ADAPTIVE RANDOMIZED COORDINATE DESCENT FOR SOLVING SPARSE SYSTEMS

Alexandru Onose[*], *Bogdan Dumitrescu*[*†]

[*] Department of Signal Processing
Tampere University of Technology
PO BOX 553, 33101, Tampere, Finland
e-mails: firstname.lastname@tut.fi

[†] Department of Automatic Control and Computers
University Politehnica of Bucharest
313 Spl. Independenţei, 060042 Bucharest, Romania
e-mail: bogdan.dumitrescu@acse.pub.ro

ABSTRACT

Randomized coordinate descent (RCD), attractive for its robustness and ability to cope with large scale problems, is here investigated for the first time in an adaptive context. We present an RCD adaptive algorithm for finding sparse least-squares solutions to linear systems, in particular for FIR channel identification. The algorithm has low and tunable complexity and, as a special feature, adapts the probabilities with which the coordinates are chosen at each time moment. We show through simulation that the algorithm has tracking properties near those of the best current methods and investigate the trade-offs in the choices of the parameters.

Index Terms— adaptive algorithm, channel identification, sparse filter, least squares, coordinate descent, randomization

1. INTRODUCTION

Randomized coordinate descent (RCD) was recently proposed and analyzed in [1, 2] for solving large optimization problems. It was shown that not only RCD has convergence speed guaranteed under general conditions (unlike deterministic versions like cyclic coordinate descent), but also it may have very low complexity.

We propose here RCD for a particular problem—finding a sparse least-squares solution to a linear system. However, the context is different than in [1], since we want an adaptive solution, not a batch one. For linear least-squares problems, cyclic coordinate descent converges, but our interest is in an adaptive, low complexity method where only few coordinates are chosen for descent at each time moment. The choice should be done cheaply, i.e. without looking at all coordinates. From this viewpoint, RCD is an ideal candidate.

Previous adaptive algorithms for sparse systems [3–7] were all deterministic and based on various ideas like using ℓ_1 regularization, greedy search, projection on convex sets. Coordinate descent was used for adaptively finding sparse solutions to linear systems in several papers [3, 8–10], using different criteria and choices of the coordinates on which descent was made (cyclic sweeps being the most popular).

Our contribution here is to propose a first adaptive RCD algorithm, minimizing directly a least-squares criterion in order to find a sparse solution to a linear system. A main innovation is in the adaptation of the probabilities that guide the RCD process, favoring the coordinates that are more likely to correspond to nonzero coefficients. Adapting probabilities during the RCD process is mentioned in [2] as a possibility, by re-evaluating the bounds on component-wise gradients (which define the probabilities). In the very recent work [11], an explicit method for changing the probabilities is given, by increasing them when the criterion decrease is larger than average; however, the context and the adaptation rule are different. For detecting the sparsity level, we rely on the Predictive Least Squares (PLS) criterion [12].

Section 2 presents the principles and the details of our algorithm. In section 3, we give evidence that our algorithm has similar behavior with previous algorithms, despite having a clearly lower complexity.

2. ADAPTIVE RANDOMIZED COORDINATE DESCENT

Our prototype problem is FIR channel identification. The channel model is

$$\sum_{i=0}^{N-1} h_i(t)u(t-i) = d(t) + \eta(t), \tag{1}$$

where $u(t)$ is the current input, $d(t)$ is the (desired) output and $\eta(t)$ is the noise. The channel model length is N and the coefficients are $h_i(t)$, possibly variable in time. We assume that the vector of coefficients is sparse, i.e. $h_i(t) \neq 0$ only for a small number L_t of indices i.

Let us denote

$$\boldsymbol{\alpha}^{(t)} = [u(t)\ u(t-1)\ \ldots\ u(t-N+1)]^T. \tag{2}$$

We use an exponential window with forgetting factor λ and define

$$\boldsymbol{A}^{(t)} = \begin{bmatrix} \sqrt{\lambda}\boldsymbol{A}^{(t-1)} \\ \boldsymbol{\alpha}^{(t)^T} \end{bmatrix}, \quad \boldsymbol{b}^{(t)} = \begin{bmatrix} \sqrt{\lambda}\boldsymbol{b}^{(t-1)} \\ d(t) \end{bmatrix}. \tag{3}$$

Assuming that $\eta(t)$ is white noise, the optimal coefficients estimate $\boldsymbol{x}^{(t)}$ can be found by minimizing the least-squares criterion

$$J(t) = \|\boldsymbol{b}^{(t)} - \boldsymbol{A}^{(t)}\boldsymbol{x}^{(t)}\|^2 \tag{4}$$

with a sparsity level that has to be detected.

2.1. Algorithm essentials

In principle, the ingredients of an adaptive RCD algorithm are simple. At the current time t, the main steps are the following.

1. Random coordinate selection. We need some probabilities π_i, $i = 0 : N-1$, that are used to choose the coordinates, with

$$\textstyle\sum_{i=0}^{N-1} \pi_i = 1. \tag{5}$$

At each time moment, we randomly choose R coordinates, selected sequentially from $0 : N-1$ with the above probabilities (hence, repetitions are allowed); we assume for simplicity that R is fixed, but it can be as well variable.

2. Descent. The chosen coordinates are used to perform optimal descent steps. Let us assume that, at time t, we have a solution with M nonzero coefficients, whose indices belong to a set $\mathcal{C}$. The residual corresponding to this solution is

$$\boldsymbol{r} = \boldsymbol{b} - \textstyle\sum_{i\in\mathcal{C}} x_i \boldsymbol{a}_i, \tag{6}$$

where $\boldsymbol{a}_i$ is the i-th column of the matrix $\boldsymbol{A}$. (To alleviate the notation, we drop the index t, implicitly referring to the current time moment.) An optimal descent step on coordinate i consists of the update

$$x_i \leftarrow x_i + \frac{\boldsymbol{a}_i^T\boldsymbol{r}}{\|\boldsymbol{a}_i\|^2} = \frac{\boldsymbol{a}_i^T(\boldsymbol{r} + x_i\boldsymbol{a}_i)}{\|\boldsymbol{a}_i\|^2}. \tag{7}$$

These operations are repeated R times, using the R randomly chosen coordinates. The residual should be updated after each modification of a coordinate; however, the implementation will avoid this update and implicitly recompute (6).

3. Selection of nonzero coefficients. Finally, since we want a sparse solution, we have to decide which coefficients are nonzero, the decision coming to effect at the next time moment. To this purpose we use an ordering of the coordinates, to be explained later, and the PLS criterion, computed for solutions formed of the first m coefficients, for all values $m = 1 : M$. Denote L the sparsity level for which the PLS criterion is minimum and $\hat{L}_t$ the sparsity level estimated for the solution at time t. Since we have noticed that small changes lead to better performance, we set

$$\hat{L}_{t+1} = \begin{cases} \hat{L}_t + 1, & \text{if } L > \hat{L}_t \\ \hat{L}_t, & \text{if } L = \hat{L}_t \\ \hat{L}_t - 1, & \text{if } L < \hat{L}_t \end{cases} \tag{8}$$

Moreover, the number of considered coefficients is taken as $M = \hat{L}_t + \Delta$, where Δ is a small constant, e.g. $\Delta = 5$; this ensures that we have enough candidates for the PLS criterion, covering the case where the sparsity levels increases. This is, excepting the way the coordinates are ordered, exactly the sparsity detection mechanism employed in [9].

The main drive for an RCD algorithm is the possibility of controlling the complexity through the number R, without affecting significantly the convergence properties. We aim to obtain an algorithm whose complexity depends rather on R than on N, as it was the case in [9] or other previous papers [3, 4].

2.2. Adapting the probabilities

The key to a successful RCD algorithm is in the choice of the probabilities π_i. In [1, 2], where sparsity is not an issue, they depend on the matrix $\boldsymbol{A}$ and they are constant. Since we seek a sparse solution, it is natural to assign larger probabilities to the coordinates with nonzero coefficients, because only they lead to a meaningful decrease of the criterion (4). Moreover, since the solution and its support may vary in time, the probabilities should be adapted in time, taking into account the latest information.

Since the matching pursuit (MP) criterion is relevant for the significance of a coordinate in the sparse solution, we propose to use it as well as a measure of the probability for drawing coordinate i. The MP criterion, computed for a coordinate i that is removed from the solution (but all other coefficients remain), has the form

$$p_i = \frac{|\boldsymbol{a}_i^T(\boldsymbol{r} + x_i\boldsymbol{a}_i)|^2}{\|\boldsymbol{a}_i\|^2}, \tag{9}$$

where $\boldsymbol{r}$ is the residual from (6). The value p_i represents the decrease of the criterion (4) when the coordinate i is introduced in the solution. As a general rule, the values p_i are large for nonzero coefficients x_i and small for zero coefficients.

We generate R random coordinates according to the probabilities π_i, as discussed in section 2.1, obtaining a sequence $\mathcal{K}$ of indices. To update the probabilities, we use a linear transformation of the values (9):

$$\pi_i \leftarrow p_{\min} + \frac{p_i}{\sum_{k\in\mathcal{K}} p_k}\left(\sum_{k\in\mathcal{K}} \pi_k - Rp_{\min}\right), \quad i \in \mathcal{K}. \tag{10}$$

This relation ensures that the scaling (5) is preserved. The value $p_{\min}$ represents the minimum value that a probability π_i can take. Such a value is necessary to ensure that each coordinate is drawn at not too long time intervals, even though the matching pursuit criterion (9) might say that its effect in the solution is negligible and so most likely that coefficient is zero. This is a precaution against sudden changes of the support. The probabilities (10) will be used at time $t+1$. The values π_i, $i \notin \mathcal{K}$, remain unchanged.

Finally, the probabilities π_i are ordered decreasingly. Due to the connection with the decrease of the criterion (4), the large probabilities are more likely to correspond to nonzero

coefficients. This order is used when building solutions with increasing sparsity level for the PLS criterion.

2.3. The adaptive RCD algorithm

The adaptive randomized coordinate descent (A-RCD) is listed in Alg. 1. We give here some explanations on the main operations.

Since the matrix $\boldsymbol{A}$ and the vectors $\boldsymbol{b}$ and $\boldsymbol{r}$ are indefinitely long, we store instead the products $\boldsymbol{\Phi} = \boldsymbol{A}^T\boldsymbol{A}$, $\boldsymbol{\Psi} = \boldsymbol{A}^T\boldsymbol{b}$. At each time moment, they are updated using the new data, in (11), see step 1 of the algorithm.

As mentioned before (6), $\mathcal{C}$ is the set of coordinates currently considered for building the sparse solution. This set is used for computing the residual $\boldsymbol{r}$ used for finding a new coefficient value in (7) and for computing the MP criterion (9). Steps 3.1–3.3 implement these relations in terms of the products $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$. (This is different from [9], where updates were used instead of recomputations of the residual.)

Steps 4 and 5 handle the probabilities as explained in section 2.2.

The size of the set $\mathcal{C}$ (to be used at the next time moment), is $\hat{L}_t + \Delta$, where $\hat{L}_t$ is the estimated sparsity level of the solution, which means that the $\hat{L}_t$ coordinates that have the largest values of the probabilities $\boldsymbol{\pi}$ form the sparse solution $\boldsymbol{x}$. As explained in section 2.1, the value $\hat{L}_t$ is computed with the help of the PLS criterion applied to sparse solutions of lengths from 1 to $M = \hat{L}_t + \Delta$, built with the coordinates with higher probabilities. Steps 6–8 contain the above operations.

Finally, since only the first M coordinates are deemed significant, the other coefficients should be forced to zero. This is done in step 9.

We have implemented A-RCD using the double residual trick proposed in [9]. Let $\mathcal{A}$ be the set of active coordinates, namely the first $\hat{L}_t$ that define the solution. If $i \in \mathcal{A}$, we compute the new coefficient x_i as in (7), but using a residual (6) built with indices from $\mathcal{A}$ instead of $\mathcal{C}$. This allows better values for the active coefficients, since the coefficients from $\mathcal{C} \setminus \mathcal{A}$, which are not considered significant, are not used; they are useful only for building the PLS criterion. For the values of the MP criterion (9), it is not so important what residual is used. Working with two residuals does not require more operations, since the value ρ from step 3.1 can be computed in two stages; first the indices from $\mathcal{A}$ are used, then this intermediate ρ is employed in step 3.2, then the remaining indices (from $\mathcal{C} \setminus \mathcal{A}$) are used for getting the final ρ.

2.4. Complexity

The only operation that has an $O(N)$ complexity is the product update in Step 1. Since a new row (2) of $\boldsymbol{A}$ is obtained by shifts, the updates (11) need not $O(N^2)$ operations, but only $O(N)$.

Generating R random indices can be done in $O(R \log N)$ operations, by building a balanced tree of cumulated probabilities. Keeping $\boldsymbol{\pi}$ sorted as required by step 5 needs also $O(R \log N)$ operations.

The computation of the coefficients and MP criterion in step 3 can be done with complexity $O(RM)$. The probability update in step 4 is only $O(R)$. Finally, the computation of the PLS criterion is cheap, requiring $O(M)$ operations.

So, the overall complexity is $O(N) + O(RM)$. Although it still depends on N, the complexity is clearly lower than for previous algorithms for the same least-squares problem [3–7, 9], where it was at least $O(MN)$, if not $O(N^2)$.

Alg. 1 (A-RCD: Adaptive randomized coordinate descent)

Parameters: R—number of random coordinates, Δ—extra length for PLS criterion, $p_{\min}$—minimum probability of a coordinate

Initialization: $\pi_i = 1/N$, $i = 0 : N-1$, $\mathcal{C} = \emptyset$, $\hat{L}_t = 0$

At each time moment t, do

1 *Update scalar products with current data*

$$\begin{aligned} \boldsymbol{\Phi} &\leftarrow \lambda\boldsymbol{\Phi} + \boldsymbol{\alpha}\boldsymbol{\alpha}^T \\ \boldsymbol{\Psi} &\leftarrow \lambda\boldsymbol{\Psi} + d(t)\boldsymbol{\alpha} \end{aligned} \tag{11}$$

2 *Generate set $\mathcal{K}$ of R random indices, using probabilities $\boldsymbol{\pi}$*

3 *for $i \in \mathcal{K}$*

3.1 *Compute $\rho = \Psi_i - \sum_{j \in \mathcal{C} \setminus \{i\}} x_j \Phi_{ij}$*

3.2 *Compute new coefficient (7): if $i \in \mathcal{C}$, $x_i = \rho / \Phi_{ii}$*

3.3 *Compute MP criterion (9): $p_i = \rho^2 / \Phi_{ii}$*

4 *Update probabilities according to (10)*

5 *Order indices in decreasing order of $\boldsymbol{\pi}$*

6 *Compute the PLS criterion for the first $\hat{L}_t + \Delta$ indices*

7 *Update $\hat{L}_t$ like in (8), with L given by the minimum PLS*

8 *Put $\mathcal{C}$ as the set of first $\hat{L}_t + \Delta$ indices*

9 *Set $x_i = 0$, for all $i \in \mathcal{K} \setminus \mathcal{C}$*

3. SIMULATIONS

We validate the performance of the randomized coordinate descent algorithm for a sparse FIR channel identification problem (1). We estimate the sparsity and the coefficient values of an L_t-sparse filter of length $N = 200$ for a variable and a constant channel, respectively. In the first case the coefficient variation is sinusoidal, each nonzero coefficient described by

$$h_i(t) = c_i \cos(2\pi f t + \phi_i). \tag{12}$$

The nonzero positions i are chosen randomly with the amplitude/constant c_i and the initial phase ϕ_i uniformly distributed

in $[0.05, 1]$ and $[0, 2\pi]$, respectively. The parameter f governing the variation speed in (12) is set to 0.0002. For the constant channel, we put $f = 0$, $\phi_i = 0$ in (12). The forgetting factor is $\lambda = 0.96$ for the variable channel and $\lambda = 0.99$ for the constant one. The filter is normed such that the average norm over all time is 1.

The input $d(t)$ is normally distributed according to $\mathcal{N}(0, 1)$ and the output is corrupted by an additive Gaussian noise with $\sigma^2 = 0.01$. We measure the performance of the algorithms in terms of the coefficient mean square error

$$\mathrm{MSE}(t) = E\{\|\boldsymbol{h} - \boldsymbol{x}\|_2^2\}, \tag{13}$$

where $\boldsymbol{x}$ is the current estimate of $\boldsymbol{h}$. It is estimated by averaging data from 1000 test runs.

The algorithms used for comparison are: RLS-SI, the sparsity informed RLS algorithm with prior knowledge of the position and number of coefficients, showing the best attainable performance; DCD-AMP, the algorithm from [9] that estimates the sparsity level using the PLS criterion with $\Delta = 5$ (this algorithm has been shown to be better than or at least comparable with those from [3, 4, 7]); A-RCD, the randomized coordinate descent algorithm presented herein, also with $\Delta = 5$; A-RCD-SI, the same algorithm, but knowing the true sparsity level (so, PLS is not used).

In Fig. 1 we present three plots with the evolution of the MSE in time. The upper figure compares the evolution of the randomized algorithms with that of the deterministic DCD-AMP algorithm from [9]. The A-RCD algorithm converges slightly slower than DCD-AMP towards an almost identical stationary MSE. If the sparsity level is known a priori, the MSE performance approaches that of RLS-SI. The test has a sudden change in the coefficient positions to exemplify the ability to track variations in the support. The convergence speed is slower after the change because for large λ the past data are forgotten very slowly.

The middle figure contains the evolution of the A-RCD algorithm for different values of the parameter R that governs the number of random descent steps. More descent steps improve the convergence speed, but a larger R also increases the numerical complexity. Thus, R is a compromise between convergence speed and numerical complexity.

The last plot in Fig. 1 contains the evolution of the MSE for different values of the minimum probability $p_{\min}$. Choosing $p_{\min}$ too small produces a slower convergence speed since some nonzero coefficients may be neglected a long time after a small value of the MP criterion (9), not unlikely at small t. For larger $p_{\min}$, the convergence speeds become similar. However, if $p_{\min}$ is too large ($p_{\min} = 0.9/N$ for instance, which makes $\boldsymbol{\pi}$ almost uniform), the performance degrades because the coefficients on the support and outside it are almost equally favored. So, the algorithm is relatively robust to the choice of $p_{\min}$ and the range of convenient values ensures also good tracking in case of support variations.

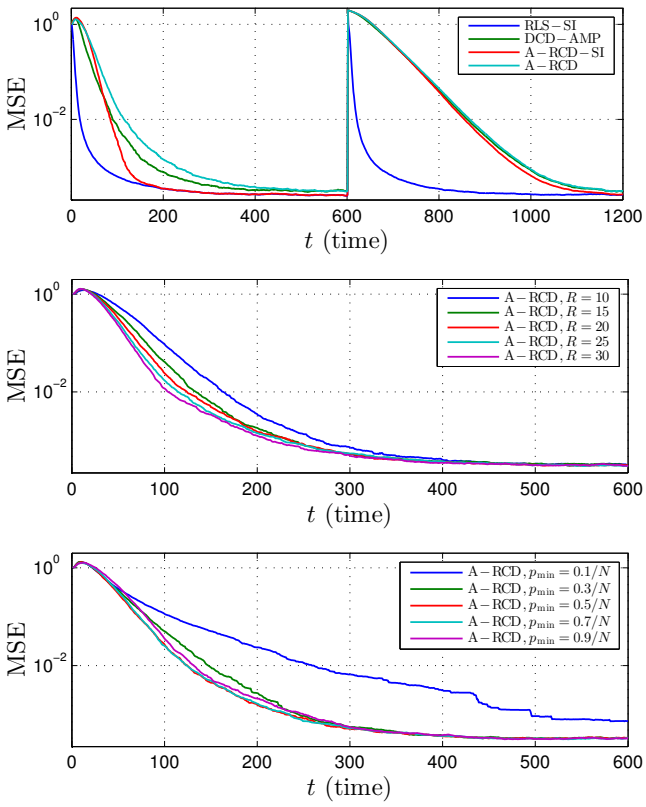

Fig. 1. MSE for a constant channel with $L_t = 5$. The parameters are $p_{\min} = 0.7/N$ and $R = 20$ if not explicitly stated.

Similar tests are performed for a sparsity level $L_t = 15$ for both the constant and the variable channels in Fig. 2 and Fig. 3, respectively. The performance remains similar for the larger sparsity and the algorithm is able to track slow varying channels.

4. CONCLUSIONS AND FUTURE WORK

We have proposed an adaptive randomized coordinate descent algorithm that adaptively updates the coordinate selection probabilities based on a MP-like criterion. It gives good performance despite its low complexity. Through extensive simulations, we have shown that the algorithm converges towards the optimal least squares solution and that the number R of descent steps governs the convergence speed.

A first glance convergence analysis, not presented here due to space limitations, suggests that the speed is inverse proportional with the number of descent steps. Further work will be dedicated to a thorough analytical proof of convergence.

REFERENCES

[1] D. Leventhal and A. S. Lewis, "Randomized Methods for Linear Constraints: Convergence Rates and Condi-

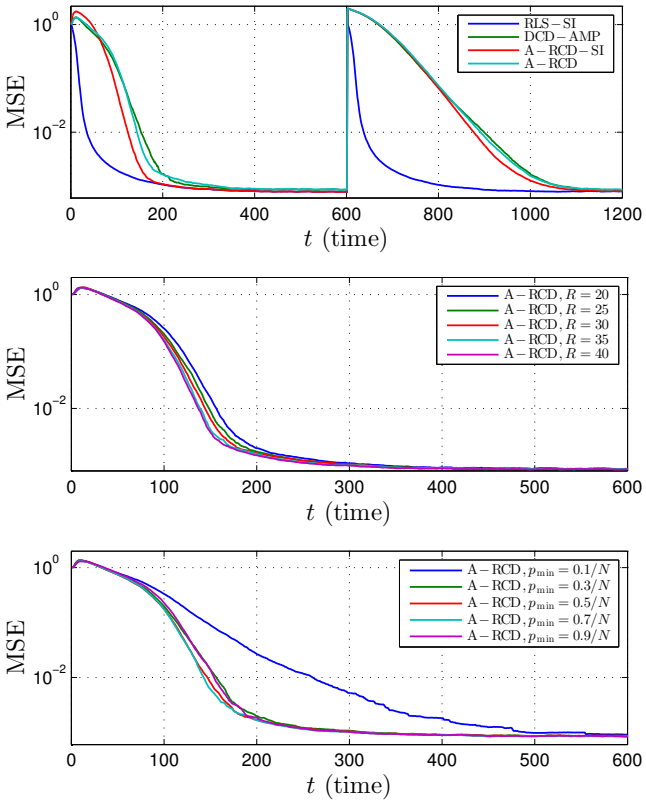

Fig. 2. MSE for a constant channel with $L_t = 15$. The parameters are $p_{\min} = 0.7/N$ and $R = 30$ if not explicitly stated.

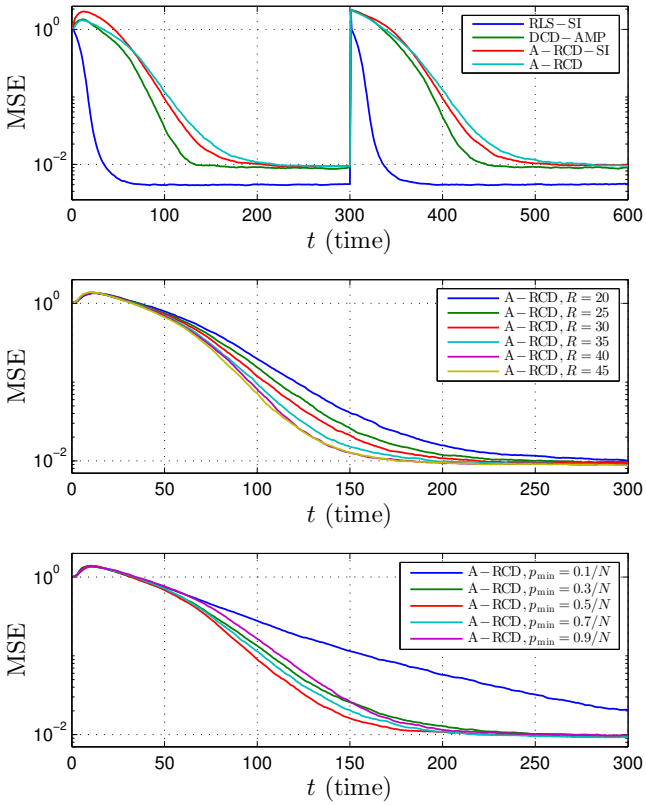

Fig. 3. MSE for a variable channel with $L_t = 15$. The parameters are $p_{\min} = 0.7/N$ and $R = 30$ if not explicitly stated.

tioning," *Math. Oper. Res.*, vol. 35, no. 3, pp. 641–654, Aug. 2010.

[2] Yu. Nesterov, "Efficiency of coordinate descent methods on huge scale optimization problems," *SIAM J. Optim.*, vol. 22, no. 2, pp. 341–362, 2012.

[3] 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 Proc.*, vol. 58, no. 7, pp. 3436–3447, July 2010.

[4] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The Sparse RLS Algorithm," *IEEE Trans. Signal Proc.*, vol. 58, no. 8, pp. 4013–4025, Aug. 2010.

[5] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online Sparse System Identification and Signal Reconstruction Using Projections Onto Weighted ℓ_1 Balls," *IEEE Trans. Signal Proc.*, vol. 59, no. 3, pp. 936–952, Mar. 2011.

[6] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, "Adaptive Algorithms for Sparse System Identification," *Signal Proc.*, vol. 91, pp. 1910–1919, 2011.

[7] B. Dumitrescu, A. Onose, P. Helin, and I. Tăbuş, "Greedy Sparse RLS," *IEEE Trans. Signal Proc.*, vol. 60, no. 5, pp. 2194–2207, May 2012.

[8] M.G. Christensen and S.H. Jensen, "The Cyclic Matching Pursuit and its Application to Audio Modeling and Coding," in *41th Asilomar Conf. Sign. Syst. Comp.*, Nov. 2007, pp. 550–554.

[9] A. Onose and B. Dumitrescu, "Adaptive matching pursuit using coordinate descent and double residual minimization," *Signal Proc.*, vol. 93, no. 11, pp. 3143–3150, 2013.

[10] Y.V. Zakharov and V.H. Nascimento, "DCD-RLS adaptive filters with penalties for sparse identification," *IEEE Trans. Signal Proc.*, vol. 61, no. 12, pp. 3198–3213, June 2013.

[11] T. Glasmachers and U. Dogan, "Accelerated coordinate descent with adaptive coordinate frequencies," in *Proc. 5th Asian Conf. Machine Learning*, 2013, vol. 29, pp. 72–86.

[12] J. Rissanen, "Order Estimation by Accumulated Prediction Errors," *J. Appl. Prob.*, vol. 23, pp. 55–61, 1986.