a two-step approach where the first step resamples the received signal and the second step makes the carrier frequency offset (CFO) compensation. The preamble and the postamble within each OFDM packet are usually filled in with linear frequency modulated (LFM) waveforms. At the receiver, we make cross correlation between the received signal and the known preamble and postamble to estimate the length of each OFDM packet and obtain an estimated Doppler scaling factor $\hat{\varepsilon}$. The equivalent received baseband signal after resampling in the first step is

$$y(t)=\sum_{i=1}^{K}\alpha_i x\left(\frac{1+\varepsilon}{1+\hat{\varepsilon}}t-\tau_i\right)e^{j2\pi f_c\left(\frac{\varepsilon-\hat{\varepsilon}}{1+\hat{\varepsilon}}t-\tau_i\right)}+n_B(t) \tag{3}$$

where $n_B(t)$ is the equivalent noise after resampling. We define the residual CFO as $f_o=f_c(\varepsilon-\hat{\varepsilon})/(1+\hat{\varepsilon})$, which is approximately seen as uniformly distributed over the whole bandwidth. In the second step, we apply the mature null-subcarrier-based scheme for CFO compensation.

Suppose the length of the CIR after sampling is N. The CIR vector can be denoted as $\boldsymbol{h}=[h_0,h_1,...,h_{N-1}]^T$. Considering the large delay spread of UWA channel and the oversampling of the OFDM receiver, $\boldsymbol{h}$ is typically sparse, indicating that only a small number of entries of $\boldsymbol{h}$ are nonzero, i.e., $K\ll N$.

We consider multicarrier UWA communications where we use an OFDM system including N_p subcarriers, among which $M(M\leqslant N_P)$ subcarriers are selected to transmit pilot symbols with the corresponding subcarrier indices denoted as $\boldsymbol{p}=[p_1,p_2,...,p_M]^T$. Suppose the transmit and receive OFDM symbols are $\boldsymbol{x}_T\in\mathbb{C}^{N_p}$ and $\boldsymbol{x}_R\in\mathbb{C}^{N_p}$, respectively. Then $\boldsymbol{x}_T(\boldsymbol{p})$ and $\boldsymbol{x}_R(\boldsymbol{p})$ denote the transmit and receive pilot symbols, respectively. Define $\boldsymbol{D}\triangleq\mathrm{diag}\{\boldsymbol{x}_T(\boldsymbol{p})\}$. Define $\boldsymbol{F_p}$ as an M by N DFT submatrix with row indices $\boldsymbol{p}$ selected from the standard DFT matrix $\boldsymbol{F}$. To ease the notation, we redefine $\boldsymbol{y}\triangleq\boldsymbol{x}_R(\boldsymbol{p})$ and $\boldsymbol{\phi}\triangleq\boldsymbol{DF_p}$. Then UWA channel estimation can be modeled as

$$\boldsymbol{y}=\boldsymbol{\phi h}+\boldsymbol{n} \tag{4}$$

where $\boldsymbol{n}\in\mathbb{C}^M$ is a noise term. Each entry of $\boldsymbol{n}$ independently obeys complex Gaussian distribution with zero mean and variance of σ_n^2, i.e., $\boldsymbol{n}\sim\mathcal{CN}(0,\sigma_n^2\boldsymbol{I}_M)$.

Since $\boldsymbol{h}$ is sparse, we can reconstruct $\boldsymbol{h}$ via $\boldsymbol{\phi}$ and $\boldsymbol{y}$ by taking advantage of CS algorithms, which mainly include orthogonal matching pursuit (OMP), subspace pursuit (SP), compressive sampling matching pursuit (CoSaMP) as well as ℓ_1-norm minimization algorithms. Due to the greedy nature of most heuristic algorithms, e.g., OMP, the sparse reconstruction performance is not globally optimal. In the following, we will introduce a Bayesian sparse reconstruction algorithm that exploits the knowledge of distribution of the sparse vector.

III. Fast Bayesian Matching Pursuit

A. Fast Bayesian Model

UWA channel can usually be modeled as a Bernoulli-Gaussian mixture random process, which means the position and the coefficients of channel taps obey Bernoulli distribution and Gaussian distribution, respectively. We introduce a binary vector $\boldsymbol{z}\triangleq[z_1,z_2,...z_N]^T$ with the same dimension as $\boldsymbol{h}$ to indicate the sparse pattern, i.e., the positions of nonzero entries of $\boldsymbol{h}$. We define $h[n]$ and $z[n]$ as the nth entry of $\boldsymbol{h}$ and $\boldsymbol{z}$, respectively. We use $z[n]\in\{0,1\}$ to indicate whether $h[n]$ is nonzero, i.e., $z[n]=1$ showing $h[n]$ is nonzero and $z[n]=0$ showing $h[n]$ is zero, so

$$h[n]\big|\{z[n]=q\}\sim\mathcal{CN}(0,\sigma_q^2). \tag{5}$$

If $z[n]=0$ $(q=0)$, we define $\sigma_0^2\triangleq 0$; otherwise if $z[n]=1$ $(q=1)$, we define $h[n]\sim\mathcal{CN}(0,\sigma_1^2)$. We have $h[n]|\{z[n]=1\}\sim\mathcal{CN}(0,\sigma_1^2)$ and $h[n]|\{z[n]=0\}\sim\mathcal{CN}(0,\sigma_0^2)$. Note that $\mathcal{P}(z[n]=1)=p_1$, $\mathcal{P}(z[n]=0)=1-p_1$. (5) can be rewritten in vector form as

$$\boldsymbol{h}|\boldsymbol{z}\sim\mathcal{CN}(0,\boldsymbol{R}_{zz}) \tag{6}$$

where $\boldsymbol{R_{zz}}=\mathrm{E}\left\{\boldsymbol{zz}^H\right\}$ is defined as the variance matrix of $\boldsymbol{z}$. It is seen that $\boldsymbol{R_{zz}}$ is a diagonal matrix with the nth diagonal entry denoted as $R_{zz}[n,n]=\sigma_{z_n}^2$.

Denote p_1 as the probability that $z[n]=1$. The probability to generate a given $\boldsymbol{z}$ is

$$\mathcal{P}(\boldsymbol{z})=p_1^K(1-p_1)^{N-K} \tag{7}$$

where $K \triangleq |\boldsymbol{z}|$ denotes the number of entries in $\boldsymbol{z}$. It can be observed that the sparsity of $\boldsymbol{h}$ is essentially controlled by p_1, since $\mathrm{E}\{K\} = Np_1$.

Define $\boldsymbol{h_z}$ as a subset of $\boldsymbol{h}$ by selecting entries of $\boldsymbol{h}$ corresponding to the positions that $z[n] = 1$. Define $\boldsymbol{\phi_z}$ as a submatrix by selecting columns of $\boldsymbol{\phi}$ corresponding to the positions that $z[n] = 1$. Assume that each entry of $\boldsymbol{h_z}$ independently obeys complex Gaussian distribution with zero mean and variance of σ_1^2, i.e., $\boldsymbol{h}_z \sim \mathcal{CN}(0, \sigma_1^2 \boldsymbol{I}_K)$. According to the Bayesian rule, the relationship between $\boldsymbol{z}$, $\boldsymbol{x}$ and $\boldsymbol{y}$ can be expressed as

$$\mathcal{P}(\boldsymbol{y}, \boldsymbol{h}|\boldsymbol{z}) = \frac{\mathcal{P}(\boldsymbol{y}, \boldsymbol{h}, \boldsymbol{z})}{\mathcal{P}(\boldsymbol{z})} = \mathcal{P}(\boldsymbol{y}|\boldsymbol{h}, \boldsymbol{z})\mathcal{P}(\boldsymbol{h}|\boldsymbol{z}). \tag{8}$$

Since $\boldsymbol{z}$ is determined by $\boldsymbol{h}$, we have

$$\mathcal{P}(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{h}) = \mathcal{P}(\boldsymbol{y}|\boldsymbol{h}, \boldsymbol{z}) = \mathcal{P}(\boldsymbol{y}|\boldsymbol{h}). \tag{9}$$

Then (8) is

$$\mathcal{P}(\boldsymbol{y}, \boldsymbol{h}|\boldsymbol{z}) = \mathcal{P}(\boldsymbol{y}|\boldsymbol{h})\mathcal{P}(\boldsymbol{h}|\boldsymbol{z}). \tag{10}$$

Therefore the model can be formulated as

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{h} \end{bmatrix} \Bigg| \boldsymbol{z} = \begin{bmatrix} \boldsymbol{y}|\boldsymbol{z} \\ \boldsymbol{h}|\boldsymbol{z} \end{bmatrix} \sim \mathcal{CN}\left(0, \begin{bmatrix} \boldsymbol{\varphi}(\boldsymbol{z}) & \boldsymbol{\phi}\boldsymbol{R}_{zz} \\ \boldsymbol{R}_{zz}\boldsymbol{\phi}^H & \boldsymbol{R}_{zz} \end{bmatrix}\right) \tag{11}$$

where

$$\boldsymbol{\varphi}(\boldsymbol{z}) \triangleq \boldsymbol{\phi}\boldsymbol{R}_{zz}\boldsymbol{\phi}^H + \boldsymbol{\sigma}_n^2 \boldsymbol{I}_M. \tag{12}$$

The MMSE estimation of $\boldsymbol{h}$ after receiving the signal $\boldsymbol{y}$ is

$$\hat{\boldsymbol{h}}_{\text{MMSE}} = \mathrm{E}\{\boldsymbol{h}|\boldsymbol{y}\} = \sum_{\boldsymbol{z}\in\boldsymbol{\Lambda}} \mathrm{E}\{\boldsymbol{h}|\boldsymbol{y}, \boldsymbol{z}\}\mathcal{P}(\boldsymbol{z}|\boldsymbol{y}). \tag{13}$$

where $\mathrm{E}\{\boldsymbol{h}|\boldsymbol{y}, \boldsymbol{z}\}$ is the expectation of $\boldsymbol{h}$ given $\boldsymbol{y}$ and $\boldsymbol{z}$, $\mathcal{P}(\boldsymbol{z}|\boldsymbol{y})$ is a posteriori probability, and $\boldsymbol{\Lambda} \subset \{0, 1\}^N$ is the all possible sets of $\boldsymbol{z}$.

We now derive $\mathrm{E}\{\boldsymbol{h}|\boldsymbol{y}, \boldsymbol{z}\}$ and $\mathcal{P}(\boldsymbol{z}|\boldsymbol{y})$. Since $\mathrm{E}\{\boldsymbol{h}|\boldsymbol{y}, \boldsymbol{z}\}$ is the MMSE estimation of $\boldsymbol{h}_z$

$$\begin{aligned} \mathrm{E}\{\boldsymbol{h}|\boldsymbol{y}, \boldsymbol{z}\} &= \sigma_1^2 \boldsymbol{\phi}_{\boldsymbol{z}}^H (\sigma_1^2 \boldsymbol{\phi}_{\boldsymbol{z}} \boldsymbol{\phi}_{\boldsymbol{z}}^H + \sigma_n^2 \boldsymbol{I}_M)^{-1} \boldsymbol{y} \\ &= \sigma_1^2 \boldsymbol{\phi}_{\boldsymbol{z}}^H \boldsymbol{\varphi}(\boldsymbol{z})^{-1} \boldsymbol{y}. \end{aligned} \tag{14}$$

According to the Bayesian rule,

$$\mathcal{P}(\boldsymbol{z}|\boldsymbol{y}) = \frac{\mathcal{P}(\boldsymbol{y}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z})}{\mathcal{P}(\boldsymbol{y})}. \tag{15}$$

Since $\mathcal{P}(\boldsymbol{y})$ can be treated as a constant and $\mathcal{P}(\boldsymbol{s})$ has been already given by (7), we have to compute $\mathcal{P}(\boldsymbol{y}|\boldsymbol{s})$. According to the multidimensional Gaussian distribution, $\mathcal{P}(\boldsymbol{y}|\boldsymbol{z})$ can be derived as

$$\mathcal{P}(\boldsymbol{y}|\boldsymbol{z}) = \frac{1}{(\sqrt{2\pi})^N |\boldsymbol{\varphi}(\boldsymbol{z})|^{\frac{1}{2}}} \exp\left(\frac{1}{2}\boldsymbol{y}^H \boldsymbol{\varphi}(\boldsymbol{z})^{-1} \boldsymbol{y}\right). \tag{16}$$

Then we have

$$\mathcal{P}(\boldsymbol{z}|\boldsymbol{y}) = \frac{p_1^K (1-p_1)^{N-K}}{(\sqrt{2\pi})^N |\boldsymbol{\varphi}(\boldsymbol{z})|^{\frac{1}{2}} \mathcal{P}(\boldsymbol{y})} \exp\left(\frac{1}{2}\boldsymbol{y}^H \boldsymbol{\varphi}(\boldsymbol{z})^{-1} \boldsymbol{y}\right). \tag{17}$$

In some applications, e.g., UWA communication system, the dimension of $\boldsymbol{h}$ is very large, which makes it impractical to compute all posterior probabilities and expectations. In order to reduce the computational complexity of Bayesian estimation, it is necessary to limit the number of possible set $\boldsymbol{\Lambda}$. We define $\boldsymbol{\Lambda}^*$ as the main support to approximate $\boldsymbol{\Lambda}$. Then (13) can be approximated by

$$\hat{\boldsymbol{h}}_{\text{AMMSE}} = \sum_{\boldsymbol{z}\in\boldsymbol{\Lambda}^*} \mathrm{E}\{\boldsymbol{h}|\boldsymbol{y}, \boldsymbol{z}\}\mathcal{P}(\boldsymbol{z}|\boldsymbol{y}). \tag{18}$$

which called Approximate MMSE (AMMSE). Therefore, how to find the main support is an important step to fast implement the Bayesian estimation.

B. Estimation of $\boldsymbol{z}$

According to Subsection A, the determination of nonzero entry's position of $\boldsymbol{h}$ is essentially the estimation of $\boldsymbol{z}$. The posterior probability of $\boldsymbol{z}$ is

$$\mathcal{P}(\boldsymbol{z}|\boldsymbol{y}) = \frac{\mathcal{P}(\boldsymbol{y}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z})}{\sum_{\boldsymbol{z}'\in\boldsymbol{\Lambda}} \mathcal{P}(\boldsymbol{y}|\boldsymbol{z}')\mathcal{P}(\boldsymbol{z}')} \tag{19}$$

Since $\boldsymbol{\Lambda}$ is large, the computational complexity of (19) is high. To reduce the computational complexity, we introduce $\boldsymbol{\Lambda}^*$ to approximate $\boldsymbol{\Lambda}$.

We define the selection metric for $\boldsymbol{z}$ as

$$\begin{aligned} v(\boldsymbol{z}) &\triangleq \ln \mathcal{P}(\boldsymbol{y}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z}) \\ &= \ln \mathcal{P}(\boldsymbol{y}|\boldsymbol{z}) + \ln \prod_{n=1}^{N} \mathcal{P}(z_n) \\ &= \ln \mathcal{P}(\boldsymbol{y}|\boldsymbol{z}) + \|z\|_0 \ln p_1 + (N - \|\boldsymbol{z}\|_0)\ln(1-p_1) \\ &= -\frac{N}{2}\ln 2\pi - \frac{1}{2}\ln\det\big(\varphi(\boldsymbol{z})\big) - \frac{1}{2}\boldsymbol{y}^H \varphi(\boldsymbol{z})^{-1}\boldsymbol{y} \\ &\quad + \|\boldsymbol{z}\|_0 \ln\frac{p_1}{1-p_1} + N\ln(1-p_1) \end{aligned} \tag{20}$$

where

$$\begin{aligned} \ln \mathcal{P}(\boldsymbol{y}|\boldsymbol{z}) = &-\frac{N}{2}\ln 2\pi - \frac{1}{2}\ln\det\big(\varphi(\boldsymbol{z})\big) \\ &- \frac{1}{2}\boldsymbol{y}^H \varphi(\boldsymbol{z})^{-1}\boldsymbol{y}. \end{aligned} \tag{21}$$

We define $\boldsymbol{z}_n$ as a vector where all entries are equal to $\boldsymbol{z}$ except the nth entry. The difference between $\boldsymbol{z}_n$ and $\boldsymbol{z}$ is denoted as

$$\Delta_n(\boldsymbol{z}_n) = v(\boldsymbol{z}_n) - v(\boldsymbol{z}). \tag{22}$$

Note that $\boldsymbol{\varphi}(\boldsymbol{0}) = \sigma_n^2 \boldsymbol{I}_M$ according to (12). At the initial state that $\boldsymbol{z} = \boldsymbol{0}$, we have

$$v(\boldsymbol{0}) = -\frac{N}{2}\ln 2\pi - M\ln\sigma_n - \frac{1}{2\sigma_n^2}\|\boldsymbol{y}\|_2^2 + N\ln(1-p_1) \tag{23}$$

In order to derive $v(\boldsymbol{z}_n)$, we first compute $\boldsymbol{\varphi}(\boldsymbol{z}_n)$ based on (12) as

$$\begin{aligned} \boldsymbol{\varphi}(\boldsymbol{z}_n) &= \boldsymbol{\phi}\boldsymbol{R}_{\boldsymbol{z}_n\boldsymbol{z}_n}\boldsymbol{\phi}^H + \sigma_n^2 \boldsymbol{I}_M = \boldsymbol{\phi}(\boldsymbol{R}_{\boldsymbol{z}\boldsymbol{z}} + \boldsymbol{A})\boldsymbol{\phi}^H \\ &= \boldsymbol{\varphi}(\boldsymbol{z}) + \sigma_1^2 \boldsymbol{\phi}_n \boldsymbol{\phi}_n^H, \end{aligned} \tag{24}$$

where $\boldsymbol{A}$ is an $N \times N$ square matrix, with all the entries equal zero except $A[n,n] = \sigma_1^2$, and $\boldsymbol{\phi}_n$ denotes the nth column of $\boldsymbol{\phi}$. According to the matrix inversion lemma, we have

$$\begin{aligned}\boldsymbol{\varphi}(\boldsymbol{z}_n)^{-1} =& \boldsymbol{\varphi}(\boldsymbol{z})^{-1} - \boldsymbol{\varphi}(\boldsymbol{z})^{-1}\boldsymbol{\phi}_n\big(\boldsymbol{\phi}_n^H\boldsymbol{\varphi}(\boldsymbol{z})^{-1}\boldsymbol{\phi}_n \\ &+ \sigma_1^{-2}\big)^{-1}\boldsymbol{\phi}_n^H\boldsymbol{\varphi}(\boldsymbol{z})^{-1}. \end{aligned} \tag{25}$$

We define

$$\boldsymbol{b}_n \triangleq \boldsymbol{\varphi}(\boldsymbol{z})^{-1}\boldsymbol{\phi}_n \tag{26}$$

and

$$\beta_n \triangleq \sigma_1^2\big(1+\sigma_1^2\boldsymbol{\phi}_n^H\boldsymbol{\varphi}(\boldsymbol{z})^{-1}\boldsymbol{\phi}_n\big)^{-1} = \sigma_1^2(1+\sigma_1^2\boldsymbol{\phi}_n^H\boldsymbol{b}_n). \tag{27}$$

Then (25) can be further denoted as

$$\boldsymbol{\varphi}(\boldsymbol{z}_n)^{-1} = \boldsymbol{\varphi}(\boldsymbol{z})^{-1} - \beta_n\boldsymbol{b}_n\boldsymbol{b}_n^H. \tag{28}$$

Suppose $\boldsymbol{z}$ is obtained by changing the entry of $\boldsymbol{z}^{\text{pre}}$ on the position n^{pre} while $\boldsymbol{z}_n$ is obtained by changing the entry of $\boldsymbol{z}$ on the position n. We define

$$\begin{aligned}\boldsymbol{b}_{n^{\text{pre}}}^{\text{pre}} &\triangleq \boldsymbol{\varphi}(\boldsymbol{z}^{\text{pre}})^{-1}\boldsymbol{\phi}_{n^{\text{pre}}} \\ \boldsymbol{b}_n^{\text{pre}} &\triangleq \boldsymbol{\varphi}(\boldsymbol{z}^{\text{pre}})^{-1}\boldsymbol{\phi}_n \\ \beta_{n^{\text{pre}}}^{\text{pre}} &\triangleq \sigma_1^2\left(1+\sigma_1^2\boldsymbol{\phi}_{n^{\text{pre}}}^H\boldsymbol{b}_{n^{\text{pre}}}^{\text{pre}}\right)^{-1}\end{aligned}$$

Then $\boldsymbol{b}_n$ can be further denoted as

$$\begin{aligned}\boldsymbol{b}_n &= \left(\boldsymbol{\varphi}\left(\boldsymbol{z}^{\text{pre}}\right)^{-1} - \beta_{n^{\text{pre}}}^{\text{pre}}\boldsymbol{b}_{n^{\text{pre}}}^{\text{pre}}\left(\boldsymbol{b}_{n^{\text{pre}}}^{\text{pre}}\right)^H\right)\boldsymbol{\phi}_n \\ &= \boldsymbol{b}_n^{\text{pre}} - \beta_{n^{\text{pre}}}^{\text{pre}}\boldsymbol{b}_{n^{\text{pre}}}^{\text{pre}}\left(\boldsymbol{b}_{n^{\text{pre}}}^{\text{pre}}\right)^H\boldsymbol{\phi}_n \end{aligned} \tag{29}$$

According to (24) and (28), we can compute the second term of (20) as

$$\begin{aligned}\ln\det\big(\boldsymbol{\varphi}(\boldsymbol{z}_n)\big) &= \ln\det\big(\boldsymbol{\varphi}(\boldsymbol{z}) + \sigma_1^2\boldsymbol{\phi}_n\boldsymbol{\phi}_n^H\big) \\ &= \ln\left(\det\big(\boldsymbol{\varphi}(\boldsymbol{z})\big)\det\big(\boldsymbol{I}_M + \sigma_1^2\boldsymbol{\varphi}(\boldsymbol{z})^{-1}\boldsymbol{\phi}_n\boldsymbol{\phi}_n^H\big)\right) \\ &= \ln\big(\det\boldsymbol{\varphi}(\boldsymbol{z})\big)\big(1+\sigma_1^2\boldsymbol{\phi}_n^H\boldsymbol{\varphi}(\boldsymbol{z})^{-1}\boldsymbol{\phi}_n\big) \\ &= \ln\det\big(\boldsymbol{\varphi}(\boldsymbol{z})\big) + \ln\big(1+\sigma_1^2\boldsymbol{\phi}_n^H\boldsymbol{\varphi}(\boldsymbol{z})^{-1}\boldsymbol{\phi}_n\big) \\ &= \ln\det\big(\boldsymbol{\varphi}(\boldsymbol{z})\big) - \ln\left(\frac{\beta_n}{\sigma_1^2}\right). \end{aligned} \tag{30}$$

The third term of (20) is

$$\begin{aligned}\boldsymbol{y}^H\boldsymbol{\varphi}(\boldsymbol{z}_n)^{-1}\boldsymbol{y} &= \boldsymbol{y}^H\big(\boldsymbol{\varphi}(\boldsymbol{z})^{-1} - \beta_n\boldsymbol{b}_n\boldsymbol{b}_n^H\big)\boldsymbol{y} \\ &= \boldsymbol{y}^H\boldsymbol{\varphi}(\boldsymbol{z})^{-1}\boldsymbol{y} - \beta_n|\boldsymbol{y}^H\boldsymbol{b}_n|^2. \end{aligned} \tag{31}$$

The fourth term of (20) is

$$\begin{aligned}\|\boldsymbol{z}_n\|_0\ln\frac{p_1}{1-p_1} &= (\|\boldsymbol{z}\|_0+1)\ln\frac{p_1}{1-p_1} \\ &= \|\boldsymbol{z}\|_0\ln\frac{p_1}{1-p_1} + \ln\frac{p_1}{1-p_1}. \end{aligned} \tag{32}$$

Finally we substitute (30), (31) and (32) into (20), we have

$$\begin{aligned}v(\boldsymbol{z}_n) =& -\frac{N}{2}\ln 2\pi - \frac{1}{2}\left(\ln\det\big(\boldsymbol{\varphi}(\boldsymbol{z})\big) - \ln\frac{\beta_n}{\sigma_1^2}\right) \\ &- \frac{1}{2}\big(\boldsymbol{y}^H\boldsymbol{\varphi}(\boldsymbol{z})^{-1}\boldsymbol{y} - \beta_n|\boldsymbol{y}^H\boldsymbol{b}_n|^2\big) + \|\boldsymbol{z}\|_0\ln\frac{p_1}{1-p_1} \\ &+ \ln\frac{p_1}{1-p_1} + N\ln(1-p_1) \\ =& \, v(\boldsymbol{z}) + \Delta_n(\boldsymbol{z}_n). \end{aligned} \tag{33}$$

Algorithm 1 Fast Bayesian Matching Pursuit

Input: $\boldsymbol{y}$, M, N, D, P_0, p_1, σ_1, σ_n, $\boldsymbol{\phi}$
Obtain an estimate of the sparsity $\hat{K}$ via (43).
Set $\boldsymbol{w} \leftarrow \boldsymbol{0}^{\hat{K}D}$, $\boldsymbol{p}_e \leftarrow \boldsymbol{0}^{\hat{K}D}$, $\boldsymbol{H}_e \leftarrow \boldsymbol{0}^{N\times\hat{K}D}$.
Obtain $v(\boldsymbol{0})$ via (23).
for $n = 1, ..., N$ **do**
  Compute $\boldsymbol{b}_n$ and β_n via (26) and (27), respectively.
  Compute $\Delta_n(\boldsymbol{z}_n)$ via (34).
  $v(\boldsymbol{z}_n) \leftarrow v(\boldsymbol{0}) + \Delta_n(\boldsymbol{z}_n)$
end for
for $d = 1 : D$ **do**
  $\Omega \leftarrow \varnothing$, $\Psi \leftarrow \{1, 2, ...N\}$.
  for $k = 1 : \hat{K}$ **do**
    if $k = 1$ **then**
      $n^* \leftarrow$ the index corresponding to the dth largest value in $\{v(\boldsymbol{z}_n), n \in \Psi\}$.
    else
      $n^* \leftarrow \arg\max_{i\in\Psi} v(\boldsymbol{z}_i)$.
    end if
    $\Omega \leftarrow \Omega\bigcup\{n^*\}$, $\Psi \leftarrow \{1, 2, ...N\}\setminus\Omega$.
    $\boldsymbol{B}_\Omega \leftarrow \{\boldsymbol{b}_i, i \in \Omega\}$.
    $\boldsymbol{H}_e[\Omega, (d-1)\hat{K}+k] = \sigma_1^2\boldsymbol{B}_\Omega^H\boldsymbol{y}$.
    $w[(d-1)\hat{K}+k] \leftarrow v(\boldsymbol{z}_{n^*})$.
    Update $\boldsymbol{b}_i$ by $\boldsymbol{b}_i \leftarrow \boldsymbol{b}_i - \beta_{n^*}\boldsymbol{b}_{n^*}\boldsymbol{b}_{n^*}^H\boldsymbol{\phi}_i, i \in \Psi$.
    Update $\beta_i, i \in \Psi$ via (27).
    Update $\Delta_n(\boldsymbol{z}_n)$ via (34).
    Update $v(\boldsymbol{z}_n)$ by $v(\boldsymbol{z}_n) \leftarrow v(\boldsymbol{z}_{n^*}) + \Delta_n(\boldsymbol{z}_n)$.
  end for
end for
Compute each entry of $\boldsymbol{p}_e$ via (39).
Compute $\mathrm{E}\{\boldsymbol{h}|\boldsymbol{y}\} = \boldsymbol{H}_e\boldsymbol{p}_e$.
Output: $\mathrm{E}\{\boldsymbol{h}|\boldsymbol{y}\}$.

Now we can obtain $\Delta_n(\boldsymbol{z}_n)$ based on (22) as

$$\Delta_n(\boldsymbol{z}_n) = \frac{1}{2}\ln\left(\frac{\beta_n}{\sigma_1^2}\right) + \frac{1}{2}\beta_n|\boldsymbol{y}^H\boldsymbol{b}_n|^2 + \ln\frac{p_1}{1-p_1}. \tag{34}$$

C. AMMSE estimation

In Section III-B, we provide the estimation of $\boldsymbol{z}$. Now we further estimate $\boldsymbol{h}$ based on the estimation of $\boldsymbol{z}$.

We define $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, ...\boldsymbol{b}_N]$ which is a matrix formed by the columns from $\boldsymbol{b}_1$ to $\boldsymbol{b}_N$. According to (26), we have

$$\boldsymbol{B} = \boldsymbol{\varphi}(\boldsymbol{z})^{-1}\boldsymbol{\phi}. \tag{35}$$

Based on (14), we define

$$\boldsymbol{h}_e = \sigma_1^2\boldsymbol{\phi}_{\boldsymbol{z}}^H\boldsymbol{\varphi}(\boldsymbol{z})^{-1}\boldsymbol{y} = \sigma_1^2\boldsymbol{B}_z^H\boldsymbol{y} \tag{36}$$

where $\boldsymbol{B}_z$ is a submatrix formed by the columns indexed by $\boldsymbol{z}$ from $\boldsymbol{B}$. According to (20), we have

$$\mathcal{P}(\boldsymbol{y}|\boldsymbol{z})\mathcal{P}(\boldsymbol{z}) = e^{v(\boldsymbol{z})}. \tag{37}$$

Substituting (37) into (19), we have

$$\mathcal{P}(\boldsymbol{z}|\boldsymbol{y}) = \frac{e^{v(\boldsymbol{z})}}{\sum_{\boldsymbol{z}'\in\Lambda}e^{v(\boldsymbol{z}')}} \approx \frac{e^{v(\boldsymbol{z})}}{\sum_{\boldsymbol{z}'\in\Lambda^*}e^{v(\boldsymbol{z}')}}. \tag{38}$$

We define

$$p_e \triangleq \frac{e^{v(\boldsymbol{z})}}{\sum_{\boldsymbol{z}' \in \Lambda^*} e^{v(\boldsymbol{z}')}}. \tag{39}$$

Finally, the AMMSE estimation of $\boldsymbol{h}$ in (18) can be rewritten as

$$\hat{\boldsymbol{h}}_{\text{AMMSE}} = \sum_{\Lambda^*} \boldsymbol{h}_e p_e. \tag{40}$$

D. Estimation of the sparsity

Since the sparsity of the channel, K, is usually known, we have to obtain an estimate of K, denoted as $\hat{K}$.

Note that $\|\boldsymbol{z}\|_0$ obeys the Binomial distribution with the mean of Np_1 and the variance of $Np_1(1-p_1)$. If $Np_1 > 5$, the Binomial distribution can be well approximated by the Gaussian distribution with the same mean and variance. Given $\hat{K}$, we define

$$\begin{aligned} P_0 &\triangleq \mathcal{P}(\|\boldsymbol{z}\|_0 > \hat{K}) = 1 - \mathcal{P}(\|\boldsymbol{z}\|_0 \leq \hat{K}) \\ &= 1 - \boldsymbol{\Phi}\left(\frac{\hat{K} - Np_1}{\sqrt{2Np_1(1-p_1)}}\right) \\ &= \frac{1}{2}\text{erfc}\left(\frac{\hat{K} - Np_1}{\sqrt{2Np_1(1-p_1)}}\right) \end{aligned} \tag{41}$$

where

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dx = \frac{1}{2}\left[1 + \text{erf}(\frac{x}{\sqrt{2}})\right]$$

denote cumulative distribution function of standard Gaussian distribution.

We set the number of inner iterations $\hat{K}$ a little bigger than the expectation $\text{E}\{\|\boldsymbol{z}\|_0\} = Np_1$, so the probability that real sparsity is bigger than estimated sparsity, i.e., $\mathcal{P}(\|\boldsymbol{z}\|_0) > \hat{K}$ will be little. The probability is

$$\mathcal{P}_0 = \mathcal{P}(\|\boldsymbol{z}\|_0 > \hat{K}) = \frac{1}{2}\text{erfc}\left(\frac{\hat{K} - Np_1}{\sqrt{2Np_1(1-p_1)}}\right) \tag{42}$$

so $\hat{K}$ can be derived as follows

$$\hat{K} = \left\lceil \text{erfc}^{-1}(2P_0)\sqrt{2Np_1(1-p_1)} + Np_1 \right\rceil. \tag{43}$$

In FBMP, we need to set P_0, e.g. $P_0 = 0.01$, then we can compute $\hat{K}$ through (43).

We summarize the above procedures in Algorithm 1. At first, we obtain an estimate of the sparsity $\hat{K}$ via (43). We set $\boldsymbol{w}$ and $\boldsymbol{p}_e$ to be zero vectors. We set $\boldsymbol{H}_e$ to be a zero matrix to store the results of $\boldsymbol{h}_e$ for all iterations. Then we obtain $v(\boldsymbol{0})$ via (23). From step 5 to step 9, we obtain $\{v(\boldsymbol{z}_l), l = 1, 2 \ldots, N\}$. From step 10 to step 27, we use D outer-loop iterations, with each iteration using the $d(d = 1, 2, \ldots, D)$th largest value of $\{v(\boldsymbol{z}_l), l = 1, 2 \ldots, N\}$ as the starting value for the inner-loop iterations from step 18 to step 26. Different starting value may lead to different final results, where the weighted average based on the probability will be computed in the end. All steps follow our precedent derivations, e.g., step 20 is based on (36) and step 22 is based on (29). After the outer-loop iterations are finished, based on (39) we can compute each entry of $\boldsymbol{p}_e$, denoted as $p_e(l),\ l = 1, 2, \ldots, \hat{K}D$ by the following equation i.e.,

$$\boldsymbol{p}_e(l) = \frac{\exp(w[l])}{\sum_{k=1}^{\hat{k}D} \exp(w[k])},\ 1 \leq l \leq \hat{K}D.$$

At last, we use $\boldsymbol{H}_e$ and $\boldsymbol{p}_e$ to compute $\text{E}\{\boldsymbol{h}|\boldsymbol{y}\}$ via (40), i.e., $\text{E}\{\boldsymbol{h}|\boldsymbol{y}\} = \boldsymbol{H}_e\boldsymbol{p}_e$.

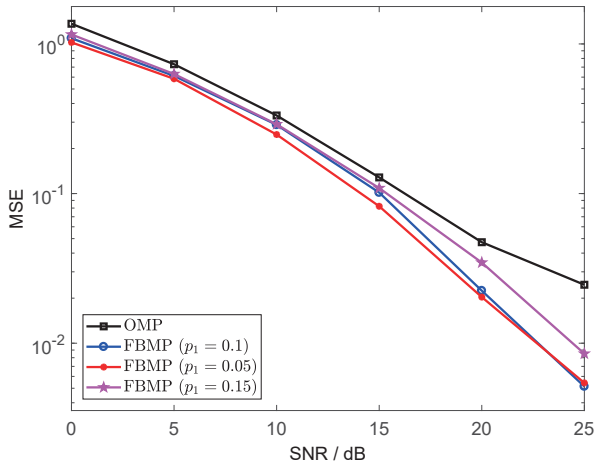

Fig. 1. Comparisons of MSE performance for FBMP and OMP.

IV. Simulation Results

For multicarrier UWA communications, we consider an OFDM system with $N_p = 512$ subcarriers and $M = 30$ pilot subcarriers. QPSK is employed for modulation. The length of UWA CIR sequence is set to $N = 100$ and the number of channel taps is set to be $K = 10$. We set $P_0 = 0.01$ and $D = 5$. We randomly generate the UWA channel 1000 times to compare the performance of OMP and FBMP. We set $p_1 = 0.05$, $p_1 = 0.1$ and $p_1 = 0.15$, which correspond to $\hat{K} = 12$, $\hat{K} = 19$ and $\hat{K} = 26$. In Fig. 1 and Fig. 2, we compare OMP with FBMP in terms of mean square error (MSE) and bit error rate (BER).

As shown in Fig. 1, FBMP outperforms OMP, especially when SNR is more than 15dB. The performance of $p_1 = 0.05$ is similar to that of $p_1 = 0.1$, which is better than that of $p_1 = 0.15$. Since $\hat{K} = 12$ when $p_1 = 0.05$ is closest to the genuine $K = 10$, the performance of $p_1 = 0.05$ is expected to be the best. However, some small errors of $\hat{K}$ may not be vital for the performance. Although $\hat{K} = 19$ when $p_1 = 0.1$ is a bit far from the genuine $K = 10$, the performance of $p_1 = 0.05$ is similar to that of $p_1 = 0.1$.

As shown in Fig. 2, we also compare the BER performance for FBMP and OMP. To achieve the same BER of 0.016, FBMP can save around 4dB SNR compared to OMP. Since in general, BER around 0.01 can be completely eliminated by additionally including channel encoding and decoding, FBMP can significantly improve the UWA channel estimation performance compared with OMP.

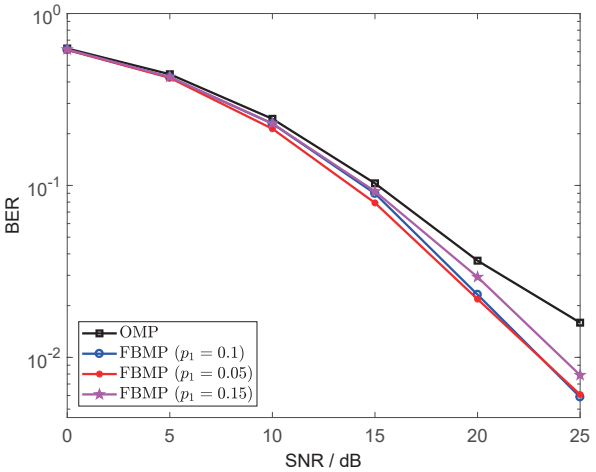

Fig. 2. Comparisons of BER performance for FBMP and OMP.

V. Conclusions

In this paper, we have studied the sparse channel estimation based on FBMP. Since UWA channel can usually be modeled as a Bernoulli-Gaussian mixture random process, we have estimated the multipath by exploiting the information of channel model under Bayesian frame. We first estimate the the position of the multipath taps, and then estimate the coefficient of multipath taps. Regarding the complexities for Bayesian estimation, we have studied the fast implementation and optimized the stop condition for the presented iterative algorithm. Simulation results show that FBMP outperforms OMP. In order to achieve the same BER of 0.016, FBMP can save around 4dB of SNR compared to OMP.

Acknowledgment

This work is supported in part by National Natural Science Foundation of China under Grant 61302097, Natural Science Foundation of Jiangsu Province under Grant BK20161428, and National Science and Technology Major Project of China under Grant 2016ZX03001016-003.

References

[1] C. R. Berger, S. Zhou, J. C. Preisig, and P. Willett, "Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1708–1721, Mar. 2010.

[2] E. Panayirci, H. Senol, M. Uysal, and H. V. Poor, "Sparse channel estimation and equalization for OFDM-based underwater cooperative systems with amplify-and-forward relaying," *IEEE Trans. Signal Process.*, vol. 64, no. 1, pp. 214–228, Jan. 2016.

[3] K. Pelekanakis and M. Chitre, "Robust equalization of mobile underwater acoustic channels," *IEEE J. Ocean. Eng.*, vol. 40, no. 4, pp. 775–784, Sep. 2015.

[4] C. Qi, X. Wang, and Y. Huang, "Selection of nonzero taps for sparse linear equalizer," in *Proc. VTC-Spring*, Nanjing, China, May 2016, pp. 1–5.

[5] Z. Yang and Y. R. Zheng, "Iterative channel estimation and turbo equalization for multiple-input multiple-output underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 41, no. 1, pp. 232–242, Feb. 2016.

[6] C. Qi, X. Wang, and L. Wu, "Underwater Acoustic Channel Estimation based on Sparse Recovery Algorithms," *IET Signal Process.*, vol. 5, no. 8, pp. 739–747, Dec. 2011.

[7] C. Qi, L. Wu, and X. Wang, "Underwater acoustic channel estimation via complex homotopy," in *Proc. IEEE International Conference on Communications (ICC)*, Ottawa, Canada, 2012, pp. 3821–3825.

[8] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, Aug. 2008.

[9] P. Schniter, L. C. Potter, and J. Ziniel, "Fast Bayesian matching pursuit," in *Proc. ITA*, San Diego, California, Jan. 2008, pp. 326–333.

[10] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Multicarrier communication over underwater acoustic channels with nonuniform Doppler shifts," *IEEE J. Ocean. Eng.*, vol. 33, no. 2, pp. 198–209, Apr. 2008.