GENERALIZED ITERATIVE THRESHOLDING FOR SPARSITY-AWARE ONLINE VOLTERRA SYSTEM IDENTIFICATION

Konstantinos Slavakis[1] *Yannis Kopsinis*[2] *Sergios Theodoridis*[2]
Georgios B. Giannakis[1] *Vassilis Kekatos*[1]

[1]University of Minnesota
Digital Technology Center
Minneapolis, USA
Emails: slavakis@dtc.umn.edu
georgios@umn.edu, kekatos@umn.edu

[2]University of Athens
Dept. Informatics & Telecomms.
Athens, Greece
Emails: kopsinis@ieee.org
stheodor@di.uoa.gr

ABSTRACT

The present paper explores the link between thresholding, one of the key enablers in sparsity-promoting algorithms, and Volterra system identification in the context of time-adaptive or online learning. A connection is established between the recently developed generalized thresholding operator and optimization theory via the concept of proximal mappings which are associated with non-convex penalizing functions. Based on such a variational analytic ground, two iterative thresholding algorithms are provided for the sparsity-cognizant Volterra system identification task: (i) a set theoretic estimation one by using projections onto hyperslabs, and (ii) a Landweber-type one. Numerical experimentation is provided to validate the proposed algorithms with respect to state-of-the-art, sparsity-aware online learning techniques.

Index Terms— Volterra, thresholding, proximal mapping, sparsity, adaptive filtering.

1. INTRODUCTION

Memory-enabled nonlinear systems are frequently encountered in various areas of engineering such as biological processes [1], and digital communications [2–4]. Truncated Volterra or polynomial series are well-established approximating models for fitting smooth nonlinear systems [3, 5]. Nevertheless, the major difficulty in such models is the "curse of dimensionality", since the number of series coefficients grows exponentially with the polynomial system's memory [3]. Kernel-inspired approaches are a popular strategy to overcome this obstacle by using elegant reproducing properties of polynomial kernels [6]. It is often the case that a parsimonious underlying physical system, or an overparametrized assumed model render the Volterra series expansion sparse, in the sense that most of its coefficients are of zero or negligible size [1]. Unfortunately, the computationally efficient polynomial-kernel-based strategies [6] face difficulties in incorporating sparsity-related information. Motivated by the previous arguments, [7] introduced an ℓ_1-driven, sparsity-aware, RLS-based approach to Volterra or polynomial system identification in the context of time-adaptive or online learning, i.e., the scenario where training data arrive sequentially, they are utilized for only a limited number of times, and the unknown system may be time-variant.

Thresholding, the operation of nullifying small components of an $L \times 1$ vector $\boldsymbol{a}$ while shrinking or leaving intact the others, is one of the key enablers of sparsity-promoting algorithms [8, 9]. It is by now well-established that the discontinuous hard thresholding (HT) often results into high-variance estimates [10–12]. On the other hand, the continuous soft thresholding (ST) operator, associated with convex ℓ_1-penalty terms, has the tendency to increase bias [10–12]. To overcome these drawbacks, alternative thresholding rules have been proposed [11–15]. These advances in thresholding operators are strongly connected to optimization tasks; they are obtained by regularizing squared error losses by usually non-convex penalties. Based on HT, the generalized thresholding (GT) operator was introduced in [16] to encompass the majority of thresholding rules [11–15]. GT was combined with a set theoretic estimation algorithm in [16] to identify a sparse system in the context of online learning.

The contribution of this paper to sparsity-cognizant online Volterra system identification is twofold. First, it establishes a novel link between GT and optimization theory by the concept of proximal mappings, associated with non-convex penalties. To justify the potential of such a viewpoint, a novel iterative thresholding Landweber-type algorithm is introduced, whose complexity scales linearly with the Volterra filter length. Second, both GT-inspired algorithms of [16] and of the Landweber-type are applied to the sparsity-aware

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thalis - UoA - Secure Wireless Nonlinear Communications at the Physical Layer.

Volterra system identification task. Numerical experimentation is provided to validate these GT-empowered techniques over state-of-the-art RLS-based methodologies [7].

2. PROBLEM STATEMENT

Given the $m \times 1$ input signal $\boldsymbol{x} = [x_1, x_2, \ldots, x_m]^\top$ ($[\cdot]^\top$ denotes transposition), the noisy output $y \in \mathbb{R}$ of a discrete Volterra model is given by the following equation [3, 5]

$$y = \sum_{d=0}^{D} \sum_{|\boldsymbol{\beta}|=d} h_*^{\boldsymbol{\beta}} x_1^{\beta_1} x_2^{\beta_2} \cdots x_m^{\beta_m} + \eta, \tag{1}$$

where η denotes noise, $\boldsymbol{\beta} := (\beta_1, \beta_2, \ldots, \beta_m) \in \mathbb{N}^m$ is a multi-index with total degree $|\boldsymbol{\beta}| := \sum_{i=1}^m \beta_i$, and $\{h_*^{\boldsymbol{\beta}} : |\boldsymbol{\beta}| \leq D\}$ are the real-valued coefficients which characterize the unknown Volterra system. Model (1) obtains a dot-product form as follows; for each $d \in \{0, 1, \ldots, D\}$, define the map $\boldsymbol{x} \mapsto \phi_d(\boldsymbol{x})$, where $\phi_d(\boldsymbol{x})$ contains all lexicographically ordered monomials of total degree d:

$$\phi_d(\boldsymbol{x}) := [x_1^d, x_1^{d-1}x_2, \ldots, x_2^d, \ldots, x_m^d]^\top \in \mathbb{R}^{\binom{m+d-1}{d}}.$$

If $\boldsymbol{a}_*^d := [h_*^{\boldsymbol{\beta}_1}, h_*^{\boldsymbol{\beta}_2}, \ldots, h_*^{\boldsymbol{\beta}_{L_d}}]^\top \in \mathbb{R}^{L_d}$, $L_d := \binom{m+d-1}{d}$, is the vector which contains the coefficients $\{h_*^{\boldsymbol{\beta}_i}\}_{i=1}^{L_d}$, with $\{\boldsymbol{\beta}_i\}_{i=1}^{L_d}$ ordered according to the previous lexicographic ordering, and if

$$\boldsymbol{a}_* := [\boldsymbol{a}_*^{0\top}, \boldsymbol{a}_*^{1\top}, \ldots, \boldsymbol{a}_*^{D\top}]^\top \in \mathbb{R}^L$$
$$\phi(\boldsymbol{x}) := [\phi_0^\top(\boldsymbol{x}), \phi_1^\top(\boldsymbol{x}), \ldots, \phi_D^\top(\boldsymbol{x})]^\top \in \mathbb{R}^L,$$

where $L := \sum_{d=0}^D L_d$, then (1) can be recast as $y = \boldsymbol{a}_*^\top \phi(\boldsymbol{x}) + \eta = \boldsymbol{a}_*^\top \boldsymbol{u} + \eta$, where the definition $\boldsymbol{u} := \phi(\boldsymbol{x})$ was introduced for simplicity.

This paper focuses on identifying $\{h_*^{\boldsymbol{\beta}} : |\boldsymbol{\beta}| \leq D\}$ or equivalently $\boldsymbol{a}_*$ given a sequence of training data $(y_n, \boldsymbol{x}_n) \subset \mathbb{R} \times \mathbb{R}^m$, or better, $(y_n, \boldsymbol{u}_n) \subset \mathbb{R} \times \mathbb{R}^L$. The $\boldsymbol{a}_*$ is assumed to be sparse, i.e., most of its components are of zero or negligible size. Moreover, to abide by the time-adaptive or online learning premises, the joint pdf of $(y_n, \boldsymbol{u}_n)$, as well as the Volterra system $\boldsymbol{a}_*$ itself, are assumed to be unknown and in general time-varying.

3. FRAGMENTS OF OPTIMIZATION THEORY: THE PROXIMAL MAPPING

Definition 1 (Proximal mapping)**.** Given a positive definite $\boldsymbol{\Gamma} \in \mathbb{R}^{L\times L}$, and a function $f : \mathbb{R}^L \to (-\infty, +\infty]$, the proximal mapping $\mathrm{Prox}_{\boldsymbol{\Gamma},f}$ is defined as the set valued operator which maps to every $\boldsymbol{a} \in \mathbb{R}^L$, the following set:

$$\mathrm{Prox}_{\boldsymbol{\Gamma},f}(\boldsymbol{a}) := \arg\min_{\boldsymbol{z}\in\mathbb{R}^L} f(\boldsymbol{z}) + \frac{1}{2}(\boldsymbol{a}-\boldsymbol{z})^\top \boldsymbol{\Gamma}(\boldsymbol{a}-\boldsymbol{z}).$$

To avoid ambiguities, the previous set of minimizers is assumed nonempty. Whenever $\boldsymbol{\Gamma} = \frac{1}{\lambda}\boldsymbol{I}_N$, for some $\lambda > 0$, then the notation $\mathrm{Prox}_{\lambda f}$ is used instead of $\mathrm{Prox}_{\boldsymbol{\Gamma},f}$.

If f is (lower semicontinuous) convex, then $\mathrm{Prox}_{\lambda f}$ becomes single-valued, with eminent applicability to signal processing tasks [17, 18]. Moreover, in the special case of $f = \iota_C$, where ι_C denotes the indicator function of C, i.e., ι_C attains the value of 0 on the closed convex set C, and $+\infty$ elsewhere, then $\forall \lambda > 0$, $\mathrm{Prox}_{\lambda\iota_C}$ is nothing but the classical (metric) projection mapping P_C onto the closed convex C.

Motivated by the soft-thresholding-based approach of [19] for modeling inaccuracies and unknown noise statistics, a *hyperslab* is defined around each datum $(y_n, \boldsymbol{u}_n)$, for some user-defined $\epsilon_n \geq 0$:

$$S_n[\epsilon_n] := \left\{\boldsymbol{a} \in \mathbb{R}^L : \left|\boldsymbol{u}_n^\top \boldsymbol{a} - y_n\right| \leq \epsilon_n\right\}, \ \forall n. \tag{2}$$

It can be verified that $S_n[\epsilon_n]$ in (2) is a closed convex set, with projection mapping given as:

$$P_{S_n[\epsilon_n]}(\boldsymbol{a}) = \boldsymbol{a} + \begin{cases} \frac{y_n-\epsilon_n-\boldsymbol{u}_n^\top\boldsymbol{a}}{\|\boldsymbol{u}_n\|^2}\boldsymbol{u}_n, & \text{if } y_n - \epsilon_n > \boldsymbol{u}_n^\top\boldsymbol{a}, \\ 0, & \text{if } |\boldsymbol{u}_n^\top\boldsymbol{a} - y_n| \leq \epsilon_n, \\ \frac{y_n+\epsilon_n-\boldsymbol{u}_n^\top\boldsymbol{a}}{\|\boldsymbol{u}_n\|^2}\boldsymbol{u}_n, & \text{if } y_n + \epsilon_n < \boldsymbol{u}_n^\top\boldsymbol{a}. \end{cases}$$

4. PENALIZED LEAST-SQUARES

The mainstream of batch sparsity-promoting algorithms utilize a number of N training data, $(\boldsymbol{y}, \boldsymbol{U}) \in \mathbb{R}^N \times \mathbb{R}^{L\times N}$, to find an exact or approximate solution, in most cases iteratively, to the following *penalized least-squares* minimization task; find

$$\arg\min_{\boldsymbol{a}\in\mathbb{R}^L} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{U}^\top\boldsymbol{a}\|^2 + \lambda\sum_{i=1}^L p(a_i), \tag{3}$$

where $p : \mathbb{R} \to [0, \infty)$ stands for a sparsity-promoting, non-decreasing, even, and generally non-convex penalty, $\lambda \in (0, \infty)$ is the regularization parameter, and a_i stands for the i-th coordinate of the vector $\boldsymbol{a}$.

Choices for p are numerous; for example, if $p(a) = \chi_{\mathbb{R}\setminus\{0\}}(|a|)$, $\forall a \in \mathbb{R}$, where $\chi_{\mathscr{A}}$ stands for the characteristic function with respect to $\mathscr{A} \subset \mathbb{R}$, then the regularization term $\sum_{i=1}^L p(a_i)$ becomes the ℓ_0-norm of $\boldsymbol{a}$. In the case where $p(a) = |a|$, $\forall a \in \mathbb{R}$, then the regularization term is the ℓ_1-norm $\|\boldsymbol{a}\|_1 := \sum_{i=1}^L |a_i|$, and (3) is the LASSO [20]. It has been observed that if some of the LASSO's regularity conditions are violated, then LASSO is sub-optimal for model selection [12, 15, 21]. Such a behavior has motivated the search for non-convex penalty functions p, which bridge the gap between the ℓ_0- and ℓ_1-norm; for example, the ℓ_γ-penalty, for $\gamma \in (0, 1)$, [10], the log- [10, 12], the SCAD [10], the MC+ [12], and the transformed ℓ_1-penalties [10].

Recently, separable counterparts of (3) are recently attracting interest due to their simplicity and scalability to high-dimensional tasks [12]. A justification for this interest is the case where $\boldsymbol{U}$ is orthogonal. By $\tilde{\boldsymbol{a}} := \boldsymbol{U}\boldsymbol{y}$, (3) is equivalent to the following separable-in-components task [10]; find

$$\mathfrak{M}(\tilde{\boldsymbol{a}}) := \arg\min_{\boldsymbol{a}\in\mathbb{R}^L} \left(\sum_{i=1}^{L} \frac{1}{2\lambda} (\tilde{a}_i - a_i)^2 + p(a_i) \right). \quad (4)$$

The connection of (4) with the proximal mapping of Def. 1 is evident: $\mathfrak{M}(\tilde{\boldsymbol{a}}) = \times_{i=1}^{L} \operatorname{Prox}_{\lambda p}(\tilde{a}_i)$, where $\times$ stands for the Cartesian product. Under certain regularity conditions on p, $\mathfrak{M}(\tilde{\boldsymbol{a}})$ becomes a singleton [10]. The mapping which takes any $\tilde{\boldsymbol{a}}$ to a solution of (4) will be called penalized least-squares thresholding operator (PLSTO).

Figs. 1(b-d), depict PLSTOs associated with some of the most commonly employed penalty functions. Due to separability in (4), only one dimension is depicted in Fig. 1. For example, if $p(a) = \left[\nu^2 - (|a| - \nu)^2 \chi_{[0,\nu)}(|a|)\right]/\nu$, $\forall a \in \mathbb{R}$, and for some $\nu > 0$, then the PLSTO is the celebrated HT [10], which is depicted in Fig. 1a together with ST, which results in the case where $p(a) = |a|$. The rest of the thresholding rules in Fig. 1b correspond to MC+ [12, 15] and SCAD [10], respectively. HT is far from being the only discontinuous PLSTO. An example is shown in Fig. 1c, by bridge thresholding (BT) [13], which relates to the ℓ_γ-penalty, $\gamma < 1$. Continuous thresholding functions, with nonlinear parts, are shown in Fig. 1(d). More specifically, the non-negative garrote [14] and representatives of the n-degree garrote thresholding are illustrated.

5. GENERALIZED THRESHOLDING MAPPING

Definition 2 (The function $\boldsymbol{i}_K$)**.** For some user-defined positive integer $K < L$, let the set-valued mapping $\mathcal{I}_K : \mathbb{R}^L \rightrightarrows \mathfrak{T}_K$, where $\forall \boldsymbol{a} \in \mathbb{R}^L$, $\mathcal{I}_K(\boldsymbol{a})$ gathers all the K-tuples in $\mathfrak{T}_K$ which identify the K largest in magnitude values of $\boldsymbol{a}$. Among these, $\boldsymbol{i}_K(\boldsymbol{a})$ is defined to be the one with the smallest indices. For example, if $\boldsymbol{a} = [1, \frac{1}{2}, \frac{1}{4}, -1, -\frac{1}{2}]^\top$, and $K = 3$, then $\mathcal{I}_K = \{(1,2,4), (1,4,5)\}$, and $\boldsymbol{i}_K(\boldsymbol{x}) = (1,2,4)$.

Definition 3 (Generalized thresholding mapping)**.** Given a penalty function p, an integer $K < L$, and a $\lambda > 0$, the i-th entry of the generalized thresholding mapping $T_{\text{GT}}^{(K)} : \mathbb{R}^L \rightrightarrows \mathbb{R}^L$ is defined as follows; $\forall \boldsymbol{a} \in \mathbb{R}^L$:

$$T_{\text{GT}}^{(K)}(\boldsymbol{a})|_i := \begin{cases} a_i, & \text{if } i \in \boldsymbol{i}_K(\boldsymbol{a}), \\ z_i \in \operatorname{Prox}_{\lambda p}(a_i), & \text{otherwise.} \end{cases} \quad (5)$$

Notice that in general $\operatorname{Prox}_{\lambda p}(a_i)$ is a set, and z_i is any element within $\operatorname{Prox}_{\lambda p}(a_i)$. Moreover, notice that $T_{\text{GT}}^{(K)}$ leaves the K-largest in absolute value components intact, as opposed to the PLSTO in (4), where penalization by p is applied to all the components of the input vector. In addition, $\operatorname{Prox}_{\lambda p}(a_i)$ is not confined to be a singleton as is the case usually in (4) [10].

Proposition 1. (a) Given $p : \mathbb{R} \to [0, \infty)$, define the loss

$$\pi_K(\boldsymbol{a}) := \sum_{i \notin \boldsymbol{i}_K(\boldsymbol{a})} p(a_i), \qquad \forall \boldsymbol{a} \in \mathbb{R}^L.$$

Then, for any $\lambda > 0$, $T_{\text{GT}}^{(K)}(\boldsymbol{a}) = \operatorname{Prox}_{\lambda \pi_K}(\boldsymbol{a})$.
(b) Define the function

$$f(\boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{U}^\top \boldsymbol{a}\|^2 + \pi_K(\boldsymbol{a}), \quad \forall \boldsymbol{a} \in \mathbb{R}^L.$$

Choose any λ such that $0 < \lambda < 1/\lambda_{\max}(\boldsymbol{U}\boldsymbol{U}^\top)$, where $\lambda_{\max}(\cdot)$ stands for the largest eigenvalue of a matrix. Then, $\forall \boldsymbol{a} \in \mathbb{R}^L$,

$$T_{\text{GT}}^{(K)}\Big(\boldsymbol{a} - \lambda \boldsymbol{U}\big(\boldsymbol{U}^\top \boldsymbol{a} - \boldsymbol{y}\big)\Big) = \operatorname{Prox}_{\frac{1}{\lambda}\boldsymbol{I}_L - \boldsymbol{U}\boldsymbol{U}^\top, f}(\boldsymbol{a}).$$

Proof. Omitted due to space limitations. □

Remark 1. The generalized thresholding mapping $T_{\text{GT}}^{(K)}$ was introduced in [16] under the following form:

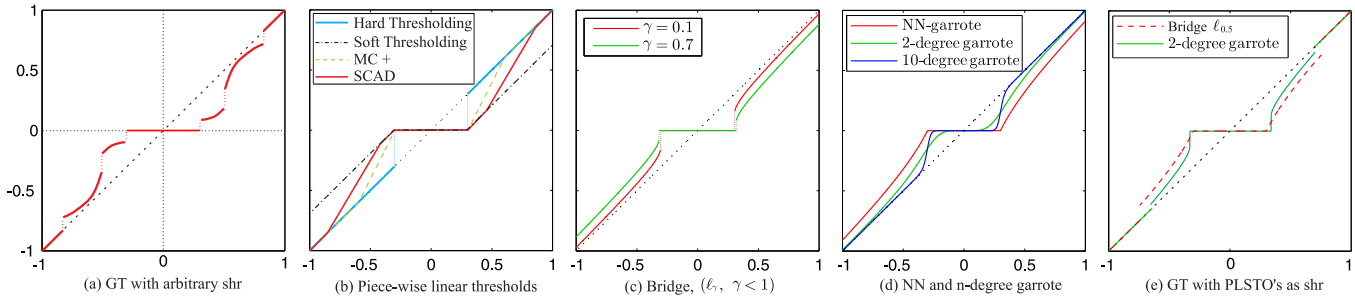
$$T_{\text{GT}}^{(K)}(\boldsymbol{a})|_i = \begin{cases} a_i, & \text{if } i \in \boldsymbol{i}_K(\boldsymbol{a}), \\ \operatorname{Shr}(a_i), & \text{otherwise,} \end{cases} \quad (6)$$

where Shr is a function such that (i) $\tau \operatorname{Shr}(\tau) \geq 0$, and (ii) $|\operatorname{Shr}(\tau)| \leq |\tau|$. An example of Shr is given in Fig. 1(a). A condition under which the definitions of (5) and (6) become equivalent is given in [10, Prop. 3.2]. More specifically, if apart from (i) and (ii), Shr is also monotonically increasing, and $\operatorname{Shr}(a) \to \infty$ as $a \to \infty$, then there exists a continuous, monotonically increasing p such that $\operatorname{Shr}(a_i) = \operatorname{Prox}_p(a_i)$ in (6) for every a_i at which Shr is continuous [10, Prop. 3.2]. A detailed study on other conditions under which the equivalence of (5) and (6) holds true, or even more interestingly, a study on the flexibility that (6) offers to thresholding techniques compared to the PLSTO of (4) is deferred to a future work.

6. ALGORITHMS

In this section, two algorithms for the sparsity-cognizant Volterra system identification task are given. The first one is based on the set theoretic estimation approach of [16], while the second one is motivated by Prop. 1.b.

Algorithm 1 (Adaptive projection-based generalized thresholding (APGT) algorithm)**.** For an arbitrary initial point, $\boldsymbol{a}_0 \in \mathbb{R}^L$, iterate the following procedure $\forall n \in \mathbb{N}$:
(**s1**) For a user-defined integer q, define the sliding window $\mathcal{W}_n := \overline{\max\{0, n-q+1\}, n}$ on the time axis, of size at most q, where $\overline{j_1, j_2}$ for two integers $j_1 \leq j_2$ stands for $\{j_1, j_1 + 1, \ldots, j_2\}$. The set $\mathcal{W}_n$ defines all the indices corresponding



(a) GT with arbitrary shr (b) Piece-wise linear thresholds (c) Bridge, $(\ell_\gamma,\ \gamma<1)$ (d) NN and n-degree garrote (e) GT with PLSTO's as shr

Fig. 1. Illustration of solutions to (4) for various choices of the penalty p, and some examples of GT.

to the hyperslabs to be processed at time instant n. Among these, identify the "active" hyperslabs $\mathcal{A}_n := \{i \in \mathcal{W}_n : P_{S_i[\epsilon_i]}(\boldsymbol{a}_n) \neq \boldsymbol{a}_n\}$, where $P_{S_i[\epsilon_i]}$ stands for the projection onto $S_i[\epsilon_i]$ (cf. Section 3). By Def. 1, $P_{S_i[\epsilon_i]} = \mathrm{Prox}_{\iota_{S_i[\epsilon_i]}} =: \mathrm{Prox}_i$, where $\iota_{S_i[\epsilon_i]}$ stands for the indicator function of $S_i[\epsilon_i]$. Moreover, for every $i \in \mathcal{A}_n$, define the weight $\omega_i^{(n)} > 0$, with $\sum_{i\in\mathcal{A}_n} \omega_i^{(n)} = 1$, to weigh the importance of the information carried by each hyperslab $S_i[\epsilon_i]$.

(**s2**) Choose any $\varepsilon' \in (0,1]$, and any $\mu_n \in [\varepsilon'\mathcal{M}_n, (2-\varepsilon')\mathcal{M}_n]$, where

$$\mathcal{M}_n := \begin{cases} \frac{\sum_{i\in\mathcal{A}_n} \omega_i^{(n)} \|\mathrm{Prox}_i(\boldsymbol{a}_n) - \boldsymbol{a}_n\|^2}{\|\sum_{i\in\mathcal{A}_n} \omega_i^{(n)} \mathrm{Prox}_i(\boldsymbol{a}_n) - \boldsymbol{a}_n\|^2}, \\ \qquad \text{if } \sum_{i\in\mathcal{A}_n} \omega_i^{(n)} \mathrm{Prox}_i(\boldsymbol{a}_n) \neq \boldsymbol{a}_n, \\ 1, \qquad \text{otherwise.} \end{cases}$$

Notice that due to convexity of $\|\cdot\|^2$, $\mathcal{M}_n \geq 1$. In general, the larger the μ_n, the larger the convergence speed of APGT.

(**s3**) Finally, the next estimate is given by

$$\boldsymbol{a}_{n+1} = \begin{cases} T_{\mathrm{GT}}^{(K)}\Big((1-\mu_n)\boldsymbol{a}_n + \mu_n \sum_{i\in\mathcal{A}_n} \omega_i^{(n)} \mathrm{Prox}_i(\boldsymbol{a}_n)\Big), \\ \qquad\qquad\qquad\qquad \text{if } \mathcal{A}_n \neq \emptyset, \\ T_{\mathrm{GT}}^{(K)}(\boldsymbol{a}_n), \qquad\qquad \text{if } \mathcal{A}_n = \emptyset. \end{cases}$$

Algorithm 2 (Adaptive generalized thresholding Landweber (AGTL) algorithm)**.** Given the sequence of training data $(y_n, \boldsymbol{u}_n)_{n\in\mathbb{N}}$, and the user-defined $q \in \mathbb{N}_*$, define the input signal matrix $\boldsymbol{U}_n = [\boldsymbol{u}_n, \boldsymbol{u}_{n-1}, \ldots, \boldsymbol{u}_{n-q+1}] \in \mathbb{R}^{L\times q}$ and the output signal vector $\boldsymbol{y}_n = [y_n, y_{n-1}, \ldots, y_{n-q+1}]^\top \in \mathbb{R}^q$. For an arbitrary initial point $\boldsymbol{a}_0$, generate the following sequence of estimates

$$\boldsymbol{a}_{n+1} = T_{\mathrm{GT}}^{(K)}\Big(\boldsymbol{a}_n - \lambda_n \boldsymbol{U}_n(\boldsymbol{U}_n^\top \boldsymbol{a}_n - \boldsymbol{y}_n)\Big),$$

where λ_n is any user-defined parameter such that $0 < \lambda_n < 1/\lambda_{\max}\big(\boldsymbol{U}_n\boldsymbol{U}_n^\top\big)$, and $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue of a matrix.

It can be verified that the computational complexities of both Algs. 1 and 2 are of order $\mathcal{O}(qL)$. This does not include the computational complexity associated with $T_{\mathrm{GT}}^{(K)}$. In Section 7, all of the employed penalizing functions p were chosen such that $T_{\mathrm{GT}}^{(K)}$ is given in closed form. The theoretical analysis of Alg. 1 can be found in [22], while the analysis and variants of Alg. 2 are deferred to a future work.

7. NUMERICAL EXAMPLES

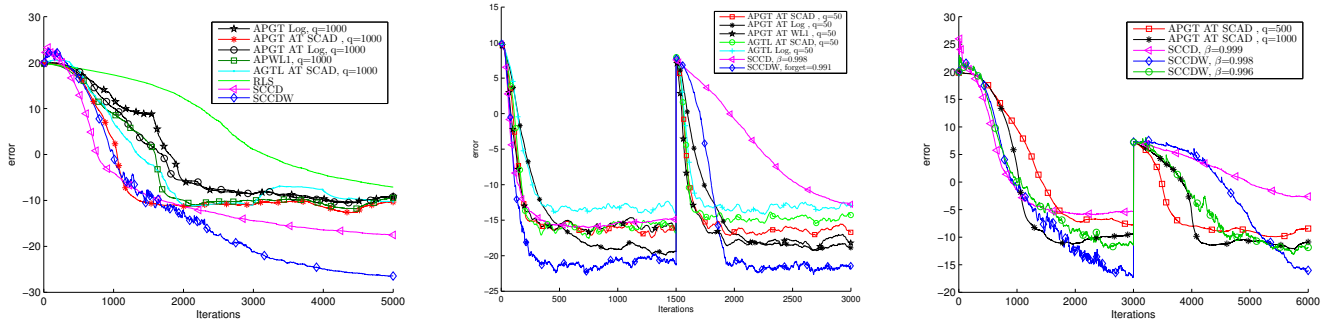
The APGT and AGTL algorithms of Section 6 are validated over the ℓ_1-penalized, RLS-driven SCCD and SCCDW methods of [7], whose complexities are of $\mathcal{O}(L^2)$. Moreover, the recently developed "APWL1" [19] is also employed, where projections onto weighted ℓ_1-balls are realized with an overall complexity of $\mathcal{O}\big((q+1)L\big)$. In all of the following examples, noise η is drawn from the class $\mathcal{N}(0, 0.1)$. In Fig. 2, vertical axes depict $\mathrm{error}_n := 10\log_{10}\sum_{r=1}^{R}\|\boldsymbol{a}_n^{(r)} - \boldsymbol{a}_*^{(r)}\|^2/R$, where $R = 10$ is the total number of realizations. In what follows, ϵ_n of APGT takes the value $\epsilon_n = 1.3\sigma_\eta^2, \forall n$.

7.1. Time-invariant case: $\boldsymbol{a}_*$ stays fixed with time

In Fig. 2a, $(m, D, L) = (13, 4, 2380)$. For each realization, 100 randomly selected components of $\boldsymbol{a}_*$ take values from $\mathcal{N}(0,1)$, while the rest are set to zero. The "APGT Log" utilizes the log-penalty of [12] ($\gamma = 10$ and $\lambda = 0.008$), whereas "APSM AT SCAD" uses the SCAD penalty, but with an adaptive thresholding (AT) strategy which renders the associated λ parameter time-adaptive ($\alpha = 12$; cf. [22]). The same AT strategy for λ is followed also in "APGT AT SCAD" and "AGTL AT SCAD". For both APGT and AGTL, $q = 10^3$ and $K = 100$. "APGT AT SCAD" shows the best performance among all employed APGT and AGTL methods, and similar convergence speed to the SCCDW of [7]. Both SCCD and SCCDW outperform all other methods in terms of steady-state error performance.

7.2. Time-varying case: $\boldsymbol{a}_*$ changes with time

Two cases are considered here; Fig. 2b corresponds to $(m, D, L) = (20, 2, 231)$, whereas Fig. 2c to the longer Volterra filter case of $(m, D, L) = (13, 4, 2380)$. For both



(a) (b) (c)

Fig. 2. (a) The unknown system $\boldsymbol{a}_*$ stays fixed with time for $(m, D, L) = (13, 4, 2380)$. (b) Time-varying $\boldsymbol{a}_*$ for $(m, D, L) = (20, 2, 231)$. (c) Time-varying $\boldsymbol{a}_*$ for $(m, D, L) = (13, 4, 2380)$.

cases, there is a time instance where $\boldsymbol{a}_*$ abruptly changes. In Fig. 2b, and for the first 1500 time instances, 10 components of $\boldsymbol{a}_*$ are drawn from $\mathcal{N}(0, 1)$, while the rest are set to zero. At the time instance 1501, 5 randomly chosen nonzero components of $\boldsymbol{a}_*$ are set to zero, while 2 randomly chosen, previously zero-valued components, take values from $\mathcal{N}(0, 1)$. For APGT and AGTL, $K = 10$ throughout the experiment. In Fig. 2c, and for the first 3000 time instances, $\boldsymbol{a}_*$ is the same as in the time-invariant case of Fig. 2a. At time 3001, 5 randomly chosen nonzero components of $\boldsymbol{a}_*$ are set to zero, while 2 randomly chosen, previously zero-valued components, take values from $\mathcal{N}(0, 1)$. For APGT and AGTL, $K = 100$ throughout the experiment.

To track the Volterra filter's changes, an RLS-inspired forgetting factor is necessary for both SCCD and SCCDW to disregard past observed values. Extensive experimentation on the values of the forgetting factor produced several indicative curves in Figs. 2b and 2c. Notice that the longer the Volterra filter is, the more distinct becomes the superior tracking ability of APGT compared to the rest of the methods. Variants of AGTL which achieve faster convergence speed and lower steady-state errors are deferred to a future work.

8. REFERENCES

[1] D. Song, H. Wang, and T. W. Berger, "Estimating sparse Volterra models using group ℓ_1-regularization," in *Proc. IEEE Intl. Conf. Engnr. in Medicine and Biology Society (EMBC)*, Buenos Aires, Argentina, Sep. 2010, pp. 4128–4131.

[2] S. Benedetto and E. Biglieri, "Nonlinear equalization of digital satellite channels," *IEEE J. Sel. Areas Commun.*, vol. SAC-1, no. 1, pp. 57–62, Jan. 1983.

[3] V. Mathews and G. Sicuranza, *Polynomial Signal Processing*, Wiley, New York, 2000.

[4] G. B. Giannakis and E. Serpedin, "A bibliography on nonlinear system identification," *Signal Process.*, vol. 81, no. 3, pp. 533–580, Mar. 2001.

[5] G. Palm and T. Poggio, "The Volterra representation and the Wiener expansion: Validity and pitfalls," *SIAM J. Appl. Math.*, vol. 33, no. 2, pp. 195–216, 1977.

[6] M. O. Franz and B. Schölkopf, "A unifying view of Wiener and Volterra theory and polynomial kernel regression," *Neural Comput.*, vol. 18, no. 12, pp. 3097–3118, 2006.

[7] V. Kekatos and G. B. Giannakis, "Sparse Volterra and polynomial regression models: Recoverability and estimation," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 5907–5920, Dec. 2011.

[8] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied Comput. Harmonic Anal.*, vol. 27, no. 3, pp. 265–274, 2009.

[9] S. Foucart, "Hard thresholding pursuit: An algorithm for compressive sensing," *SIAM J. Numer. Anal.*, vol. 49, no. 6, pp. 2543–2563, 2011.

[10] A. Antoniadis, "Wavelet methods in statistics: Some recent developments and their applications," *Statist. Surveys*, vol. 1, pp. 16–55, 2007.

[11] Y. She, "Thresholding-based iterative selection procedures for model selection and shrinkage," *Electr. J. Statist.*, vol. 3, pp. 384–415, 2009.

[12] R. Mazumder, J. H. Friedman, and T. Hastie, "SPARSENET: Coordinate descent with nonconvex penalties," *J. Amer. Statist. Assoc.*, vol. 106, no. 495, pp. 1125–1138, Sept. 2011.

[13] I. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–135, 1993.

[14] H.-Y. Gao, "Wavelet shrinkage denoising using the non-negative garrote," *J. Comput. Graph. Statist.*, vol. 7, no. 4, pp. 469–488, Dec. 1998.

[15] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *Annals Statist.*, vol. 38, no. 6, pp. 894–942, 2010.

[16] Y. Kopsinis, K. Slavakis, S. Theodoridis, and S. McLaughlin, "Generalized thresholding sparsity-aware algorithm for low complexity online learning," in *Proc. ICASSP*, Kyoto: Japan, 2012, pp. 3277–3280.

[17] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer-Verlag, 2011.

[18] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2011.

[19] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted ℓ_1-balls," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 936–952, Mar. 2011.

[20] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. Royal. Statist. Soc. B.*, vol. 58, no. 1, pp. 267–288, 1996.

[21] H. Zou, "The adaptive LASSO and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 101, pp. 1418–1429, Dec. 2006.

[22] K. Slavakis, Y. Kopsinis, S. Theodoridis, and S. McLaughlin, "Generalized thresholding and online sparsity-aware learning in a union of subspaces," arXiv: `http://arxiv.org/abs/1112.0665`.