SPARSE SLIDING-WINDOW RLS ADAPTIVE FILTER WITH DYNAMIC REGULARIZATION

Y. V. Zakharov

University of York
Department of Electronics
York, UK

V. H. Nascimento

Univ. of São Paulo
Dept. of Electronic Systems Eng.
São Paulo, Brazil

ABSTRACT

Recently, a sliding-window RLS (SRLS) adaptive algorithm based on dynamic regularization, involving a time- and tap-varying diagonal loading (VDL), has been proposed, which is equivalent to the proportionate affine projection algorithm (PAPA) used for sparse identification. The complexity of this SRLS algorithm is significantly lower than the PAPA complexity. However, its identification performance (the same as the PAPA performance) does not approach that of the oracle SRLS algorithm. We propose here a new version of the SRLS-VDL algorithm that does achieve next-to-oracle performance. We arrive at this algorithm by minimizing the least squares cost function with a penalty. We also propose a modified penalty that takes into account both the sparsity of the unknown system and additive noise. Numerical results with speech signals in an acoustic echo cancellation scenario show that the proposed algorithm outperforms other sparse estimation techniques and its performance is close to the oracle performance.

I. INTRODUCTION

Compared to the classical least squares (LS) identification, identification algorithms that exploit system sparsity can achieve a higher accuracy. Multiple sparse adaptive filters have been proposed in the literature and significant attention in adaptive sparse identification has recently been devoted to the application in echo cancelation [1]–[5]. In general, the complexity of sparse adaptive algorithms is higher than that of their LS counterparts. Therefore, reduction of the algorithm complexity is an important issue.

In this paper, we consider adaptive filtering for sparse identification. For this purpose, proportionate affine projection algorithms (PAPAs) are often used [4]. To achieve a high identification performance, the projection order of such algorithms should be high [6], which results in high complexity. It has recently been shown that PAPAs are equivalent to specific sliding-window RLS (SRLS) adaptive algorithms, namely SRLS algorithms with dynamic regularization based on time- and tap-varying diagonal loading [6]. A low-complexity SRLS algorithm was proposed in [6], exploiting properties of the dichotomous coordinate descent (DCD) algorithm [7]; we will call it here the SRLS-VDL1-DCD algorithm. The SRLS-VDL1-DCD performance approaches the PAPA performance, but with a lower complexity which is linear in the filter length and independent of the projection order. However, the PAPA performance is not close to the oracle performance, i.e., the best performance that can be obtained when the zero filter taps of the unknown system are perfectly known. Therefore, the SRLS-VDL1-DCD algorithm in [6] also cannot approach the oracle performance. In [8], we proposed an exponentially-weighted RLS algorithm with time- and tap-varying diagonal loading with a complexity linear in the filter length; the algorithm exploits an ℓ_0-penalty to attract the sparsity. However, RLS with exponential weight has a long tail that limits its tracking capability.

In this paper, we propose a new version of the SRLS adaptive algorithm with dynamic diagonal loading that combines the best properties of algorithms in [6] and [8]. Instead of starting from the PAPA recursion as in [6], we arrive at our algorithm starting from the optimization of the least squares cost function with penalties, using ideas similar to those previously used for deriving the exponentially-weighted RLS algorithm with dynamic diagonal loading in [8]. We also propose a modified penalty function (defining the diagonal loading) that allows taking into account both the sparsity of the unknown system and additive noise. Numerical results conducted with speech signals in a typical acoustic echo cancelation scenario show that the proposed (we will call it SRLS-VDL2-DCD) algorithm outperforms other techniques, in particular it has fast convergence and tracking, and its performance is close to the oracle performance.

Notation: Capital and small bold fonts denote matrices and vectors, respectively; e.g., $\mathbf{X}$ is a matrix and $\mathbf{x}$ a vector. Their elements are denoted as $X_{n,p}$ and x_n, respectively. $(\cdot)^T$ is the matrix transpose. $\mathbf{I}_M$ is an $M \times M$ identity matrix. $\mathbf{0}_{N,N}$ denotes an $N \times N$ matrix with all-zero entries.

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

II. SIGNAL MODEL AND REGULARIZED ADAPTIVE FILTERING SETUP

The signal model corresponding to system identification scenarios is described by the relationship

$$d(n) = \mathbf{h}_\circ^T(n)\mathbf{x}(n) + \nu(n), \quad (1)$$

where $\nu(n)$ is a zero-mean Gaussian random noise with variance σ^2. The vector $\mathbf{x}(n) = [x(n)\ x(n-1)\ \dots\ x(n-N+1)]^T$ represents the input to the system with unknown impulse response $\mathbf{h}_\circ(n)$ of length N, and the vector $\mathbf{h}_\circ(n)$ is sparse.

The adaptive filter at time n tries to find a vector $\mathbf{h}(n)$ minimizing the cost function

$$J[\mathbf{h}(n)] = f_{\text{LS}}[\mathbf{h}(n)] + f[\mathbf{h}(n)]. \quad (2)$$

The term $f_{\text{LS}}[\mathbf{h}(n)]$ is the LS fitting error, whereas the term $f[\mathbf{h}(n)]$ is a penalty that incorporates *a priori* information on the true solution. In the case of a sparse vector $\mathbf{h}(n)$, often the following penalties are used:

$$f[\mathbf{h}(n)] = \tau||\mathbf{h}(n)||_p, \quad 0 \le p \le 1, \quad (3)$$

where $||\mathbf{h}(n)||_p$ is the ℓ_p-norm:

$$||\mathbf{h}(n)||_p = \left(\sum_{k=0}^{N-1} |h_k(n)|^p\right)^{1/p},$$

$h_k(n)$ is the kth element of $\mathbf{h}(n)$, and τ is a regularization parameter controlling the balance between the LS fitting and the penalty. In noisy conditions, the penalty $f[\mathbf{h}(n)] = \varepsilon||\mathbf{h}(n)||_2^2$ is often used, where the regularization parameter ε relates to the noise level.

Let $\mathbf{x}(n)$ be an $N \times 1$ regressor vector and $d(n)$ be a desired signal at time instant n. Denote $\mathbf{X}(n) = [\mathbf{x}(n), \dots, \mathbf{x}(n-M+1)]^T$ and $\mathbf{d}(n) = [d(n), \dots, d(n-M+1)]^T$, where M is the sliding window length. The LS term of the cost function using a sliding window can be expressed as

$$\frac{1}{2}||\mathbf{X}(n)\mathbf{h}(n) - \mathbf{d}(n)||_2^2,$$

which after omitting terms independent of $\mathbf{h}(n)$ can be represented as

$$f_{\text{LS}}[\mathbf{h}(n)] = \frac{1}{2}\mathbf{h}^T(n)\mathbf{R}(n)\mathbf{h}(n) - \mathbf{h}^T(n)\mathbf{b}(n),$$

where $\mathbf{R}(n) = \mathbf{X}^T(n)\mathbf{X}(n)$ and $\mathbf{b}(n) = \mathbf{X}^T(n)\mathbf{d}(n)$.

In [6], starting from a generalized PAPA recursion, we arrived at an adaptive filtering algorithm that possesses the same performance as the PAPA, but with a complexity significantly lower than that of the PAPA. Besides, the complexity does not depend on the PAPA projection order M; thus, M can be high to improve the performance. The adaptive algorithm proposed in [6] is a sliding-window RLS (SRLS) algorithm with tap- and time-varying diagonal loading. It is shown in Table I; we will call it SRLS-VDL1-DCD.

Table I. SRLS-VDL1-DCD adaptive filter

	Initialization: $\hat{\mathbf{h}}(0) = \mathbf{0}_{N,1}$, $\hat{\mathbf{h}}(-1) = \mathbf{0}_{N,1}$, $\tilde{\hat{\mathbf{h}}}(0) = \mathbf{0}_{N,1}$, $\mathbf{r}(0) = \mathbf{0}_{N,1}$, $\mathbf{R}(0) = \mathbf{0}_{N,N}$
	for $n = 1, 2, \dots$
1.	$\mathbf{R}(n) = \mathbf{X}^T(n)\mathbf{X}(n)$
2.	Update the matrix $\mathbf{W}(n)$
3.	$\tilde{y}(n) = \tilde{\mathbf{h}}^T(n-1)\mathbf{x}(n)$
4.	$\tilde{y}_M(n) = \tilde{\mathbf{h}}^T(n-1)\mathbf{x}(n-M)$
5.	$\tilde{e}(n) = d(n) - \tilde{y}(n)$
6.	$\tilde{e}_M(n) = d(n-M) - \tilde{y}_M(n)$
7.	$\mathbf{r}(n) = \mathbf{r}(n-1) + \tilde{e}(n)\mathbf{x}(n) - \tilde{e}_M(n)\mathbf{x}(n-M) + \mathbf{W}(n)\hat{\mathbf{h}}(n-1) - \mathbf{W}(n-1)\hat{\mathbf{h}}(n-2) - \Delta\mathbf{W}(n)\tilde{\mathbf{h}}(n-1)$
8.	Solve $[\mathbf{R}(n) + \mathbf{W}(n)]\Delta\mathbf{h}(n) = \mathbf{r}(n)$ using DCD iterations to obtain $\Delta\tilde{\mathbf{h}}(n)$ and update $\mathbf{r}(n)$ and $\tilde{\mathbf{h}}(n) = \tilde{\mathbf{h}}(n-1) + \Delta\tilde{\mathbf{h}}(n)$
9.	$\hat{\mathbf{h}}(n) = (1-\mu)\hat{\mathbf{h}}(n-1) + \mu\tilde{\mathbf{h}}(n)$

The matrix $\mathbf{W}(n)$ is a diagonal matrix which differs for different PAPA adaptive filters. In particular, in the improved PAPA [4], [9], the kth diagonal element of $\mathbf{W}(n)$ is given by $w_k(n) = \tau/q_k(n)$, where $\tau > 0$ is a regularization parameter,

$$q_k(n) = \frac{1-\kappa}{2N} + \frac{1+\kappa}{2}\frac{|\hat{h}_k(n-1)|}{||\hat{\mathbf{h}}(n-1)||_1 + \epsilon}, \quad (4)$$

$-1 \le \kappa < 1$, and $\epsilon > 0$ is a small number added to avoid division by zero. The matrix $\Delta\mathbf{W}(n)$ is given by $\Delta\mathbf{W}(n) = \mathbf{W}(n) - \mathbf{W}(n-1)$. The step-size parameter μ is the same as in the PAPA ($\mu \in [0, 1]$). The system of equations at step 8 is solved using DCD iterations that do not require any multiplications or divisions [10]. They require about $2N_uN$ additions, where N_u is the number of DCD iterations. The algorithm is based on a fixed-point representation of the solution in an amplitude interval $[-H, +H]$ with M_b bits; see more details on the DCD algorithm, e.g., in [10], [11]. Without the weight update, the complexity of the SRLS-VDL1-DCD algorithm is $11N$ multiplications and $(10 + 2N_u)N$ additions [6], i.e., the complexity is linear in the filter length N.

III. OPTIMIZATION WITH A PENALTY

To find the optimal vector $\mathbf{h}(n)$ minimizing the cost function in (2), we compute its subgradient and make it equal zero. The subgradient of the cost function is given by [1]

$$\bigtriangledown_{\mathbf{h}(n)} J[\mathbf{h}(n)] = -\mathbf{b}(n) + \mathbf{R}(n)\mathbf{h}(n) + \mathbf{W}[\mathbf{h}(n)]\mathbf{h}(n), \quad (5)$$

where the first two terms are due to the LS fitting component and the third term is due to the penalty. For the penalty in (3), we have

$$\mathbf{W}[\mathbf{h}(n)] = \frac{p\tau}{2}\text{diag}[|\mathbf{h}(n)|^{p-2}]. \quad (6)$$

Then, the solution to the optimization problem at every time instant n can be found as a solution to the system of

equations

$$\{\mathbf{R}(n) + \mathbf{W}[\mathbf{h}(n)]\}\,\mathbf{h}(n) = \mathbf{b}(n), \quad n > 0. \tag{7}$$

The system in (7) is nonlinear and finding its solution is a complicated problem.

To simplify the problem, we use the approach from [8]. Firstly, we replace the matrix $\mathbf{W}[\mathbf{h}(n)]$ by a matrix $\mathbf{W}(n) = \text{diag}\{\mathbf{w}(n)\}$ that does not depend on $\mathbf{h}(n)$, but instead depends on $\hat{\mathbf{h}}(n-1)$, where $\hat{\mathbf{h}}(n-1)$ is an estimate of $\mathbf{h}(n-1)$. As a result, we convert the system of nonlinear equations to a system of linear equations as is done in [5], [8]. However, following [8] and different from [5], we only approximate the matrix $\mathbf{W}[\mathbf{h}(n)]$, i.e., the diagonal loading, that is, we use the approximation

$$\bigtriangledown_{\mathbf{h}(n)} f[\mathbf{h}(n)] = \mathbf{W}[\mathbf{h}(n)]\mathbf{h}(n) \approx \mathbf{W}(n)\mathbf{h}(n), \tag{8}$$

where $\mathbf{W}(n) = \mathbf{W}[\mathbf{h}(n-1)]$. Secondly, we use the DCD algorithm [7] to iteratively solve the linear system by reusing the solution found at the previous time instant $(n-1)$ as a *warm-start* for the solution at time n. This solver is proved to be efficient for accurate solving of linear systems with a low complexity that is linear in the filter length N [10], [11].

We start with deriving an algorithm that reuses the solution found at the previous instant $(n-1)$ as a *warm-start* for finding a solution at instant n when the diagonal matrix $\mathbf{W}(n)$ is arbitrarily varying in time. The derivation follows steps similar to that in [10] and [12].

When using an estimate $\hat{\mathbf{h}}(n-1)$ of $\mathbf{h}(n-1)$ as a *warm-start* for finding an estimate $\hat{\mathbf{h}}(n)$ of $\mathbf{h}(n)$, the new estimate can be represented as $\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \Delta\hat{\mathbf{h}}$. The increment vector $\Delta\hat{\mathbf{h}}$ is then found as a solution to the system $[\mathbf{R}(n) + \mathbf{W}(n)]\Delta\mathbf{h} = \mathbf{c}(n|n-1)$ with $\mathbf{c}(n|n-1)$ being the residual vector

$$\mathbf{c}(n|n-1) = \mathbf{b}(n) - \mathbf{R}(n)\hat{\mathbf{h}}(n-1) - \mathbf{W}(n)\hat{\mathbf{h}}(n-1). \tag{9}$$

An iterative solver of linear systems, in addition to an approximate solution $\hat{\mathbf{h}}(n-1)$, typically provides as a bypass product another residual vector [13]:

$$\begin{aligned}\mathbf{c}(n-1|n-1) &= \mathbf{b}(n-1) - \mathbf{R}(n-1)\hat{\mathbf{h}}(n-1) \\ &- \mathbf{W}(n-1)\hat{\mathbf{h}}(n-1).\end{aligned} \tag{10}$$

We use the vector $\mathbf{c}(n-1|n-1)$ for computing the vector $\mathbf{c}(n|n-1)$. We also take into account that, for SRLS algorithms, the matrix $\mathbf{R}(n) = \mathbf{R}(n-1) + \Delta\mathbf{R}(n)$ and vector $\mathbf{b}(n) = \mathbf{b}(n-1) + \Delta\mathbf{b}(n)$ are varying in time as follows [14], [15]:

$$\begin{aligned}\mathbf{R}(n) &= \mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n) - \mathbf{x}(n-M)\mathbf{x}^T(n-M), \\ \mathbf{b}(n) &= \mathbf{b}(n-1) + d(n)\mathbf{x}(n) - d(n-M)\mathbf{x}(n),\end{aligned}$$

Table II. SRLS-VDL2-DCD adaptive filter

	Initialization: $\hat{\mathbf{h}}(0) = \mathbf{0}_{N,1}$, $\mathbf{c}_r(0) = \mathbf{0}_{N,1}$, $\mathbf{R}(0) = \mathbf{0}_{N,N}$
	for $n = 1, 2, \ldots$
1.	Update the matrix $\mathbf{R}(n)$
2.	$y(n) = \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$
3.	$e(n) = d(n) - y(n)$
4.	$y_M(n) = \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n-M)$
5.	$e_M(n) = d(n-M) - y_M(n)$
6.	$\mathbf{c}(n) = \mathbf{c}_r(n-1) + e(n)\mathbf{x}(n) - e_M(n)\mathbf{x}(n-M)$ $-\,[\mathbf{W}(n) - \mathbf{W}(n-1)]\,\hat{\mathbf{h}}(n-1)$
7.	Solve $[\mathbf{R}(n) + \mathbf{W}(n)]\Delta\mathbf{h}(n) = \mathbf{c}(n)$ to obtain $\Delta\hat{\mathbf{h}}$ and $\mathbf{c}_r(n)$, and update $\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \Delta\hat{\mathbf{h}}(n)$
8.	Update the weight vector $\mathbf{w}(n)$

and, thus,

$$\Delta\mathbf{R}(n) = \mathbf{x}(n)\mathbf{x}^T(n) - \mathbf{x}(n-M)\mathbf{x}^T(n-M), \tag{11}$$

$$\Delta\mathbf{b}(n) = d(n)\mathbf{x}(n) - d(n-M)\mathbf{x}(n-M). \tag{12}$$

After some algebra (see more details in [10]), we obtain the relationship between $\mathbf{c}(n-1|n-1)$ and $\mathbf{c}(n|n-1)$:

$$\begin{aligned}\mathbf{c}(n|n-1) &= \mathbf{c}(n-1|n-1) + e(n)\mathbf{x}(n) \\ &- e_M(n)\mathbf{x}(n-M) \\ &- [\mathbf{W}(n) - \mathbf{W}(n-1)]\,\hat{\mathbf{h}}(n-1),\end{aligned} \tag{13}$$

where $e(n)$ is the *a priori* estimation error, $e(n) = d(n) - y(n)$, $e_M(n) = d(n-M) - y_M(n)$, $y(n)$ is the adaptive filter output at time instant n: $y(n) = \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$, and $y_M(n) = \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n-M)$. The structure of the adaptive filter can now be expressed as a set of steps shown in Table II.

Consider the computational complexity of this algorithm. For updating the matrix $\mathbf{R}(n)$ at step 1, for regressors $\mathbf{x}(n)$ with no structure, $\mathcal{O}(N^2)$ operations per sample are required. However, for regressors with a time-shifted structure, i.e., for transversal adaptive filters, the complexity is $\mathcal{O}(N)$ [10]. The other steps of the algorithm (apart from step 7) have a complexity $\mathcal{O}(N)$. For transversal adaptive filters, the overall complexity of the algorithm is $\mathcal{O}(N)$ operations per sample; more specifically, without the weight update, the algorithm requires $7N$ multiplications and $(8 + 2N_u)N$ additions per sample, which is lower than the complexity of the SRLS-VDL1-DCD algorithm. The weight update depends on the penalty, and its complexity is linear in the filter length.

IV. MODIFIED PENALTY FUNCTION

The ℓ_0-norm penalty is known to attract sparsity. It is ideal for noiseless sparse identification and is given by (3) with $p = 0$. However, the ℓ_0-norm penalty is very sensitive to noise, and as the noise is always present in real measurements, it is more practical to use a penalty function that

includes two terms:

$$f(\mathbf{h}) = f_s(\mathbf{h}) + f_\nu(\mathbf{h}), \tag{14}$$

where the first term attracts sparsity and is a modification of the ℓ_0-norm penalty as follows

$$f_s(\mathbf{h}) = \tau \sum_{k=0}^{N-1} \xi(h_k), \tag{15}$$

where h_k are elements of the vector $\mathbf{h}$ and

$$\xi(h) = \begin{cases} \frac{|h|}{T}, & \text{if } |h| < T, \\ 1, & \text{otherwise.} \end{cases} \tag{16}$$

Such modification introduces a transition zone $[-T, T]$ for small values of h_k most probably affected by the noise. In the transition zone, the penalty (3) with $p = 1$ is used, which also attracts sparsity, but is less sensitive to noise. The second term in (14) is the ℓ_2-norm penalty

$$f_\nu(\mathbf{h}) = \frac{\varepsilon}{2} \sum_{k=0}^{N-1} |h_k|^2, \tag{17}$$

with $\varepsilon > 0$ being a regularization parameter. This penalty is often used for regularization in adaptive filters [4], [14], [15]. The gradient of $f(\mathbf{h})$ is then given by

$$[\nabla_{\mathbf{h}} f(\mathbf{h})]_k = \varepsilon h_k + \begin{cases} -\frac{\tau}{T}, & \text{if } -T < h_k < 0, \\ \frac{\tau}{T}, & \text{if } 0 < h_k < T, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

This can also be represented as

$$[\nabla_{\mathbf{h}} f(\mathbf{h})]_k = \varepsilon h_k + \begin{cases} \frac{\tau}{T} \frac{h_k}{|h_k|}, & \text{if } |h_k| < T, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

Elements of the weight vector $\mathbf{w}(n)$ (the diagonal loading) are then computed as

$$w_k(n) = \varepsilon + \begin{cases} \frac{\tau}{T(|\hat{h}_k(n-1)|+\delta)}, & \text{if } |\hat{h}_k| < T, \\ 0, & \text{otherwise,} \end{cases} \tag{20}$$

where δ is a small positive constant added to prevent division by zero.

V. NUMERICAL RESULTS

This section presents results of computer simulation with real speech signals as the input to adaptive filters. The speech signal is sampled at a rate of 8 kHz and its samples are represented as 14-bit fixed-point numbers. We compare the Mean Square Deviation (MSD) performance of the proposed SRLS-VDL2-DCD algorithm against the oracle SRLS algorithm, SRLS-VDL1-DCD from [6], and the SRLS-VDL1-DCD algorithm with the weights (20) proposed in this paper. The MSD is calculated as $\text{MSD}(n) = ||\mathbf{h}_\circ(n) - \hat{\mathbf{h}}(n)||_2^2$ and plotted against the time index n. In the simulation, the impulse response $\mathbf{h}_\circ(n)$ is kept constant during the first 8000 instants (samples) and then changes. The change is in both positions and values of the non-zero taps. The plots start at iteration 7500, since we are mainly interested in the behavior of the adaptive filter after the change; this demonstrates the tracking capability of the adaptive filter.

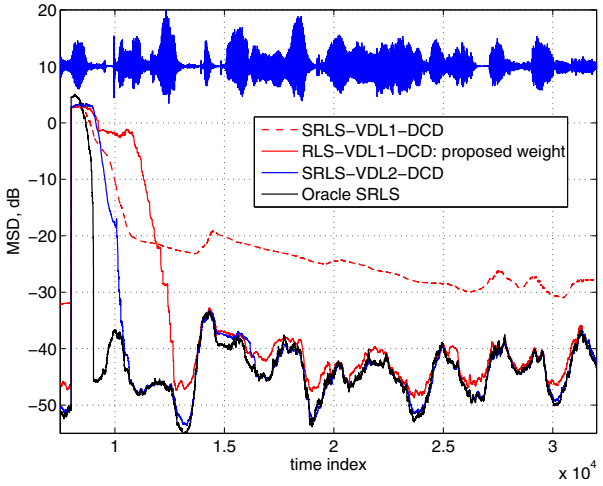

Fig. 1. MSD performance of adaptive filters with speech as the input signal. Parameters of the scenario: $N = 512$, $K = 100$, SNR$= 45$ dB.

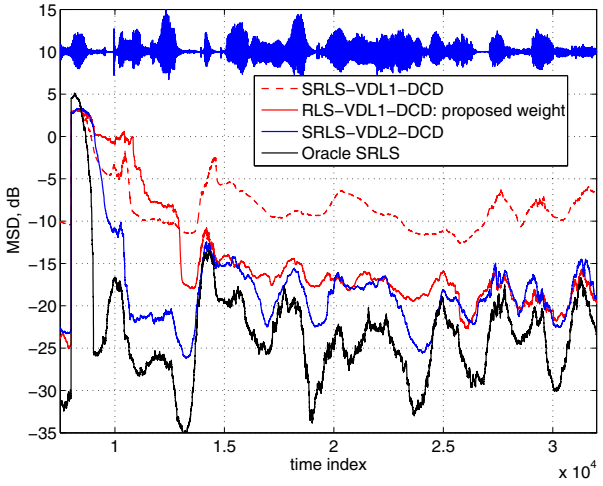

Fig. 2. MSD performance of adaptive filters with speech as the input signal. Parameters of the scenario: $N = 512$, $K = 100$, SNR$= 25$ dB.

Positions of the K non-zero elements in $\mathbf{h}_\circ(n)$ are chosen randomly. In our simulation, we use $K = 100$ and $N = 512$. The non-zero elements of $\mathbf{h}_\circ(n)$ are generated as independent Gaussian zero-mean random numbers of fixed variance and then $\mathbf{h}_\circ(n)$ is normalized to have the norm equal to one: $||\mathbf{h}_\circ(n)||_2 = 1$.

Fig. 1 and Fig. 2 show the MSD performance of the adaptive filters at a signal-to-noise ratio (SNR) of respectively 45 dB and 25 dB. The SNR is computed as the ratio of the powers of the clean signal $\mathbf{x}^T(n)\mathbf{h}_\circ(n)$ and of the noise.

Adaptation in the oracle SRLS algorithm is based on a perfectly known support (i.e., positions of non-zero elements) of $\mathbf{h}_o(n)$. The algorithm only performs the SRLS recursion with a sliding window of length $M = 1024$ for the non-zero taps and ignores the zero taps. The oracle SRLS algorithm provides the best performance: it requires the shortest period to arrive at a steady-state MSD level. We use the oracle performance as a benchmark for the other algorithms.

The SRLS-VDL1-DCD algorithm from [6] is based on the weighting corresponding to the Improved PAPA [4]; we set $M = 1024$, while the other parameters are adjusted to guarantee the best performance (they are the same as in [6]). For the SRLS-VDL2-DCD algorithm proposed in this paper, for the simulation shown in Fig. 1 the parameters are set to: $M = 1024$, $\varepsilon = 10^4$, $\delta = 10^{-4}$, $\tau = 10^6$, $T = 10^{-3}$, $H = 1$, $N_u = 16$, and $M_b = 15$. For the simulation in Fig. 2, with lower SNR, we increased the value of T to 10^{-2} (i.e., an increase proportional to that of the noise standard deviation). It can be seen that this algorithm outperforms the other adaptive algorithms considered and its MSD closely follows that of the oracle SRLS algorithm.

It can be seen that the SRLS-VDL1-DCD algorithm with original weights (from [6]) approaches an MSD around -30 dB in Fig. 1 (-10 dB in Fig. 2). However, this is significantly higher than the MSD levels obtained by both the SRLS-VDL2-DCD algorithm and the SRLS-VDL1-DCD algorithm using the new weights (20). We can see that the algorithm and weights proposed here result in performance much closer to that of the oracle.

Comparing the SRLS-VDL2-DCD algorithm with the SRLS-VDL1-DCD algorithm using the new weights (20), we see that the former converges in less iterations for the same steady-state MSD. In addition to the improvement in performance, Sec. III showed that the SRLS-VDL2-DCD algorithm requires less computations than the SRLS-VDL1-DCD algorithm.

VI. CONCLUSIONS

In this paper, we have proposed a novel SRLS adaptive filtering algorithm for sparse identification; the algorithm is based on dynamic diagonal loading. It provides MSD performance close to the oracle SRLS performance and outperforms other sparse adaptive filters in an acoustic echo cancellation scenario. For transversal adaptive filters, the complexity of the proposed algorithm is linear in the filter length and comparable to that of other DCD based adaptive filters. Moreover, the proposed algorithm, being based on dichotomous coordinate descent iterations, is well suited to implementation in finite precision, in particular on FPGAs [11].

REFERENCES

[1] B. D. Rao and B. Song, "Adaptive filtering algorithms for promoting sparsity," in *Proceedings IEEE Int. Conf. Acoustics, Speech and Signal Processing, ICASSP'2003*, 2003, pp. VI–361–364.

[2] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The sparse RLS algorithm," *IEEE Trans. Signal Processing*, vol. 58, no. 8, pp. 4013–4025, 2010.

[3] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *Proceedings IEEE Int. Conf. Acoustic, Speech and Signal Processing, ICASSP'2010*, 2010, pp. 3734–3737.

[4] C. Paleologu, J. Benesty, and S. Ciochina, "Sparse adaptive filters for echo cancellation," *Synthesis Lectures on Speech and Audio Processing*, vol. 6, no. 1, pp. 1–124, 2010.

[5] E. M. Eksioglu and A. K. Tanc, "RLS algorithm with convex regularization," *IEEE Signal Processing Letters*, vol. 18, no. 8, pp. 470–473, 2011.

[6] Y. Zakharov and V. Nascimento, "Sliding-window RLS low-cost implementation of proportionate affine projection algorithms," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 12, pp. 1815–1824, 2014.

[7] Y. V. Zakharov and T. C. Tozer, "Multiplication-free iterative algorithm for LS problem," *Electonics Letters*, vol. 40, no. 9, pp. 567–569, 2004.

[8] Y. V. Zakharov and V. H. Nascimento, "Sparse RLS adaptive filter with diagonal loading," in *46th Asilomar Conference on Signals, Systems and Computers*, 2012.

[9] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Orlando, FL, USA*, 2002, vol. 2, pp. 1881–1884.

[10] Y. Zakharov, G. White, and J. Liu, "Low complexity RLS algorithms using dichotomous coordinate descent iterations," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3150–3161, July 2008.

[11] J. Liu, Y. V. Zakharov, and B. Weaver, "Architecture and FPGA design of dichotomous coordinate descent algorithms," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 56, no. 11, pp. 2425–2438, 2009.

[12] J. Liu and Y. Zakharov, "Low complexity dynamically regularised RLS algorithm," *Electronics Letters*, vol. 44, no. 14, pp. 886–885, 2008.

[13] G. H. Golub and C. F. Van Loan, *Matrix computations*, The Johns Hopkins University Press, Baltimore, 3rd edition, 1996.

[14] S. Haykin, *Adaptive filtering*, Prentice Hall, 2nd Edition edition, 1991.

[15] A. H. Sayed, *Fundamentals of adaptive filtering*, Hoboken, N. J.: Wiley, 2003.