Selection of Nonzero Taps for Sparse Linear Equalizer

Chenhao Qi, Xin Wang and Yongming Huang
Key Laboratory of Underwater Acoustic Signal Processing of Ministry of Education
School of Information Science and Engineering
Southeast University, Nanjing 210096, China
Email: qch@seu.edu.cn

***Abstract*—The selection of nonzero taps for sparse linear equalizer under the criterion of minimum mean square error (MMSE) is investigated. In some applications such as underwater acoustic communications, the computational resource in terms of the number of nonzero channel taps is given in existing channel equalizers. In this context, the joint determination of positions and weights of nonzero taps of sparse equalizer is considered and then formulated as a subset selection problem. A fast algorithm that uses two levels of loops is proposed to iteratively update each entry of the subset. The computational complexity of the proposed algorithm is analyzed. To make the work comprehensive, the sparse equalizer design where the number of nonzero taps is not given is also investigated. Simulation results show that around 60% equalizer taps can be saved with no more than 1dB of performance loss. Moreover, compared to the OMP algorithm, the proposed algorithm can save 33% equalizer taps achieving the same bit error rate (BER) of 10^{-3}.**

***Index Terms*—Compressed sensing (CS), channel equalizer, finite impulse response (FIR), sparse equalizer**

I. Introduction

The multipath propagation in wireless environment causes the inter-symbol interference (ISI), which usually needs the channel equalization at the receiver. For wireless channels with large delay spread, the number of equalizer taps may grow up to several hundreds, which increase the hardware complexity and energy consumption. The complexity of linear channel equalizers is proportional to the square of the number of nonzero taps in the equalizers. To reduce the complexity, one potential candidate is the use of sparse equalizer that has fewer nonzero taps than conventional channel equalizers.

Several methods for sparse equalizer design have already been presented. In [1], the determination of taps positions for sparse equalizer is considered. After the positions of nonzero taps are obtained, the weights of these nonzero taps are then figured out. In [2], a new framework for the design of sparse multi-input multi-output (MIMO) equalizer is proposed, where the orthogonal matching pursuit (OMP) algorithm is employed to determine the positions and weights of the nonzero taps. In [3], a method for designing sparse filters in the minimax sense using a mixture of reweighted ℓ_1-norm minimization and greedy iterations has been presented. In [4], [5], the sparse equalizer design under a quadratic constraint is developed and a branch-and-bound algorithm that uses backward selection is described with emphasis on its efficient implementation. Examples in wireless channel equalization and minimum-variance distortionless-response beamforming show that the backward selection algorithm in [4], [5] yields optimally sparse designs in many instances while also highlighting the benefits of sparse design. In [6], a low-complexity sparse frequency-domain equalizer exploits the sparsity observed in the graded index multimode fiber MIMO channel. In [7], a fractional-norm constrained blind adaptive algorithm is p-resented for sparse channel equalization, which essentially improves on the minimization of the constant modulus criteria by adding a sparsity-inducing ℓ_p-norm penalty.

In this paper, we study the selection of nonzero taps for sparse linear equalizer under the criterion of minimum mean square error (MMSE). Consider that in some applications, i.e., underwater acoustic communications, the computational resource in terms of the number of complex-valued multipliers for channel equalization is given. Instead of completely replacing the existing channel equalizers with new ones, a compromised choice is to improve the performance of the existing architecture by optimizing the tap positions and tap weights, where the optimization procedures are provided by additional block that can be connected to the existing equalizer. In this context where the number of nonzero taps is known, we formulate the joint determination of positions and weights of nonzero taps of sparse equalizer as a subset selection problem. We propose a fast algorithm that uses two levels of loops to iteratively update each entry of the subset. The computational complexity of the proposed algorithm is analyzed. Then we also investigate the sparse equalizer design where the number of nonzero taps is not given. In our simulations, we compare the performance of our proposed algorithm with the OMP algorithm used in [2].

The remainder of this paper is organized as follows. Section II formulates sparse equalizer design as a joint determination of positions and weights problem. Section III investigates the sparse equalizer design under two settings, one assuming the number of nonzero taps is known and the other assuming it is unknown. Simulation results are provided in Section IV. Finally, conclusions are provided in Section V.

The notations used in this paper are defined as follows. Symbols for matrices (upper case) and vectors (lower case)

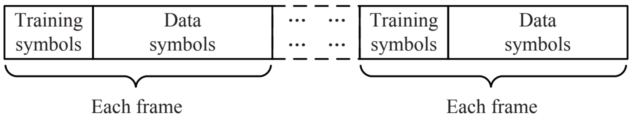

Fig. 1. The transmission structure considered in this paper.

are in boldface. $(\cdot)^T$, $(\cdot)^H$, diag$\{\cdot\}$, $\boldsymbol{I}_L$, $\|\boldsymbol{a}\|_2$, $\mathcal{CN}$ and $\emptyset$, denote the matrix transpose, conjugate transpose (Hermitian), the diagonal matrix, the identity matrix of size L, ℓ_2-norm of a vector $\boldsymbol{a}$, the complex Gaussian distribution and the empty set, respectively. $O(\cdot)$ denotes the order of complexity.

II. Problem formulation

We consider the wireless transmission in units of frames. As shown in Fig. 1, each frame with the length of N_f consists of N_t training symbols and N_d data symbols, where $N_f = N_t + N_d$. Suppose the channel is block fading, which means that the channel coherence time is larger than the length of each frame. We model the wireless multipath channel as a finite-impulse response (FIR) filter with the channel impulse response (CIR) to be $\boldsymbol{h} = [h(0), h(1), \ldots, h(L)]^T$, which is static within each frame. At the receiver, we adopt frame-by-frame processing for channel equalization. Once receiving the training symbols, the channel equalizer adjusts the tap weights to combat the ISI caused by the multipath propagation.

At time t of each frame, we input $N_e (N_e \le N_t)$ received training symbols, denoted as $\boldsymbol{y}_t = [y(t), y(t-1), \ldots, y(t-N_e+1)]^T$, to the channel equalizer, where $y(i)$ denotes the received symbols including training symbols and data symbols at time instant i. We have

$$\boldsymbol{y}_t = \boldsymbol{H}\boldsymbol{x}_t + \boldsymbol{\eta}_t \tag{1}$$

where $\boldsymbol{x}_t = [x(t), x(t-1), \ldots, x(t-N_e-L+1)]^T$ is the transmitted symbols, $\boldsymbol{\eta}_t = [\eta(t), \eta(t-1), \ldots, \eta(t-N_e+1)]^T \sim \mathcal{CN}(\boldsymbol{0}, \sigma_\eta^2 \boldsymbol{I}_{N_e})$ is the additive white Gaussian noise (AWGN) with zero mean and variance being σ_η^2, and

$$\boldsymbol{H} = \begin{bmatrix} h(0) & h(1) & \cdots & h(L) & \cdots & \cdots & 0 \\ 0 & h(0) & h(1) & \cdots & h(L) & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h(0) & h(1) & \cdots & h(L) \end{bmatrix} \tag{2}$$

is a Toeplitz channel matrix with N_e rows and $N_e + L$ columns. Then the output correlation matrix is denoted as

$$\boldsymbol{R}_{yy} \triangleq E[\boldsymbol{y}_t \boldsymbol{y}_t^H]. \tag{3}$$

$\boldsymbol{R}_{yy}$ will be full rank if we make expectation using a large enough window.

In a linear FIR equalizer, the error between the transmitted training symbol and the received training symbol at time t can be denoted as

$$e(t) = x(t) - \boldsymbol{w}^H \boldsymbol{y}_t \tag{4}$$

where $\boldsymbol{w}$ is a column vector of length N_e and represents the complex-valued weight of the channel equalizer. According to the criterion of MMSE, the cost function is

$$\xi \triangleq E[|e(t)|^2] = \sigma_x^2 + \boldsymbol{w}^H \boldsymbol{R}_{yy} \boldsymbol{w} - \boldsymbol{w}^H \boldsymbol{r}_\triangle - \boldsymbol{r}_\triangle^H \boldsymbol{w} \tag{5}$$

where

$$\sigma_x^2 \triangleq E[x(t) x^H(t)] \tag{6}$$

and

$$\boldsymbol{r}_\triangle \triangleq E[\boldsymbol{y}_t x^H(t)]. \tag{7}$$

The objective of this MMSE problem is

$$\min_{\boldsymbol{w}} \xi. \tag{8}$$

We set the derivative of ξ over $\boldsymbol{w}$ to be zero and obtain the optimal equalizer weights, denoted as

$$\boldsymbol{w}_{\text{opt}} = \boldsymbol{R}_{yy}^{-1} \boldsymbol{r}_\triangle. \tag{9}$$

We substitute (9) in (5), obtaining the minimized cost function as

$$\xi_{\min} = \sigma_x^2 - \boldsymbol{r}_\triangle^H \boldsymbol{R}_{yy}^{-1} \boldsymbol{r}_\triangle. \tag{10}$$

Since $\boldsymbol{R}_{yy}$ is a Hermitian matrix, we denote the Cholesky factorization as $\boldsymbol{R}_{yy} \triangleq \boldsymbol{L}\boldsymbol{L}^H$, where $\boldsymbol{L}$ is a lower-triangular matrix. We rewrite (5) as

$$\xi = \sigma_x^2 + \boldsymbol{w}^H \boldsymbol{L}\boldsymbol{L}^H \boldsymbol{w} - \boldsymbol{w}^H \boldsymbol{L}\boldsymbol{L}^{-1} \boldsymbol{r}_\triangle - \boldsymbol{r}_\triangle^H (\boldsymbol{L}^{-1})^H \boldsymbol{L}^H \boldsymbol{w} \tag{11}$$

and then we have

$$\xi = \underbrace{\sigma_x^2 - \boldsymbol{r}_\triangle^H (\boldsymbol{L}^{-1})^H \boldsymbol{L}^{-1} \boldsymbol{r}_\triangle}_{\xi_{\min}} + \underbrace{\left\| \boldsymbol{L}^H \boldsymbol{w} - \boldsymbol{L}^{-1} \boldsymbol{r}_\triangle \right\|_2^2}_{\triangleq \xi_{\text{excess}}}. \tag{12}$$

It's observed that $\xi_{\min}$ is a constant independent of $\boldsymbol{w}$ and $\boldsymbol{w}_{\text{opt}}$ will lead ξ_{excess} to be zero. Any choice of $\boldsymbol{w}$ different from $\boldsymbol{w}_{\text{opt}}$ will increase ξ_{excess}. We define the signal to noise ratio (SNR) as

$$\gamma \triangleq \frac{\sigma_x^2}{\xi} = \frac{\sigma_x^2}{\xi_{\min} + \xi_{\text{excess}}}. \tag{13}$$

With $\boldsymbol{w}_{\text{opt}}$, the SNR is maximized to be

$$\gamma_{\text{opt}} \triangleq \frac{\sigma_x^2}{\xi_{\min}} \ge \gamma. \tag{14}$$

Then the performance loss in units of dB can be defined as

$$\alpha \triangleq 10 \log_{10} \gamma_{\text{opt}} - 10 \log_{10} \gamma = 10 \log_{10} \left(1 + \frac{\xi_{\text{excess}}}{\xi_{\min}} \right). \tag{15}$$

III. Sparse Equalizer Design

In this section, we consider the sparse equalizer design in terms of $\boldsymbol{w}$. In particular, $\boldsymbol{w}$ is sparse, where only $S (S \ll N_e)$ taps of $\boldsymbol{w}$ are nonzero. Therefore, $\boldsymbol{w}_{\text{opt}}$ corresponding to γ_{opt} may not be achieved by sparse equalizer. In order to improve γ, the design of $\boldsymbol{w}$ including the selection of S nonzero tap positions and the determination of the corresponding S weights, is investigated under two following different settings, one assuming S is known and the other assuming S is unknown.

A. The number of nonzero taps is given

In some applications, i.e., underwater acoustic communications [8]–[10], the architecture of channel equalizer, e.g., the number of complex-valued multipliers, is given. In other words, the number of nonzero taps of the equalizer, denoted as S, is known. Instead of completely replacing the equalizers with new ones, a compromised choice considering the cost of equalizer upgrade is to improve the performance of the existing architecture by optimizing the tap positions and weights of $\boldsymbol{w}$, where the optimization procedures are provided by additional block that can be connected to the existing equalizer. In this case, the design of sparse equalizer is to maximize γ given S, which is essentially to minimize ξ_{excess} and can be formulated as

$$\min_{\boldsymbol{w}} \left\| \boldsymbol{L}^H \boldsymbol{w} - \boldsymbol{L}^{-1} \boldsymbol{r}_{\triangle} \right\|_2 \quad \text{s.t.} \quad \|\boldsymbol{w}\|_0 = S \tag{16}$$

where $\|\boldsymbol{w}\|_0$ denotes the number of nonzero entries of $\boldsymbol{w}$. Note that although the cost function of (16) is quadratic, the constraint of (16) is non-convex. Hence (16) can not be solved by the existing optimization tools. Suppose $\boldsymbol{p}$ is a set of S ascending positive integers and denotes the indices of the S nonzero entries of $\boldsymbol{w}$ corresponding to the S nonzero tap positions of the sparse equalizer. Then (16) is equivalent to

$$\min_{\boldsymbol{w}_{\boldsymbol{p}}} \left\| \boldsymbol{L}_{\boldsymbol{p}}^H \boldsymbol{w}_{\boldsymbol{p}} - \boldsymbol{L}^{-1} \boldsymbol{r}_{\triangle} \right\|_2 \tag{17}$$

where $\boldsymbol{w}_{\boldsymbol{p}}$ and $\boldsymbol{L}_{\boldsymbol{p}}$ denote the S nonzero entries of $\boldsymbol{w}$ and the corresponding S rows of $\boldsymbol{L}$, respectively. For this unconstrained ℓ_2-norm minimization problem, the optimal weights can be derived by the least squares (LS) method as

$$\hat{\boldsymbol{w}}_{\boldsymbol{p}} = (\boldsymbol{L}_{\boldsymbol{p}} \boldsymbol{L}_{\boldsymbol{p}}^H)^{-1} \boldsymbol{L}_{\boldsymbol{p}} \boldsymbol{L}^{-1} \boldsymbol{r}_{\triangle}. \tag{18}$$

In fact, $\boldsymbol{L}_{\boldsymbol{p}} \boldsymbol{L}_{\boldsymbol{p}}^H$ can be fast computed based on $\boldsymbol{R}_{yy}$ by selecting a submatrix of $\boldsymbol{R}_{yy}$ with both of rows and columns indexed by $\boldsymbol{p}$. Since $\boldsymbol{L}\boldsymbol{L}^{-1} = \boldsymbol{I}_{N_e}$, $\boldsymbol{L}_{\boldsymbol{p}}\boldsymbol{L}^{-1}$ is a submatrix of $\boldsymbol{I}_{N_e}$ with rows indexed by $\boldsymbol{p}$, where $\boldsymbol{I}_{N_e}$ is the identity matrix of dimension N_e. We denote the achieved minimum of (17) by $\hat{\boldsymbol{w}}_{\boldsymbol{p}}$ as

$$\xi_p = \left\| \boldsymbol{L}_{\boldsymbol{p}}^H \hat{\boldsymbol{w}}_{\boldsymbol{p}} - \boldsymbol{L}^{-1} \boldsymbol{r}_{\triangle} \right\|_2. \tag{19}$$

The objective of (16) is to find a best subset $\boldsymbol{p}$ from Ω so that

$$\min_{\boldsymbol{p} \subset \Omega} \xi_p \tag{20}$$

where

$$\Omega \triangleq \{\boldsymbol{q} | \boldsymbol{q} \subset \Lambda, \|\boldsymbol{q}\|_0 = S\} \tag{21}$$

contains all possibilities of $\boldsymbol{p}$ and $\Lambda \triangleq \{1, 2, \ldots, N_e\}$ is a positive integer set. If N_e and S are not small enough, Ω will be a huge set. For example, if $N_e = 100$ and $S = 10$, $\|\Omega\|_0 = \binom{100}{10} = 1.7 \times 10^{13}$. It's impossible for the receiver to store Ω into the memory and exhaustively check each entry of Ω until the best $\boldsymbol{p} \subset \Omega$ is found.

Although we may use orthogonal matching pursuit (OMP) to sequentially obtain each entry of $\boldsymbol{p}$, where the number of iterations of OMP is set to be S, OMP may fast converge to a local optimum due to the greedy essence of OMP. Since the number of nonzero entries is explicitly known, it is better to take advantage of this knowledge at the start of the algorithm, i.e., starting with a random feasible subset and iteratively refining the subset until an optimal or near-optimal subset is achieved.

Now we propose a fast algorithm to design $\boldsymbol{p}$ for (20), as shown in Algorithm 1. At first, we input the parameters N_e, S, T_1 and T_2, where T_1 and T_2 represent the number of outer-loop iterations and inner-loop iterations, respectively. Each outer-loop iteration starting from step 3 to step 15 includes T_2 inner-loop iterations that start from step 5 to step 13. Then at step 2, a zero matrix $\boldsymbol{D}$ is initialized to store the optimized subsets after running inner-loop iterations. Each row of $\boldsymbol{D}$ stores an optimized subset $\boldsymbol{p}$, with the corresponding ξ_p stored in $\boldsymbol{r}$ that is initialized to be a zero vector. At each outer-loop iteration, we start by randomly generating a subset $\boldsymbol{p} \subset \Omega$. By introducing the randomness to the algorithm so that it starts from different initial subset, we can avoid that the algorithm falls into local optimums. As T_1 increases to infinity, the algorithm will converge to the global optimum.

We use each inner-loop iteration to obtain an optimized subset. An auxiliary vector $\boldsymbol{p}^*$ is used to record the subset obtained from the previous inner-loop iteration. If $\boldsymbol{p}$ is exactly the same as $\boldsymbol{p}^*$ as shown in step 6, which implies we did not get a new subset, it is meaningless to continue the inner-loop iterations because the results thereafter will be exactly the same. Then we break from the inner-loop iterations, indicated by step 7. Specially, we have to reset $\boldsymbol{p}^*$ by $\boldsymbol{p}^* \Leftarrow \boldsymbol{0}^S$ before starting each inner-loop iteration. In this way, we can save the CPU running time and therefore improve the efficiency of the algorithm by skipping the same routine.

Algorithm 1 - Design of Sparse Linear Equalizer Given the Number of Nonzero Taps

---

1: Input: N_e, S, T_1, T_2.
2: Initializations: $\boldsymbol{D} \Leftarrow \boldsymbol{0}^{T_1 \times S}$. $\boldsymbol{r} \Leftarrow \boldsymbol{0}^{T_1}$.
3: **for** $l = 1, 2, \ldots, T_1$
4: randomly generate $\boldsymbol{p} \subset \Omega$. $\boldsymbol{p}^* \Leftarrow \boldsymbol{0}^S$.
5: **for** $n = 1, 2, \ldots, T_2$
6: **if** $\boldsymbol{p} = \boldsymbol{p}^*$
7: break.
8: **end if**
9: $\boldsymbol{p}^* \Leftarrow \boldsymbol{p}$.
10: **for** $k = 1, 2, \ldots, S$
11: Obtain $\hat{\boldsymbol{p}}_{\boldsymbol{p},k}$ according to (23). $\boldsymbol{p} \Leftarrow \hat{\boldsymbol{p}}_{\boldsymbol{p},k}$.
12: **end for**(k)
13: **end for**(n)
14: $D(l) \Leftarrow \boldsymbol{p}$. $r(l) \Leftarrow \xi_p$.
15: **end for**(l)
16: $t = \arg\min_{i=1,2,\ldots,T_1} r(i)$.
17: Output $\boldsymbol{p}_o = D(t)$ as the designed nonzero tap positions.
18: Substitute $\boldsymbol{p}_o$ into (18) to design the weights of taps.

The sequential update of each entry of $\boldsymbol{p}$ is indicated from step 10 to step 12. For $k = 1, 2, \ldots, S$, given the latest $\boldsymbol{p}$ from the last inner-loop iteration, we update the kth entry of $\boldsymbol{p}$ with the best one selected from

$$\Phi = \Omega \setminus \{p(i)|i = 1, 2, \ldots, S, i \neq k\}, \tag{22}$$

where $p(i)$ denotes the ith entry of $\boldsymbol{p}$. Mathematically, the resultant subset $\hat{\boldsymbol{p}}_{\boldsymbol{p},k}$ with the update of the kth entry is given by

$$\hat{\boldsymbol{p}}_{\boldsymbol{p},k} = \arg \min_{\substack{\tilde{\boldsymbol{p}} \\ \tilde{p}(i)=p(i),\ i=1,2,\ldots,S, i\neq k \\ \tilde{p}(i)\in\Phi}} \xi_{\tilde{\boldsymbol{p}}}. \tag{23}$$

After we obtain $\hat{\boldsymbol{p}}_{\boldsymbol{p},k}$ for given $\boldsymbol{p}$ and k, we update $\boldsymbol{p}$ by $\boldsymbol{p} \Leftarrow \hat{\boldsymbol{p}}_{\boldsymbol{p},k}$. Once we finish all of outer-loop iterations, we select the minimum from $\boldsymbol{r}$ and output the corresponding row of $\boldsymbol{D}$ as the optimized subset $\boldsymbol{p}_o$ corresponding to the designed nonzero tap positions, indicated by step 16 and step 17. We then substitute $\boldsymbol{p}_o$ into (18) to design the weights of nonzero taps.

The computational complexity to obtain ξ_p given a subset $\boldsymbol{p}$ in (19) is $O(S^2)$. Therefore, the computational complexity for Algorithm 1 is

$$O(T_1 T_2 S^3 (N_e - S + 1)) \tag{24}$$

suppose that T_2 is small enough. If T_2 is large, the inner-loop iterations will terminate itself by procedures from step 6 to step 8, leading to even lower complexity than (24).

B. The number of nonzero taps is not given

In most applications, the number of nonzero taps is not given, which means that we have to find an appropriate S as well as to select S nonzero tap positions and determine the corresponding S weights of $\boldsymbol{w}$. Given the tolerance of performance loss α, we can obtain the upper bound for ξ_{excess} as

$$\epsilon \triangleq (10^{\frac{\alpha}{10}} - 1)\xi_{\min} \tag{25}$$

according to (15). In this case, the objective of sparse equalizer design is to minimize the number of nonzero taps within the tolerance of performance loss. It can be formulated as

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|_0 \quad \text{s.t.} \quad \left\|\boldsymbol{L}^H \boldsymbol{w} - \boldsymbol{L}^{-1} \boldsymbol{r}_\triangle\right\|_2 < \epsilon \tag{26}$$

which can be convex relaxed [11] as

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|_1 \quad \text{s.t.} \quad \left\|\boldsymbol{L}^H \boldsymbol{w} - \boldsymbol{L}^{-1} \boldsymbol{r}_\triangle\right\|_2 < \epsilon. \tag{27}$$

(27) has already been extensively studied by compressed sensing (CS) literatures and can be relaxed as a convex optimization problem where existing optimization solver can be applied. Specially, $\boldsymbol{L}^H$ is a square matrix with the same number of rows and columns; while in CS, the rows of measurement matrix is much fewer than the columns.

IV. Simulation results

We consider each frame consisting of $N_t = 500$ training symbols and $N_d = 2000$ data symbols with quadrature phase-shift keying (QPSK) modulation. The CIR length of the channel is set to be 31, where $L = 30$. A channel equalizer with the length of $N_e = 100$ is used to combat the channel distortion. To perform Algorithm 1, we set the number of outer-loop iterations $T_1 = 5$ and the number of inner-loop iterations $T_2 = 8$.

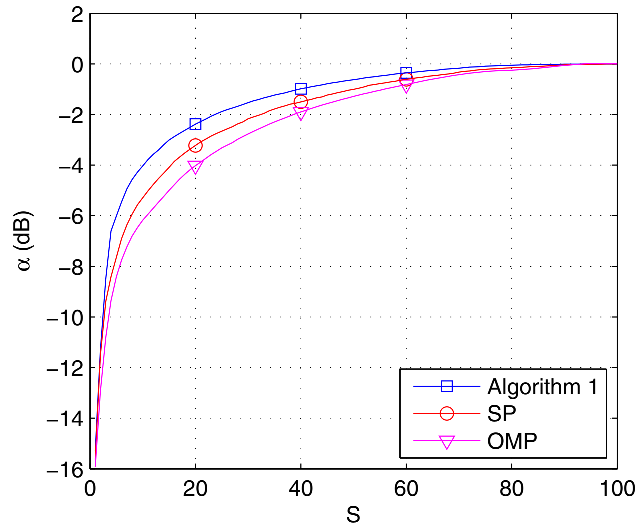

Fig. 2. Comparisons of performance loss for Algorithm 1, SP and OMP with different number of nonzero taps of sparse equalizer.

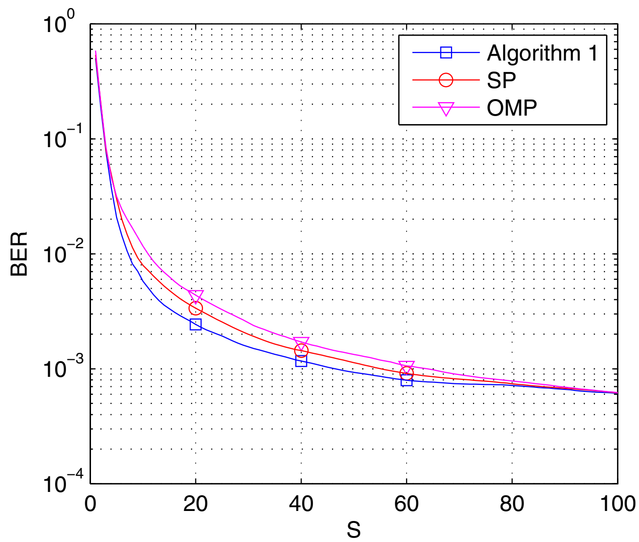

Fig. 3. Comparisons of BER for Algorithm 1, SP and OMP with different number of nonzero taps of sparse equalizer.

As shown in Fig. 2, sparse equalizer can approach the performance of non-sparse equalizer with much fewer taps. The performance loss using Algorithm 1 is less than 1dB at $S = 40$, while totally 60 equalizer taps with a percentage of 60% can be saved. Moreover, compared to the OMP algorithm used in [2] and subspace pursuit (SP) [12], Algorithm 1 can further reduce the performance loss. At $S = 20$, around 2dB performance improvement can be achieved by Algorithm 1 over OMP.

As shown in Fig. 3, we also compare the bit error rate (BER) performance for Algorithm 1, SP and OMP. SP performs better than OMP, while Algorithm 1 performs best among the three. At BER of 10^{-3}, around 20 equalizer taps with a percentage of 33% can be saved using Algorithm 1 instead of OMP.

V. Conclusions

We have studied the selection of nonzero taps for sparse linear equalizer under the criterion of MMSE. In this context where the number of nonzero taps of sparse equalizer is known, we have formulated the joint determination of positions and weights of nonzero taps of sparse equalizer as a subset selection problem. We have proposed a fast algorithm that uses two levels of loops to iteratively update each entry of the subset. We have analyzed the computational complexity of the proposed algorithm. We have also investigated the sparse equalizer design where the number of nonzero taps is not given. Simulation results have shown that around 60% equalizer taps can be saved, while the performance loss is no more than 1dB. Moreover, compared to the OMP algorithm, the proposed algorithm can save 33% equalizer taps achieving the same BER of 10^{-3}.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61302097, by the National High Technology Research and Development Program of China (863 Program) under the Grant 2012AA01A504, by the Ph.D. Programs Foundation of the Ministry of Education of China under Grant 20120092120014, and by the Fundamental Research Funds for the Central Universities.

References

[1] G. Kutz and D. Raphaeli, "Determination of tap positions for sparse equalizers," *IEEE Trans. Commun.*, vol. 55, no. 9, pp. 1712–1724, Sep. 2007.

[2] A. Gomaa and N. Al-Dhahir, "A new design framework for sparse FIR MIMO equalizers," *IEEE Trans. Commun.*, vol. 59, no. 8, pp. 2132–2140, Aug. 2011.

[3] C. Rusu and B. Dumitrescu, "Iterative reweighted ℓ_1 design of sparse fir filters," *Signal Process.*, vol. 92, no. 4, pp. 905–911, Apr. 2012.

[4] D. Wei and A. V. Oppenheim, "A branch-and-bound algorithm for quadratically-constrained sparse filter design," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 1006–1018, Feb. 2013.

[5] D. Wei, C. K. Sestok, and A. V. Oppenheim, "Sparse filter design under a quadratic constraint: Low-complexity algorithms," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 857–870, Feb. 2013.

[6] K. Shi and B. C. Thomsen, "Sparse adaptive frequency domain equalizers for mode-group division multiplexing," *J. Lightwave Technol.*, vol. 33, no. 2, pp. 311–317, Jan. 2015.

[7] S. Khalid and S. Abrar, "Blind adaptive algorithm for sparse channel equalisation using projections onto ℓ_p-ball," *Electr. Lett.*, vol. 51, no. 18, pp. 1422–1424, Sep. 2015.

[8] 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.

[9] 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.

[10] C. Qi and L. Wu, "Sparse channel estimation for wavelet-based underwater acoustic communications," *Trans. Emerg. Telecommun. Technol.*, vol. 23, no. 8, pp. 764–776, Dec. 2012.

[11] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1030–1051, Mar. 2006.

[12] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.