RLS Algorithm with Convex Regularization

Ender M. Eksioglu, *Member, IEEE* and A. Korhan Tanc, *Student Member, IEEE*

***Abstract*—In this letter the RLS adaptive algorithm is considered in the system identification setting. The RLS algorithm is regularized using a general convex function of the system impulse response estimate. The normal equations corresponding to the convex regularized cost function are derived, and a recursive algorithm for the update of the tap estimates is established. We also introduce a closed-form expression for selecting the regularization parameter. With this selection of the regularization parameter, we show that the convex regularized RLS algorithm performs as well as, and possibly better than, the regular RLS when there is a constraint on the value of the convex function evaluated at the true weight vector. Simulations demonstrate the superiority of the convex regularized RLS with automatic parameter selection over regular RLS for the sparse system identification setting.**

***Index Terms*—Adaptive filter, RLS, convex regularization, sparsity, l1 norm, l0 norm. *EDICS*: SAS-SYST, SAS-ADAP.**

I. Introduction

The last decade has seen a flurry of activities in regularization of an otherwise ill-posed inverse problem by a convex, most of the time sparsity based prior. The sparsity prior utilizes the knowledge that the object to be recovered is sparse in a certain, known representation. The replacement of the nonconvex ℓ_0 pseudo-norm as a count for sparsity with the convex ℓ_1 norm has led to new data acquisition paradigms introduced under Compressive Sensing [1], and it has found numerous applications including sparse channel estimation [2].

These advances in sparse signal representation have also impacted sparse adaptive system identification. In [3], the authors propose to modify the LMS cost function by addition of a convex approximation for the ℓ_0 norm penalty. The resulting sparsity enhancing LMS variant is called as the ℓ_0-LMS. The authors of [4] propose to regularize the LMS cost function by adding an ℓ_1 norm term or a log-sum term. They have recently considered the regularization of the LMS algorithm by a general convex function [5]. ℓ_1-norm regularized recursive least squares (RLS) adaptive algorithms have also been suggested in the literature. The SPARLS algorithm [6] presents an expectation-maximization (EM) approach for sparse system identification. The authors of [7] propose the application of an online coordinate descent algorithm together with the least-squares cost function penalized by an ℓ_1-norm term. Another RLS algorithm for sparse system identification is proposed in [8], where the RLS cost function is regularized by adding a weighted ℓ_1 norm of the current system estimate. Adaptive sparse system identification has been recently successfully extended to nonlinear systems [9], [10].

In this letter we consider regularization of the RLS cost function in a manner alike to the approach as outlined in [8]. However, here the regularizing term is defined as a general convex function of the system estimate, rather than being defined specifically as the weighted ℓ_1 norm. This generalization allows utilization of any convex function for regularization, which permits one to exploit a much more general class of prior knowledge about the system to be identified, rather than being limited only to sparsity. We develop the update algorithm for the convex regularized RLS using results from subgradient calculus. Additionally, we develop conditions on the proper selection of the regularization parameter. We prove that if the regularization parameter is selected accordingly, the convex regularized RLS algorithm performs as well as, if not better than, the regular RLS algorithm in terms of the mean square deviation (MSD) of the tap estimates. We consider ℓ_1 norm and smoothed ℓ_0 norm as examples for regularizing convex functions. Simulations demonstrate that the resulting ℓ_1-RLS and ℓ_0-RLS algorithms outperform the regular RLS in the sparse system identification setting.

The authors are with the Department of Electronics and Communications Engineering, Istanbul Technical University, Istanbul, Turkey (Phone/fax: +90-212-2853623, e-mail: {eksioglue,tanc}@itu.edu.tr).

II. Convex Regularized RLS Algorithm

We first review the adaptive input-output system identification setting.

$$y_n = \mathbf{w}^T \mathbf{x}_n + \eta_n \tag{1}$$

$\mathbf{w} = [w_0, w_1, \ldots, w_{N-1}]^T \in \mathbb{R}^N$ is the impulse response for the FIR system to be identified. $\mathbf{x}_n = [x_n, x_{n-1}, \ldots x_{n-N+1}]^T \in \mathbb{R}^N$ is the input vector where x_n is the input signal. y_n is the desired output signal, and η_n denotes the observation noise at time n. The estimate for the system tap vector at time n is given by $\mathbf{w}_n = [w_{0,n}, w_{1,n}, \ldots, w_{N-1,n}]^T \in \mathbb{R}^N$. The regular RLS cost function with exponential forgetting factor λ is defined as

$$\mathcal{E}_n = \sum_{m=0}^{n} \lambda^{n-m} (e_m)^2. \tag{2}$$

Here, e_n is the instantaneous error between the desired output and estimated system output.

$$e_n = y_n - \mathbf{w}_n^T \mathbf{x}_n = (\mathbf{w}^T - \mathbf{w}_n^T)\mathbf{x}_n + \eta_n \tag{3}$$

We modify the RLS cost function by the addition of convex function of the instantaneous system estimate. This convex penalty function can be chosen to reflect any prior knowledge about the true system, including but not limited to sparsity.

$$J_n = \frac{1}{2}\mathcal{E}_n + \gamma_n f(\mathbf{w}_n) \tag{4}$$

$f : \mathbb{R}^N \to \mathbb{R}$ is a general convex function. $\gamma_n \geqslant 0$ is the possibly time-varying regularization parameter which governs the compromise between the effect of the regularizing convex function term and the estimation error. We wish to find the optimal system tap vector $\widehat{\mathbf{w}}_n$ which minimizes the regularized cost function J_n. For convex and nondifferentiable functions subgradient analysis offers a substitute for the gradient when finding this minimum [11]. At any point $\boldsymbol{\nu}$ where the convex function f fails to be differentiable, there exist possibly many valid subgradient vectors. All the subgradients together are called as the subdifferential of f and is designated by $\partial f(\boldsymbol{\nu})$. We denote a subgradient vector of f at $\boldsymbol{\nu}$ with $\nabla^s f(\boldsymbol{\nu}) \in \partial f(\boldsymbol{\nu})$. A valid subgradient vector of J_n with respect to $\mathbf{w}_n$ can be written as follows, by using the fact that $\mathcal{E}_n$ is differentiable everywhere.

$$\nabla^s J_n = \frac{1}{2}\nabla\mathcal{E}_n + \gamma_n \nabla^s f(\mathbf{w}_n) \tag{5}$$

One theorem from the subdifferential calculus states that a point $\widehat{\boldsymbol{\nu}} \in \mathbb{R}^N$ minimizes a convex function f if and only if $\mathbf{0} \in \partial f(\widehat{\boldsymbol{\nu}})$, that is if $\mathbf{0}$ is a subgradient of f at $\widehat{\boldsymbol{\nu}}$ [11]. Hence, to find the optimal $\widehat{\mathbf{w}}_n$ which minimizes J_n we set the subgradient of J_n as given in (5) equal to $\mathbf{0}$. After evaluating the gradient $\nabla\mathcal{E}_n$ and setting the subgradient $\nabla^s J_n$ equal to $\mathbf{0}$, the relation for the i^{th} term reads as follows.

$$\sum_{m=0}^{n} \lambda^{n-m}\Big\{y_m - \sum_{k=0}^{N-1} \hat{w}_{k,n} x_{m-k}\Big\}x_{m-i} = \gamma_n \big\{\nabla^s f(\widehat{\mathbf{w}}_n)\big\}_i \tag{6}$$

The relations for all $i = 0, \ldots, N-1$ can be written together in a matrix form as a set of modified normal equations.

$$\boldsymbol{\Phi}_n \widehat{\mathbf{w}}_n = \mathbf{r}_n - \gamma_n \nabla^s f(\widehat{\mathbf{w}}_n) \tag{7}$$

$\boldsymbol{\Phi}_n \in \mathbb{R}^{N\times N}$ is the deterministic autocorrelation matrix estimate for the input signal x_n.

$$\boldsymbol{\Phi}_n = \sum_{m=0}^{n} \lambda^{n-m}\mathbf{x}_m\mathbf{x}_m^T = \lambda\boldsymbol{\Phi}_{n-1} + \mathbf{x}_n\mathbf{x}_n^T \tag{8}$$

$\mathbf{r}_n \in \mathbb{R}^N$ is the deterministic cross-correlation estimate vector between y_n and x_n.

$$\mathbf{r}_n = \sum_{m=0}^{n} \lambda^{n-m} y_m \mathbf{x}_m = \lambda\mathbf{r}_{n-1} + y_n\mathbf{x}_n \tag{9}$$

$\boldsymbol{\Phi}_n$ and $\mathbf{r}_n$ both have rank-one update equations associated with them. For the right hand side of (7) a new variable $\boldsymbol{\theta}_n$ can be defined.

$$\boldsymbol{\theta}_n = \mathbf{r}_n - \gamma_n \nabla^s f(\widehat{\mathbf{w}}_n) \tag{10}$$

The update equation (9) and the definition (10) together lead to an update equation for $\boldsymbol{\theta}_n$. Assuming that γ_{n-1} and $\nabla^s f(\widehat{\mathbf{w}}_{n-1})$ do not change considerably over a single time step, this update equation can be approximately written as

$$\boldsymbol{\theta}_n \approx \lambda\boldsymbol{\theta}_{n-1} + y_n\mathbf{x}_n - \gamma_{n-1}(1-\lambda)\nabla^s f(\widehat{\mathbf{w}}_{n-1}). \tag{11}$$

We define the inverse of the autocorrelation matrix by $\mathbf{P}_n = \boldsymbol{\Phi}_n^{-1}$. Using the matrix inversion lemma and (8), there is a well-known update equation for $\mathbf{P}_n$.

$$\mathbf{P}_n = \lambda^{-1}\Big\{\mathbf{P}_{n-1} - \mathbf{k}_n\mathbf{x}_n^T\mathbf{P}_{n-1}\Big\} \tag{12}$$

$\mathbf{k}_n$ is the gain vector defined as $\mathbf{k}_n = \frac{\mathbf{P}_{n-1}\mathbf{x}_n}{\lambda + \mathbf{x}_n^T\mathbf{P}_{n-1}\mathbf{x}_n}$. The normal equation (7) can be rewritten as follows.

$$\widehat{\mathbf{w}}_n = \mathbf{P}_n\boldsymbol{\theta}_n \tag{13}$$

Algorithm 1 Convex Regularized-RLS (CR-RLS) algorithm.

$\lambda,\ \delta,\ x_n,\ y_n,\ \widehat{\mathbf{w}}_{-1} = \mathbf{0},\ \mathbf{P}_{-1} = \delta^{-1}\mathbf{I}_N$ ▷ inputs

for $n := 0, 1, 2, \ldots$ **do** ▷ time recursion

$\quad \mathbf{k}_n = \dfrac{\mathbf{P}_{n-1}\mathbf{x}_n}{\lambda + \mathbf{x}_n^T\mathbf{P}_{n-1}\mathbf{x}_n}$ ▷ gain vector

$\quad \widehat{\xi}_n = y_n - \widehat{\mathbf{w}}_{n-1}^T\mathbf{x}_n$ ▷ a priori error

$\quad \mathbf{P}_n = \dfrac{1}{\lambda}\Big[\mathbf{P}_{n-1} - \mathbf{k}_n\mathbf{x}_n^T\mathbf{P}_{n-1}\Big]$

$\quad \widehat{\mathbf{w}}_n = \widehat{\mathbf{w}}_{n-1} + \widehat{\xi}_n\mathbf{k}_n - \gamma_{n-1}(1-\lambda)\mathbf{P}_n\nabla^s f(\widehat{\mathbf{w}}_{n-1})$

end for ▷ end of recursion

After evaluating (13) using the recursions (11) and (12), we come up with the following update equation for $\widehat{\mathbf{w}}_n$.

$$\widehat{\mathbf{w}}_n = \widehat{\mathbf{w}}_{n-1} + \mathbf{k}_n\widehat{\xi}_n - \gamma_{n-1}(1-\lambda)\mathbf{P}_n\nabla^s f(\widehat{\mathbf{w}}_{n-1}) \tag{14}$$

where $\widehat{\xi}_n = y_n - \widehat{\mathbf{w}}_{n-1}^T\mathbf{x}_n$ is the a priori estimation error. Let us remember the update equation for the standard a priori RLS algorithm.

$$\widetilde{\mathbf{w}}_n = \widetilde{\mathbf{w}}_{n-1} + \mathbf{k}_n\widetilde{\xi}_n = \widetilde{\mathbf{w}}_{n-1} + \mathbf{k}_n\big(y_n - \widetilde{\mathbf{w}}_{n-1}^T\mathbf{x}_n\big) \tag{15}$$

Equation (14) differs from the standard RLS algorithm update equation (15) with the inclusion of the rightmost term. Equation (14) summarizes an adaptive algorithm which calculates (approximately) the solution to the convex regularized normal equation as given in (7). We entitle this adaptive RLS based algorithm as the "Convex Regularized-RLS" (CR-RLS). The CR-RLS algorithm is summarized in Algorithm 1.

III. Selection of the Regularization Parameter

The cost function in (4) includes the $\gamma_n f(\mathbf{w}_n)$ penalty term to put to use some a priori knowledge about the true system. The convex function f formalizes this a priori information. We assume that this a priori information is in the form of a constraint on the true system parameters $\mathbf{w}$ given as follows,

$$f(\mathbf{w}) \le \rho \tag{16}$$

where ρ denotes an upper bound constant. $\widehat{\mathbf{w}}_n$ is the solution to the convex regularized normal equation (7). $\widetilde{\mathbf{w}}_n$ is the solution to the nonregularized normal equation given as $\boldsymbol{\Phi}_n\widetilde{\mathbf{w}}_n = \mathbf{r}_n$ or $\widetilde{\mathbf{w}}_n = \mathbf{P}_n\mathbf{r}_n$. We denote the deviation of the system estimates from the true system parameters as $\widehat{\boldsymbol{\epsilon}}_n = \widehat{\mathbf{w}}_n - \mathbf{w}$ and $\widetilde{\boldsymbol{\epsilon}}_n = \widetilde{\mathbf{w}}_n - \mathbf{w}$. From (7) it follows that

$$\widehat{\boldsymbol{\epsilon}}_n = \widetilde{\boldsymbol{\epsilon}}_n - \gamma_n\mathbf{P}_n\nabla^s f(\widehat{\mathbf{w}}_n). \tag{17}$$

The instantaneous square deviation for $\widehat{\boldsymbol{\epsilon}}_n$ is calculated below,

$$\begin{aligned}\widehat{\mathcal{D}}_n &= \widehat{\boldsymbol{\epsilon}}_n^T\widehat{\boldsymbol{\epsilon}}_n = \|\widehat{\boldsymbol{\epsilon}}_n\|_2^2 \\ &= \widetilde{\mathcal{D}}_n - 2\gamma_n\nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n\widetilde{\boldsymbol{\epsilon}}_n + \gamma_n^2\|\mathbf{P}_n\nabla^s f(\widehat{\mathbf{w}}_n)\|_2^2\end{aligned} \tag{18}$$

where $\widetilde{\mathcal{D}}_n = \|\widetilde{\boldsymbol{\epsilon}}_n\|_2^2$. Equation (18) leads to the following theorem.

Theorem 1. $\widehat{\mathcal{D}}_n \le \widetilde{\mathcal{D}}_n$ if $\gamma_n \in \big[0, \max(\hat{\gamma}_n, 0)\big]$, where

$$\hat{\gamma}_n = 2\frac{\nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n\widetilde{\boldsymbol{\epsilon}}_n}{\|\mathbf{P}_n\nabla^s f(\widehat{\mathbf{w}}_n)\|_2^2}. \tag{19}$$

Proof: From (18) it is obvious that $\widehat{\mathcal{D}}_n \leq \widetilde{\mathcal{D}}_n$ as long as $\gamma_n^2\|\mathbf{P}_n\nabla^s f(\widehat{\mathbf{w}}_n)\|_2^2 - 2\gamma_n\nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n\widetilde{\boldsymbol{\epsilon}}_n \leq 0$. This condition can be rewritten as

$$\gamma_n^2\|\mathbf{P}_n\nabla^s f(\widehat{\mathbf{w}}_n)\|_2^2 \leq 2\gamma_n\nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n\widetilde{\boldsymbol{\epsilon}}_n. \tag{20}$$

We only allow $\gamma_n \geq 0$, hence when $\nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n\widetilde{\boldsymbol{\epsilon}}_n < 0$ the above inequality holds only for $\gamma_n = 0$ and becomes an equality. If $\nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n\widetilde{\boldsymbol{\epsilon}}_n \geq 0$, for the inequality to hold γ_n can be any value between 0 and $\hat{\gamma}_n$ as given in (19). □

Theorem 1 states that the CR-RLS algorithm provides an MSD as low as, and possibly lower than, that of the regular RLS algorithm, if γ_n is chosen using (19). However, it is not possible to evaluate $\hat{\gamma}_n$ in (19), because it refers to $\widetilde{\boldsymbol{\epsilon}}_n$ and hence to $\mathbf{w}$. Now we will try to find a calculable approximation to $\hat{\gamma}_n$ by replacing $\widetilde{\boldsymbol{\epsilon}}_n$. $\widetilde{\boldsymbol{\epsilon}}_n$ can be rewritten as,

$$\begin{aligned}\widetilde{\boldsymbol{\epsilon}}_n &= \widetilde{\mathbf{w}}_n - \mathbf{w} = (\widehat{\mathbf{w}}_n - \mathbf{w}) + (\widetilde{\mathbf{w}}_n - \widehat{\mathbf{w}}_n)\\ &= \widehat{\boldsymbol{\epsilon}}_n + \boldsymbol{\epsilon}'_n\end{aligned} \tag{21}$$

where $\boldsymbol{\epsilon}'_n = \widetilde{\mathbf{w}}_n - \widehat{\mathbf{w}}_n$. At this stage $\hat{\gamma}_n$ of (19) becomes

$$\hat{\gamma}_n = 2\,\frac{\nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n(\widehat{\boldsymbol{\epsilon}}_n + \boldsymbol{\epsilon}'_n)}{\|\mathbf{P}_n\nabla^s f(\widehat{\mathbf{w}}_n)\|_2^2}. \tag{22}$$

There are two terms in the right hand side nominator of (22). The term $\nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n\boldsymbol{\epsilon}'_n$ is calculable. $\boldsymbol{\epsilon}'_n$ employs $\widetilde{\mathbf{w}}_n$, the calculation of which would only require $\mathcal{O}(N)$ additional operations per time step in Algorithm 1. The second term is

$$\nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n\widehat{\boldsymbol{\epsilon}}_n = \nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n(\widehat{\mathbf{w}}_n - \mathbf{w}). \tag{23}$$

From the definition of the subgradient for a convex function f [11] and using (16) the following holds.

$$\nabla^s f(\widehat{\mathbf{w}}_n)^T(\widehat{\mathbf{w}}_n - \mathbf{w}) \geqslant f(\widehat{\mathbf{w}}_n) - f(\mathbf{w}) \geqslant f(\widehat{\mathbf{w}}_n) - \rho \tag{24}$$

Assuming the input is white and n is large enough, the following inequality can be deduced using (24).

$$\nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n\widehat{\boldsymbol{\epsilon}}_n \geqslant \frac{\mathrm{tr}(\mathbf{P}_n)}{N}\big(f(\widehat{\mathbf{w}}_n) - \rho\big) \tag{25}$$

Here, $\mathrm{tr}(\cdot)$ denotes the matrix trace operator. With (25), the $\hat{\gamma}_n$ expression in (22) modifies into

$$\hat{\gamma}_n \geqslant \gamma'_n = 2\frac{\frac{\mathrm{tr}(\mathbf{P}_n)}{N}\big(f(\widehat{\mathbf{w}}_n) - \rho\big) + \nabla^s f(\widehat{\mathbf{w}}_n)^T\mathbf{P}_n\boldsymbol{\epsilon}'_n}{\|\mathbf{P}_n\nabla^s f(\widehat{\mathbf{w}}_n)\|_2^2}. \tag{26}$$

Equation (26) presents a calculable approximation γ'_n for $\hat{\gamma}_n$ in the case of white input. The instantaneous regularization parameter can be automatically updated by $\gamma_n \in \big[0, \max(\gamma'_n, 0)\big]$ as suggested by Theorem 1, where γ'_n is calculated using (26). The operational complexity of Algorithm 1 with automatic γ_n update via (26) will be $\mathcal{O}(N^2)$ per iteration, just like the regular RLS.

IV. Simulation Results

We will employ two sparsity inducing convex penalty functions in the CR-RLS algorithm and analyze their performances in sparse system identification. The true measure of sparsity is the ℓ_0 pseudo-norm, which is known to be a nonconvex function. One obvious convex relaxation option for the ℓ_0 sparsity measure is the ℓ_1 norm. For this choice $f(\mathbf{w}) = \|\mathbf{w}\|_1 = \sum_{k=0}^{N-1}|w_k|$, where a corresponding subgradient is calculated as $\nabla^s(\|\mathbf{w}\|_1) = \mathrm{sgn}(\mathbf{w})$ [4], [8]. Here $\mathrm{sgn}(\cdot)$ is the component-wise sign function. The CR-RLS algorithm resulting from this choice of f is equivalent to the ℓ_1-RLS algorithm as outlined in [8].

Another choice for convexly relaxing ℓ_0 is the approximation as given below [3],

$$\|\mathbf{w}\|_0 \approx f^\beta(\mathbf{w}) = \sum_{k=0}^{N-1}\big(1 - e^{-\beta|w_k|}\big) \tag{27}$$

where β is an appropriate constant. A subgradient for (27) is approximately calculated as [3],

$$\nabla^s f^\beta(\mathbf{w})_k \approx \begin{cases}\beta\,\mathrm{sgn}(w_k) - \beta^2 w_k, & |w_k| \leqslant \frac{1}{\beta}\\ 0, & \text{elsewhere}\end{cases}. \tag{28}$$

This cost function with the corresponding subgradient has been utilized in the LMS context, and the resulting algorithm has been called as the ℓ_0-LMS [3]. Fittingly, we entitle the novel algorithm which results from utilizing the cost function (27) in the CR-RLS approach as the ℓ_0-RLS algorithm.

In the experiments the true system function $\mathbf{w}$ has a total of $N = 64$ taps, where only S of them are nonzero. The nonzero coefficients are positioned randomly and take their values from a $\mathcal{N}(0, \frac{1}{S})$ distribution. The input signal is $x_n \sim \mathcal{N}(0, 1)$, and measurement noise is $\eta_n \sim \mathcal{N}(0, \sigma^2)$, where σ^2 is chosen to fulfill the desired SNR. The CR-RLS algorithms are realized in two different modes, first with constant $\gamma_n = \gamma$ and secondly with automatic γ_n selection using (26). For constant case, γ is found as the optimum value which results in minimum steady-state MSD using repeated simulations. For the automatic case $\gamma_n = \max(\gamma'_n, 0)$, where γ'_n is calculated at each time instant via (26). The ρ value is taken to be the true value of $f(\mathbf{w})$, that is for ℓ_1-RLS $\rho = \|\mathbf{w}\|_1$ and for ℓ_0-RLS $\rho = \|\mathbf{w}\|_0$. We also implement the regular RLS, the SPARLS of [6] [1] and an oracle RLS algorithm. For SPARLS the algorithm parameters are fine-tuned as to result in minimum steady-state MSD. The oracle RLS is the regular RLS algorithm where the positions of the true nonzero system parameters are known. For all algorithms $\lambda = 0.995$, $\delta = 1$, and each simulation setting is averaged over 2000 independent realizations. For SPARLS $\alpha = 0.005$ and for ℓ_0-RLS $\beta = 50$.

In the first experiment we realize the algorithms for $S = 4$ and SNR $= 20$ dB. For ℓ_1-RLS the optimum $\gamma = 1.5$, for ℓ_0-RLS the optimum $\gamma = 0.2$ and for SPARLS $\gamma = 150$. We plot the variation of the MSD versus iteration number in Fig.1. The oracle RLS has the best performance as expected. On the other hand, CR-RLS algorithms present considerable improvement over the regular RLS. The ℓ_0-RLS has better performance than ℓ_1-RLS and SPARLS, and it is not very far off from the oracle. The CR-RLS variants with automatic regularization parameter selection converge to almost the same MSD values as the CR-RLS algorithms with the ad hoc, optimally selected γ. Hence, we can state that (26) presents a viable systematic method for automatically selecting γ_n in the white input case, rather than resorting to improvisation of a parameter value for each simulation setting.

[1] The authors of [6] did generously share their code for simulations.

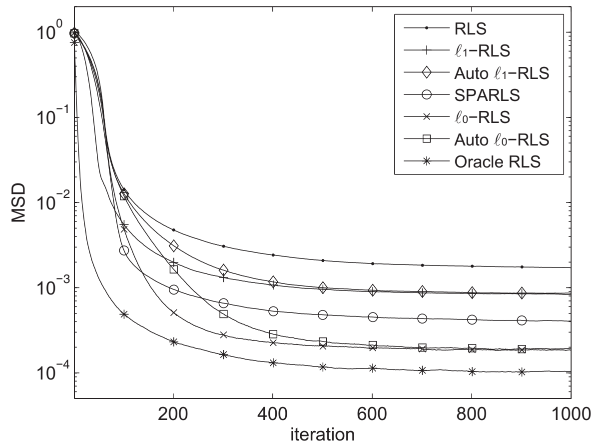

Fig. 1. Performance of different algorithms for $S = 4$ and SNR $= 20$ dB.

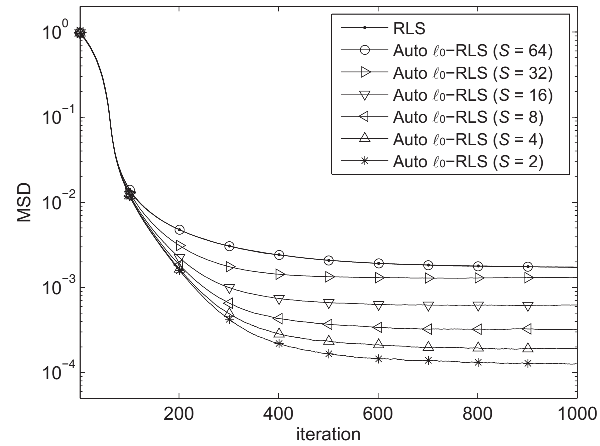

Fig. 2. Performance of auto ℓ_0-RLS and RLS under SNR $= 20$ dB and for varying S.

TABLE I
STEADY-STATE MSD FOR DIFFERENT SPARSITY VALUES.

SNR = 20 dB	$S = 2$	$S = 4$	$S = 8$	$S = 64$
RLS	1.7×10^{-3}	1.7×10^{-3}	1.7×10^{-3}	1.7×10^{-3}
Auto ℓ_1-RLS	7.6×10^{-4}	8.6×10^{-4}	1.1×10^{-3}	1.7×10^{-3}
SPARLS [6]	2.3×10^{-4}	4.1×10^{-4}	6.2×10^{-4}	1.7×10^{-3}
Auto ℓ_0-RLS	1.3×10^{-4}	1.9×10^{-4}	3.2×10^{-4}	1.7×10^{-3}
Oracle RLS	5.0×10^{-5}	1.0×10^{-4}	2.0×10^{-4}	1.7×10^{-3}

As a second experiment we consider the effect of the sparsity on the algorithm performance. We simulate the algorithms with SNR $=$ 20 dB and for $S = 2, 4, 8$ and 64, where $S = 64$ corresponds to a completely non-sparse system. The respective parameters for SPARLS are $\gamma = \{150, 150, 100, 0\}$. The steady-state MSD values at the end of 1000 iterations for the algorithms are given in Table 1. Table 1 shows that RLS performance does not vary with sparsity. Performance of the other algorithms deteriorate with decreasing sparsity, where for $S = 64$ all MSD values become equivalent. The CR-RLS algorithms have better performance than RLS when sparsity is present, and they gracefully converge to the RLS algorithm with decreasing sparsity. We also did simulations for $S = 4$ where ρ is chosen nonideally as $\rho = 10 \times f(\mathbf{w})$. The steady-state MSD for ℓ_0-RLS comes out as 3.3×10^{-4}, and for ℓ_1-RLS it comes out as 1.7×10^{-3}. These results suggest that rough selection of ρ leads to a detoriation of performance for the CR-RLS algorithms, and it can be stated that for very large ρ values the CR-RLS algorithms approach the regular RLS.

In Fig. 2, we plot the MSD curves of auto ℓ_0-RLS and RLS with varying S for SNR $=$ 20 dB. The curves affirm that auto ℓ_0-RLS performs better when sparsity is present and converges to the regular RLS when sparsity vanishes. There is no need for tweaking any parameters for the automatic CR-RLS algorithms depending on the simulation scenario.

V. CONCLUSIONS

In this letter we introduced a convex regularized RLS approach for adaptive system identification, when there is a priori information about the true system formalized in the form of a convex function. We develop the update steps for the resulting algorithm by employing subgradient analysis on the convex regularized cost function. We also present a closed-form expression for the automatic selection of the regularization parameter in the case of white input. Simulation results suggest that the automatic parameter selection works almost as well as optimizing a constant regularization parameter manually. Simulations also show that the introduced ℓ_1-RLS and ℓ_0-RLS algorithms with automatic parameter selection show better performance than RLS in the sparse setting, and that they gracefully converge to the regular RLS algorithm when sparsity vanishes.

REFERENCES

[1] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
[2] C. R. Berger, Z. Wang, J. Huang, and S. Zhou, "Application of compressive sensing to sparse channel estimation," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 164–174, Nov. 2010.
[3] Y. Gu, J. Jin, and S. Mei, "l_0 norm constraint LMS algorithm for sparse system identification," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 774–777, Sept. 2009.
[4] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proc. ICASSP*, 19–24 April 2009, pp. 3125–3128.
[5] Y. Chen, Y. Gu, and A. O. Hero, "Regularized Least-Mean-Square Algorithms," *ArXiv e-prints*, Dec. 2010, [Online]. Available: http://arxiv.org/abs/1012.5066v2.
[6] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The sparse RLS algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4013–4025, Aug. 2010.
[7] D. Angelosante, J.A. Bazerque, and G.B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the ℓ_1-norm," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3436–3447, July 2010.
[8] E. M. Eksioglu, "Sparsity Regularized RLS Adaptive Filtering," *IET Signal Process.*, Feb. 2011, to be published.
[9] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, "Adaptive algorithms for sparse system identification," *Signal Process.*, vol. 91, no. 8, pp. 1910–1919, Aug. 2011.
[10] G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh, "An adaptive greedy algorithm with application to nonlinear communications," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 2998–3007, June 2010.
[11] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific, Cambridge, Massachusetts, 2003.