Reweighted l_1-norm penalized LMS for sparse channel estimation and its analysis ☆

Omid Taheri [a], Sergiy A. Vorobyov [a,b,*]

[a] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada T6G 2V4
[b] School of Electrical Engineering, Aalto University, FI-00076 Aalto, Finland

ARTICLE INFO

ABSTRACT

A new reweighted l_1−norm penalized least mean square (LMS) algorithm for sparse channel estimation is proposed and studied in this paper. Since standard LMS algorithm does not take into account the sparsity information about the channel impulse response (CIR), sparsity-aware modifications of the LMS algorithm aim at outperforming the standard LMS by introducing a penalty term to the standard LMS cost function which forces the solution to be sparse. Our reweighted l_1−norm penalized LMS algorithm introduces in addition a reweighting of the CIR coefficient estimates to promote a sparse solution even more and approximate l_0−pseudo−norm closer. We provide in depth quantitative analysis of the reweighted l_1−norm penalized LMS algorithm. An expression for the excess mean square error (MSE) of the algorithm is also derived which suggests that under the right conditions, the reweighted l_1−norm penalized LMS algorithm outperforms the standard LMS, which is expected. However, our quantitative analysis also answers the question of what is the maximum sparsity level in the channel for which the reweighted l_1−norm penalized LMS algorithm is better than the standard LMS. Simulation results showing the better performance of the reweighted l_1−norm penalized LMS algorithm compared to other existing LMS-type algorithms are given.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

The least mean square (LMS) algorithm is very well known in the field of adaptive signal processing [1,2]. It belongs to the class of stochastic gradient algorithms. The attractive feature of the LMS algorithm is that it does not need extensive stochastic knowledge of the channel and the input data sequence unlike some other parameter estimation methods such as the recursive least squares (RLS) and Kalman filter. While RLS and Kalman filter need to know the covariance matrix of the input data sequence, the LMS algorithm only requires an approximate estimate of the largest eigenvalue of the covariance matrix for proper selection of the step size that guarantees the convergence. The LMS algorithm is being employed in a wide variety of applications in signal processing and communications including system identification [3], echo cancelation [4], channel estimation [5], adaptive communication line enhancement [6], etc. A particular application considered in this paper is that of estimating a finite impulse response (FIR) channel. The choice of the channel estimation algorithm for use in a communication system comes down to the available

information about the statistics of the system, the desired performance of the estimation algorithm, as well as the complexity of the estimation process.

The standard recursive parameter estimation algorithms do not assume any information about the specific structure of the channel being estimated. However, being aware of the channel structure one can modify the standard algorithms in order to have a better estimate of the channel. In this paper, we are concerned with a class of channels where the channel impulse response (CIR) is sparse. A time sparse discrete-time signal is the one with only a few nonzero entries. In general, the domain that the signal is sparse in does not necessarily have to be the time domain. Other sparsity bases can also be used and are represented by an $N \times N$ orthogonal matrix where N is the length of the signal.

Sparsity-aware modifications of the LMS algorithm have been presented in the signal processing literature in the past few years. The methods introduced in [7,8] add a penalty term to the standard LMS error function which is designed in a way to force the solution to be sparse. A penalty in the form of the l_0-pseudo-norm of the CIR is used in [8], while [7] uses the l_1−norm. In [9], the mean square convergence and stability analysis for one of the algorithms in [7] for the case of white input signals is presented. A performance analysis of the l_0-pseudo-norm constraint LMS algorithm of [8] is given in [10]. In [11,12], variations of the algorithms in [7] are introduced. In [11], the filter coefficients are updated in a transform domain which leads to faster convergence for non-white inputs. In [13], the idea of using a weighted l_1−norm penalty for the purpose of sparse system identification is presented without any convergence analysis. Moreover, sparsity promoting partial update LMS algorithms have been recently developed in [14].

The authors of [15] introduce a scheme that employs two sequential adaptive filters for communication line or network echo cancelers. The method exploits the sparseness of the CIR and uses two sequential LMS type structures which are both shorter than the largest delay of the channel. A family of the so-called natural gradient estimation algorithms is also studied in [16]. It is shown that the class of sparse LMS algorithms presented has faster convergence rate.

Sparse diffusion schemes are presented in [17,18] that provide adaptive algorithms for distributed learning in networks. In [17], projection methods over hyperslabs and weighted l_1-balls are presented and analyzed for distributed learning. Penalized cost functions are used in [18] to enforce the sparsity of the solution. Among the penalty terms considered is the weighted l_1−norm penalty of [7]. Convergence analysis for the distributed adaptive algorithm is also given in [18] for a convex penalty term.

Other channel estimation algorithms have also been modified to either better adapt to a sparse channel or achieve the same performance as the corresponding standard algorithms with lower complexity. Time and norm-weighted least absolute shrinkage and selection operator (LASSO) where weights obtained from RLS algorithm has been presented in [19]. A greedy RLS algorithm designed for finding sparse solutions to linear systems has been presented in [20], and it has been demonstrated that it has better performance than the standard RLS algorithm for estimating sparse time-varying FIR channels. A compressed sensing (CS)-based Kalman filter has been developed in [21] for estimating signals with time varying sparsity pattern.[1]

In this paper, we first derive the reweighted l_1−norm penalized LMS algorithm which is based on modifying the LMS error (objective) function by adding the l_1−norm penalty term and also introducing a reweighting of the CIR coefficients.[2] Then the main contribution follows that is the in depth study of the convergence and excess mean square error (MSE) analysis of the reweighted l_1−norm penalized LMS algorithm. It is worth mentioning that the analytic arguments in [18] can be applied to a centralized learning problem as well as a diffusion network. In this way, it is also possible to prove the mean square stability of the reweighted l_1−norm penalized LMS algorithm in a different manner than presented in this paper. Our simulation results show that the proposed algorithm outperforms the standard LMS as well as the penalized sparsity-aware LMS algorithms of [7] and approve our theoretical studies.

The rest of the paper is organized as follows. Section 2 reviews the system model used and the standard LMS algorithm. In Section 3, the reweighted l_1−norm penalized LMS algorithm is introduced. An analytical study of the convergence of the reweighted l_1−norm penalized LMS algorithm as well as its excess MSE is given in Section 4. Simulation results comparing the performance of different sparsity-aware LMS algorithms are given in Section 5. Section 6 concludes the paper.

2. System model and preliminaries

2.1. Standard LMS

The system model of a general communication system used in this paper is shown in Fig. 1. The standard LMS algorithm is used to estimate the actual CIR of the communication system, i.e., the vector $\boldsymbol{w}$ in Fig. 1. Let us introduce as well other notations in the figure. An estimate of the actual CIR vector $\boldsymbol{w}$ at the time step k is denoted as $\boldsymbol{w}_k$. The system's input data vector is $\boldsymbol{x}_k$, n_k is the additive noise, d_k is the desired response, and e_k is the error signal. The CIR is assumed to be of length N, and therefore, $\boldsymbol{w} \triangleq [w_1 \ w_2 \ \dots \ w_N]^T$, $\boldsymbol{w}_k \triangleq [w_{1,k} \ w_{2,k} \ \dots \ w_{N,k}]^T$, and $\boldsymbol{x}_k \triangleq [x_k \ x_{k-1} \ \dots \ x_{k-N+1}]^T$, where $(\cdot)^T$ stands for the vector transposition. As shown in Fig. 1

$$d_k = \boldsymbol{w}^T \boldsymbol{x}_k + n_k$$
$$e_k \triangleq d_k - \boldsymbol{w}_k^T \boldsymbol{x}_k. \tag{1}$$

The noise samples n_k are assumed to be independent and identically distributed (i.i.d.) with zero mean and variance

---

[1] CS is the theory that considers the problem of sparse signal recovery from a few measurements [22,23]. The number of measurements in CS is a lot smaller than the overall dimension of the signal.

[2] Some preliminary results (the method and some simulation results) have been reported in the conference contribution [24].
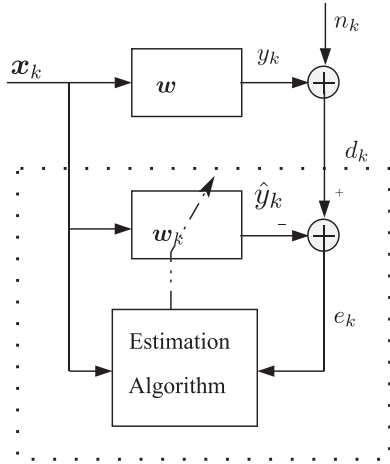
Fig. 1. Block diagram of a communication system.

of σ_n^2. Also, the input data sequence $\boldsymbol{x}_k$ and the additive noise samples n_k are assumed to be independent.

In standard LMS, the cost function is $L_k \triangleq (1/2)e_k^2$, and it is minimized using the gradient descent algorithm [1]. The update equation of the standard LMS algorithm can be derived from the above-mentioned cost function as

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \mu \frac{\partial L_k}{\partial \boldsymbol{w}_k} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k \qquad (2)$$

where μ is the step size of the iterative algorithm. To make sure that the LMS algorithm converges, μ is chosen such that $0 < \mu < \lambda_{\max}^{-1}$ with $\lambda_{\max}$ being the maximum eigenvalue of the covariance matrix of $\boldsymbol{x}_k$, i.e., $\boldsymbol{R} \triangleq E[\boldsymbol{x}_k \boldsymbol{x}_k^T]$. For the purpose of convergence analysis of the LMS algorithm, a coefficient error vector is usually defined as

$$\boldsymbol{v}_k \triangleq \boldsymbol{w}_k - \boldsymbol{w}. \qquad (3)$$

The data vector $\boldsymbol{x}_k$ is assumed to be independent of the coefficient error vector $\boldsymbol{v}_k$. The excess MSE denoted as ξ_k is defined as $\xi_k \triangleq E[(\boldsymbol{v}_k^T \boldsymbol{x}_k)^2]$. It can be further expanded as

$$\xi_k = E[\boldsymbol{v}_k^T \boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{v}_k]. \qquad (4)$$

In (4), $\boldsymbol{v}_k^T \boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{v}_k$ is a scalar, and therefore, it is equal to its trace, denoted hereafter as tr$\{\cdot\}$. Also, since tr$\{\boldsymbol{v}_k^T \boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{v}_k\} = $ tr$\{\boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{v}_k \boldsymbol{v}_k^T\}$ and the two mathematical operators of matrix trace and expectation are interchangeable we can simplify (4) as

$$\xi_k = E[\text{tr}\{\boldsymbol{v}_k^T \boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{v}_k\}] = \text{tr}\{\boldsymbol{R} E[\boldsymbol{v}_k \boldsymbol{v}_k^T]\}. \qquad (5)$$

Let us introduce the matrix $\boldsymbol{R}_v \triangleq \lim_{k \to \infty} E[\boldsymbol{v}_k \boldsymbol{v}_k^T]$ and the vector $\xi \triangleq \lim_{k \to \infty} \xi_k$. Then we have from (5) that $\xi = \text{tr}\{\boldsymbol{R}\boldsymbol{R}_v\}$. Moreover, the excess MSE can be found as [7]

$$\xi = \frac{\eta}{2-\eta} \sigma_n^2 \qquad (6)$$

where

$$\eta \triangleq \mu \, \text{tr}\{\boldsymbol{R}(\boldsymbol{I} - \mu \boldsymbol{R})^{-1}\}. \qquad (7)$$

3. Reweighted l_1−norm penalized LMS algorithm

In the standard LMS algorithm, the fact that the cost function is convex guarantees that the gradient descent algorithm converges to the optimum point under the aforementioned condition on μ. The standard LMS algorithm assumes no structural information about the signal/system to be estimated. Taking any structural information into account, one should be able to modify the algorithm and benefit by lower estimation error, faster convergence, or lower algorithm complexity. In this paper, we are interested in the case when the CIR is sparse. For a CIR $\boldsymbol{w}$ to be sparse in some sparsity domain $\boldsymbol{\Psi}$ most of the coefficients in the vector representation of $\boldsymbol{w}$ in this domain $\boldsymbol{\Psi}$ should be zeros or insignificant in value. Several sparsity-aware modifications of the standard LMS have been introduced in the literature [7–13,24].

The reweighted l_1−norm minimization for sparse signal recovery has a better performance than the standard l_1−norm minimization that is usually employed in the CS literature [25]. It is due to the fact that a properly reweighted l_1−norm approximates the l_0-pseudo-norm, which actually needs to be minimized, better than the l_1−norm. Therefore, one approach to enforce the sparsity of the solution for the sparsity-aware LMS-type algorithms is to introduce the reweighted l_1−norm penalty term in the cost function [24].[3] Our reweighted l_1−norm penalized LMS algorithm considers a penalty term proportional to the reweighted l_1−norm of the coefficient vector. The corresponding cost function can be written as

$$L_k^{rl_1} \triangleq (1/2)e_k^2 + \gamma_r \|\boldsymbol{s}_k \boldsymbol{w}_k\|_1 \qquad (8)$$

where $\|\cdot\|_1$ stands for the l_1−norm of a vector and γ_r is the weight associated with the penalty term and elements of the $1 \times N$ row vector $\boldsymbol{s}_k$ are set to

$$[\boldsymbol{s}_k]_i = \frac{1}{\epsilon_r + |[\boldsymbol{w}_{k-1}]_i|}, \quad i = 1, \dots, N \qquad (9)$$

with ϵ_r being some positive number and $[\cdot]_i$ denoting the i-th entry of a vector. The update equation can be derived by differentiating (8) with respect to the vector of CIR coefficients and using the gradient descent principle shown in (2). The resulting update equation is

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_r \frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|} \qquad (10)$$

where $\rho_r = \mu \gamma_r$ and sgn$(\cdot)$ is the sign function which operates on every component of the vector separately and it is zero for $x=0$, 1 for $x>0$, and -1 for $x<0$. The absolute value operator as well as the sgn$(\cdot)$ and the division operator in the last term of (10) are all component-wise. Therefore, the i-th element of sgn$(\boldsymbol{w}_k)/(\epsilon_r + |\boldsymbol{w}_{k-1}|)$ is $[\text{sgn}(\boldsymbol{w}_k)]_i/(\epsilon_r + |[\boldsymbol{w}_{k-1}]_i|)$. Note that although the weight vector $\boldsymbol{s}_k$ changes in every stage of this sparsity-aware LMS algorithm, it does not depend on $\boldsymbol{w}_k$, and the cost function $L_k^{rl_1}$ is convex. Therefore, the reweighted l_1−norm penalized LMS algorithm is guaranteed to converge to the global minimum under some conditions. Thus, we study the convergence of the proposed algorithm in the next section.

---

[3] The other approach is to use the l_p-pseudo-norm penalty term with $0 < p < 1$ which is introduced in the simulations section.

4. Convergence study of the reweighted l_1-norm penalized LMS method

The reweighted l_1-norm penalized LMS algorithm follows the logic that the penalty term resembling the l_0-pseudo-norm of the coefficient vector forces the solution of the modified LMS algorithm to be sparse. The cost function of the reweighted l_1-norm penalized LMS algorithm is given in (8), while the update equation is given in (10).

4.1. Mean convergence

We first study the mean convergence of the reweighted l_1-norm penalized LMS algorithm. The update equation for the coefficient error vector of the l_1-norm penalized LMS $\boldsymbol{v}_k$ can be written as

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k + \mu\big((\boldsymbol{w}^T - \boldsymbol{w}_k^T)\boldsymbol{x}_k + n_k\big)\boldsymbol{x}_k - \rho_r \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}$$
$$= \boldsymbol{v}_k - \mu\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k + \mu n_k\boldsymbol{x}_k - \rho_r \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}. \quad (11)$$

Since $\boldsymbol{v}_k^T\boldsymbol{x}_k$ is a scalar which is equal to $\boldsymbol{x}_k^T\boldsymbol{v}_k$, (11) can be rewritten as

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k - \mu\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k + \mu n_k\boldsymbol{x}_k - \rho_r \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}. \quad (12)$$

From (12) we can derive the evolution equation for $E[\boldsymbol{v}_k]$. Since n_k and $\boldsymbol{x}_k$ are independent and n_k is assumed to have zero mean, we have $E[\mu n_k\boldsymbol{x}_k] = 0$. Then the evolution equation is

$$E[\boldsymbol{v}_{k+1}] = (\boldsymbol{I} - \mu\boldsymbol{R})E[\boldsymbol{v}_k] - \rho_r E\left[\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\right]. \quad (13)$$

It is easy to see that the term $\mathrm{sgn}(\boldsymbol{w}_k)/(\epsilon_r + |\boldsymbol{w}_{k-1}|)$ is bounded below and above element-wise as follows:

$$\frac{-\mathbf{1}}{\epsilon_r} \leq \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|} \leq \frac{\mathbf{1}}{\epsilon_r} \quad (14)$$

where $\mathbf{1}$ is the vector with all of its entries set to one. Indeed, -1 is always less than or equal to $\mathrm{sgn}(\boldsymbol{w}_k)$, while 1 is always larger than or equal to $\mathrm{sgn}(\boldsymbol{w}_k)$. Moreover, $|\boldsymbol{w}_{k-1}|$ and ϵ_r are always non-negative, which means that the denominator of the middle term in (14) is always larger than or equal to the denominator of the right and left terms of (14), which means that (14) always holds true.

We can further see that $\rho_r E[\mathrm{sgn}(\boldsymbol{w}_k)/(\epsilon_r + |\boldsymbol{w}_{k-1}|)]$ is bounded between $(-\rho_r/\epsilon_r)\mathbf{1}$ and $(\rho_r/\epsilon_r)\mathbf{1}$. This bound on the second term on the right-hand side of (14) is helpful for studying the mean convergence of the reweighted l_1-norm penalized LMS algorithm. The following theorem establishes our main result on the mean convergence of the reweighted l_1-norm penalized LMS algorithm.

Theorem 1. If the maximal eigenvalue of the matrix $\boldsymbol{I} - \mu\boldsymbol{R}$ is smaller than 1, then the mean coefficient error vector $E[\boldsymbol{v}_k]$ is bounded as $k \to \infty$.

Proof. Let $\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$ be the eigenvalue decomposition of $\boldsymbol{R}$. Eq. (13) can be rewritten as

$$E[\boldsymbol{c}_{k+1}] = (\boldsymbol{I} - \mu\boldsymbol{\Lambda})E[\boldsymbol{c}_k] - \boldsymbol{w}_k' \quad (15)$$

where

$$\boldsymbol{c}_k \triangleq \boldsymbol{Q}^T\boldsymbol{v}_k$$
$$\boldsymbol{w}_k' \triangleq \rho_r\boldsymbol{Q}^T E\left[\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\right]. \quad (16)$$

Let also $\boldsymbol{q}$ be the vector whose i-th entry is the sum of the absolute values of the elements in the i-th row of the matrix $\boldsymbol{Q}^T$. The variable q_m is defined as the maximum element of the vector $\boldsymbol{q}$. The vector $\boldsymbol{Q}^T\mathrm{sgn}(\boldsymbol{w}_k)$ is thus bounded between $q_m\mathbf{1}$ and $-q_m\mathbf{1}$. Therefore, the variable $\boldsymbol{w}_k'$ in (16) is bounded between $(\rho_r q_m/\epsilon_r)\mathbf{1}$ and $(-\rho_r q_m/\epsilon_r)\mathbf{1}$.

It is easy to see from (15) that

$$E[\boldsymbol{c}_{k+M}] = (\boldsymbol{I} - \mu\boldsymbol{\Lambda})^M E[\boldsymbol{c}_k]$$
$$- \sum_{m=0}^{M-1}(\boldsymbol{I} - \mu\boldsymbol{\Lambda})^m \boldsymbol{w}_{k+M-m-1}'. \quad (17)$$

Moreover, since $\boldsymbol{\Lambda}$ and correspondingly $\boldsymbol{I} - \mu\boldsymbol{\Lambda}$ are diagonal matrices, the convergence behavior of every element of the vector $E[\boldsymbol{c}_{k+M}]$ can be studied separately.

Let λ_i be the i-th diagonal element of the matrix $\boldsymbol{\Lambda}$. From (17), we have

$$[E[\boldsymbol{c}_{k+M}]]_i = (1 - \mu\lambda_i)^M [E[\boldsymbol{c}_k]]_i$$
$$- \sum_{m=0}^{M-1}(1 - \mu\lambda_i)^m [\boldsymbol{w}_{k+M-m-1}']_i. \quad (18)$$

Since the largest eigenvalue of $\boldsymbol{I} - \mu\boldsymbol{R}$ is smaller than 1, then all the diagonal elements $1 - \mu\lambda_i$ are smaller than 1. Also note that the i-th entry of the vector $\boldsymbol{w}_k'$ is bounded between $\rho_r q_m/\epsilon_r$ and $-\rho_r q_m/\epsilon_r$. Therefore, by letting $M \to \infty$, the sum on the right-hand side of (18) is a geometric series with a common ratio of $1 - \mu\lambda_i$ and is bounded between $\rho_r q_m/(\mu\lambda_i\epsilon_r)$ and $-\rho_r q_m/(\mu\lambda_i\epsilon_r)$.

The other term on the right-hand side of (18) approaches zero as $M \to \infty$. As a result, $[E[\boldsymbol{c}_{k+M}]]_i$ as well as the whole vector $E[\boldsymbol{c}_{k+M}]$ are bounded when $M \to \infty$. Since according to (16) $E[\boldsymbol{c}_k]$ is a rotated version of $E[\boldsymbol{v}_k]$, the coefficient error vector $\boldsymbol{v}_k$ is also bounded in mean. Therefore, if the largest eigenvalue of $\boldsymbol{I} - \mu\boldsymbol{R}$ is smaller than 1, then $E[\boldsymbol{v}_k]$ is bounded as $k \to \infty$. □

Note that the condition in Theorem 1 is the same as the mean convergence condition for the standard LMS algorithm which has the following evolution equation for $E[\boldsymbol{v}_k]$:

$$E[\boldsymbol{v}_{k+1}] = (\boldsymbol{I} - \mu\boldsymbol{R})E[\boldsymbol{v}_k]. \quad (19)$$

4.2. Excess MSE

We now turn to the excess MSE calculation for the reweighted l_1-norm penalized LMS algorithm. Using the expression in (11) for $\boldsymbol{v}_{k+1}$, the variable $\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T$ can be written as follows:

$$\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T = \left(\boldsymbol{v}_k - \mu\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k + \mu n_k\boldsymbol{x}_k - \rho_r\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\right)$$
$$\times \left(\boldsymbol{v}_k^T - \mu\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T + \mu n_k\boldsymbol{x}_k^T - \rho_r\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r + |\boldsymbol{w}_{k-1}^T|}\right). \quad (20)$$

Expanding the right-hand side of (20) and then taking expectation on both sides results in the following equation:

$$
\begin{aligned}
E\left[\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T\right] &= E\left[\boldsymbol{v}_k\boldsymbol{v}_k^T\right] - \mu\left(E\left[\boldsymbol{v}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right] + E\left[\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{v}_k^T\right]\right) \\
&+ \mu^2 E\left[n_k^2\boldsymbol{x}_k\boldsymbol{x}_k^T\right] + \mu\left(E\left[n_k\boldsymbol{v}_k\boldsymbol{x}_k^T\right] + E\left[n_k\boldsymbol{x}_k\boldsymbol{v}_k^T\right]\right) \\
&- \mu^2\left(E\left[n_k\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{x}_k^T\right] + E\left[n_k\boldsymbol{x}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right]\right) + \mu^2 E\left[\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right] \\
&- \rho_r\left(E\left[\boldsymbol{v}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r + |\boldsymbol{w}_{k-1}^T|}\right] + E\left[\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\boldsymbol{v}_k^T\right]\right) \\
&+ \mu\rho_r\left(E\left[\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r + |\boldsymbol{w}_{k-1}^T|}\right] + E\left[\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right]\right) \\
&- \mu\rho_r\left(E\left[n_k\boldsymbol{x}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r + |\boldsymbol{w}_{k-1}^T|}\right] + E\left[n_k\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\boldsymbol{x}_k^T\right]\right) \\
&+ \rho_r^2\left(E\left[\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r + |\boldsymbol{w}_{k-1}^T|}\right]\right).
\end{aligned}
\tag{21}
$$

It is worth noting that due to the independence of the additive noise n_k of the data and coefficient error vectors and due to the fact that the additive noise is zero mean, we have

$$
\begin{aligned}
E\left[n_k\boldsymbol{v}_k\boldsymbol{x}_k^T\right] &= E[n_k]E\left[\boldsymbol{v}_k\boldsymbol{x}_k^T\right] = 0 \\
E\left[n_k\boldsymbol{x}_k\boldsymbol{v}_k^T\right] &= E[n_k]E\left[\boldsymbol{x}_k\boldsymbol{v}_k^T\right] = 0 \\
E\left[n_k\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{x}_k^T\right] &= E[n_k]E\left[\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{x}_k^T\right] = 0 \\
E\left[n_k\boldsymbol{x}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right] &= E[n_k]E\left[\boldsymbol{x}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right] = 0 \\
E\left[n_k\boldsymbol{x}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r + |\boldsymbol{w}_{k-1}^T|}\right] &= E\left[n_k\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\boldsymbol{x}_k^T\right] = 0.
\end{aligned}
$$

Since for Gaussian input sequences $E[\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T]$ can be shown to be equal to $2\boldsymbol{R}E[\boldsymbol{v}_k\boldsymbol{v}_k^T]\boldsymbol{R} + \boldsymbol{R}\,\mathrm{tr}\{\boldsymbol{R}E[\boldsymbol{v}_k\boldsymbol{v}_k^T]\}$ (see, for example, equation (12) of [26] and the derivation of equation (35) in [27]) in (21), the expression for $E[\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T]$ can be derived as in the following equation:

$$
\begin{aligned}
E\left[\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T\right] &= E\left[\boldsymbol{v}_k\boldsymbol{v}_k^T\right] - \mu\left(E\left[\boldsymbol{v}_k\boldsymbol{v}_k^T\right]\boldsymbol{R} + \boldsymbol{R}E\left[\boldsymbol{v}_k\boldsymbol{v}_k^T\right]\right) + \mu^2\sigma_n^2\boldsymbol{R} \\
&+ \mu^2\left(2\boldsymbol{R}E\left[\boldsymbol{v}_k\boldsymbol{v}_k^T\right]\boldsymbol{R} + \boldsymbol{R}\,\mathrm{tr}\{\boldsymbol{R}E\left[\boldsymbol{v}_k\boldsymbol{v}_k^T\right]\}\right) \\
&- \rho_r\left((\boldsymbol{I}-\mu\boldsymbol{R})E\left[\boldsymbol{v}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r + |\boldsymbol{w}_{k-1}^T|}\right] + E\left[\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\boldsymbol{v}_k^T\right](\boldsymbol{I}-\mu\boldsymbol{R})\right) \\
&+ \rho_r^2\left(E\left[\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r + |\boldsymbol{w}_{k-1}^T|}\right]\right).
\end{aligned}
\tag{22}
$$

Let $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ be defined as

$$
\begin{aligned}
\boldsymbol{A}_k &\triangleq \rho_r\left((\boldsymbol{I}-\mu\boldsymbol{R})E\left[\boldsymbol{v}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r + |\boldsymbol{w}_{k-1}^T|}\right]\right. \\
&\left.+ E\left[\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\boldsymbol{v}_k^T\right](\boldsymbol{I}-\mu\boldsymbol{R})\right)
\end{aligned}
\tag{23}
$$

and

$$
\boldsymbol{B}_k \triangleq \rho_r^2\left(E\left[\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r + |\boldsymbol{w}_{k-1}^T|}\right]\right).
\tag{24}
$$

Then, (22) can be rewritten as

$$
\begin{aligned}
E[\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T] &= E[\boldsymbol{v}_k\boldsymbol{v}_k^T] \\
&- \mu(E[\boldsymbol{v}_k\boldsymbol{v}_k^T]\boldsymbol{R} + \boldsymbol{R}E[\boldsymbol{v}_k\boldsymbol{v}_k^T]) + \mu^2\sigma_n^2\boldsymbol{R} \\
&+ \mu^2(2\boldsymbol{R}E[\boldsymbol{v}_k\boldsymbol{v}_k^T]\boldsymbol{R} + \boldsymbol{R}\,\mathrm{tr}\{\boldsymbol{R}E[\boldsymbol{v}_k\boldsymbol{v}_k^T]\}) \\
&- \boldsymbol{A}_k + \boldsymbol{B}_k.
\end{aligned}
\tag{25}
$$

Letting $k\to\infty$ in (25), we obtain

$$
\begin{aligned}
\boldsymbol{R}_v &= \boldsymbol{R}_v - \mu(\boldsymbol{R}_v\boldsymbol{R} + \boldsymbol{R}\boldsymbol{R}_v) + \mu^2\sigma_n^2\boldsymbol{R} \\
&+ \mu^2(2\boldsymbol{R}\boldsymbol{R}_v\boldsymbol{R} + \boldsymbol{R}\,\mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\}) + \lim_{k\to\infty}(\boldsymbol{B}_k - \boldsymbol{A}_k).
\end{aligned}
\tag{26}
$$

Crossing out $\boldsymbol{R}_v$ from both sides of (26) and then dividing the resulting equation by μ, we find that

$$
\begin{aligned}
\boldsymbol{R}_v\boldsymbol{R} &+ \boldsymbol{R}\boldsymbol{R}_v - 2\mu\boldsymbol{R}\boldsymbol{R}_v\boldsymbol{R} \\
&= \mu\boldsymbol{R}(\sigma_n^2 + \mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\}) + \frac{1}{\mu}\lim_{k\to\infty}(\boldsymbol{B}_k - \boldsymbol{A}_k).
\end{aligned}
\tag{27}
$$

Breaking $2\mu\boldsymbol{R}\boldsymbol{R}_v\boldsymbol{R}$ into the sum of two identical terms and then factoring out $\boldsymbol{R}\boldsymbol{R}_v$ and $\boldsymbol{R}_v\boldsymbol{R}$, we also obtain

$$
\begin{aligned}
\boldsymbol{R}\boldsymbol{R}_v(\boldsymbol{I}-\mu\boldsymbol{R}) &+ (\boldsymbol{I}-\mu\boldsymbol{R})\boldsymbol{R}_v\boldsymbol{R} \\
&= \mu\boldsymbol{R}(\sigma_n^2 + \mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\}) + \frac{1}{\mu}\lim_{k\to\infty}(\boldsymbol{B}_k - \boldsymbol{A}_k).
\end{aligned}
\tag{28}
$$

Multiplying both sides of (28) by $(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}$ from right, the following can be derived:

$$
\begin{aligned}
\boldsymbol{R}\boldsymbol{R}_v &+ (\boldsymbol{I}-\mu\boldsymbol{R})\boldsymbol{R}_v\boldsymbol{R}(\boldsymbol{I}-\mu\boldsymbol{R})^{-1} \\
&= \mu\boldsymbol{R}(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}(\sigma_n^2 + \mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\}) \\
&+ \frac{1}{\mu}\lim_{k\to\infty}(\boldsymbol{B}_k - \boldsymbol{A}_k)(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}.
\end{aligned}
\tag{29}
$$

Note that $\sigma_n^2 + \mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\}$ here is a scalar. Taking the trace of the two sides of (29), we have

$$
\begin{aligned}
\mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\} &+ \mathrm{tr}\{(\boldsymbol{I}-\mu\boldsymbol{R})\boldsymbol{R}_v\boldsymbol{R}(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\} \\
&= \mu(\sigma_n^2 + \mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\})\mathrm{tr}\{\boldsymbol{R}(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\} \\
&+ \frac{1}{\mu}\lim_{k\to\infty}\mathrm{tr}\{(\boldsymbol{B}_k - \boldsymbol{A}_k)(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\}.
\end{aligned}
\tag{30}
$$

$\mathrm{tr}\{(\boldsymbol{I}-\mu\boldsymbol{R})\boldsymbol{R}_v\boldsymbol{R}(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\}$ equals $\mathrm{tr}\{\boldsymbol{R}_v\boldsymbol{R}(\boldsymbol{I}-\mu\boldsymbol{R})(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\}$ which in turn is equal to $\mathrm{tr}\{\boldsymbol{R}_v\boldsymbol{R}\}$. Therefore, Eq. (30) can be simplified as follows:

$$
\begin{aligned}
\mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\} &+ \mathrm{tr}\{\boldsymbol{R}_v\boldsymbol{R}\} \\
&= \mu(\sigma_n^2 + \mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\})\mathrm{tr}\{\boldsymbol{R}(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\} \\
&+ \frac{1}{\mu}\lim_{k\to\infty}\mathrm{tr}\{(\boldsymbol{B}_k - \boldsymbol{A}_k)(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\}.
\end{aligned}
\tag{31}
$$

Since $\mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\} = \mathrm{tr}\{\boldsymbol{R}_v\boldsymbol{R}\}$, we can further rewrite (31) as

$$
\begin{aligned}
\mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\}&\left(2 - \mu\,\mathrm{tr}\{\boldsymbol{R}(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\}\right) \\
&= \mu\sigma_n^2\mathrm{tr}\{\boldsymbol{R}(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\} \\
&+ \frac{1}{\mu}\lim_{k\to\infty}\mathrm{tr}\{(\boldsymbol{B}_k - \boldsymbol{A}_k)(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\}.
\end{aligned}
\tag{32}
$$

Having in mind that the excess MSE ξ is found to be $\xi = \mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\}$, we obtain from (32) the following expression for ξ:

$$
\xi = \mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_v\} = \frac{\eta}{2-\eta}\sigma_n^2 + \frac{\beta-\alpha}{\mu(2-\eta)}
\tag{33}
$$

where $\eta \triangleq \mu\,\mathrm{tr}\{\boldsymbol{R}(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\}$, $\beta \triangleq \lim_{k\to\infty}\beta_k$, $\alpha \triangleq \lim_{k\to\infty}\alpha_k$, $\beta_k \triangleq \mathrm{tr}\{\boldsymbol{B}_k(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\}$, and $\alpha_k \triangleq \mathrm{tr}\{\boldsymbol{A}_k(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\}$.

We now further examine variables β_k and α_k. The matrix $\boldsymbol{B}_k(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}$ can be expressed as

$$
\begin{aligned}
\boldsymbol{B}_k&(\boldsymbol{I}-\mu\boldsymbol{R})^{-1} \\
&= \rho_r^2\left(E\left[\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_r + |\boldsymbol{w}_{k-1}|}\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r + |\boldsymbol{w}_{k-1}^T|}\right](\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\right).
\end{aligned}
\tag{34}
$$

Using (34), we obtain

$$
\beta_k = \text{tr}\{\boldsymbol{B}_k(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\}
$$
$$
= \rho_r^2\left(E\left[\text{tr}\left\{\frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\right\}\right]\right). \quad (35)
$$

Moreover, β_k in (35) can also be written as

$$
\beta_k = \rho_r^2\left(E\left[\text{tr}\left\{\frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\right\}\right]\right). \quad (36)
$$

The matrix $\boldsymbol{I}-\mu\boldsymbol{R}$ is symmetric, and its eigenvalue decomposition can be written as $\boldsymbol{I}-\mu\boldsymbol{R}=\boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{U}^T$ with $\boldsymbol{U}$ being an orthonormal matrix of eigenvectors and $\boldsymbol{\Gamma}$ being a diagonal matrix of eigenvalues. Therefore, $(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}=\boldsymbol{U}\boldsymbol{\Gamma}^{-1}\boldsymbol{U}^T$ and β_k from Eq. (36) can be written as

$$
\beta_k = \rho_r^2\left(E\left[\text{tr}\left\{\frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}\boldsymbol{U}\boldsymbol{\Gamma}^{-1}\boldsymbol{U}^T\frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\right\}\right]\right)
$$
$$
= \rho_r^2\left(E\left[\text{tr}\left\{\boldsymbol{\Gamma}^{-1}\boldsymbol{U}^T\frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}\boldsymbol{U}\right\}\right]\right). \quad (37)
$$

Let λ_{max} be the largest eigenvalue of the covariance matrix $\boldsymbol{R}$. Also, let μ be small enough such that $(1-\mu\lambda_{max})^{-1}$ is positive. In (37), since $\boldsymbol{\Gamma}^{-1}$ is a diagonal matrix whose diagonal elements are all non-negative and less than or equal to $(1-\mu\lambda_{max})^{-1}$, we have

$$
\beta_k \leq \frac{\rho_r^2}{1-\mu\lambda_{max}}
$$
$$
\times\left(E\left[\text{tr}\left\{\boldsymbol{U}^T\frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}\boldsymbol{U}\right\}\right]\right). \quad (38)
$$

Note that

$$
\text{tr}\left\{\boldsymbol{U}^T\frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}\boldsymbol{U}\right\}
$$
$$
= \text{tr}\left\{\frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}\boldsymbol{U}\boldsymbol{U}^T\frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\right\}
$$
$$
= \frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}\frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r+|\boldsymbol{w}_{k-1}|}
$$
$$
\leq \frac{\text{sgn}(\boldsymbol{w}_k^T)\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r^2} \leq \frac{N}{\epsilon_r^2}. \quad (39)
$$

Substituting (39) in (38), the following bound on β_k can be finally obtained:

$$
\beta_k \leq \frac{N\rho_r^2}{\epsilon_r^2(1-\mu\lambda_{max})}. \quad (40)
$$

Moreover, β_k in (37) can also be written as

$$
\beta_k = \rho_r^2(E[\text{tr}\{\boldsymbol{z}_k^T\boldsymbol{z}_k\}]) = \rho_r^2(E[\|\boldsymbol{z}_k\|_2^2]) \quad (41)
$$

where $\boldsymbol{z}_k$ is defined as

$$
\boldsymbol{z}_k \triangleq \boldsymbol{\Gamma}^{-1/2}\boldsymbol{U}^T\frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r+|\boldsymbol{w}_{k-1}|} \quad (42)
$$

and $\|\cdot\|_2$ stands for the Euclidean norm of a vector. Therefore, it can be seen from (41) that β_k is non-negative. Since, β_k is upper bounded and non-negative, so is β.

The variable α_k can be derived as

$$
\alpha_k = \text{tr}\{\boldsymbol{A}_k(\boldsymbol{I}-\mu\boldsymbol{R})^{-1}\} = \rho_r\left(E\left[\text{tr}\left\{\boldsymbol{v}_k\frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}+\frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\boldsymbol{v}_k^T\right\}\right]\right)
$$
$$
= 2\rho_r\left(E\left[\text{tr}\left\{\boldsymbol{v}_k\frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}\right\}\right]\right)
$$
$$
= 2\rho_r\left(E\left[\text{tr}\left\{\boldsymbol{w}_k\frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}\right\}\right]-E\left[\text{tr}\left\{\boldsymbol{w}\frac{\text{sgn}(\boldsymbol{w}_k^T)}{\epsilon_r+|\boldsymbol{w}_{k-1}^T|}\right\}\right]\right). \quad (43)
$$

Assuming that $\lim_{k\to\infty}E[\text{sgn}(\boldsymbol{w}_k)]=\text{sgn}(\boldsymbol{w})$ which is a common assumption and it is, for example, the same as in [7], α_k in (43) can be written as

$$
\alpha_k = 2\rho_r\left(E\left[\left\|\frac{\boldsymbol{w}_k}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\right\|_1\right]-E\left[\left\|\frac{\boldsymbol{w}}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\right\|_1\right]\right). \quad (44)
$$

Defining $\beta' \triangleq \beta/\rho_r^2$, and $\alpha' \triangleq \alpha/\rho_r$, the excess MSE equation of (33) can be rewritten as

$$
\xi = \frac{\eta}{2-\eta}\sigma_n^2+\frac{\beta'\rho_r}{\mu(2-\eta)}\left(\rho_r-\frac{\alpha'}{\beta'}\right) \quad (45)
$$

where β' is non-negative and upper bounded by $N/\epsilon_r^2(1-\mu\lambda_{max})$, and α' is given as

$$
\alpha' = \lim_{k\to\infty}2\left(E\left[\left\|\frac{\boldsymbol{w}_k}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\right\|_1\right]\right.
$$
$$
\left.-E\left[\left\|\frac{\boldsymbol{w}}{\epsilon_r+|\boldsymbol{w}_{k-1}|}\right\|_1\right]\right). \quad (46)
$$

It can be seen from (45) that if α' is positive, then choosing ρ_r in a way that $\rho_r < \alpha'/\beta'$ can lead to the excess MSE of the reweighted l_1−norm penalized LMS algorithm being smaller than that of the standard LMS algorithm given in (6). The following example shows how the value of α' varies with respect to the sparsity level of the CIR that is being estimated.

Example 1: A time sparse CIR of length $N=16$ whose sparsity level varies from $S=1$ to $S=16$ is considered in this example. The nonzero entries of the CIR take the values of 1 or -1 with equal probabilities each equal to half. In order to ensure a constant value for the term $\eta\sigma_n^2/(2-\eta)$ in the excess MSE equation of (45) for different values of sparsity S, σ_n^2 is a constant set to 0.01. The step size μ is set to 0.05, while $\rho_r = 5 \times 10^{-4}$ and $\epsilon_r = 0.05$ in (10). Elements of the training sequence $\boldsymbol{x}_k$ are chosen with equal probability from the set $\{1, -1\}$. Table 1 shows the value of α' after 250

Table 1
Value of α' for different sparsity levels.

S	1	2	3	4	5	6	7	8
α'	3.23	2.99	2.74	2.45	2.11	1.74	1.32	0.89
S	9	10	11	12	13	14	15	16
α'	0.39	-0.17	-0.79	-1.46	-2.23	-3.10	-4.07	-5.20

iterations of the reweighted l_1−norm penalized LMS algorithm for different sparsity levels.

The results in Table 1 show that as the CIR becomes less and less sparse, i.e., as S increases, α' becomes smaller to a point that it takes a negative value. Therefore, based on (45) we can expect a smaller excess MSE for the reweighted l_1−norm penalized LMS algorithm compared to that of the standard LMS algorithm providing that the sparsity level is small enough so that α' is positive.

5. Simulation results

In this section we compare the performance of different channel estimation algorithms for several scenarios. The algorithms being considered here are the ZA-LMS and RZA-LMS algorithms of [7] as well as the proposed reweighted l_1−norm penalized LMS algorithm and the l_p-pseudo-norm penalized LMS algorithm [24]. The standard LMS algorithm is also included for comparison in our simulation figures. The performance of the so-called *oracle* LMS is reported in the first simulation example as a lower bound for all sparsity-aware algorithms. In oracle LMS, the positions of the nonzero taps of the CIR are assumed to be known before hand.

The cost function of ZA-LMS can be written as $L_k^{\mathrm{ZA}} \triangleq (1/2)e_k^2 + \gamma_{\mathrm{ZA}} \| \boldsymbol{w}_k \|_1$, where γ_{ZA} is the weight associated with the penalty term. The CIR is assumed to be sparse in the time domain and the cost function L_k^{ZA} is convex. The algorithm has the following update equation:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{ZA}} \operatorname{sgn}(\boldsymbol{w}_k) \tag{47}$$

where $\rho_{\mathrm{ZA}} \triangleq \mu \gamma_{\mathrm{ZA}}$.

The RZA-LMS algorithm uses a logarithmic penalty term. The modified cost function of the algorithm is $L_k^{\mathrm{RZA}} \triangleq (1/2) e_k^2 + \gamma_{\mathrm{RZA}} \sum_{i=1}^{N} \log(1 + [\boldsymbol{w}_k]_i / \epsilon'_{\mathrm{RZA}})$, where $[\boldsymbol{w}_k]_i$ is the i-th element of the vector $\boldsymbol{w}_k$ and γ_{RZA} and ϵ'_{RZA} are some positive numbers. Note that the same penalty term is also used, for example, in [28]. The update equation for the RZA-LMS is

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{RZA}} \frac{\operatorname{sgn}(\boldsymbol{w}_k)}{1 + \epsilon_{\mathrm{RZA}}|\boldsymbol{w}_k|} \tag{48}$$

where $\rho_{\mathrm{RZA}} \triangleq \mu \gamma_{\mathrm{RZA}} \epsilon_{\mathrm{RZA}}$ and $\epsilon_{\mathrm{RZA}} \triangleq 1/\epsilon'_{\mathrm{RZA}}$. Note that the cost function of the RZA-LMS method is not convex that makes the convergence and consistency analysis problematic.

Although only time domain sparsity is considered in [7], the ZA-LMS algorithm, for example, can be easily extended to an arbitrary sparsity basis. Let $\boldsymbol{\Psi}$ be the $N \times N$ orthonormal matrix denoting a specific sparsity basis. The CIR $\boldsymbol{w}$ is sparse in the sparsity domain $\boldsymbol{\Psi}$ if its representation in $\boldsymbol{\Psi}$, that is, the vector $\boldsymbol{\Psi}\boldsymbol{w}$, has only few nonzero components. The ZA-LMS cost function can be rewritten then as $L_k^{\mathrm{ZA}} \triangleq (1/2)e_k^2 + \gamma_{\mathrm{ZA}} \| \boldsymbol{\Psi}\boldsymbol{w}_k \|_1$, and the update equation becomes

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{ZA}} \operatorname{sgn}(\boldsymbol{\Psi}\boldsymbol{w}_k)\boldsymbol{\Psi} \tag{49}$$

where $\operatorname{sgn}(\boldsymbol{\Psi}\boldsymbol{w}_k)$ as well as $\operatorname{sgn}(\boldsymbol{\Psi}\boldsymbol{w}_k)\boldsymbol{\Psi}$ are row vectors.

In [24], we considered the l_p-pseudo-norm of $\boldsymbol{w}_k$ with $0 < p < 1$ as the penalty term introduced into the cost function of the standard LMS. The cost function of the l_p-pseudo-norm penalized LMS is then expressed as $L_k^{l_p} \triangleq (1/2)e_k^2 + \gamma_p \| \boldsymbol{w}_k \|_p$, where $\| \cdot \|_p$ stands for the l_p-pseudo-norm of a vector and γ_p is the corresponding weight term.

Using gradient descent, the update equation based on (5) can be derived as

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{p}} \frac{(\| \boldsymbol{w}_k \|_p)^{1-p} \operatorname{sgn}(\boldsymbol{w}_k)}{|\boldsymbol{w}_k|^{(1-p)}} \tag{50}$$

where $\rho_{\mathrm{p}} = \mu \gamma_{\mathrm{p}}$. In practice, we need to impose an upper bound on the last term in (50) in the situation when an entry of $\boldsymbol{w}_k$ approaches zero, which is the case for a sparse CIR. Then the update equation (50) is modified as

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{p}} \frac{(\| \boldsymbol{w}_k \|_p)^{1-p} \operatorname{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{p}} + |\boldsymbol{w}_k|^{(1-p)}} \tag{51}$$

where ϵ_{p} is a value which is used to upper bound the last term in (50).

5.1. Simulation example 1: time sparse channel estimation

In this example, we consider the problem of estimating a CIR of length $N=16$. The CIR is assumed to be sparse in the time domain. Two different sparsity levels of $S=1$ and $S=4$ are considered. The positions of the nonzero taps in the CIR are chosen randomly. The value of each nonzero tap is a zero mean Gaussian random variable with a variance of 1.

Two different signal-to-noise ratio (SNR) values of 10 dB and 20 dB are considered. For the l_p-pseudo-norm penalized LMS algorithm, p is chosen to be 1/2 with $\epsilon_{\mathrm{p}} = 0.05$ and $\rho_{\mathrm{p}} = 2 \times 10^{-4}$. The parameters of the reweighted l_1−norm penalized LMS algorithm are set to $\rho_{\mathrm{r}} = 2 \times 10^{-4}$ and $\epsilon_{\mathrm{r}} = 0.05$. For the ZA-LMS and the RZA-LMS algorithms, $\rho_{\mathrm{ZA}} = 5 \times 10^{-4}$, $\rho_{\mathrm{RZA}} = 4 \times 10^{-3}$, and $\epsilon_{\mathrm{RZA}} = 25$. Parameter values for the ZA-LMS and RZA-LMS algorithms are optimized through simulations. The step size is set to $\mu = 0.05$ for all algorithms. The measure of performance is the MSE between the actual and estimated CIR. Simulation results are averaged over 10 000 simulation runs to smooth out the curves.

Fig. 2 shows the MSE versus the number of iterations for different estimation algorithms for the case when the sparsity level is $S=1$. It is expected that the oracle LMS outperforms all sparsity-aware algorithms as well as the standard LMS. The simulation results conform it. Outside the oracle LMS, it can be seen that for both SNR values tested, the l_p-pseudo-norm penalized LMS algorithm has the best performance followed by the reweighted l_1−norm penalized LMS algorithm, and then by the RZA-LMS, ZA-LMS, and standard LMS algorithms. The MSEs of the RZA-LMS and reweighted l_1−norm penalized LMS algorithms are close to each other. As the SNR increases, the performance of all the algorithms tested improves as expected. Also, it can be seen in Fig. 2 that the performance gap between the MSE of the standard LMS algorithm and the MSEs for the rest of the algorithms increases as SNR increases. The l_p-pseudo-norm penalized LMS and reweighted l_1−norm penalized LMS algorithms have faster convergence rate compared to the standard LMS algorithm.

Fig. 3 shows the simulation results for the case when the sparsity level is set to $S=4$. The parameter choices for all the algorithms tested are the same as in the previous case. Most of the observations from Fig. 2 also hold for this case of increased sparsity level. However, increasing the sparsity level of the CIR leads to a decrease in the
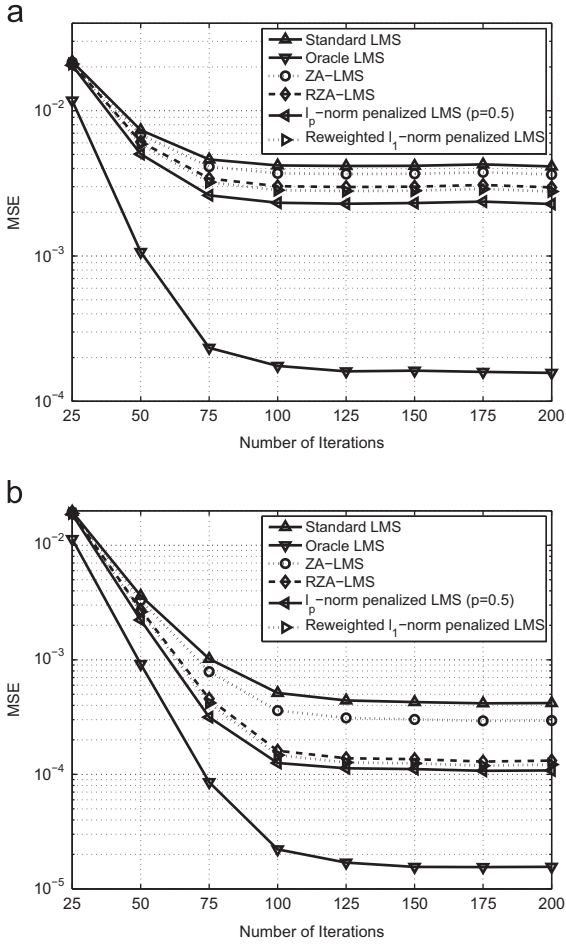
Fig. 2. Example 1, Case 1: MSEs of different estimation algorithms versus number of iterations ($S=1$). (a) 10 dB SNR. (b) 20 dB SNR.

Fig. 3. Example 1, Case 2: MSEs of different estimation algorithms versus number of iterations ($S=4$). (a) 10 dB SNR. (b) 20 dB SNR.

performance gap between the sparsity-aware LMS algorithms and the standard LMS algorithm.

Overall, the proposed reweighted l_1−norm penalized LMS algorithm performs better than the RZA-LMS and significantly better than the RA-LMS. Both the proposed reweighted l_1−norm penalizes LMS and the RA-LMS algorithms use l_1−norm penalty for enforcing sparsity, but the proposed algorithm uses the reweighting on the top. Thus, the corresponding performance improvement of the proposed algorithm as compared to the RA-LMS algorithm is due to the reweighting only. The RZA-LMS algorithm uses a different nonconvex penalty term, and it is proper to compare it to the other proposed l_p-pseudo-norm ($p < 1$) penalized LMS algorithm, where the penalty term is also nonconvex. We can see the significant performance improvement for the other proposed algorithm versus the RZA-LMS algorithm.

5.2. Simulation example 2: arbitrary sparsity basis

The ZA-LMS and RZA-LMS algorithms in the form derived in [7] are only applied to the case when the channel is sparse in the time domain. However, these algorithms as well as the l_p-pseudo-norm penalized LMS
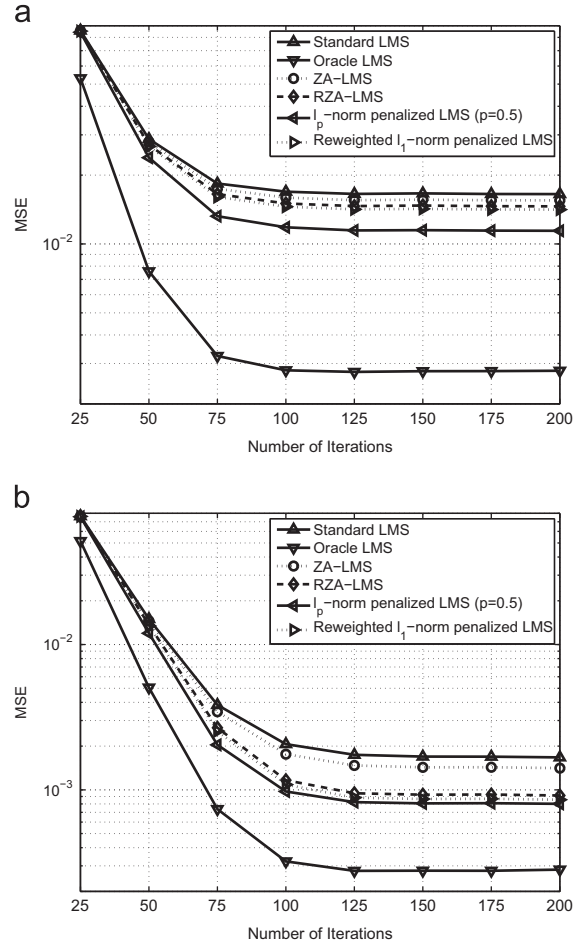
and reweighted l_1−norm penalized LMS algorithms can be modified to accommodate the case of an arbitrary sparsity basis. Consider the ZA-LMS algorithm in the case when the CIR is sparse in a sparsity domain denoted by $\mathbf{\Psi}$. The CIR representation in $\mathbf{\Psi}$, i.e., the vector $\mathbf{w}_\Psi = \mathbf{\Psi w}$, is a sparse vector and it has a few nonzero entries. The corresponding update equation for the ZA-LMS algorithm is given by (49).

The update equation for the reweighted l_1−norm penalized LMS algorithm becomes

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu e_k \mathbf{x}_k - \rho_\mathrm{r} \frac{\mathrm{sgn}(\mathbf{\Psi w}_k)\mathbf{\Psi}}{\epsilon_\mathrm{r} + |\mathbf{\Psi w}_{k-1}|}. \tag{52}$$

Finally, the modified update equation of the l_p-pseudo-norm penalized LMS algorithm can be derived as

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu e_k \mathbf{x}_k - \rho_\mathrm{p} \frac{(\|\mathbf{\Psi w}_k\|_p)^{1-p} \mathrm{sgn}(\mathbf{\Psi w}_k)\mathbf{\Psi}}{\epsilon_\mathrm{p} + |\mathbf{\Psi w}_k|^{(1-p)}}. \tag{53}$$

In this simulation example, a CIR of length $N=16$ with the sparsity level of $S=2$ is being estimated which is sparse in the discrete cosine transform (DCT) domain. The positions of nonzero taps in the DCT domain are chosen randomly. The values of the nonzero elements in the DCT domain are set to 1 or -1 with the same probabilities each

equal to half. The algorithms being compared here are the ZA-LMS, RZA-LMS, l_p-pseudo-norm penalized LMS, reweighted l_1−norm penalized LMS, and standard LMS algorithms. As in the first simulation scenario, two different SNR values of 10 and 20 dBs are tested. Parameter choices for the 10 dB SNR case are as follows. For the l_p-pseudo-norm penalized LMS algorithm, $p=1/2$, $\epsilon_p = 0.05$, and $\rho_p = 2 \times 10^{-4}$. Parameters of the reweighted l_1−norm penalized LMS algorithm are $\rho_r = 2 \times 10^{-4}$ and $\epsilon_r = 0.05$. For the ZA-LMS and the RZA-LMS algorithms, the values are $\rho_{ZA} = 5 \times 10^{-4}$, $\rho_{RZA} = 4 \times 10^{-3}$, and $\epsilon_{RZA} = 25$. The step size μ is set to 0.05. For the 20 dB SNR case, ρ_r, ρ_p, and ρ_{RZA} are reduced by half.

The MSE curves in Fig. 4 are averaged over 10 000 simulation runs. The same conclusions as in Simulation Example 1 hold here as well. For the SNR of 10 dB SNR, the l_p-pseudo-norm penalized LMS algorithm outperforms all the other algorithms followed by the reweighted l_1−norm penalized LMS algorithm, and then by the RZA-LMS and ZA-LMS algorithms. However, when the SNR is set to 20 dB, the reweighted l_1−norm penalized LMS and RZA-LMS algorithms show a better performance than the l_p-pseudo-norm penalized LMS algorithm.

5.3. Simulation example 3: effect of sparsity level on the performance of the reweighted l_1−norm penalized LMS algorithm

In this example, we study the effect that the increasing sparsity level of CIR has on the performance of the reweighted l_1−norm penalized LMS algorithm. A CIR is assumed to be sparse in the time domain and it is of length $N=16$. The sparsity level varies from 2 to 8. The positions of the nonzero taps of the CIR are chosen randomly and the values of nonzero taps are set to 1 or -1 with equal probability each equal to half. Parameters of the reweighted l_1−norm penalized LMS algorithm are $\rho_r = 2 \times 10^{-4}$ and $\epsilon_r = 0.05$. The step size μ is set to 0.05. Variance of the additive noise term n_k is $\sigma_n^2 = 0.01$. Excess MSE is used as a performance measure in this example. We have chosen a constant variance σ_n^2 for the noise in order to make sure that the standard LMS algorithm has the same
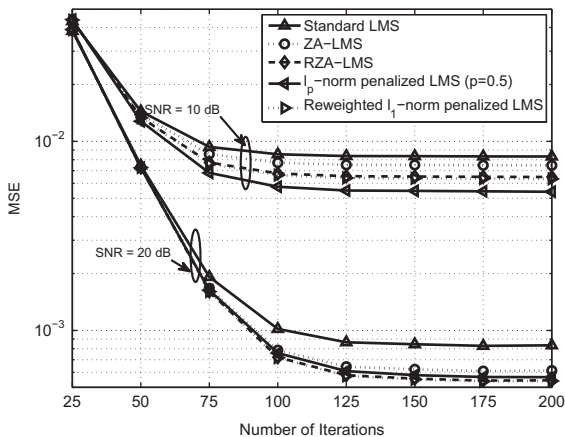

Fig. 4. Example 2: MSEs of estimation algorithms versus number of iterations for a DCT sparse channel with $S=2$.
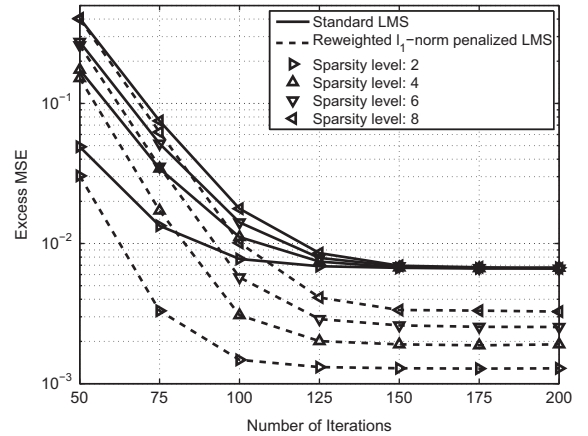

Fig. 5. Example 3: Excess MSE versus number of iterations.

excess MSE regardless of the sparsity level of the channel. The excess MSE curves are averaged over 10 000 simulation runs. According to (5), the excess MSE can be derived as $\xi_k = \text{tr}\{\boldsymbol{R}E[\boldsymbol{v}_k\boldsymbol{v}_k^T]\}$. In this simulation example with $\boldsymbol{x}_k$ being an i.i.d. binary phase-shift keying (BPSK) sequence, the covariance matrix $\boldsymbol{R}$ becomes identity, and therefore, ξ_k can be evaluated as $\text{tr}\{E[\boldsymbol{v}_k\boldsymbol{v}_k^T]\}$.

Fig. 5 shows the excess MSE versus the number of iterations for the standard LMS and reweighted l_1−norm penalized LMS algorithms when the CIR sparsity level is varied from 2 to 8. It can be seen that the standard LMS algorithm results in the same excess MSE regardless of the sparsity level of the CIR. However, the excess MSE of the reweighted l_1−norm penalized LMS algorithm increases with increasing sparsity level which is due to the fact that the value of α' in Eq. (46) is decreasing. For example, α' is equal to 2.7, 2.3, 2.0, and 1.6 for sparsity levels of 2, 4, 6, and 8, respectively, after 150 iterations. It can be also seen that in all cases, the reweighted l_1−norm penalized LMS algorithm outperforms the standard LMS algorithm.

6. Conclusions

Sparse channel estimation problem has been considered in this paper and the reweighted l_1−norm penalized LMS algorithm has been introduced and analyzed. Quantitative analysis of the reweighted l_1−norm penalized LMS algorithm and the attainable excess MSE have been presented. The excess MSE result shows that the reweighted l_1−norm penalized LMS algorithm outperforms the standard LMS algorithm for the case of sparse CIR. The analysis has enabled us also to answer the question of what is the maximum sparsity level in the channel for which the reweighted l_1−norm penalized LMS algorithm is better than the standard LMS. Update equations of the reweighted l_1−norm penalized LMS, ZA-LMS, and the l_p-pseudo-norm penalized LMS algorithms have been generalized to the case of an arbitrary sparsity basis. Simulation results for the DCT sparse channel are given along with simulation results for the time sparse channel. The performance of the reweighted l_1−norm penalized LMS algorithm has been compared to that of the standard LMS, ZA-LMS, RZA-LMS

algorithms, and our earlier proposed l_p-pseudo-norm penalized LMS algorithm through computer simulations. These results show that the reweighted l_1-norm penalized LMS algorithm outperforms the standard LMS, ZA-LMS, and RZA-LMS algorithms in all examples. It is also worth mentioning that variable step size is known to lead to better steady state error and therefore, better performance. Thus, as a further extension, the variable step size feature can be easily added to the proposed algorithm in the same way as it has been added to the RA-LMS in [29].

References

[1] B. Widrow, S.D. Stearns, Adaptive Signal Processing, Prentice Hall, Englewood Cliffs, NJ, 1985.
[2] S. Haykin, Adaptive Filter Theory, Prentice Hall, Upper Saddle River, NJ, 2002.
[3] N.J. Bershad, J.M. Bermudez, J.Y. Tourneret, Stochastic analysis of the LMS algorithm for system identification with subspace inputs, IEEE Trans. Signal Process. 56 (March (3)) (2008) 1018–1027.
[4] H.I.K. Rao, B. Farhang-Boroujeny, Fast LMS/Newton algorithms for stereophonic acoustic echo cancellation, IEEE Trans. Signal Process. 57 (August (8)) (2009) 2919–2930.
[5] S. Coleri, M. Ergen, A. Puri, A. Bahai, Channel estimation techniques based on pilot arrangement in OFDM systems, IEEE Trans. Broadcast. 48 (September (3)) (2002) 223–229.
[6] O. Macchi, N. Bershad, M. Mboup, Steady-state superiority of LMS over LS for time-varying line enhancer in noisy environment, IEE Proc. Radar Signal Process. 138 (August (4)) (1991) 354–360.
[7] Y. Chen, Y. Gu, A.O. Hero, Sparse LMS for system identification, in: Proceedings of IEEE ICASSP, Taipei, Taiwan, April 2009, pp. 3125–3128.
[8] Y. Gu, J. Jin, S. Mei, l_0 norm constraint LMS algorithm for sparse system identification, IEEE Signal Process. Lett. 16 (September (9)) (2009) 774–777.
[9] K. Shi, P. Shi, Convergence analysis of sparse LMS algorithms with l_1-norm penalty based on white input signal, Signal Process. 90 (December (12)) (2010) 3289–3293.
[10] G. Su, J. Jin, Y. Gu, J. Wang, Performance analysis of l_0 norm constraint least mean square algorithm, IEEE Trans. Signal Process. 60 (May (9)) (2012) 2223–2235.
[11] K. Shi, X. Ma, Transform domain LMS for sparse system identification, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Dallas, USA, March 2010, pp. 3714–3717.
[12] J. Yang, G.E. Sobelman, Sparse LMS with segment zero attractors for adaptive estimation of sparse signals, in: Proceedings of IEEE Asia Pacific Conference on Circuits and Systems, Kuala Lumpur, Malaysia, December 2010, pp. 422–425.
[13] Y. Murakami, M. Yamagishi, M. Yukawa, I. Yamada, A sparse adaptive filtering using time-varying soft-thresholding techniques, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Dallas, USA, March 2010, pp. 3734–3737.
[14] O. Taheri, S.A. Vorobyov, Decimated least mean squares for frequency sparse channel estimation, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Kyoto, Japan, March 25–30, 2012, pp. 3181–3184.
[15] N.J. Bershad, A. Bist, Fast coupled adaptation for sparse impulse responses using a partial Haar transform, IEEE Trans. Signal Process. 53 (March (3)) (2005) 966–976.
[16] R.K. Martin, W.A. Sethares, R.C. Williamson, C.R. Johnson, Exploiting sparsity in adaptive filters, IEEE Trans. Signal Process. 50 (August (8)) (2002) 1883–1894.
[17] S. Chouvardas, K. Slavakis, Y. Kopsinis, S. Theodoridis, A sparsity promoting adaptive algorithm for distributed learning, IEEE Trans. Signal Process. 60 (October (10)) (2012) 5412–5425.
[18] P. Di Lorenzo, A.H. Sayed, Sparse distributed learning based on diffusion adaptation, IEEE Trans. Signal Process. 61 (March (6)) (2013) 1419–1433.
[19] D. Angelosante, J.A. Bazerque, G.B. Giannakis, Online adaptive estimation of sparse signals: where RLS meets the l_1-norm, IEEE Trans. Signal Process. 58 (July) (2010) 3436–3447.
[20] B. Dumitrescu, A. Onose, P. Helin, I. Tabus, Greedy sparse RLS, IEEE Trans. Signal Process. 60 (May (5)) (2012) 2194–2207.
[21] N. Vaswani, Kalman filtered compressed sensing, in: Proceedings of IEEE ICIP, San Diego, USA, October 2008, pp. 893–896.
[22] E.J. Candes, M.B. Wakin, An introduction to compressive sampling, IEEE Signal Process. Mag. 25 (March (2)) (2008) 21–30.
[23] D. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (April) (2006) 1289–1306.
[24] O. Taheri, S.A. Vorobyov, Sparse channel estimation with l_p-norm and reweighted l_1-norm penalized least mean squares, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Prague, Czech Republic, May 2011, pp. 2864–2867.
[25] E.J. Candes, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted l_1 minimization, J. Fourier Anal. Appl. 14 (December (5)) (2008) 877–905.
[26] S.C. Douglas, W. Pan, Exact expectation analysis of the LMS adaptive filter, IEEE Trans. Signal Process. 43 (December (12)) (1995) 2863–2871.
[27] L.L. Horowitz, K.D. Senne, Performance advantage of complex LMS for controlling narrow-band adaptive arrays, IEEE Trans. Acoust. Speech Signal Process. 29 (June (3)) (1981) 722–736.
[28] B. Liu, M.D. Sacchi, Minimum weighted norm interpolation of seismic records, Geophysics 69 (November–December (6)) (2004) 1560–1568.
[29] M.O. Bin Saeed, A. Zerguine, A variable step size strategy for sparse system identification, in: Proceedings of 10th International Multi-Conference on Systems, Signals, and Devices, Hammamet, Tunisia, March 2013, pp. 1–4.