Regularized Multipath Matching Pursuit for Sparse Channel Estimation in Millimeter Wave Massive MIMO System

Jun Tao, *Student Member, IEEE*, Chenhao Qi, *Senior Member, IEEE*, and Yongming Huang, *Senior Member, IEEE*

***Abstract*—Sparse channel estimation is investigated for millimeter wave massive MIMO systems, where a base station equipped with a uniform planar array serves several single-antenna users. At first, the 2-D multiuser channel estimation is formulated as several sparse recovery problems. Then a regularized multipath matching pursuit (RMMP) algorithm is proposed for sparse channel estimation. Compared to the existing multipath matching pursuit (MMP) algorithm, a regularization step is introduced in RMMP to screen the candidate paths, which can reduce the computational complexity as well as the storage overhead. Simulation results show that the proposed RMMP algorithm outperforms the existing orthogonal matching pursuit and orthogonal least squares algorithms. In particular, RMMP has the same sparse channel estimation performance as MMP while the computational complexity of the former is much lower than the latter.**

***Index Terms*—Millimeter wave communications, channel estimation, sparse recovery, massive MIMO, compressed sensing.**

I. Introduction

Millimeter wave (mmWave) massive multi-input multi-output (MIMO) [1], which is capable of achieving significant increase in data rate based on its wider bandwidth, has been considered as one of key techniques for next generation wireless communications [2]. The increased frequency leads to decreased wavelength and therefore the feasibility of integrating a large antenna array into the small dimension [3]. On the other hand, the large antenna array of massive MIMO achieving directional beamforming can provide sufficient gain to compensate for severe path loss during mmWave signal transmission.

However, the implementation of mmWave massive MIMO communications in practice is not a trivial task. One challenge is the efficient channel estimation to acquire channel state information (CSI). Conventional MIMO channel estimation methods may not be directly applicable in mmWave massive MIMO systems because of its substantially greater number of antennas and thus the unaffordable channel estimation complexity. Regarding the sparse feature of mmWave MIMO channels [3], compressed sensing (CS) techniques can be leveraged to effectively estimate mmWave channels [1]. In [4], the orthogonal matching pursuit (OMP) algorithm is used to perform the sparse channel estimation in downlink multiuser mmWave massive MIMO systems. In [5], the basis pursuit denoising (BPDN) algorithm is adopted for sparse spatial channel recovery in outdoor mmWave massive MIMO systems.

In this letter, we consider the sparse channel estimation for mmWave massive MIMO systems, where a base station (BS) equipped with an uniform planar array (UPA) serves several single-antenna users. We first formulate the two dimensional multiuser channel estimation as several sparse recovery problems. Then we propose a regularized multipath matching pursuit (RMMP) algorithm for sparse channel estimation. Compared to the existing multipath matching pursuit (MMP) algorithm [6], we introduce a regularization step to screen the candidate paths, which can reduce the computational complexity as well as the storage overhead.

The notations are defined as follows. Symbols for matrices (upper case) and vectors (lower case) are in boldface. $(\cdot)^{-1}$, $\boldsymbol{I}_L$, $\mathbb{C}^{M\times N}$, $\otimes$, $\mathrm{vec}(\cdot)$, $\boldsymbol{A}(l)$ and $\mathcal{CN}$, denote the matrix inverse, identity matrix of size L, set of $M \times N$ complex-valued matrices, kronecker product, vectorization, lth column of a matrix $\boldsymbol{A}$ and complex Gaussian distribution, respectively.

II. Problem Formulation

We consider an mmWave massive MIMO system which includes a BS and U single-antenna users. The BS is equipped with an UPA which has M_v antennas in vertical and M_h antennas in horizontal, resulting in totally $M \triangleq M_v M_h$ antennas. According to the popular 3D Saleh-Valenzuela mmWave channel model [1], the channel between the BS and the $u(u = 1, 2, \ldots, U)$th user, denoted as $\boldsymbol{h}_u \in \mathbb{C}^M$, can be expressed as

$$\boldsymbol{h}_u = \sqrt{\frac{M}{L_u}} \sum_{i=1}^{L_u} g_{u,i} \boldsymbol{\alpha}(M_h, \theta_{u,i}) \otimes \boldsymbol{\alpha}(M_v, \varphi_{u,i}) \tag{1}$$

where L_u and $g_{u,i}$ denote the total number of resolvable paths and the channel gain of the ith path, respectively. The physical azimuth and physical elevation of the ith path are denoted as $\Theta_{u,i}$ and $\Phi_{u,i}$, respectively. We define $\theta_{u,i} = \frac{d_h}{\lambda}\sin\Theta_{u,i}$ and $\varphi_{u,i} = \frac{d_v}{\lambda}\sin\Phi_{u,i}$, where d_h and d_v denote the antenna interval in horizontal and in vertical, respectively. For simplicity, we set $d_v = d_h = \lambda/2$, where λ is the wavelength of mmWave signal. Both $\theta_{u,i}$ and $\varphi_{u,i}$ obey the uniform distribution [−0.5, 0.5]. The steering vector $\boldsymbol{\alpha}(M, \theta)$ is then

Manuscript received July 5, 2018; revised August 3, 2018; accepted August 5, 2018. Date of publication August 14, 2018; date of current version February 19, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61871119 and Grant 61701307, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20161428. The associate editor coordinating the review of this paper and approving it for publication was C.-K. Wen. *(Corresponding author: Chenhao Qi.)*

The authors are with the School of Information Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: qch@seu.edu.cn).

Digital Object Identifier 10.1109/LWC.2018.2865472

defined as

$$\boldsymbol{\alpha}(M,\theta) = \frac{1}{\sqrt{M}}\left[1, e^{-j2\pi\theta}, \ldots, e^{-j2\pi\theta(M-1)}\right]^T. \tag{2}$$

We consider uplink channel estimation in mmWave massive MIMO systems, where the users send mutually orthogonal pilot sequences and the BS uses the received pilot sequences for pilot-assisted channel estimation. We suppose that each user repeatedly sends a pilot sequence with length of U for K times during the uplink channel estimation. Note that we may simply set $K = 1$. But if we use larger K indicating more pilots are used, we can improve the channel estimation performance. The channel is supposed to be constant during $V \triangleq KU$ time slots. We define a pilot matrix $\boldsymbol{P} \in \mathbb{C}^{U\times U}$, which includes the U orthogonal pilot sequences sent from U users. For the $k(k = 1, 2, \ldots, K)$th repetitive pilot transmission, the BS employs a phase shifter network $\boldsymbol{F}_k \in \mathbb{C}^{U\times M}$ for uplink signal combining. Normally the phase shifter network works as an analog precoder for downlink transmission. But for uplink transmission, it works as a signal combiner. Then the pilot sequences received by the BS are denoted as

$$\boldsymbol{Y}_k = \boldsymbol{F}_k\boldsymbol{H}\boldsymbol{P} + \boldsymbol{F}_k\boldsymbol{N}_k \tag{3}$$

where $\boldsymbol{H} \triangleq [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_U] \in \mathbb{C}^{M\times U}$ is the mmWave MIMO uplink channel matrix from the users to the BS. $\boldsymbol{N}_k$ is an additive white Gaussian noise (AWGN) matrix, where each entry of $\boldsymbol{N}_k$ independently obeys complex Gaussian distribution with zero mean and variance of σ^2. Due to the orthogonality of $\boldsymbol{P}$, i.e., $\boldsymbol{P}\boldsymbol{P}^H = \boldsymbol{I}_U$, we multiply $\boldsymbol{Y}_k$ with $\boldsymbol{P}^H$ obtaining

$$\boldsymbol{R}_k \triangleq \boldsymbol{Y}_k\boldsymbol{P}^H = \boldsymbol{F}_k\boldsymbol{H} + \widetilde{\boldsymbol{N}}_k \tag{4}$$

where $\widetilde{\boldsymbol{N}}_k \triangleq \boldsymbol{F}_k\boldsymbol{N}_k\boldsymbol{P}^H$. We define

$$\boldsymbol{G} = \boldsymbol{D}(M_h) \otimes \boldsymbol{D}(M_v) \tag{5}$$

where $\boldsymbol{D}(M)$ is essentially the DFT matrix of size M as

$$\boldsymbol{D}(M) = \left[\boldsymbol{\alpha}(M,0), \boldsymbol{\alpha}\Big(M, \frac{1}{M}\Big), \ldots, \boldsymbol{\alpha}\Big(M, \frac{(M-1)}{M}\Big)\right]^H. \tag{6}$$

Then the mmWave beamspace channel matrix can be denoted as

$$\boldsymbol{H}^b \triangleq \boldsymbol{G}\boldsymbol{H} \tag{7}$$

which exhibits the sparsity due to the limited scatters in mmWave channel [7]. Since $\boldsymbol{G}^H\boldsymbol{G} = \boldsymbol{I}_M$, we have

$$\boldsymbol{H} = \boldsymbol{G}^H\boldsymbol{H}^b. \tag{8}$$

Substituting (8) into (4), we have

$$\boldsymbol{R}_k = \boldsymbol{F}_k\boldsymbol{G}^H\boldsymbol{H}^b + \widetilde{\boldsymbol{N}}_k. \tag{9}$$

Define $\boldsymbol{C}_k \triangleq \boldsymbol{F}_k\boldsymbol{G}^H$. Then (9) can be rewritten as

$$\boldsymbol{R}_k = \boldsymbol{C}_k\boldsymbol{H}^b + \widetilde{\boldsymbol{N}}_k. \tag{10}$$

After pilot sequences are repeatedly transmitted for K times by all the users, we can combine $\boldsymbol{R}_k, k = 1, 2, \ldots, K$ together into a matrix $\boldsymbol{R}$ which has V rows and U columns as

$$\boldsymbol{R} = [\boldsymbol{R}_1^T, \boldsymbol{R}_2^T, \ldots, \boldsymbol{R}_K^T]^T = \boldsymbol{C}\boldsymbol{H}^b + \widetilde{\boldsymbol{N}} \tag{11}$$

where

$$\boldsymbol{C} \triangleq [\boldsymbol{C}_1^T, \boldsymbol{C}_2^T, \ldots, \boldsymbol{C}_K^T]^T \in \mathbb{C}^{V\times M}, \tag{12}$$

$$\widetilde{\boldsymbol{N}} \triangleq [\widetilde{\boldsymbol{N}}_1^T, \widetilde{\boldsymbol{N}}_2^T, \ldots, \widetilde{\boldsymbol{N}}_K^T]^T \in \mathbb{C}^{V\times U}. \tag{13}$$

The $u(u = 1, 2, \ldots, U)$th column of $\boldsymbol{R}$, denoted as $\boldsymbol{r}_u$, can be expressed as

$$\boldsymbol{r}_u = \boldsymbol{C}\boldsymbol{h}_u^b + \widetilde{\boldsymbol{n}}_u \tag{14}$$

where $\boldsymbol{h}_u^b$ and $\widetilde{\boldsymbol{n}}_u$ are the uth column of $\boldsymbol{H}^b$ and $\widetilde{\boldsymbol{N}}$, respectively. In fact, $\boldsymbol{h}_u^b = \boldsymbol{G}\boldsymbol{h}_u$ is usually named as a beamspace channel vector from the uth user to the BS. Note that the mmWave beamspace channel is essentially sparse [7], i.e., $\boldsymbol{h}_u^b$ is sparse, indicating that most entries of $\boldsymbol{h}_u^b$ are zero while only a small number of entries are nonzero. In this context, we can use $\boldsymbol{r}_u$ and $\boldsymbol{C}$ to reconstruct $\boldsymbol{h}_u^b$ via sparse recovery algorithms, including OMP [8], OLS [9], MMP and etc. Although MMP has high precision of sparse recovery, the computational complexity is also high. Moreover, MMP requires the sparsity to be known. Thus we propose the RMMP algorithm, which does not need the knowledge of the sparsity and has lower computational complexity than MMP.

Algorithm 1 Regularized Multipath Matching Pursuit

```
Input: $\boldsymbol{C}$, $\boldsymbol{r}_u$, $\sigma$ and $N$.
Initialization: $\boldsymbol{S} \leftarrow \emptyset$, $\boldsymbol{Z} \leftarrow \boldsymbol{r}_u$, $\zeta \leftarrow \|\boldsymbol{r}_u\|_2$ and $N_c \leftarrow 1$.
while $\zeta > \sigma^2$ do
    Set $\boldsymbol{T} \leftarrow \emptyset$, $\boldsymbol{B} \leftarrow \emptyset$ and $N_w \leftarrow 0$.
    for $i = 1{:}N_c$ do
        Obtain $\boldsymbol{\Omega}$ via (19). $N_R \leftarrow \|^\bullet\|_0$.
        for $n = 1{:}N_R$ do
            $\boldsymbol{t} \leftarrow \boldsymbol{S}(i) \cup \boldsymbol{\Omega}(n)$.
            if $t \not\subset T$ then
                $\boldsymbol{T} \leftarrow [\boldsymbol{T}, \boldsymbol{t}]$. $N_w \leftarrow N_w + 1$.
                Obtain $\boldsymbol{b}$ via (20). $\boldsymbol{B} \leftarrow [\boldsymbol{B}, \boldsymbol{b}]$.
            end if
        end for
    end for
    $\boldsymbol{S} \leftarrow \boldsymbol{T}$. $\boldsymbol{Z} \leftarrow \boldsymbol{B}$. $N_c \leftarrow N_w$.
    Obtain $i_{\min}$ via (22).
    $\zeta \leftarrow \|\boldsymbol{Z}(i_{\min})\|_2^2$.
end while
$\boldsymbol{s}_{\min} \leftarrow \boldsymbol{S}(i_{\min})$.
Compute $\hat{\boldsymbol{h}}_u^b \leftarrow \boldsymbol{C}^\dagger(\boldsymbol{s}_{\min})\boldsymbol{r}_u$.
Output: $\hat{\boldsymbol{h}}_u^b$.
```

III. Regularized Multipath Matching Pursuit

As shown in **Algorithm 1**, we propose the RMMP algorithm. Compared to the existing MMP algorithm [6], we first remove the requirement that the sparsity should be known, and then reduce the computational complexity of MMP by introducing a regularization step to reduce the number of candidate atoms, where the atom is essentially a column of $\boldsymbol{C}$ according to the naming rule of CS community.

The input to **Algorithm 1** is four parameters including $\boldsymbol{C}$, $\boldsymbol{r}_u$, σ and N, where N ($N \leq M$) representing the number of

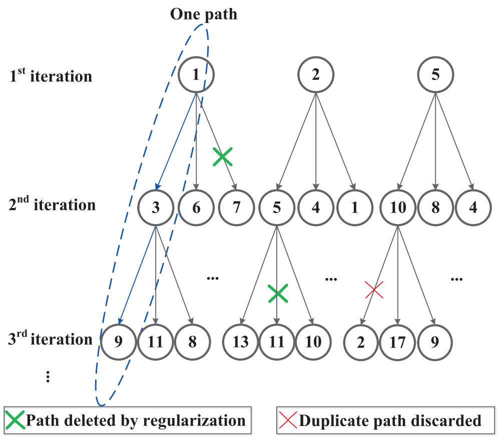

Fig. 1. Illustration of the RMMP algorithm.

paths has already been defined in the MMP algorithm. Unlike the OMP algorithm that only keeps the best candidate atom having the largest projection with the current residue at each iteration, RMMP as well as MMP keeps N best candidate atoms, resulting in N different set of atoms including several atoms already selected in the previous iterations and a candidate atom selected in the current iteration. As shown in Fig. 1, three paths are deleted, including $\{1, 7\}$, $\{2, 5, 11\}$ and $\{5, 10, 2\}$. The first two are deleted because of the regularization, and the last one is discarded because of repetition. At the lth iteration, we initialize a temporary matrix $\boldsymbol{T}$, where $\boldsymbol{T}$ is used to store the sequentially generated surviving paths based on $\boldsymbol{S}$ for the $(l+1)$th iteration. For example, $\boldsymbol{S}$ has 8 columns at the 2nd iteration, while $\boldsymbol{T}$ has 22 columns accordingly.

At first, we make initialization for $\boldsymbol{S}$, $\boldsymbol{Z}$, ζ and N_c, where ζ denotes the power of the residue and N_c denotes the number of surviving path in the previous iteration. If the power of the residue is larger than the noise power σ^2, we preform the iteration.

At each iteration, we initialize two temporary matrices $\boldsymbol{T}$ and $\boldsymbol{B}$, where $\boldsymbol{T}$ is used to store the sequentially generated surviving paths based on $\boldsymbol{S}$ for the next iteration, and $\boldsymbol{B}$ is used to store the residue of each surviving path corresponding to $\boldsymbol{T}$. For the $i(i = 1, 2, \dots, N_c)$th surviving path, we first make a projection of the current residue $\boldsymbol{Z}(i)$ on the dictionary matrix $\boldsymbol{C}$ as

$$\boldsymbol{q} = \boldsymbol{C}^H \boldsymbol{Z}(i). \tag{15}$$

Then we sort the set $\{|q(1)|, |q(2)|, \dots, |q(M)|\}$ in descending order, obtaining a new set $\boldsymbol{q}^{\text{sort}}$. We denote the set of column indices of $\boldsymbol{C}$ corresponding to $q^{\text{sort}}(1)$, $q^{\text{sort}}(2), \dots, q^{\text{sort}}(N)$ as $\boldsymbol{\Lambda}$. In the existing MMP algorithm [6], each entry of $\boldsymbol{\Lambda}$ is considered to be a candidate atom. However, the computational complexity is high if all the entries of $\boldsymbol{\Lambda}$ are kept. Now we propose to use a step of regularization to screen the candidate paths, which is essentially to narrow the size of $\boldsymbol{\Lambda}$. The step of regularization can be expressed as

$$\begin{aligned} &\max_{\boldsymbol{\Gamma}} \sum_{l\in\boldsymbol{\Gamma}} q^{\text{sort}}(l) \\ &\text{s.t.}\ \ q^{\text{sort}}(i) \le \beta q^{\text{sort}}(j),\ \forall i,j\in\boldsymbol{\Gamma},\ \ i\ne j \\ &\qquad \boldsymbol{\Gamma}\subseteq\{1,2,\dots,N\}, \end{aligned} \tag{16}$$

where $\beta > 0$ is a scalable factor to determine the size of $\boldsymbol{\Lambda}$. Larger β leads to larger size of $\boldsymbol{\Lambda}$. To solve (16), we first compute N different sets as

$$\begin{aligned} \boldsymbol{\Gamma}_k = \{i \mid q^{\text{sort}}(i) \ge q^{\text{sort}}(k)/\beta,\ \ i = k, k+1, \dots, N\}, \\ k = 1, 2, \dots, N, \end{aligned} \tag{17}$$

and then obtain

$$\boldsymbol{\Gamma}_{\max} = \max_{\boldsymbol{\Gamma}_k} \sum_{l\in\boldsymbol{\Gamma}_k} q^{\text{sort}}(l),\ \ k = 1, 2, \dots, N. \tag{18}$$

We denote

$$\boldsymbol{\Omega} = \boldsymbol{\Lambda}(\boldsymbol{\Gamma}_{\max}) \tag{19}$$

which is essentially a subset of $\boldsymbol{\Lambda}$ with entries selected from $\boldsymbol{\Lambda}$ according to $\boldsymbol{\Gamma}_{\max}$. Compared to $\boldsymbol{\Lambda}$, the size of $\boldsymbol{\Omega}$ can be smaller and therefore leading to the reduction of the computational complexity as well as the storage overhead. As shown in Fig. 1, some paths are deleted after finishing the step of regularization.

Note that the same path repeatedly stored in $\boldsymbol{S}$ will cause the reduction of the efficiency with respect to the storage and computation. Therefore, we have to further screen the candidate paths by making duplicate check for each path. For each entry of $\boldsymbol{\Omega}$, we sequentially generate a new path as $t \leftarrow S(i) \cup \boldsymbol{\Omega}(n)$, indicated by Step 8 of **Algorithm 1**. We check if $\boldsymbol{t}$ is a duplicate path that already exists in $\boldsymbol{T}$. If $\boldsymbol{t}$ is not a duplicate path, i.e., $\boldsymbol{t} \not\subset \boldsymbol{T}$, we store it in $\boldsymbol{T}$ as $T \leftarrow [T, t]$ and increase the path counter N_w by one; otherwise, we discard $\boldsymbol{t}$. As shown in Fig. 1, $\{5, 10, 2\}$ is a duplicate path and should be discarded since there already exists $\{2, 5, 10\}$ in $\boldsymbol{T}$. As $\{5, 10, 2\}$ and $\{2, 5, 10\}$ lead to the same residue and the exactly same routine for the following iterations, it is meaningless to store $\{5, 10, 2\}$ in $\boldsymbol{T}$. If $\boldsymbol{t}$ is not a duplicate path, we compute the residue of $\boldsymbol{t}$ as

$$\boldsymbol{b} = \boldsymbol{r}_u - \boldsymbol{C}(\boldsymbol{t})\boldsymbol{C}^{\dagger}(\boldsymbol{t})\boldsymbol{r}_u \tag{20}$$

where $\boldsymbol{C}(\boldsymbol{t})$ represents a submatrix of $\boldsymbol{C}$ with columns selected according to $\boldsymbol{t}$, and

$$\boldsymbol{C}^{\dagger}(\boldsymbol{t}) = \left(\boldsymbol{C}^H(\boldsymbol{t})\boldsymbol{C}(\boldsymbol{t})\right)^{-1}\boldsymbol{C}^H(\boldsymbol{t}) \tag{21}$$

represents the pseudo inverse of $\boldsymbol{C}(\boldsymbol{t})$. Then we store $\boldsymbol{b}$ in $\boldsymbol{B}$ as $\boldsymbol{B} \leftarrow [\boldsymbol{B}, \boldsymbol{b}]$.

After all of N_c surviving paths are iteratively treated, we update ζ to be the minimum power of the residual of all surviving paths, i.e., $\zeta \leftarrow \|\boldsymbol{Z}(i_{\min})\|_2^2$, where $i_{\min}$ is obtained as

$$i_{\min} = \arg\min_{i=1,2,\dots,N_c} \|\boldsymbol{Z}(i)\|_2^2. \tag{22}$$

If $\zeta \le \sigma^2$, implying that it is almost impossible to distinguish the signal from the noise, we stop the iterations and select the path having the minimum power of the residual as $\boldsymbol{s}_{\min} \leftarrow \boldsymbol{S}(i_{\min})$. Finally, we output the estimate of the sparse channel $\boldsymbol{h}_u^b$, i.e., $\hat{\boldsymbol{h}}_u^b \leftarrow \boldsymbol{C}^{\dagger}(\boldsymbol{s}_{\min})\boldsymbol{r}_u$.

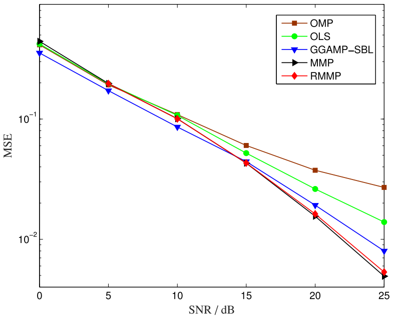

Fig. 2. MSE comparisons for different sparse recovery algorithms in terms of SNR.

TABLE I

COMPARISONS OF COMPUTATIONAL COMPLEXITY IN TERMS OF RUNNING TIME FOR DIFFERENT ALGORITHMS (IN 10^{-3} SECOND)

SNR(dB)	0	5	10	15	20	25
OMP	0.242	0.268	0.323	0.393	0.448	0.513
OLS	1.387	2.545	4.965	6.775	8.063	9.483
GGAMP-SBL	28.21	24.95	21.94	21.04	19.84	17.54
RMMP	9	46	160	377	772	1265
MMP	10	75	412	1637	4461	9233

IV. SIMULATION RESULTS

The considered mmWave massive MIMO system includes a BS equipped with $M_h = M_v = 8$ ($M = 64$) antennas and $U = 4$ single-antenna users. The number of resolvable paths in mmWave channel is set to be $L_u = 5$, while $g_{u,1} \sim \mathcal{CN}(0, 1)$ and $g_{u,i} \sim \mathcal{CN}(0, 0.01)$ for $i = 2, \ldots, 5$. For uplink channel estimation, we set $K = 4$ and take $V = UK = 16$ time slots to transmit pilot sequences for pilot assisted channel estimation. Each entry of the uplink phase shifter network $\boldsymbol{F}_k \in \mathbb{C}^{U\times M}$ is randomly selected from a candidate set $\{1/V, e^{-j\frac{2\pi}{M}}/V, \ldots, e^{-j\frac{2\pi(M-1)}{M}}/V\}$. We set $\beta = 2$ and $N = 3$ for RMMP.

As shown in 2, we compare the mean square error (MSE) for different sparse recovery algorithms in terms of signal-to-noise ratio (SNR). RMMP, MMP, GGAMP-SBL [10], OLS and OMP are included in the comparisons. It is seen that the performance of RMMP is almost the same as that of MMP. Both RMMP and MMP outperform GGAMP-SBL, OLS and OMP in high SNR region. The performance of GGAMP-SBL is better than RMMP in low SNR region because of the large EM iterations and the damping, while RMMP stops the iteration earlier due to the large noise power. To achieve the same MSE of 0.014, RMMP can save 4.5dB and 2dB of SNR compared to OLS and GGAMP-SBL, respectively. As shown in Table I, as SNR increases, the computational complexity in terms of running time for different algorithms also increases. The reason is that given the same signal power, as SNR increases, σ^2 decreases, which implies we have to run more iterations and therefore it takes longer time. It is seen from Table I that both OMP and OLS are much faster than GGAMP-SBL, RMMP and MMP. In particular, the running time of RMMP is much less than that of MMP, e.g., the ratio of the former over the latter is only 13.7% at SNR of 25dB. As shown in Table II, MSE comparisons in terms of different β have been provided where SNR $= 20$dB and $L_u = 5$. It is seen that as β increases, the MSE of RMMP decreases, where more atoms are selected with more surviving path to improve the MSE performance.

TABLE II

MSE COMPARISONS FOR RMMP IN TERMS OF β FOR RMMP

β	1.5	2	2.5	3
MSE	0.01506	0.01341	0.01288	0.01284

TABLE III

COMPARISONS OF STORAGE OVERHEAD FOR OMP, OLS, MMP AND RMMP

Number of iterations	OMP	OLS	MMP	RMMP
1st iteration	1	1	1	1
2nd iteration	1	1	3	3
3rd iteration	1	1	9	8
4th iteration	1	1	26	22
5th iteration	1	1	73	52

As shown in Table III, we provide a table comparing the storage overhead of OMP, OLS, MMP and RMMP. As the number of iterations increases, the difference between RMMP and MMP gets larger. The storage overhead of RMMP decreases around 30% compared with that of MMP in the 5th iteration.

V. CONCLUSION

In this letter, we have proposed a RMMP algorithm for sparse channel estimation. Future work will be continued with the focus on hybrid precoding following the sparse channel estimation in mmWave massive MIMO systems.

REFERENCES

[1] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436–453, Apr. 2016.

[2] P. Wang, Y. Li, L. Song, and B. Vucetic, "Multi-gigabit millimeter wave wireless communications for 5G: From fixed access to cellular networks," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 168–178, Jan. 2015.

[3] T. S. Rappaport *et al.*, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.

[4] A. Alkhateeb, G. Leus, and R. W. Heath, "Compressed sensing based multi-user millimeter wave systems: How many measurements are needed?" in *Proc. IEEE ICASSP*, Brisbane, QLD, Australia, Apr. 2015, pp. 2909–2913.

[5] D. E. Berraki, S. M. D. Armour, and A. R. Nix, "Application of compressive sensing in sparse spatial channel recovery for beamforming in mmWave outdoor systems," in *Proc. IEEE WCNC*, Istanbul, Turkey, Apr. 2014, pp. 887–892.

[6] S. Kwon, J. Wang, and B. Shim, "Multipath matching pursuit," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2986–3001, May 2014.

[7] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 831–846, Oct. 2014.

[8] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[9] J. Wen, J. Wang, and Q. Zhang, "Nearly optimal bounds for orthogonal least squares," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5347–5356, Oct. 2017.

[10] M. Al-Shoukairi, P. Schniter, and B. D. Rao, "A GAMP based low complexity sparse Bayesian learning algorithm," *IEEE Trans. Signal Process.*, vol. 66, no. 2, pp. 294–308, Jan. 2018.