ADAPTIVE CYCLIC AND RANDOMIZED COORDINATE DESCENT FOR THE SPARSE TOTAL LEAST SQUARES PROBLEM

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

[†] Department of Automatic Control and Computers
University Politehnica of Bucharest
313 Spl. Independenţei, 060042 Bucharest, Romania

[*] Department of Signal Processing
Tampere University of Technology
PO BOX 553, 33101, Tampere, Finland

ABSTRACT

Coordinate descent (CD) is a simple and general optimization technique. We use it to solve the sparse total least squares problem in an adaptive manner, working on the ℓ_1-regularized Rayleigh quotient function. We propose two algorithmic approaches for choosing the coordinates: cyclic and randomized. In both cases, the number of CD steps per time instant is a parameter that can serve as a trade-off between complexity and performance. We present numerical experiments showing that the proposed algorithms can approach stationary error near that of the oracle. The randomized algorithm is slightly better than the cyclic one with respect to convergence speed.

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

1. INTRODUCTION

The total least squares (TLS) problem associated with an overdetermined linear system assumes that both the matrix of the system and the right hand side are affected by noise, unlike the least squares (LS) problem where the matrix is considered perfectly known. In signal processing, a problem matching the TLS setup is FIR channel identification with additive noise not only on the output, but also on the input. Our purpose here is to study adaptive algorithms for the sparse TLS problem.

The work on adaptive TLS (ATLS) algorithms started more than two decades ago and can be loosely split on two classes, the Rayleigh quotient function associated with TLS being minimized in both. The first class contains LMS-like algorithms, which advance with fixed step size and whose complexity is $O(N)$ per time instant. Gradient descent was recently analyzed in detail in [1], where other relevant work is cited. Other LMS-like algorithms can be found in [2–4]. The second class contain algorithms more similar to RLS, where an optimal step is performed on a chosen direction and the covariance matrix of the input is needed. The complexity is $O(N)$ for FIR channel identification, but would be $O(N^2)$ in general for covariance matrix update. The direction of the Kalman gain is used in [5], while [6] appeals to the direction of the current data; the latter algorithm has a lower complexity.

All the above work is on full solutions. For sparse TLS, we can cite only batch algorithms. A block coordinate descent method is used on an ℓ_1-regularized objective in [7], while several greedy algorithms are presented in [8].

We propose here two types of sparse ATLS algorithms using coordinate descent (CD) on the ℓ_1-regularized Rayleigh quotient function. The difference is in the order in which the coordinates are taken: cyclic or randomized. In both cases, the number of CD steps per time instant is a parameter of the algorithm, that can be used to trade-off complexity and performance. The innovation consists in the specific form of the CD steps and in the combination of techniques for obtaining performance nearing that of the oracle algorithm. In particular, the adaptation of probabilities from [9] is modified for the specific of the ATLS behavior.

The contents of the paper is as follows. After presenting the ATLS problem in section 2, we show how optimal CD steps can be computed efficiently for the considered objective in section 3. Then, in section 4, we describe the details of our algorithms. Section 5 is dedicated to simulations, showing the performance of our algorithms.

2. ADAPTIVE TLS PROBLEM

The basic problem considered here is to identify the parameters of a linear model with noise on both input and output

$$[\boldsymbol{\alpha}^{(t)} + \boldsymbol{\eta}_i^{(t)}]^T \boldsymbol{x} = \beta^{(t)} + \eta_o^{(t)}, \tag{1}$$

where $\boldsymbol{\alpha}^{(t)}$ is the input vector at time t and $\beta^{(t)}$ the corresponding output. FIR channel identification is a particular case. The input and the output are available via measurements affected by additive Gaussian noise with variance σ_i^2 and σ_o^2, respectively; we denote $\gamma = \sigma_o^2/\sigma_i^2$. The parameter vector $\boldsymbol{x}$, possibly variable in time, is unknown. We assume that the

This work was supported by the Romanian National Authority for Scientific Research, CNCS - UEFISCDI, project number PN-II-ID-PCE-2011-3-0400. E-mails: alex.onose@gmail.com, bogdan.dumitrescu@acse.pub.ro.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

length N vector $\boldsymbol{x}$ is sparse, i.e. only a few of its elements are nonzero, their locations being unknown.

We use an exponential window with forgetting factor λ and define

$$\boldsymbol{A}^{(t)} = \begin{bmatrix} \sqrt{\lambda}\boldsymbol{A}^{(t-1)} \\ \boldsymbol{\alpha}^{(t)^T} \end{bmatrix}, \quad \boldsymbol{b}^{(t)} = \begin{bmatrix} \sqrt{\lambda}\boldsymbol{b}^{(t-1)} \\ \beta^{(t)} \end{bmatrix}. \tag{2}$$

In the adaptive algorithms, where the whole data cannot be stored, we will use

$$\boldsymbol{\Phi}^{(t)} = \boldsymbol{A}^{(t)^T}\boldsymbol{A}^{(t)}, \;\; \boldsymbol{\psi}^{(t)} = \boldsymbol{A}^{(t)^T}\boldsymbol{b}^{(t)}. \tag{3}$$

instead of (2).

The solution of the estimation problem based on (1) is the (structured) total least squares (TLS) solution of the system $\boldsymbol{A}^{(t)}\boldsymbol{x}^{(t)} = \boldsymbol{b}^{(t)}$. One way of finding it is to minimize the Rayleigh quotient function [1,5]

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

with a sparsity constraint on $\boldsymbol{x}^{(t)}$.

3. COORDINATE DESCENT

The adaptive algorithms that we propose are based on coordinate descent (CD). We start by studying CD in the batch case.

3.1. CD for TLS

We derive here formulas for the optimal CD step on coordinate i for the function (4).

Let us assume that, at time t, we have a solution approximation $\boldsymbol{x}$. The residual corresponding to this solution is (we drop the time index)

$$\boldsymbol{r} = \boldsymbol{b} - \textstyle\sum_{j=0}^{N-1} x_j \boldsymbol{a}_j, \tag{5}$$

where $\boldsymbol{a}_j$ is the j-th column of the matrix $\boldsymbol{A}$. Denote

$$\tilde{\boldsymbol{r}} = \boldsymbol{b} - \textstyle\sum_{j\neq i} x_j \boldsymbol{a}_j = \boldsymbol{r} + x_i \boldsymbol{a}_i \tag{6}$$

the residual without the contribution of the i-th coordinate. Denote

$$\tilde{\gamma} = \gamma + \textstyle\sum_{j\neq i} x_j^2. \tag{7}$$

Isolating the i-th coordinate, the function (4) has the form

$$J(x_i) = \frac{\|\tilde{\boldsymbol{r}} - x_i\boldsymbol{a}_i\|^2}{x_i^2 + \tilde{\gamma}}. \tag{8}$$

The gradient of this function with respect to x_i is

$$\frac{\partial J}{\partial x_i} = 2\frac{c_0 x_i^2 + c_1 x_i + c_2}{(x_i^2 + \tilde{\gamma})^2}, \tag{9}$$

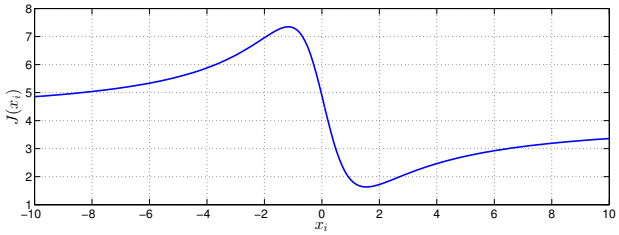

Fig. 1. Typical form of function (8).

where

$$\begin{aligned} c_0 &= \boldsymbol{a}_i^T\tilde{\boldsymbol{r}} \\ c_1 &= \tilde{\gamma}\|\boldsymbol{a}_i\|^2 - \|\tilde{\boldsymbol{r}}\|^2 \\ c_2 &= -\tilde{\gamma}\boldsymbol{a}_i^T\tilde{\boldsymbol{r}} = -\tilde{\gamma}c_0 \end{aligned} \tag{10}$$

The gradient is zero for the values

$$\xi_\pm = \frac{-c_1 \pm \sqrt{c_1^2 - 4c_0c_2}}{2c_0} \tag{11}$$

Since c_0 and c_2 have opposite signs, one of these values is positive and the other negative.

The function (8) has finite (and equal) values at $\pm\infty$. The zeros of the gradient correspond to a minimum and a maximum. See figure 1 for a typical shape of the function. Since the zeros have opposite signs, we can distinguish the minimum by looking at the sign of the gradient for $x_i = 0$, i.e. at the sign of c_2. So, the optimal value of the coordinate x_i when the other coordinates are fixed is

$$\xi = \begin{cases} \max(\xi_+, \xi_-), & \text{if } c_2 < 0 \\ \min(\xi_+, \xi_-), & \text{if } c_2 > 0 \end{cases} \tag{12}$$

The decrease of the function (8) when x_i goes from zero to its optimal value is

$$\Delta J = \frac{\|\tilde{\boldsymbol{r}}\|^2}{\tilde{\gamma}} - \frac{\|\tilde{\boldsymbol{r}} - \xi\boldsymbol{a}_i\|^2}{\xi^2 + \tilde{\gamma}} = \frac{\boldsymbol{a}_i^T\tilde{\boldsymbol{r}}\cdot\xi}{\tilde{\gamma}} \tag{13}$$

(Note that indeed $\Delta J \geq 0$, due to the choice (12) and the values (10).)

3.2. CD for ℓ_1-regularized TLS

If the parameter vector $\boldsymbol{x}$ from (1) is sparse, then it can be estimated by minimizing the ℓ_1-regularized TLS objective

$$\hat{J}(\boldsymbol{x}) = J(\boldsymbol{x}) + \mu\|\boldsymbol{x}\|_1, \tag{14}$$

where $J(\boldsymbol{x})$ is defined in (4) and μ is a positive weight. Optimal coordinate descent is still possible and we explain here how it can be done exactly.

Let us assume that the optimal value (12) of the i-th coordinate is positive when minimizing the TLS objective (8); the

case $\xi < 0$ is similar. CD on the i-th coordinate for (4) means minimizing

$$\hat{J}(x_i) = J(x_i) + \mu|x_i| + \text{const.} \tag{15}$$

Due to the shape of function (8) (see again figure 1), the value $\hat{\xi}$ at which $\hat{J}(x_i)$ is minimum is such that $\hat{\xi} < \xi$. The discontinuity of the derivative of (15) in the origin means that either $\partial\hat{J}/\partial\hat{\xi} = 0$ for some $\hat{\xi} > 0$ or $\hat{\xi} = 0$.

To minimize (15) we attempt to solve, for $x_i \geq 0$, the equation

$$\frac{\partial J}{\partial x_i} + \mu = 0. \tag{16}$$

In view of (9), this means finding the roots of the polynomial

$$q(x_i) = \mu(x_i^2 + \gamma)^2 + 2(c_0 x_i^2 + c_1 x_i + c_2). \tag{17}$$

Let ζ be the largest real root that is smaller than ξ; if there is no such root, we formally take $\zeta = -\infty$. Then, the minimum of (15) is obtained for

$$\hat{\xi} = \max(\zeta, 0). \tag{18}$$

This is the typical soft thresholding for ℓ_1-regularized functions. Note that (17) is a fourth order polynomial, whose roots can be found through explicit formulas. Since the coefficient of x_i^3 is zero, the formulas are somewhat simpler than in the general case. The number of operations is around 50, in any case less than usual values of N.

4. ADAPTIVE TLS ALGORITHMS

Algorithm 1 presents the details of our CD adaptive approach for solving the TLS problem. At each time instant a number of R coordinates are updated, with $R < N$. There are two choices of coordinates that we considered. One is the usual cyclic CD, with the remark that a full sweep is done over several time instants. The other is randomized CD, similar to the least-squares solution from [9]. Besides the choice of coordinates, the other operations are identical.

Step 5 contains the update of the data-defined matrices from (3), whose elements are sufficient for all subsequent calculations. The update needs $O(N^2)$ operations in general, but only $O(N)$ in the case of FIR channel identification, where the first $N-1$ elements of vector $\boldsymbol{\alpha}^{(t)}$ are generated by shifting $\boldsymbol{\alpha}^{(t-1)}$. For efficient computation we permanently update the values of the squared norm of the current solution $\boldsymbol{x}$ and residual $\boldsymbol{r}$. Step 6 updates $\|\boldsymbol{r}\|^2$ with the error given by the current equation.

The loop 7 contains the R CD steps. The solution norm $\|\boldsymbol{x}\|^2$ is used in step 9 to compute the useful quantity $\tilde{\gamma}$ from (7) and updated in step 16 with the new value of the current coordinate; in the long run, such an update may be numerically unsafe and so the recomputation of $\|\boldsymbol{x}\|^2$ once every several time instants is recommendable.

Algorithm 1: Cyclic and randomized adaptive TLS

1 Main parameters: R, number of CD steps per time moment; λ, forgetting factor; γ, noise variances ratio
2 Initialize $\boldsymbol{x} = 0$, $\|\boldsymbol{x}\|^2 = 0$, $\|\boldsymbol{r}\|^2 = 0$, $\pi_\ell = 1/N$, $\nu_\ell = \nu_{\text{av}}/N$, $w_\ell = 1$, $\ell = 0 : N-1$
3 cyclic: $i \leftarrow 0$
4 **for** $t = 1, 2, \ldots$ **do**
5 Update data products
$$\boldsymbol{\Phi} \leftarrow \lambda\boldsymbol{\Phi} + \boldsymbol{\alpha}^{(t)}\boldsymbol{\alpha}^{(t)^T}$$
$$\boldsymbol{\psi} \leftarrow \lambda\boldsymbol{\psi} + \beta^{(t)}\boldsymbol{\alpha}^{(t)}$$
6 Update residual norm
$\|\boldsymbol{r}\|^2 \leftarrow \lambda\|\boldsymbol{r}\|^2 + [\beta^{(t)} - \boldsymbol{\alpha}^{(t)^T}\boldsymbol{x}]^2$
7 **for** $k = 0 : R-1$ **do**
8 randomized: generate random i using $\boldsymbol{\pi}$
9 $\tilde{\gamma} \leftarrow \gamma + \|\boldsymbol{x}\|^2 - x_i^2$
10 Compute c_0 with (19)
11 Compute $\|\tilde{\boldsymbol{r}}\|^2$ with (20)
12 $c_1 = \tilde{\gamma}\phi_{ii} - \|\tilde{\boldsymbol{r}}\|^2$
13 $c_2 = -\tilde{\gamma}c_0$
14 Compute new x_i with (18) and the construction before it, and weight defined by (21), (25), (26)
15 Compute probability related quantity (22)
16 $\|\boldsymbol{x}\|^2 \leftarrow \tilde{\gamma} + x_i^2 - \gamma$
17 $\|\boldsymbol{r}\|^2 \leftarrow \|\tilde{\boldsymbol{r}}\|^2 - 2c_0 x_i + \phi_{ii}x_i^2$
18 cyclic: $i \leftarrow (i+1) \bmod N$
19 Compute new probabilities $\boldsymbol{\pi}$ with (23), (24)

Using the compacted data (3), the coefficients (10) that determine the size of the CD step can be efficiently computed. From (6) we obtain

$$c_0 = \boldsymbol{a}_i^T\tilde{\boldsymbol{r}} = \psi_i - \sum_{j\neq i}\phi_{ij}x_j. \tag{19}$$

The same relation gives

$$\|\tilde{\boldsymbol{r}}\|^2 = \|\boldsymbol{r} + x_i\boldsymbol{a}_i\|^2 = \|\boldsymbol{r}\|^2 + 2c_0x_i - \phi_{ii}x_i^2. \tag{20}$$

Hence, steps 9–13 produce the coefficients (10). The cost is only $O(N)$, given by the computation of c_0.

Before describing the other operations, let us point out that TLS is more challenging than LS, since the objective (4) favors higher values of the solution $\boldsymbol{x}$ than the LS objective, which is only the numerator of (4). Hence, especially in the beginning of the TLS adaptive process, it is possible that a coordinate may get a high value, if such a value does not affect too much the residual, but decreases the objective by its sheer magnitude. Such an event is less likely if the ℓ_1 regularization weight from (14) is large. However, a large weight would bias the TLS solution. A solution, see e.g. [10], is to take in (15)

$$\mu = \mu_0 \cdot w_i \tag{21}$$

where w_i is 1 if the coefficient x_i is likely to be small and decreases to zero as the coefficient magnitude is likely to be larger. We will give later the exact relation for w_i.

The probabilities $\boldsymbol{\pi}$ associated with the coordinates choices in step 8 of Algorithm 1 have two components. One is taken similarly to the randomized RLS algorithm from [9], i.e. it is proportional to the decrease (13) of the TLS objective produced by a CD step and hence to the quantity

$$p_i = \frac{c_0 \xi}{\tilde{\gamma}}, \tag{22}$$

where ξ is the optimal TLS step given by (12). So, the first component is meant to ensure that the coordinates with large contribution to the objective are selected more often then the others. The second component, denoted ν_i, tries to ensure a certain fairness in the coordinate selection; this is different from the minimum probability technique used in [9] and has the purpose of promoting the selection of coordinates not selected for a long time, which is more dangerous for TLS than for LS. Denoting $\mathcal{S}$ the set of coordinates chosen at time t, we use the update rule

$$\nu_i^{(t+1)} = \begin{cases} 0 & \text{if } i \in \mathcal{S} \\ \nu_i^{(t)} + \frac{\sum_{\ell \in \mathcal{S}} \nu_\ell^{(t)}}{N - |\mathcal{S}|} & \text{if } i \notin \mathcal{S} \end{cases} \tag{23}$$

So, a selected coordinate sees its probability drastically decreased for the next time instants, while the others benefit from an increase that keeps the overall sum constant. We define ν_{av} the average value of the second component, hence $\sum_\ell \nu_\ell = N\nu_{\text{av}}$. The overall probabilities are defined via

$$\pi_i = \nu_i + \frac{p_i}{\sum_{\ell=0}^{N-1} p_\ell} \left(1 - N\nu_{\text{av}}\right). \tag{24}$$

Coming back to the weights (21), we propose a logarithmic function [11] that depends on the probabilities (24)

$$w_i(\pi_i) = \begin{cases} 1 & \text{if } \pi_i - \nu_i \le \tau \\ \frac{\log_2(g_\upsilon) - \log_2(g_\tau + \vartheta)}{\log_2(g_\upsilon) - \log_2(g_\tau)} & \text{if } \tau < \pi_i - \nu_i < \upsilon \\ 0 & \text{if } \upsilon \le \pi_i - \nu_i \end{cases} \tag{25}$$

where $\vartheta = (g_\upsilon - g_\tau)\frac{\pi_i - \nu_i - \tau}{\upsilon - \tau}$. The constants g_υ, g_τ mostly define the shape of the function; υ is the probability threshold over which the coefficient is considered surely nonzero, while τ is the probability threshold under which a coefficient is considered surely zero. For the cyclic version of the algorithm we use the same weighting scheme, since the relation between the probabilities and the magnitude of the coefficients is the same. Finally, to prevent large changes of the weight, induced by the above mentioned possible sudden growth of a coefficient, we set

$$w_i^{(t)} = \rho w_i^{(t-1)} + (1 - \rho) w_i(\pi_i), \tag{26}$$

where ρ is a constant near to 1. Thus, the values produced by the logarithmic law (25) have small importance at a certain time instant and only repeated occurrence of similar values may drive the weight towards zero or one. (A zero weight would be especially dangerous for a coefficient that should be zero but gets accidentally a large value.)

The above weight calculation and the operations described in section 3.2 allow the update of the current coordinate x_i in step 14 of the algorithm. The remaining steps update other useful values, including $\|\boldsymbol{r}\|^2$ in step 17 by reversing (20).

The number of operations per time instant is $O(RN)$, with a small constant multiplying RN. So, as advocated in [9], the number of CD steps R can effectively serve as a trade-off between complexity and performance.

5. SIMULATIONS

We name ATLS-C and ATLS-R our algorithms, the first being the cyclic and the second the randomized versions of Algorithm 1. The parameters common to all simulations are as follows. The average probability from (24) is $\pi_{\text{av}} = 0.7/N$. The logarithmic function used to compute the weights (25) is defined with $g_\tau = 2$, $g_\upsilon = 4$. The constant from (26) is $\rho = 0.99$. The other parameters, namely ℓ_1 regularization penalty from (21) and the thresholds τ, υ used to compute the weights are specified for each simulation setup. For comparison we use the oracle RTLS-O, which implements the algorithm from [6] knowing the locations of the nonzero elements. RLS-O is the recursive least-squares oracle.

We consider two simulation scenarios, both for an FIR channel with $N = 200$ coefficients of which L are nonzero. In the first, the channel is constant; in the second, the channel is variable and the coefficients have a sinusoidal variation with a period of 5000 samples. The magnitude of the coefficients is generated randomly and the magnitude of the vector of coefficients is normalized to 1. The positions of the nonzero coefficients are randomly chosen. In both scenarios, there are two types of noise. We take either $\sigma_i = \sigma_o = 0.01$ (hence $\gamma = 1$) or $\sigma_i = 0.05$, $\sigma_o = 0.01$ (hence $\gamma = 0.2$)

Figure 2 presents the evolution of solution MSE for the constant channel with $L = 5$ coefficients. The horizontal green line marks the stationary level attained by the RTLS algorithm from [6] (that assumes a full solution) at about $t = 1500$; the full curve is not drawn due to its very slow convergence. The algorithms using the randomized coordinate selection converge consistently faster than their cyclic counterparts. The stationary error approaches that of the TLS oracle and is lower than that of the LS oracle, especially if the inputs are noisier.

Figure 3 presents the evolution of the solution MSE for the variable channel for different sparsity levels, $L = 5$ and $L = 15$. The algorithms are robust and are able to track slow changes in the coefficients values. For a low sparsity level the performance approaches that of the LS oracle while for larger sparsity levels the performance degrades slightly.

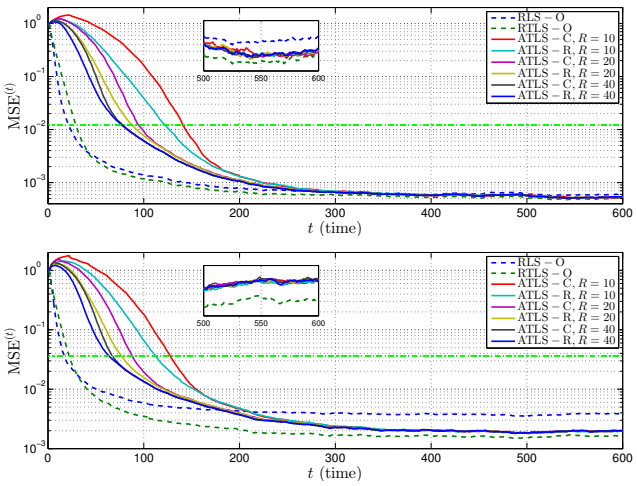

Fig. 2. MSE for a constant channel with $L = 5$, $\lambda = 0.99$, $\mu_0 = 4$, $\tau = 0.05/N$, $\upsilon = 0.15/N$ and (up) $\gamma = 1$, (down) $\gamma = 0.2$.

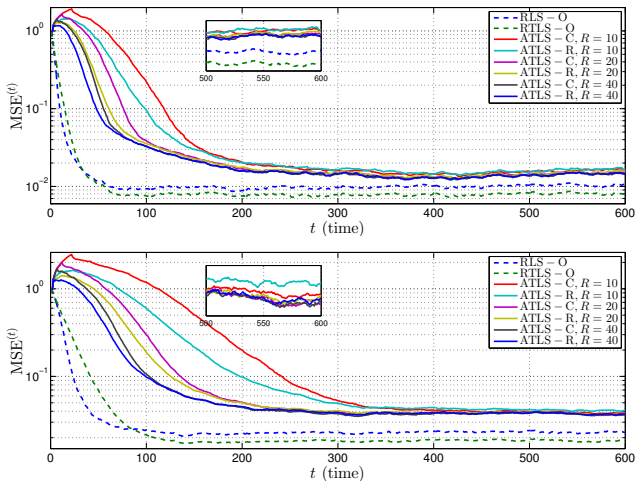

Fig. 3. MSE for a variable channel with $\lambda = 0.96$, $\gamma = 0.2$, $\tau = 0.15/N$, $\upsilon = 0.25/N$ and (up) $L = 5$, $\mu_0 = 8$, (down) $L = 15$, $\mu_0 = 5$.

6. CONCLUSIONS AND FUTURE WORK

We have proposed two adaptive, sparsity aware algorithms for solving the total least squares problem. Both are based on coordinate descent on the ℓ_1-regularized Rayleigh quotient criterion function. The coordinates for each update are selected using either a cyclic or probabilistic approach. For the probabilistic approach, the probabilities are updated online based on the decrease of the criterion.

Despite their low computational burden, the proposed algorithms have good performance, approaching that of the oracle TLS. The number R of descent steps governs the convergence speed and serves a tradeoff between complexity and performance.

Further work will be dedicated towards the mathematical analysis of the algorithm performance. It will also aim to provide an online adaptation of the ℓ_1 penalty weight and of the probability thresholds.

REFERENCES

[1] R. Arablouei, S. Werner, and K. Dogancay, "Analysis of the Gradient-Descent Total Least-Squares Adaptive Filtering Algorithm," *IEEE Trans. Signal Proc.*, vol. 62, no. 5, pp. 1256–1264, Mar. 2014.

[2] K. Gao, M. Omair Ahmad, and M.N.S. Swamy, "A Constrained Anti-Hebbian Learning Algorithm for Total Least-Squares Estimation with Applications to Adaptive FIR and IIR Filtering," *IEEE Trans. Circ. Syst. II*, vol. 41, no. 11, pp. 718–729, Nov. 1994.

[3] D.Z. Feng, Z. Bao, and L.C. Jiao, "Total Least Mean Squares Algorithm," *IEEE Trans. Signal Proc.*, vol. 46, no. 8, pp. 2122–2130, Aug. 1998.

[4] S.E. Jo and S.W. Kim, "Consistent Normalized Least Mean Square Filtering With Noisy Data Matrix," *IEEE Trans. Signal Proc.*, vol. 53, no. 6, pp. 2112–2123, June 2005.

[5] C.E. Davila, "An Efficient Recursive Total Least Squares Algorithm for FIR Adaptive Filtering," *IEEE Trans. Signal Proc.*, vol. 42, no. 2, pp. 268–280, Feb. 1994.

[6] D.Z. Feng, X.D. Zhang, D.X. Chang, and W.X. Zheng, "A Fast Recursive Total Least Squares Algorithm for Adaptive FIR Filtering," *IEEE Trans. Signal Proc.*, vol. 52, no. 10, pp. 2729–2737, Oct. 2004.

[7] H. Zhu, G. Leus, and G.B. Giannakis, "Sparsity-Cognizant Total Least-Squares for Perturbed Compressive Sampling," *IEEE Trans. Signal Proc.*, vol. 59, no. 5, pp. 2002–2016, May 2011.

[8] B. Dumitrescu, "Sparse Total Least Squares: Analysis and Greedy Algorithms," *Lin. Alg. Appl.*, vol. 438, no. 6, pp. 2661–2674, Mar. 2013.

[9] A. Onose and B. Dumitrescu, "Adaptive Randomized Coordinate Descent for Solving Sparse Systems," in *EUSIPCO*, Lisbon, Portugal, 2014.

[10] 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.

[11] H. Zou and R. Li, "One-Step Sparse Estimates in Nonconcave Penalized Likelihood Models," *Annals Stat.*, vol. 36, no. 4, pp. 1509–1533, 2008.