RECURSIVE LEAST-SQUARES ALGORITHMS FOR SPARSE SYSTEM MODELING

Hamed Yazdanpanah, and Paulo S. R. Diniz, Fellow, IEEE

Universidade Federal do Rio de Janeiro DEL/Poli & PEE/COPPE/UFRJ P.O. Box 68504, Rio de Janeiro, RJ, 21941-972, Brazil

ABSTRACT

In this paper, we propose some sparsity aware algorithms, namely the Recursive least-Squares for sparse systems (S-RLS) and l_0 -norm Recursive least-Squares (l_0 -RLS), in order to exploit the sparsity of an unknown system. The first algorithm, applies a discard function on the weight vector to disregard the coefficients close to zero during the update process. The second algorithm, employs the sparsity-promoting scheme via some non-convex approximations to the l_0 -norm. In addition, we consider the respective versions of these algorithms in data-selective versions in order to reduce the update rate. Simulation results show similar performance when comparing the proposed algorithms with standard Recursive Least-Squares (RLS) algorithm while the proposed algorithms require lower computational complexity.

Index Terms— adaptive filtering, data-selective, sparsity, discard function, sparsity-promoting scheme

1. INTRODUCTION

Sparse signals and systems are found in a wide diversity of areas and scenarios such as communications, control, acoustics, spectral sensing, channel equalization, echo cancellation, and system identification. Recently, it has been realized that by exploiting signal sparsity, significant improvement in convergence rate and steady-state performance can be obtained. Unfortunately, traditional adaptive algorithms such as leastmean square (LMS) based algorithms and the recursive least squares (RLS) [1,2] do not take into consideration the sparsity in the signal or system models. However, many extensions of the classical algorithms were proposed aiming at exploiting sparsity.

A recent approach to exploit the system sparsity is achieved by utilizing *discard function* [3]. Algorithms employing the discard function avoid updating the coefficients close to zero, as a result reducing the computational burden. This idea is motivated by the inherent relative importance of the estimated coefficients in practical applications. In fact, in a sparse system a few coefficients have most of the energy, whereas the other coefficients are close to zero. As by-product, algorithms applying the discard function require lower computational complexity than the traditional algorithms.

Another interesting approach to exploit sparsity is to include a *sparsity-promoting penalty* function into the original objective function of classical algorithms [4]. For this purpose, most algorithms apply the l_1 -norm as the sparsitypromoting penalty [5–9], but recently an approximation to the l_0 -norm has shown some advantages [4, 10–12]. Adding this penalty function to the RLS cost function increases the computational complexity. Also, there are some other algorithms to exploit the sparsity such as *proportionate* algorithms [13–16] leading to even higher computational complexity.

In this paper, we introduce some sparse-aware RLS algorithms employing the discard function and the l_0 -norm approximation. The first proposed algorithm, the RLS for sparse systems (S-RLS), gives low weight to the coefficients close to zero and exploits system sparsity with low computational complexity whereas the second algorithm, the l_0 -norm RLS (l_0 -RLS), has higher computational complexity. In both algorithms, in order to reduce further the computational load we apply a data-selective strategy [17] leading to the dataselective S-RLS (DS-S-RLS) and the data-selective l_0 -RLS (DS- l_0 -RLS) algorithms. That is, the proposed algorithms update the weight vector if the output estimation error is larger than a prescribed value. Applying this strategy, both algorithms attain lower computational complexity compared to the RLS algorithm.

The rest of this paper is organized as follows. The proposed S-RLS and l_0 -RLS algorithms along with their dataselective versions are presented in Sections 2 and 3, respectively. Simulations are presented in Section 4 and Section 5 contains the conclusions.

2. RECURSIVE LEAST-SQUARES ALGORITHM EXPLOITING SPARSITY

In Subsection 2.1, we derive the S-RLS algorithm that exploits the sparsity of the estimated parameters by giving low weight to the small coefficients. For this purpose, the strategy consists in multiplying the coefficients of the sparse filter which are close to zero by a small constant. Then, in Subsection 2.2, we include a discussion of some characteristics of



Fig. 1. (a) Discard function $f_{\epsilon}(w)$ for $\epsilon = 10^{-4}$; (b) Geman-McClure function for $\beta = 5$.

the proposed algorithm. Subsection 2.3 briefly describes the DS-S-RLS algorithm.

2.1. Derivation of the S-RLS algorithm

Defining the *discard function* $f_{\epsilon} : \mathbb{R} \to \mathbb{R}$ for the positive constant ϵ as follows

$$f_{\epsilon}(w) = \begin{cases} w & \text{if } |w| > \epsilon \\ 0 & |w| \le \epsilon \end{cases}$$
(1)

where function f_{ϵ} discards the values of w which are close to zero. The parameter ϵ indicates how close to zero a coefficient should be discarded. The value of ϵ is chosen based on some *a priori* information about the relevance of a given coefficient to the sparse system. The function $f_{\epsilon}(w)$ is illustrated in Figure 1(a) for $\epsilon = 10^{-4}$. As can be observed, $f_{\epsilon}(w)$ is not differentiable at $\pm \epsilon$. To address the non-differentiable property of $f_{\epsilon}(w)$, we consider the derivative of $f_{\epsilon}(w)$ at $+\epsilon$ and $-\epsilon$ as equal to the left and the right derivatives, respectively. Therefore, the derivative of $f_{\epsilon}(w)$ at $\pm \epsilon$ is zero. Denote the *discard vector function* $\mathbf{f}_{\epsilon} : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$ as $\mathbf{f}_{\epsilon}(\mathbf{w}) = [f_{\epsilon}(w_0), \cdots, f_{\epsilon}(w_N)]^T$.

The objective function of the S-RLS algorithm is given as follows

$$\min \xi^d(k) = \sum_{i=0}^k \lambda^{k-i} [d(i) - \mathbf{x}^T(i) \mathbf{f}_\epsilon(\mathbf{w}(k))]^2 \quad (2)$$

where $d(i) \in \mathbb{R}$ is the desired signal and $\mathbf{x}(i) \in \mathbb{R}^{N+1}$ is the input-signal vector,

$$\mathbf{x}(k) = [x(k) \ x(k-1) \ \cdots \ x(k-N)]^T$$
 (3)

The parameter λ is an exponential weighting factor that should be selected in the range $0 \ll \lambda \leq 1$. Note that the objective function is quadratic with respect to the discard function which turn leads to many equivalent maps of the weight function.

By differentiating $\xi^d(k)$ with respect to $\mathbf{w}(k)$, we obtain

$$\frac{\partial \xi^d(k)}{\partial \mathbf{w}(k)} = -2\sum_{i=0}^k \lambda^{k-i} \mathbf{F}_{\epsilon}(\mathbf{w}(k)) \mathbf{x}(i) [d(i) - \mathbf{x}^T(i) \mathbf{f}_{\epsilon}(\mathbf{w}(k))] \quad (4)$$

where $\mathbf{F}_{\epsilon}(\mathbf{w}(k))$ is the Jacobian matrix of $\mathbf{f}_{\epsilon}(\mathbf{w}(k))$. By equating the above equation to zero, we will find the optimal vector $\mathbf{w}(k)$ that minimizes the least-square error, as follows

$$-\sum_{i=0}^{k} \lambda^{k-i} \mathbf{F}_{\epsilon}(\mathbf{w}(k)) \mathbf{x}(i) \mathbf{x}^{T}(i) \mathbf{f}_{\epsilon}(\mathbf{w}(k)) + \sum_{i=0}^{k} \lambda^{k-i} \mathbf{F}_{\epsilon}(\mathbf{w}(k)) \mathbf{x}(i) d(i) = 0$$
(5)

Therefore,

$$\mathbf{f}_{\epsilon}(\mathbf{w}(k)) = \left[\sum_{i=0}^{k} \lambda^{k-i} \mathbf{F}_{\epsilon}(\mathbf{w}(k)) \mathbf{x}(i) \mathbf{x}^{T}(i)\right]^{-1} \times \sum_{i=0}^{k} \lambda^{k-i} \mathbf{F}_{\epsilon}(\mathbf{w}(k)) \mathbf{x}(i) d(i)$$
(6)

Note that $\mathbf{F}_{\epsilon}(\mathbf{w}(k))$ is a diagonal matrix with diagonal entries equal to zero or one. Indeed, for the components of $\mathbf{w}(k)$ whose absolute values are larger than ϵ , their corresponding entries on the diagonal matrix $\mathbf{F}_{\epsilon}(\mathbf{w}(k))$ are one, whereas the remaining entries are zero. Hence,

$$\begin{aligned} \mathbf{F}_{\epsilon}(\mathbf{w}(k))\mathbf{x}(i)\mathbf{x}^{T}(i) &= \mathbf{F}_{\epsilon}^{2}(\mathbf{w}(k))\mathbf{x}(i)\mathbf{x}^{T}(i) \\ &= \mathbf{F}_{\epsilon}(\mathbf{w}(k))(\mathbf{x}^{T}(i)\mathbf{F}_{\epsilon}(\mathbf{w}(k)))^{T}\mathbf{x}^{T}(i) \\ &= \mathbf{F}_{\epsilon}(\mathbf{w}(k))\mathbf{x}(i)\mathbf{x}^{T}(i)\mathbf{F}_{\epsilon}(\mathbf{w}(k)) \end{aligned}$$
(7)

By utilizing (7) in (6) and replacing $\mathbf{f}_{\epsilon}(\mathbf{w}(k))$ by $\mathbf{w}(k+1)$, we get

$$\mathbf{w}(k+1) = \left[\sum_{i=0}^{k} \lambda^{k-i} \mathbf{F}_{\epsilon}(\mathbf{w}(k)) \mathbf{x}(i) \mathbf{x}^{T}(i) \mathbf{F}_{\epsilon}(\mathbf{w}(k))\right]^{-1} \\ \times \sum_{i=0}^{k} \lambda^{k-i} \mathbf{F}_{\epsilon}(\mathbf{w}(k)) \mathbf{x}(i) d(i) = \mathbf{R}_{D,\epsilon}^{-1}(k) \mathbf{p}_{D,\epsilon}(k)$$
(8)

where $\mathbf{R}_{D,\epsilon}(k)$ and $\mathbf{p}_{D,\epsilon}(k)$ are called the deterministic correlation matrix of the input signal and the deterministic crosscorrelation vector between the input and desired signals, respectively. Whenever *i*-th diagonal entry of matrix $\mathbf{F}_{\epsilon}(\mathbf{w}(k))$ is zero, it is replaced by a small power-of-two (e.g., 2^{-5}) multiplied by the sign of the component $w_i(k)$ in order to avoid that matrix $\mathbf{R}_{D,\epsilon}(k)$ becomes ill conditioned. Then using the

Table 1. S-RLS Algorithm									
$\mathbf{S}_{D,\epsilon}(-1) = \delta \mathbf{I}$									
where δ can be inverse of the input signal power estimate									
$\mathbf{p}_{D,\epsilon}(-1) = [0\ 0\ \cdots\ 0]^T$									
$\mathbf{w}(-1) = [1 \ 1 \ \cdots \ 1]^T$									
Do for $k \ge 0$									
$\mathbf{S}_{D,\epsilon}(k)$ as in Equation (9)									
$\mathbf{p}_{D,\epsilon}(k) = \lambda \mathbf{p}_{D,\epsilon}(k-1) + \mathbf{F}_{\epsilon}(\mathbf{w}(k))\mathbf{x}(k)d(k)$									
$\mathbf{w}(k+1) = \mathbf{S}_{D,\epsilon}(k)\mathbf{p}_{D,\epsilon}(k)$									
end									

matrix inversion lemma, the inverse of the deterministic correlation matrix can then be calculated in the following form

$$\mathbf{S}_{D,\epsilon}(k) = \mathbf{R}_{D,\epsilon}^{-1}(k) = \frac{1}{\lambda} \Big[\mathbf{S}_{D,\epsilon}(k-1) - \frac{\mathbf{S}_{D,\epsilon}(k-1)\mathbf{F}_{\epsilon}(\mathbf{w}(k))\mathbf{x}(k)\mathbf{x}^{T}(k)\mathbf{F}_{\epsilon}(\mathbf{w}(k))\mathbf{S}_{D,\epsilon}(k-1)}{\lambda + \mathbf{x}^{T}(k)\mathbf{F}_{\epsilon}(\mathbf{w}(k))\mathbf{S}_{D,\epsilon}(k-1)\mathbf{F}_{\epsilon}(\mathbf{w}(k))\mathbf{x}(k)} \Big]$$
(9)

Therefore,

$$\mathbf{w}(k+1) = \mathbf{S}_{D,\epsilon}(k)\mathbf{p}_{D,\epsilon}(k) \tag{10}$$

2.2. Discussion of the S-RLS algorithm

The update equation of the S-RLS algorithm is similar to the update equation of the RLS algorithm, but the former gives importance only to the subset of coefficients of $\mathbf{w}(k)$ whose absolute values are larger than ϵ . The matrix $\mathbf{F}_{\epsilon}(\mathbf{w}(k))$ defines the important coefficients of $\mathbf{w}(k)$.

Unlike classical RLS algorithm in which the initialization of the weight vector is often chosen as $\mathbf{w}(0) = \mathbf{0}$, this same procedure cannot be applied to the proposed algorithm. Indeed, for the S-RLS algorithm, each of the coefficients should be initialized as $|w_i(0)| > \epsilon$ for $i = 0, 1, \dots, N$.

2.3. DS-S-RLS algorithm

In this subsection, our goal is to reduce the update rate of the S-RLS algorithm. In fact, when the current weight vector is acceptable, i.e., the output estimation error is small, we can save computational resources by avoiding new update. The data selective S-RLS (DS-S-RLS) algorithm updates whenever the output estimation error is larger than a prescribed value $\overline{\gamma}$, i.e., when $|e(k)| = |d(k) - \mathbf{w}^T(k)\mathbf{x}(k)| > \overline{\gamma}$. Therefore, the DS-S-RLS algorithm reduces the computational complexity by avoiding updates whenever the estimate is acceptable.

3. *l*₀-NORM RECURSIVE LEAST-SQUARES ALGORITHM

In the previous section, we exploit the systems sparsity utilizing discard function. Another interesting approach to exploit the system sparsity can be derived by using l_0 -norm [4] leading to the l_0 -RLS algorithm. However, the resulting optimization problem of l_0 -norm has difficulties due to the discontinuity of the l_0 -norm. Thus, we use the Geman-McClure function (GMF) [18] to approximate the l_0 -norm. The GMF is given as follows, Figure 1(b),

$$G_{\beta}(\mathbf{w}) = \sum_{i=0}^{N} (1 - \frac{1}{1 + \beta |w(i)|})$$
(11)

where $\beta \in \mathbb{R}_+$ is a parameter responsible for controlling the agreement between the quality of the approximation and smoothness of G_β . The gradient of G_β is defined as follows

$$\nabla G_{\beta}(\mathbf{w}) \triangleq \mathbf{g}_{\beta}(\mathbf{w}) \triangleq [g_{\beta}(w(0)) \cdots g_{\beta}(w(N))]^T$$
 (12)

where $g_{\beta}(w) \triangleq \frac{\partial G_{\beta}(w)}{\partial w}$ is given by

$$g_{\beta}(w) = \frac{\beta \operatorname{sgn}(w)}{(1+\beta|w|)^2}$$
(13)

where $sgn(\cdot)$ is the sign function.

The objective function of the l_0 -RLS algorithm is given by

$$\min \sum_{i=0}^{k} \lambda^{k-i} [d(i) - \mathbf{x}^{T}(i)\mathbf{w}(k)]^{2} + \alpha \|\mathbf{w}(k)\|_{0}$$
(14)

where $\alpha \in \mathbb{R}_+$ is the weight given to the l_0 -norm penalty. Replacing $\|\mathbf{w}(k)\|_0$ by its approximation, we obtain

$$\min\sum_{i=0}^{k} \lambda^{k-i} [d(i) - \mathbf{x}^{T}(i)\mathbf{w}(k)]^{2} + \alpha G_{\beta}(\mathbf{w}(k))$$
(15)

By differentiating the above equation and equating the result to zero, we get

$$\mathbf{w}(k) = \left[\sum_{i=0}^{k} \lambda^{k-i} \mathbf{x}(i) \mathbf{x}^{T}(i)\right]^{-1} \times \left(\left(\sum_{i=0}^{k} \lambda^{k-i} \mathbf{x}(i) d(i)\right) - \frac{\alpha}{2} \mathbf{g}_{\beta}(\mathbf{w}(k))\right) = \mathbf{R}_{D}^{-1}(k) \left(\mathbf{p}_{D}(k) - \frac{\alpha}{2} \mathbf{g}_{\beta}(\mathbf{w}(k))\right)$$
(16)

Using the matrix inversion lemma, the update equation of the l_0 -RLS algorithm is given as follows

$$\mathbf{w}(k) = \mathbf{S}_D(k) \left(\mathbf{p}_D(k) - \frac{\alpha}{2} \mathbf{g}_\beta(\mathbf{w}(k-1)) \right)$$
(17)

where a same strategy as the PASTd (projection approximation subspace tracking with deflation) [19] is employed and $\mathbf{g}_{\beta}(\mathbf{w}(k))$ is replaced by $\mathbf{g}_{\beta}(\mathbf{w}(k-1))$ in order to form the recursion. Also, $\mathbf{p}_D(k)$ and $\mathbf{S}_D(k)$ are given as follows

$$\mathbf{p}_D(k) = \lambda \mathbf{p}_D(k-1) + d(k)\mathbf{x}(k) \tag{18}$$

$$\mathbf{S}_{D}(k) = \frac{1}{\lambda} \Big[\mathbf{S}_{D}(k-1) - \frac{\mathbf{S}_{D}(k-1)\mathbf{x}(k)\mathbf{x}^{T}(k)\mathbf{S}_{D}(k-1)}{\lambda + \mathbf{x}^{T}(k)\mathbf{S}_{D}(k-1)\mathbf{x}(k)} \Big]$$
(19)

Table 2. The coefficients of unknown systems w_0 and w'_0

									2	0	0				
\mathbf{w}_o	24e-2	2e-8	-23e-2	-3e-7	5e-1	-1e-9	2e-1	1e-7	-5e-8	12e-6	1e-8	-5e-6	4e-6	-1e-7	-2e-1
\mathbf{w}_o'	2e-7	-21e-10	17e-8	21e-8	-3e-7	24e-2	7e-1	2e-1	33e-2	-6e-1	-5e-7	18e-9	-5e-7	21e-8	-11e-8

Note that Equation (17) does not require to be iterated.

Remark 1: Similarly to the discussion in Subsection 2.3, the DS- l_0 -RLS algorithm for sparse systems can be derived by implementing an update in the l_0 -RLS algorithm whenever the output estimation error is larger than a prescribed value $\overline{\gamma}$, reducing the update rate.

4. SIMULATIONS

In this section, RLS, S-RLS, lo-RLS, Adaptive Sparse Variational Bayes iterative scheme based on Laplace prior (ASVB-L) [20], Zero-Attracting LMS (ZA-LMS), DS-S-RLS, DS l_0 -RLS, data selective ZA-LMS (DS-ZA-LMS), and dataselective ASVB-L (DS-ASVB-L) algorithms are tested to identify three unknown sparse systems of order 14. The first model is an arbitrary sparse system \mathbf{w}_o , the second model is a block sparse system \mathbf{w}_{o}^{\prime} , and the third model, $\mathbf{w}_{o}^{\prime\prime}$, is a sparse system which its coefficients changes at 500th and 1000th iterations. The coefficients of \mathbf{w}_{o} and \mathbf{w}_{o}' are listed in Table 2. The input is an autoregressive signal generated by x(k) = 0.95x(k-1) + n(k-1). The signal-to-noise ratio (SNR) is set to be 20 dB, meaning that the noise variance is $\sigma_n^2 = 0.01$. The bound on the estimation error is set to be $\overline{\gamma} = \sqrt{5\sigma_n^2}$. The initial vector $\mathbf{w}(0)$ and λ are $[1, \dots, 1]^T$ and 0.97, respectively. The parameter δ is 0.2 and the constant ϵ is chosen as 0.015. For DS- l_0 -RLS and l_0 -RLS algorithms, the parameters α and β are chosen as 0.005 and 5, respectively. The parameters μ and ρ in ZA-LMS algorithm are chosen as 0.01 and 0.0005, respectively. The depicted learning curves represent the results of averaging of the outcomes of 4000 trials. Figures 2(a), 2(b), and 2(c) show the learning curves for the RLS, S-RLS, l₀-RLS, ASVB-L, ZA-LMS algorithms to identify the unknown systems \mathbf{w}_o , \mathbf{w}'_o , and \mathbf{w}''_o , respectively. Figure 2(d) illustrates the learning curves for the DS-ZA-LMS, DS-S-RLS, DS-l₀-RLS, and DS-ASVB-L algorithms to identify the unknown system \mathbf{w}_{o} . The average number of updates implemented by the DS-ZA-LMS, DS-S-RLS, DS- l_0 -RLS, and DS-ASVB-L algorithms are 44.5%, 10.3%, 9.8%, and 7.9%, respectively.

Observe that, in every scenario we tested, the S-RLS and the l_0 -RLS algorithms performed as well as the RLS algorithm. The DS-S-RLS and DS- l_0 -RLS algorithms have lower computational complexity. As can be seen, their performance is close to the DS-ASVB-L algorithm while having lower computational complexity. Also, note that the DS-ZA-LMS algorithm has larger update rate and higher MSE compared to the proposed algorithms.



Fig. 2. The learning curves of the RLS, the S-RLS, the l_0 -RLS, the ASVB-L, and ZA-LMS algorithms applied to identify: (a) \mathbf{w}_o ; (b) \mathbf{w}'_o ; (c) \mathbf{w}''_o . (d): The learning curves of the DS-ZA-LMS, the DS-S-RLS, the DS- l_0 -RLS, and the DS-ASVB-L algorithms applied on \mathbf{w}_o .

5. CONCLUSIONS

In this paper, we have proposed the S-RLS and the l_0 -RLS algorithms to exploit the sparsity in the involved signal models. Also, we have employed the data-selective strategy to implement an update when the output estimation error is larger than a pre-described positive value leading to reduced update rate and lower computational complexity. The simulation results have shown the excellent performance of the proposed algorithms as compared to the standard RLS algorithm being competitive with the recent proposed state-of-the-art ASVB-L algorithm which requires much more computations.

6. REFERENCES

- P.S.R. Diniz, Adaptive Filtering: Algorithms and Practical Implementation, Springer, New York, USA, 4th edition, 2013.
- [2] A.H. Sayed, Fundamentals of Adaptive Filtering, Wiley-IEEE, New York, USA, 2003.
- [3] H. Yazdanpanah, P.S.R. Diniz, and M.V.S. Lima, "A simple set-membership affine projection algorithm for sparse system modeling," in 24th European Signal Processing Conference (EUSIPCO 2016), Budapest, Hungary, Sept. 2016, pp. 1798–1802.
- [4] M.V.S. Lima, T.N. Ferreira, W.A. Martins, and P.S.R. Diniz, "Sparsity-aware data-selective adaptive filters," *IEEE Transactions on Signal Processing*, vol. 62, no. 17, pp. 4557–4572, Sept. 2014.
- [5] R. Meng, R.C. de Lamare, and V.H. Nascimento, "Sparsity-aware affine projection adaptive algorithms for system identification," in *Sensor Signal Processing for Defence*, London, U.K., Sept. 2011, pp. 1–5.
- [6] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted l₁ balls," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 936– 952, March 2011.
- [7] Y. Chen, Y. Gu, and A.O. Hero, "Regularized leastmean-square algorithms," arXiv, pp. 1–9, Dec. 2010.
- [8] Y. Chen, Y. Gu, and A.O. Hero, "Sparse LMS for system identification," in *IEEE International Conference* on Acoustics, Speech and Signal Processing (ICASSP 2009), Taipei, Taiwan, April 2009, pp. 3125–3128.
- [9] B. Babadi, N. Kalouptsidis, and V. Tarokh, "Sparls: The sparse rls algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4013–4025, Aug. 2010.
- [10] M.V.S. Lima, I. Sobron, W.A. Martins, and P.S.R. Diniz, "Stability and MSE analyses of affine projection algorithms for sparse system identification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, Florence, Italy, May 2014, pp. 6399–6403.

- [11] M.V.S. Lima, W.A. Martins, and P.S.R. Diniz, "Affine projection algorithms for sparse system identification," in *IEEE International Conference on Acoustics, Speech* and Signal Processing (ICASSP 2013), Vancouver, Canada, May 2013, pp. 5666–5670.
- [12] Y. Gu, J. Jin, and S. Mei, "l₀ norm constraint LMS algorithm for sparse system identification," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 774–777, Sept. 2009.
- [13] D.L. Duttweiler, "Proportionate normalized least-meansquares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, Sept. 2000.
- [14] J. Benesty and S.L. Gay, "An improved PNLMS algorithm," in *IEEE International Conference on Acoustics*, *Speech and Signal Processing (ICASSP 2002)*, Dallas, USA, May 2002, vol. 2, pp. 1881–1884.
- [15] S.L. Gay, "An efficient, fast converging adaptive filter for network echo cancellation," in *Thirty-Second Asilomar Conference on Signals, Systems amp, Computers* (ACSSC 1998), 1998, vol. 1, pp. 394–398.
- [16] S. Werner, J.A. Apolinario Jr., and P.S.R. Diniz, "Setmembership proportionate affine projection algorithms," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, no. 1, pp. 1–10, Jan. 2007.
- [17] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.-F. Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Processing Letters*, vol. 5, no. 5, pp. 111– 114, May 1998.
- [18] D. Geman and G. Reynolds, "Nonlinear image recovery with half-quadratic regularization," *IEEE Transactions* on *Image Processing*, vol. 4, no. 7, pp. 932–946, July 1995.
- [19] X. Wang and H.V. Poor, Wireless Communication Systems: Advanced Techniques for Signal Reception, Prentice Hall, Upper Saddle River, NJ, 2004.
- [20] K.E. Themelis, A.A. Rontogiannis, and K.D. Koutroumbas, "A variational bayes framework for sparse adaptive estimation," *IEEE Transactions on Signal Processing*, vol. 62, no. 18, pp. 4723–4736, Sept. 2014.