ADAPTIVE IDENTIFICATION OF SPARSE SYSTEMS USING THE SLIM APPROACH

George-Othon Glentis

Department of Informatics and Telecommunications, University of Peloponnese, Greece

ABSTRACT

In this paper, a novel time recursive implementation of the Sparse Learning via Iterative Minimization (SLIM) algorithm is proposed, in the context of adaptive system identification. The proposed scheme exhibits fast convergence and tracking ability at an affordable computational cost. Numerical simulations illustrate the achieved performance gain in comparison to other existing adaptive sparse system identification techniques.

Index Terms— Adaptive system identification, Sparse systems, SLIM algorithm

1. INTRODUCTION

With the advent of compressing sensing adaptive sparsity aware algorithms for system identification have been emerged, offering significant advantage over the classic schemes [1, 2]. Under certain mild assumptions about the system model and the driven signals, these methods provide more accurate estimates of the sought parameters and exhibit better performance with respect to the speed of convergence, the tracking ability and often the complexity burden (see among others [3–9] and the references therein).

A non parametric sparse estimation method, the Sparse Learning via Iterative Minimization (SLIM) algorithm, has recently been proposed in the context of radar imaging [10]. The SLIM algorithm was subsequently utilized in various applications, such as spectral analysis and Synthetic Aperture Radar (SAR) imaging, channel estimation of underwater communications channels, multistatic active sonar signal processing and wideband source location [11–16]. It was demonstrated that, compared to other competitive schemes, the SLIM algorithm offers a superior performance at an affordable computational cost.

The SLIM algorithm has been designed for batch processing. Direct application of the original scheme [10] for adaptive processing results to an exceptional increase in the computational load. Moreover, processing successive blocks of data independently, may result to inferior performance, as prior information related to the sought parameters is not incorporated into the current estimates.

In this paper, a novel time recursive implementation of the SLIM algorithm is proposed in the context of adaptive system identification. Reminiscing our earlier work on the Iterative Adaptive Approach (IAA) algorithm [17], a time recursive scheme is built up for the SLIM algorithm, able of updating the sought parameters on a sample by sample basis. A fast implementation is derived, offering a significant reduction in the required computational load. The proposed adaptive scheme offers improved convergence speed and tracking ability compared to other existing schemes [4, 7], at an affordable computational cost. The superior performance of the proposed approach is illustrated by means of computer simulation.

2. THE SLIM ALGORITHM

Let $x(k) \in \Re$, $k = 0, 1, 2, \ldots$ be the input signal of a Finite Impulse Response (FIR) system and $y(k)$ be the measured output signal described by

$$y(k) = \mathbf{x}_M^T(k)\mathbf{c}_M + \eta(k), \tag{1}$$

where $\mathbf{c}_M = [c_1\ c_2\ \ldots\ c_M]^T$ is a vector that carries the system parameters, $\mathbf{x}_M(k) = [x(k)\ x(k-1)\ \ldots\ x(k-M+1)]^T$ is the regressor, whereas $\eta(k) \in \Re$ denotes the disturbance signal, with $(\cdot)^T$ denoting the transpose. The parameter M is related to the memory of the system and determines the maximum number of taps to be considered in a specific application. If the number of the non-zero elements in $\mathbf{c}_M$ is much smaller than M, then the system is said to be sparse. Given a set of measurements at time instant n, we construct the input-output data relationship as

$$\mathbf{y}_L(n) = \mathbf{X}_L(n)\mathbf{c}_M(n) + \boldsymbol{\eta}_L(n) \tag{2}$$

where

$$\mathbf{y}_L(n) = \begin{bmatrix} y(n-L+1) \\ \vdots \\ y(n-1) \\ y(n) \end{bmatrix}, \quad \mathbf{X}_L(n) = \begin{bmatrix} \mathbf{x}_M^T(n-L+1) \\ \vdots \\ \mathbf{x}_M^T(n-1) \\ \mathbf{x}_M^T(n) \end{bmatrix} \tag{3}$$

and $\boldsymbol{\eta}_L(n) = [\ \eta(n-L+1)\ \ldots\ \eta(n-1)\ \eta(n)\ ]^T$, with $L > 0$ denoting the number of available measurements at time instant n. The SLIM algorithm introduced in [10] is formed by minimizing, with respect to $\mathbf{c}_M(n)$ and $\sigma^2(n)$, the regularized cost function

$$L\log(\sigma^2(n)) + \frac{1}{\sigma^2(n)}||\mathbf{y}_L(n) - \mathbf{X}_L(n)\mathbf{c}_M(n)||_2^2 +$$

$$\frac{2}{p}\sum_{\kappa=1}^{M}\left(|c_\kappa(n)|^p - 1\right) \qquad (4)$$

where $0 < p \leq 1$ is a parameter that controls the level of sparsity. $\sigma^2(n)$ denotes an estimate of the variance of the noise signal $\eta(k)$. The last term in (4) is a penalty term, which in the case when $p = 1$ it reduces to $2||\mathbf{c}_M(n)||_1 - 2M$, while as $p \rightarrow 0$ it becomes $2\sum_{\kappa=1}^{M}\log\left(|c_\kappa(n)|\right)$. SLIM is a non parametric sparse system identification method capable of producing estimates of $\mathbf{c}_M(n)$ and $\sigma^2(n)$ using a coupled iterative procedure. It can be interpreted as a maximum a posteriori (MAP) method and existing sparse estimation schemes [18, 19] can be viewed as special cases (see [10] for further details). The SLIM-0 is formed by iterating (5)-(9), for $i = 0, 1, \ldots$

$$\begin{aligned}
w_\kappa^{(i)}(n) &= |c_\kappa^{(i-1)}(n)|^{2-p}, \quad \kappa = 1, 2, \ldots, M && (5)\\
\mathbf{W}_M^{(i)}(n) &= \mathrm{diag}\left\{w_1^{(i)}(n),\ w_2^{(i)}(n),\ \ldots,\ w_M^{(i)}(n)\right\} && (6)\\
\mathbf{R}_L^{(i)}(n) &= \mathbf{X}_L(n)\mathbf{W}_M^{(i)}(n)\mathbf{X}_L^T(n) + \sigma^{2(i-1)}(n)\mathbf{I}_L && (7)\\
\mathbf{c}_M^{(i)}(n) &= \mathbf{W}_M^{(i)}(n)\mathbf{X}_L(n)^T\left[\mathbf{R}_L^{(i)}(n)\right]^{-1}\mathbf{y}_L(n) && (8)\\
\sigma^{2(i)}(n) &= \frac{1}{L}||\mathbf{y}_L(n) - \mathbf{X}_L(n)\mathbf{c}_M^{(i)}(n)||_2^2 && (9)
\end{aligned}$$

until practical convergence, with $\mathrm{diag}\{.\}$ denoting a diagonal matrix. $\mathbf{c}_M^{(0)}(n)$ is initialized using the ordinary Least Squares (LS) or the Ridge Regression LS (RR-LS) [20] and $\sigma^{2(0)}(n)$ is set equal to a small positive value. The brute force implementation of the SLIM algorithm requires approximately $ML^2 + L^3/3$ operations per iteration, and usually 10-15 iterations are sufficient for convergence.

We hereafter focus on the particular case when $p = 0$, resulting to an estimation scheme known as the SLIM-0, algorithm [12, 14–16], noting that the presented work is also valid for a different choice of p. Experimental evidence supports the claim that SLIM-0 produces sparser estimates compared to other possible SLIM alternatives resulting from a different choice of p (see also [14–16] for the derivation of the SLIM-0 algorithm using a hierarchical Bayesian reasoning).

3. THE TIME RECURSIVE SLIM-0 ALGORITHM

When adaptive processing is considered, the SLIM-0 may be applied directly on each block of data in (3), processing consecutive and overlapped data blocks one at a time, as being in batch mode, noting however that this form of updating will result in an unnecessarily heavy work load. Reminiscing our earlier work on the time recursive implementation of the IAA algorithm in the context of spectral estimation [17], we propose a time recursive updating of the SLIM-0 algorithm, where estimates at time instant $(n-1)$ are used for the initialization of the corresponding recursions at time instant n. Moreover, as time evolves and upon convergence, $\mathbf{c}_M^{(i)}(n) \approx \mathbf{c}_M^{(i)}(n-1)$. This fact allows for the use of a single SLIM-0 iteration each time a new set of measurements is available, i.e., $i = 1$ and $\mathbf{W}_M^{(0)}(n) \approx \mathbf{W}_M^{(1)}(n-1)$. We further assume that the noise variance is constant and it is given, i.e. $\sigma^2(n) = \sigma_0^2$. This is a commonly adopted assumption in existing adaptive methods for sparse system identification [4, 7]. The assumption that $\sigma^2(n)$ is constant and known beforehand, can be relaxed and it is here adopted solely for reasons of comparison with existing schemes [4, 7].

Table 1. The Time Recursive SLIM-0 Algorithm

$$\begin{aligned}
\mathbf{R}_L(n) &= \mathbf{X}_L(n)\mathbf{W}_M(n-1)\mathbf{X}_L^T(n) + \sigma_0^2\mathbf{I}_L && (1)\\
\mathbf{c}_M(n) &= \mathbf{W}_M(n-1)\mathbf{X}_L^T(n)\left[\mathbf{R}_L(n)\right]^{-1}\mathbf{y}_L(n) && (2)\\
w_\kappa(n) &= |c_\kappa(n)|^2 + w_0(n), \quad \kappa = 1, \ldots, M && (3)\\
\mathbf{W}_M(n) &= \mathrm{diag}\left\{w_1(n),\ \ldots,\ w_M(n)\right\} && (4)
\end{aligned}$$

The resulting algorithm, referred to hereafter as the Time Recursive SLIM-0 (TR-SLIM-0) algorithm is tabulated in Table 1. A small positive constant, denoted by $w_0(n)$, has been inserted in the equation that determines the updating of the weights. Otherwise, the system coefficients whose magnitude is close to zero at a specific time instant could not be able to adapt in the case of variation in either the location or in the magnitude of the sought system parameters. Thus, $w_\kappa(n) = |c_\kappa(n)|^2 + w_0(n)$, $\kappa = 1, 2, \ldots, M$ with $w_0(n) > 0$ is used instead of (5). A reasonable choice for $w_0(n)$ is the use of the (smoothed) normalized, mean squared deviation between the values of two successive estimates of $\mathbf{c}_M(n)$, given by

$$w_0(n) = \frac{||\boldsymbol{\delta}_M(n)||_2^2}{||\boldsymbol{\gamma}_M(n)||_2^2} \qquad (10)$$

where

$$\begin{aligned}
\boldsymbol{\delta}_M(n) &= \nu\boldsymbol{\delta}_M(n-1) + (1-\nu)|\mathbf{c}_M(n) - \mathbf{c}_M(n-1)|^2\\
\boldsymbol{\gamma}_M(n) &= \nu\boldsymbol{\gamma}_M(n-1) + (1-\nu)|\mathbf{c}_M(n)|^2 && (11)
\end{aligned}$$

and where $\nu \in (0, 1]$ is a smoothing factor. Upon convergence, $w_0(n)$ is expected to be small enough, but not equal zero. When the system undergoes variations either in the location or in the magnitude of the non-zero coefficients, $w_0(n)$ will move towards larger values. In this way, the elements of $\mathbf{c}_M(n)$ that have been settled into some location close to zero, will be able to re-adapt again following the system variation.

3.1. Fast Implementation

The computational complexity of the proposed TR-SLIM-0 algorithm is $\mathcal{O}\left(ML^2 + L^3\right)$ which is still higher than that of other existing $\mathcal{O}\left(M^2\right)$ or $\mathcal{O}\left(LM\right)$ schemes [4, 7]. Although $\mathbf{X}_L(n)$ is a Toeplitz matrix, $\mathbf{R}_L(n)$ does not enjoy any particular structure. Introducing some further approximations in the

TR-SLIM-0 algorithm, a fast updating scheme is derived. The computational complexity of the proposed fast implementation is reduced to $\mathcal{O}\left(L^2\right)$.

The data matrix (3) can be partitioned as

$$\mathbf{X}_L(n) = \begin{bmatrix} \mathbf{X}_{L-1}(n-1) \\ \mathbf{x}_M^T(n) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{x}}_L(n) \\ \mathbf{X}_{L-1}(n) \end{bmatrix}, \qquad (12)$$

where $\tilde{\mathbf{x}}_L(n) \triangleq \mathbf{x}_M(n-L+1)$. Using (12), $\mathbf{R}_L(n)$ defined by (1) of Table 1, is partitioned as

$$\mathbf{R}_L(n) = \begin{bmatrix} \mathbf{R}_{L-1}^f & \mathbf{r}_{L-1}^f \\ \mathbf{r}_{L-1}^T & r^{fo} \end{bmatrix} \qquad (13)$$

where, $r^{fo} = \mathbf{x}_M^T(n)\mathbf{W}_M(n-1)\mathbf{x}_M^T(n) + \sigma_0^2$, and

$$\begin{aligned} \mathbf{R}_{L-1}^f &= \mathbf{X}_{L-1}(n-1)\mathbf{W}_M(n-1)\mathbf{X}_{L-1}^T(n-1) + \\ &\quad \sigma_0^2\mathbf{I}_{L-1} \qquad (14) \\ \mathbf{r}_{L-1}^f &= \mathbf{X}_{L-1}(n-1)\mathbf{W}_M(n-1)\mathbf{x}_M^T(n). \qquad (15) \end{aligned}$$

$\mathbf{R}_L(n-1)$, which is the covariance matrix at the previous time instant $(n-1)$, is partitioned using (12) as

$$\mathbf{R}_L(n-1) = \begin{bmatrix} \times & \times \\ \times & \mathbf{R}_{L-1}^b \end{bmatrix} \qquad (16)$$

where

$$\begin{aligned} \mathbf{R}_{L-1}^b &= \mathbf{X}_{L-1}(n-1)\mathbf{W}_M(n-2)\mathbf{X}_{L-1}^T(n-1) + \\ &\quad \sigma_0^2\mathbf{I}_{L-1}, \qquad (17) \end{aligned}$$

with $\times$ denoting terms of no interest. Clearly, (14) and (17) imply that $\mathbf{R}_{L-1}^f \neq \mathbf{R}_{L-1}^b$, as $\mathbf{W}_M(n-1) \neq \mathbf{W}_M(n-2)$, and thus further simplifications in the matrix manipulation is prohibited. We notice however that $\mathbf{R}_{L-1}^b$ and $\mathbf{R}_{L-1}^f$ can be related by a relatively simple expression, provided that at each time instant n, a small fraction of the diagonal elements in $\mathbf{W}_M(n-1)$ is only reloaded, while the rest are kept unchanged. The weight reloading strategy can be organized in a cyclic way [21, 22], where a fixed number of weights, say m, are only reloaded at each time instant, and where $\mathbf{W}_M(n-1)$ is replaced by the cyclically updated counterpart $\mathbf{W}_M^C(n-1)$. Using the cyclically reloaded weight matrix $\mathbf{W}_L^C(n-1)$, a low rank relationship between $\mathbf{R}_{L-1}^f$ and $\mathbf{R}_{L-1}^b$ is established as

$$\begin{aligned} \mathbf{R}_{L-1}^f &= \mathbf{R}_{L-1}^b + \\ &\quad \sum_{i=1}^{m} \Delta^{w(i)}\mathbf{x}_{L-1}^{k_i}(n-1)\mathbf{x}_{L-1}^{k_i T}(n-1) \qquad (18) \end{aligned}$$

where

$$\Delta^{w(i)} \triangleq w_{k_i}(n-1) - w_{k_i}(n-2),$$

with $\mathbf{x}_{L-1}^{k_i}(n-1)$ denoting the k_i-th column of $\mathbf{X}_{L-1}(n-1)$, $i = 1, 2 \ldots m$.

Suppose that at time instant $(n-1)$ the inverse $[\mathbf{R}_L(n-1)]^{-1}$ has been computed. Using (16) and the matrix inversion lemma for partitioned matrices ((4.160) in [1]), the inverse $[\mathbf{R}_{L-1}^b]^{-1}$ is computed as (see also [22, 23])

$$\begin{bmatrix} 0 & \mathbf{0}_{L-1}^T \\ \mathbf{0}_{L-1} & [\mathbf{R}_{L-1}^b]^{-1} \end{bmatrix} = [\mathbf{R}_L(n-1)]^{-1} - \mathbf{B}_L\mathbf{B}_L^T \qquad (19)$$

where $\mathbf{B}_L = [\mathbf{R}_L(n-1)]^{-1}\mathbf{e}_L^1 / \sqrt{\mathbf{e}_L^{1T}[\mathbf{R}_L(n-1)]^{-1}\mathbf{e}_L^1}$, with $\mathbf{e}_L^1$ denoting the first column of the identity matrix $\mathbf{I}_L$. $[\mathbf{R}_L^f]^{-1}$ is subsequently updated using (18) and the matrix inversion lemma for modified matrices ((4.148) in [1]). Finally, $[\mathbf{R}_L(n)]^{-1}$ is obtained using (13)-(15) and the matrix inversion lemma for partitioned matrices ((4.159) in [1]). The resulting algorithm is termed hereafter as the Cyclically weight reloaded TR-SLIM-0 (CTR-SLIM-0). The computational complexity of the proposed scheme is given by

$$(4+2m)(L^2+L) + 5M + 5\phi(M+L)$$

where m is the number of the cyclic steps applied at each time instant, with $\phi(x)$ denoting the computational cost of performing a Fast Fourier Transform (FFT) of size equal to x. The L^2 term results from the computations involved in the application of the matrix inversion lemmas in (13) and in (18), the evaluation of (19), as well as from the various matrix vector products. The term $\phi(M+L)$ results from the efficient implementation of the Toeplitz vector products involved in (2) of Table 1 and in (15), using the FFT.

4. SIMULATIONS

To illustrate the performance of the proposed algorithm, the identification of a time-varying sparse system with memory $M = 200$ and $S = 20$ non-zero coefficients is considered. Each non-zero component in (1) varies according to the first order auto-regressive model [24], as

$$c_i(n) = \alpha c_i(n-1) + \sqrt{1-|\alpha|^2}v_i(n)$$

where $\alpha = J_o(2\pi f_D T_s)$ with $J_o(.)$ denoting the zeroth-order Bessel function and $v_i(n)$ being a unit variance white noise signal. The normalized Doppler frequency is set equal to $f_D T_s = 0.005$, resulting in rapid time variation. The location of the active (non-zero) system coefficients is selected randomly, with an abrupt change occurring in the middle of the experiment, where the non-zero coefficients are relocated in new, randomly selected positions. The input signal $x(n)$ is a unit variance white noise signal. The additive disturbance is a white noise signal with variance $\sigma_0^2 = 0.1$. The mean squared deviation of the estimated system coefficients, defined as $\mathrm{MSD}(n) = \frac{\mathbb{E}\left[||\mathbf{c}_M(n) - \mathbf{c}_M^o(n)||_2^2\right]}{\mathbb{E}\left[||\mathbf{c}_M^o(n)||_2^2\right]}$ is used as a performance index, with $\mathbf{c}_M^o(n)$ denoting the true system coefficients, and where $\mathbb{E}[.]$ denotes the expectation operator, which

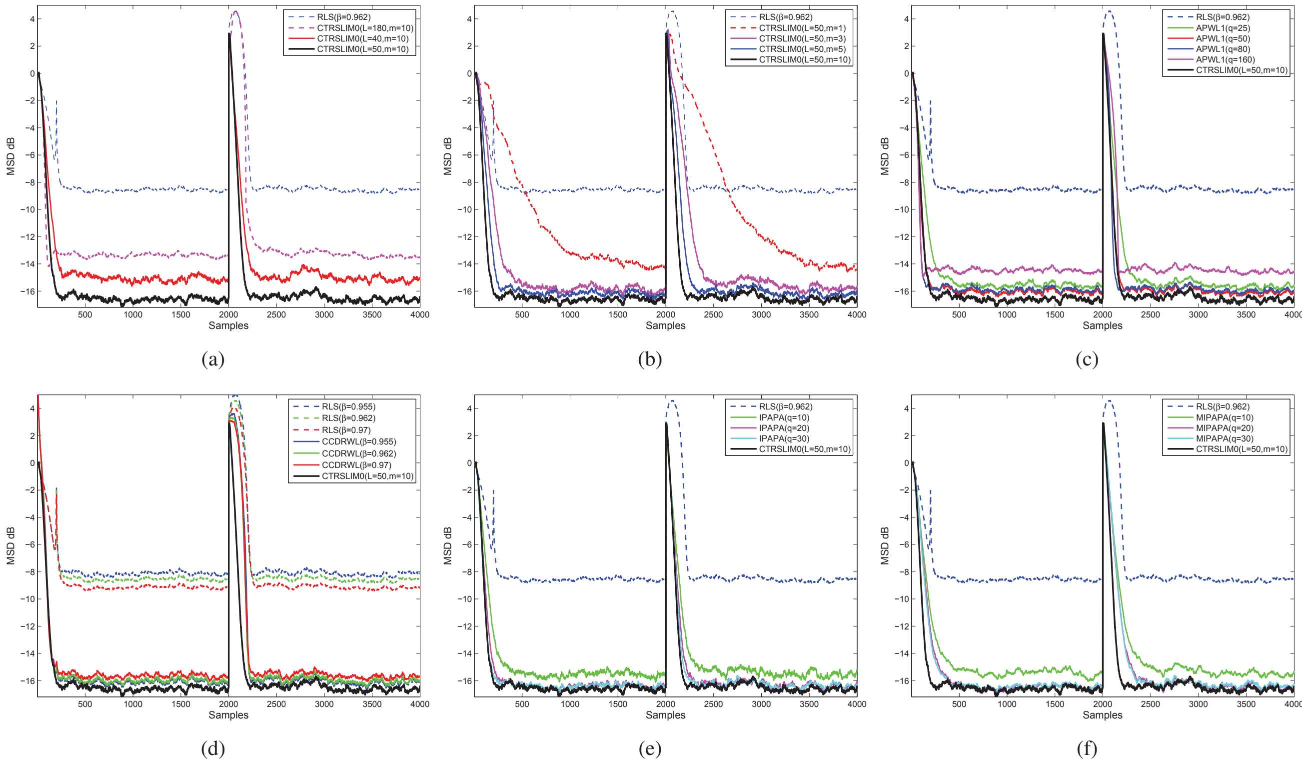

Fig. 1. (a) CTR-SLIM-0: MSD for different values of L when $m = 10$ (b) CTR-SLIM-0: MSD for different values of m when $L = 50$. (c) CTR-SLIM-0 against APWL1. (d) CTR-SLIM-0 against CCDRWL, (e) CTR-SLIM-0 against IPAPA, and (f) CTR-SLIM-0 against MIPAPA

is approximated by averaging over 100 independent experiments. The performance of the proposed CTR-SLIM-0 algorithm is illustrated in Figs. 1(a) and 1(b), for different values of L and m. The smoothing parameter in (11) is set equal to $\nu = 0.9$. The best performance is achieved when the sliding window data size is set equal to $L = 50$ and the number of the cyclically updated weights at each time instant is $m = 10$. Compared against the exponentially forgetting windowing Recursive LS (RLS) algorithm [2] with a forgetting factor set equal to $\beta = 0.962$, the proposed algorithm provides an improvement of about 8dB in the estimated MSD.

The performance of the APWL1 algorithm [7] for different values q of the used hyperslabs[1] is shown in Fig. 1(c). The choice of $q = 80$ results in the best performance, which is however outperformed by the proposed scheme, further noting that the APWL1 requires the knowledge of the true number of the non-zero elements in $\mathbf{c}_M(n)$, which is a rather strong assumption in the context of system identification.

The performance of the time recursive implementation of the adaptive LASSO algorithm using the cycled coordinate descent approach [4], implemented as it is detailed in Algorithm 2 (CCDRWL) of [8], using the exponentially forgetting RLS estimates as weights is illustrated in Fig. 1(d). The regularization parameter λ_n is computed as it is has been proposed in [7], with a parameter C tuned for the best performance at $C = 0.0005$. Clearly, the proposed algorithm exhibits better convergence and tracking characteristics than those obtained using the CCDRWL approach.

The performance of the IPAPA and the MIPAPA algorithm [25] is illustrated in Fig. 1(e) and 1(f), respectively, for various values of the projection order (here designated by q). The value of the parameter α is set equal to $\alpha = 0.9$ in both algorithms, while μ is set equal to $\mu = 1$ and to $\mu = 0.1$ for the IPAPA and the MIPAPA algorithms, respectively. When the IPAPA is considered, the choice of $q = 30$ results in an almost similar performance as that obtained by the proposed scheme, noting however that in this case the cost of implementing the IPAPA algorithm is about 3.5 times more than that of the CTR-SLIM-0 approach ($m = 10$). On the contrary, the MIMAPA implementation ($q = 30$) is about 2 times cheaper than that of the proposed scheme, at the expense of some degradation in the rate of convergence.

It is worth noting that the assumption that the noise variance $\sigma^2(n)$ is constant and is given, has been adopted solely for reasons of comparison with available schemes [4, 7, 8]. It

[1]The remaining parameters of the APWL1 algorithm are set as it is described in the Matlab implementation available by the authors [7] in http://cgi.di.uoa.gr/~stheodor/SPAL.zip

can be relaxed allowing for pure adaptive estimation of the noise variance. This is the subject of ongoing research and results will appear in forthcoming publications.

5. CONCLUSION

In this paper, a novel time recursive implementation of the SLIM algorithm is proposed in the context of adaptive system identification. A time recursive scheme is built up for the SLIM algorithm, able of updating the sought parameters on a sample by sample basis. A fast implementation is derived, based on cyclically reloading of the weight elements, offering a significant reduction in the required computational load. The proposed adaptive algorithm offers improved convergence speed and tracking ability compared to other existing schemes at an affordable computational cost. The performance of the proposed approach is illustrated by means of computer simulation, in the context of adaptive identification of rapidly time varying linear systems.

Acknowledgement

This work was supported through a grant (THALES PROTOMI) in the framework of the O.P. Education and Lifelong Learning from Community (ESF) and national funds, grant number MIS 377322.

6. REFERENCES

[1] N. Kalouptsidis, *Signal Processing Systems: Theory and Design*, Wiley-Interscience, 1997.

[2] A. H. Sayed, *Adaptive Filters*, Wiley-IEEE, 2008.

[3] Y. C. Eldar and G. Kutyniok, Eds., *Compressed Sensing: Theory and Applications*, Cambridge University Press, 2012.

[4] 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, 2010.

[5] B.Babadi, N. Kalouptsidis, and V.Tarokh, “Sparls: The sparse rls algorithm,” *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4013 4025, 2010.

[6] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, “Adaptive algorithms for sparse system identification,” *Signal Processing*, vol. 91, pp. 213–234, 2011.

[7] Y. Kopsinis, K. Slavakis, and S. Theodoridis, “Online sparse system identification and signal reconstruction using projections onto weighted ℓ_1 balls,” *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 905930, 2011.

[8] V. Kekatos and G. B. Giannakis, “Sparse Volterra and Polynomial Regression Models: Recoverability and Estimation,” *IEEE Trans. Signal Process.*, vol. 59, pp. 5907–5920, 2011.

[9] S. Adalbjornsson, G.O. Glentis, and A. Jakobsson, “Efficient block and time-recursive estimation of sparse volterra systems,” in *IEEE Statistical Signal Processing Workshop*, 2012, pp. 173 – 176.

[10] X. Tan, W. Roberts, J. Li, and P. Stoica, “Sparse Learning via Iterative Minimization With Application to MIMO Radar Imaging,” *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1088–1101, March 2011.

[11] D. Vu, L. Xu, M. Xue, and J. Li, “Nonparametric Missing Sample Spectral Analysis and Its Applications to Interrupted SAR,” *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 1, pp. 1–14, Feb. 2012.

[12] G.-O. Glentis, K. Zhao, A. Jakobsson, and J. Li, “Non-Parametric High-Resolution SAR Imaging,” *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1614–1624, April 2013.

[13] D. Vu D. Vu. Xue, X. Tan, and J. Li, “A bayesian approach to sar imaging,” *Digital Signal Processing*, vol. 23, no. 3, pp. 852–858, 2013.

[14] J. Ling, K. Zhao, J. Li, and M.L. Nordenvaad, “Multi-input multi-output underwater communications over sparse and frequency modulated acoustic channels,” *J. Acoustical Society of America*, vol. 129, no. 3, pp. 3640 – 3651, 2011.

[15] K. Zhao, J. Liang, J. Karlsson, and J. Li, “Enchanced multistatice active sonar signal processing,” *J. Acoustical Society of America*, vol. 134, no. 1, pp. 300–310, 2013.

[16] L. Xu, K. Zhao, J. Li, and P. Stoica, “Wideband sourse localization using sparse learing via iterative minimization,” *Signal Processing*, vol. 93, pp. 3504–3514, 2013.

[17] G.-O. Glentis and A. Jakobsson, “Time-Recursive IAA Spectral Estimation,” *IEEE Signal Process. Lett.*, vol. 18, no. 2, pp. 111–114, Feb. 2011.

[18] E. J. Candes, M. B. Wakin, and S. Boyd, “Enhancing Sparsity by Reweighted l1 Minimization,” *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, Dec. 2008.

[19] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado, “Subset Selection in Noise Based on Diversity Measure Minimization,” *IEEE Trans. Signal Process.*, pp. 760–770, 2003.

[20] G. H. Golub and C. F. Van Loan, *Matrix Comptutations*, Johns Hopkins University Press, Baltimore, MD., 1984.

[21] S. Gay, “Dynamically regularized fast rls with application to echo cancellation,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, vol. 2, pp. 957–960.

[22] H. Koeppl, G. Kubin, and G. Paoli, “Bayesian methods for sparse rls adaptive filters,” in *Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, 2003, vol. 2, pp. 1273–1277.

[23] J. Karlsson, W. Rowe, L. Xu, G. Glentis, and J. Li, “Fast Missing-Data IAA by Low Rank Completion,” in *The 38th International Conference on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 26-31 2013.

[24] E. Vlachos, A. Lalos, and K. Berberidis, “Stochastic gradient puruit for adaptive equalization of sparse multipath channels,” *IEEE J. Emenging and Selected Topics in Ciruits and Systems*, vol. 2, no. 3, pp. 413–423, 2012.

[25] C. Paleologu, S. Ciochina, and J. Benesty, “An efficient proportionate affine projection algorithm for echo cancellation,” *IEEE Signal Process. Lett.*, vol. 17, no. 2, pp. 165–168, Feb. 2010.