Nonlinear Signal Processing and its Applications to Telecommunications

Gerasimos Mileounis*

Department of Informatics and Telecommunications
National and Kapodistrian University of Athens
TYPA Buildings, University Campus, 15784 Athens, Greece
gmil@di.uoa.gr

Abstract. This dissertation is primarily concerned with the estimation of nonlinear communication systems that are modeled by Volterra series. The major methods used for estimating the unknown channel parameters can be classified into two main categories: training-based and blind. First, orthobasis representation and training-based identification through the respective Fourier series are investigated for most modulated signals of interest. Next, higher order cumulants are used for the blind identification of nonlinear channels. The proposed algorithms for blind nonlinear channel estimation take advantage of the inherent sparseness of the higher order cumulants of common communication signals. Then, sparse Volterra channels are employed to mitigate the enormous computational complexity of the full Volterra channels. Sparse Volterra channels are approached by two newly developed sparse adaptive (greedy and ℓ_1-regularized) algorithms. Last, the problem of blind sparse channel estimation is formulated by modifying the Expectation-Maximization framework to accommodate channel sparsity.

Keywords: Volterra sereis, Higher-Order-Statistics, Adaptive filters, Blind identification, Nonlinear compressed channel sensing.

1 Introduction

Nonlinear behavior is observed in almost all digital communication systems including satellite, telephone channels, mobile cellular communications, wireless LAN devices, radio and TV stations, digital magnetic systems and so forth. In those cases, possible remedies based on linear approximations degrade system performance. Significant benefits in the performance of a digital communication system are expected when appropriate nonlinear models, methods and algorithms are developed, taking into account nonlinear effects.

Nonlinear systems have been systematically studied in the past [1], but they have not been widely used in communications due to their computational complexity. Computational complexity depends on several factors, including: (1) dimension of the unknown parameter vector, (2) sparseness characteristics of the parameter vector, (3) degree of nonlinearity, (4) number of available measurements, and (5) the probability density function of the input. The motivation

* Dissertation Advisor: Prof. Nicholas Kalouptsidis.

of this dissertation is to develop methods and algorithms that considerably reduce the computational complexity and increase the performance of existing algorithms for nonlinear channel estimation, by using the above factors in a beneficial manner.

We study nonlinearities in communication systems using polynomial filters, a special class of which is Volterra series. More precisely, we propose new estimation techniques and apply them to linear and nonlinear communication channels. Our research efforts focus on two areas: (1) nonlinear channel estimation using higher order statistics, and (2) adaptive and blind algorithms for sparse channel estimation.

Initially, we develop training based algorithms for the identification of (passband and baseband) Volterra channels modulated by QAM, PSK and OFDM inputs [2]. When the Volterra channel is excited by QAM or PSK inputs, multivariate orthogonal polynomials are used to estimate the unknown parameters. Closed form expressions are established for baseband Volterra channels driven by i.i.d complex Gaussian (OFDM) signals.

Blind methods identify the unknown channel merely based on the received signal, without consuming any of the available channel capacity. However, blind nonlinear channel estimation is a hard problem and the development of blind methods remains at a very preliminary stage dealing with special model subclasses. We investigate sparseness of the higher-order output cumulants in order to simplify the blind identification problem of two different nonlinear models: (1) passband and baseband Hammerstein channels excited by common communication signals [3], and (2) linear-quadratic Volterra with complex random inputs [4].

Volterra models employ a large number of parameters, to adequately represent many real-world systems, which increases exponentially with the order of nonlinearity and memory length. For this reason, their applicability is limited to weak nonlinearities, e.g. only up to third order. Therefore, there is a strong need to decrease the parameter space by only considering those parameters that actually contribute to the output. This observation lead us to the exploitation of sparse Volterra models which constitute a major component of this dissertation.

The use of adaptive filtering is crucial in applications like communications where channel measurements arrive sequentially and in many cases the channel response is time-varying. All adaptive algorithms in the literature for nonlinear channel estimation treat each parameter equally and identify the complete set of parameters. The major drawback of estimating the complete set of parameters is the large computational/implementation cost. In this dissertation, we investigate the performance gains that can be achieved if insignificant parameters are ignored. Two Novel adaptive algorithms are developed that recursively update the parameters of interest. The first adaptive algorithm combines the Expectation-Maximization and Kalman filtering [5], whereas the second one relies on greedy methods [6]. Finally, using the Expectation-Maximization framework, we address the problem of blind identification of sparse linear and nonlinear channels [7].

This summary is organized as follows. Firstly, Chapter 2 establishes the necessary background needed in the sequential chapters. Section 3 deals with training-based methods for the identification of Volterra channels. Sections 4 and 5 tackle

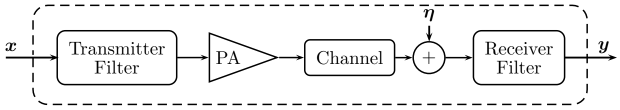

Fig. 1. Nonlinear communication system

the problem of blind identification in Hammerstein channels and second order Volterra systems, respectively, by making use of Higher-Order output Statistics. Two different adaptive training-based algorithms for the estimation of sparse channels are presented in Sections 6 and 7. Section 8 proposes a blind identification algorithm for the estimation of sparse channels. Finally, Section 9 presents the overall conclusion of this summary.

2 Background

Modern high-speed communication systems are frequently operated over nonlinear channels with memory. Most transmitters are equipped with Power Amplifiers (PAs) operating close to saturation to achieve power efficiency [8]. To properly analyze a communication system, like the one of Fig. 1, the nonlinear effects caused by the presence of PAs must be combined with the transmitting, receiving and channel filters.

One of the most popular models that are applied for the description of nonlinear phenomena are Volterra series [1] and allows us to capture these combined effects. Causal discrete Volterra series of finite-order have the following form (also referred to as *passband Volterra*):

$$y(n) = \sum_{p=1}^{P} \sum_{\tau_1=0}^{M_p} \cdots \sum_{\tau_p=0}^{M_p} h_p(\tau_1, \ldots, \tau_p) \left[\prod_{i=1}^{p} x(n-\tau_i) \right]. \tag{1}$$

where $x(n)$ and $y(n)$ are the system input and output respectively. The function $h_p(\tau_1, \ldots, \tau_p)$ is called the *pth-order Volterra kernel* of the system. P is the highest order of nonlinearity while M_p is the pth-order system memory, note that the Volterra model of Eq. (1) becomes linear when $P = 1$.

In many communication systems the signal bandwidth is very carefully defined depending on the application. The receiver filter is used to eliminate signal components outside the desired bandwidth. Therefore the output signal only contains spectral components near the carrier frequency ω_c. This leads to the *baseband Volterra* system [8, Ch. 14], given by

$$y(n) = \sum_{p=0}^{\lfloor \frac{P-1}{2} \rfloor} \sum_{\tau_1=0}^{M_{2p+1}} \cdots \sum_{\tau_{2p+1}=0}^{M_{2p+1}} h_{2p+1}(\tau_1, \ldots, \tau_{2p+1}) \prod_{i=1}^{p+1} x(n-\tau_i) \prod_{j=p+2}^{2p+1} x^*(n-\tau_j) \tag{2}$$

where $\lfloor \cdot \rfloor$ denote the floor operation. The above representation only considers odd-order powers with one more unconjugated input than conjugated input.

This way the output does not create spectral components outside the frequency band of interest.

The key feature of Volterra series is that the nonlinearity is due to multiple products of delayed input values, while the kernel coefficients appear linearly in the output. This allows us to rewrite them as a linear regression model using Kronecker products. Indeed, consider the passband case of Eq. (1) and let $\boldsymbol{x}_{M_1}^{(1)}(n) = [x(n), x(n-1), \cdots, x(n-M_1)]^T$ and the pth-order Kronecker power $\boldsymbol{x}_{M_p}^{(p)}(n) = \boldsymbol{x}_{M_1}^{\otimes p}(n)$. The Kronecker power contains all pth-order products of the input. Likewise $\boldsymbol{h} = \left[\boldsymbol{h}_1(\cdot), \cdots, \boldsymbol{h}_p(\cdot)\right]^T$ is obtained by treating the p-dimensional kernel as a $(M_p)^p$ column vector. We now rewrite the output of Eq. (1) as follows

$$y(n) = \left[\boldsymbol{x}_{M_1}^T(n) \cdots \boldsymbol{x}_{M_p}^{(p)T}(n)\right] \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_p \end{bmatrix} = \boldsymbol{x}^T(n)\boldsymbol{h}. \tag{3}$$

Because of this property, linear estimation techniques can be exploited in the identification of Volterra coefficients.

3 Nonlinear system identification using othogonal bases and cumulants

In this section baseband and passband nonlinear channels featuring PSK, QAM and OFDM modulation are considered. Channel estimation is performed using multivariate orthogonal polynomials of complex variables.

An i.i.d complex valued signal is orthogonalizable [9] and there are various ways to construct associated orthogonal bases. A common construction relies on one dimensional orthogonal base and its separable extension to higher dimensions. One dimensional polynomials constructed by the Gram-Schmidt procedure is a notable case. Multidimensional orthogonal polynomials are formed as products of one dimensional orthogonal polynomials $(P_k(x_i)$, where k is the degree of the polynomial and $x_i \equiv x(n-i))$ [9]. For the monomials (in one variable) $\left\{x_n^i\right\}_{i=0}^p$ associated with passband Volterra models, we introduce a degree ordering $1 < x_n < \ldots < x_n^p$. For monomials in two variables $(x_n x_n^*)$ related to baseband models we apply the graded lexicographic ordering [2].

By definition the multivariate orthogonal polynomials, $Q_{\boldsymbol{i}_{1:p}}^{(p)}(\boldsymbol{x}(n))$ of degree p, are orthogonal to all lower orders and to the same order. The passage to the original Volterra kernels from the orthogonal coefficients is effected by the following expression

$$\begin{aligned} \mathbb{E}[y(n)P_{\tau_1}^*(x_{i_1}) \cdots P_{\tau_k}^*(x_{i_k})] &= \pi(\boldsymbol{i}_{1:k})h_k(i_1,\ldots,i_k)\|P_{\tau_1}(x_{i_1})\|_{\ell_2}^2 \cdots \|P_{\tau_k}(x_{i_k})\|_{\ell_2}^2 \\ &+ \sum_{v=1}^{\lfloor\frac{P-k}{2}\rfloor} E\left\{\mathcal{H}_{k+2v}(\underline{z_n})Q_{i_1\ldots i_k}^{(*p)}(\boldsymbol{x}(n))\right\} \end{aligned}$$

where $\mathcal{H}_v(.)$ is the homogeneous term of order v. Similarly for the baseband case. The identification process starts by estimating the highest order kernel which has no contribution from other kernels and moving downwards.

If the input is approximately complex white Gaussian (OFDM case) the relevant orthogonal polynomials are the Hermite polynomials. The method described above is applicable. Alternately cumulant operators can be used. Expressions invoking cumulants are much simpler because cumulants are equivalent to multiples of Hermite moments.

Theorem 1 *Consider the baseband Volterra model* (2)*. The cross-cumulant of* $y(n)$ *with* $(p+1)$ *conjugated copies of the input and* p *unconjugated copies of the input is given by*

$$c_{y,\boldsymbol{x}^*}{}^{(p+1)}_{(p)}(\boldsymbol{\tau}_{1:p+1}, \boldsymbol{\tau}_{p+2:2p+1}) = p!(p+1)!\gamma_{1,1}^{2p+1} h_{2p+1}(\boldsymbol{\tau}_{1:p+1}, \boldsymbol{\tau}_{p+2:2p+1})$$
$$+ \sum_{v=1}^{\lfloor \frac{P-2p-1}{2} \rfloor} \frac{(p+1+v)!(p+v)!}{v!} \gamma_{1,1}^{2p+1+v} \sum_{k_1} \cdots \sum_{k_v} h_{2p+1+2v}(\boldsymbol{\tau}_{1:p+1}, \boldsymbol{k}_{1:v}, \boldsymbol{\tau}_{p+2:2p+1}, \boldsymbol{k}_{1:v})$$

$\gamma_{1,1}$ *represents the variance of* $x(n)$ *and* $\boldsymbol{k}_{1:v} = (k_1, \ldots, k_v)$*.*

Detailed proof of the above theorem is given in [2]. The algorithm identifies the highest order kernel first. Then the lower order kernels are identified recursively using the previous estimated kernels and cross-cumulants information.

4 Blind identification of Hammerstein channels

This section considers the blind identification problem of Hammerstein channels. The Hammerstein model corresponds to a diagonal Volterra model, since all off-diagonal elements are zero.

Baseband: The proposed method starts by expressing the baseband Hammerstein model (similarly for passband) as a linear multichannel system of the form

$$y(n) = \sum_{i=0}^{q_1} \mathbf{b}(i)\mathbf{w}(n-i) + \eta(n)$$

where

$$\mathbf{w}(n) = \begin{pmatrix} x(n) & \cdots & |x(n)|^{2p}x(n) \end{pmatrix}^T, \quad \mathbf{b}(i) = \begin{pmatrix} h_1(i) \; h_3(i) & \cdots & h_{2p+1}(i) \end{pmatrix}$$

and ^{T} denotes matrix transpose. We assume that the linear kernel has the largest memory $q_1 > q_{2l+1} \; \forall \; 1 \leq l \leq p$. Moreover, we find it convenient to impose the following normalization $h_1(q_1) = 1$.

The output cumulant of order $(k+l)$, with k unconjugate and l conjugate output lags, is given by:

$$c_y{}^{(l)}_{(k)}(\tau_1, \ldots, \tau_{k+l-1}) = \sum_{i=0}^{q_1} \mathbf{b}(i) \otimes \mathbf{b}(i+\tau_1) \otimes \cdots \otimes \mathbf{b}^*(i+\tau_{k+l-1}) \, \boldsymbol{\Gamma}_{\mathbf{w}}{}^{(l)}_{(k)},$$

where ${\mathbf{\Gamma_w}}^{(l)}_{(k)}$ is the input intensity vector (zero lag cumulant) of order $k+l$ of $\mathbf{w}(n)$. Then, parameter estimation relies on the following equation and the solution of a system of linear equations

$$\tilde{c}_{y}{}^{(l)}_{(k)}(q_1,\tau) = c_{y}{}^{(l)}_{(k)}(q_1,\ldots,q_1,\tau) = \mathbf{b}^*(\tau)\left(\overline{\mathbf{\Gamma}}_{\mathbf{w}}{}^{(l)}_{(k)}\right)_{s\times s}\mathbf{b}^T(0), \quad s=p+1.$$

For PSK inputs we always consider cumulants with an equal number of conjugate/unconjugate copies of the output. Whereas, for QAM inputs we may employ output cumulants with unequal number of unconjugate/conjugate entries of the output. In this manner we reduce significantly the order of the output cumulants.

Passband: Prakriya et al. [10] have proved that if the order of nonlinearity is p, then the only non-zero multilinear function of $c_{y}{}^{(1)}_{(p)}(\tau_1,\ldots,\tau_p)$ will be the one which includes the linear part p times and the pth-order term one time. Based on this remark, we can estimate the linear and the pth-order kernel as follows:

$$\begin{aligned} h_1(\tau) &= \tilde{c}_{y}{}^{(1)}_{(p)}(\tau,q_p)/\,\tilde{c}_{y}{}^{(1)}_{(p)}(q_1,q_p) \\ h_p(\tau) &= \tilde{c}_{y}{}^{(1)}_{(p)}(q_1,\tau)/\big(h_1(0)\,\mathrm{cum}\{\underbrace{x(n),\ldots,x(n)}_{p\text{ copies of }x(n)},x^{*p}(n)\}\big). \end{aligned}$$

So far we have estimated the first and last kernel of the passband Hammerstein channel by combining the techniques in [11, 10]. The kernels sandwiched between the linear kernel and the pth-order term can be obtained through the following recursion.

Theorem 2 *Consider a passband Hammerstein model. For $2 \le k \le p$, the following equation holds:*

$$\tilde{c}_{y}{}^{(1)}_{(k)}(q_1,\tau) = \sum_{\mu=0}^{p-k}\mathrm{cum}\{\underbrace{x^{1+\mu}(n),x(n),\ldots,x(n)}_{k\ copies\ of\ x(n)},x^{*(k+\mu)}(n)\}h_{1+\mu}(0)h_{k+\mu}(\tau).$$

The proof is supplied in [3]. Theorem 2 is based on the fact that the linear kernel and the pth-order kernel are identified first then the kernel of order $k=p-1$ is calculated. This process is iterated until $k=2$. The above technique is applicable to Hammerstein channels excited by PSK inputs of arbitrary order. However when the channel is excited by QAM inputs, the procedure is limited to quadratic Hammerstein channels.

5 Blind identification of second order Volterra systems with complex random inputs

In this section blind identification methods for second order Volterra systems excited by complex valued random variables are developed. The proposed blind identification method relies on output cumulants of order up to 4. The computation of these cumulants and the resulting expressions are provided in [4]. If

Table 1. Algorithms for blind Volterra identification

Algorithm 1 $(q_1 > q_2)$	**Algorithm 2** $(q_1 = q_2 = q)$
Require : $h_1(0) = 1$	**Require :** $h_1(0) = 1,\ h_2(0,0) = 0$
1: $\gamma_{4,0} = \dfrac{c_{y(4)}^{(0)}(q_1,0,0)^2}{c_{y(4)}^{(0)}(q_1,q_1,0)}$	1: $\gamma_{4,0} = \dfrac{c_{y(4)}^{(0)}(q,q,0)^3}{c_{y(4)}^{(0)}(q,q,q)^2}$
2: $\gamma_{1,1} = \dfrac{c_{y(1)}^{(1)}(q_1)\, c_{y(4)}^{(0)}(q_1,0,0)}{c_{y(4)}^{(0)}(q_1,q_1,0)}$	2: $\gamma_{1,1} = \dfrac{c_{y(1)}^{(1)}(q)\, c_{y(4)}^{(0)}(q,q,0)}{c_{y(4)}^{(0)}(q,q,q)}$
3: $h_1(\tau) = \dfrac{c_{y(4)}^{(0)}(q_1,\tau,0)}{c_{y(4)}^{(0)}(q_1,0,0)}$, $c_{y(3)}^{(0)}(q_1,q_1) = h_2(0,0)\gamma_{4,0}h_1^2(q_1)$	3: $h_1(\tau) = \dfrac{c_{y(4)}^{(0)}(q,q,\tau)}{c_{y(4)}^{(0)}(q,q,0)}, h_2(q,q) = \dfrac{c_{y(3)}^{(0)}(q,q)}{2\gamma_{4,0}h_1(q)}$
4: $h_2(\tau,\tau) = \dfrac{c_{y(3)}^{(0)}(q_1,\tau) - \gamma_{4,0}h_2(0,0)h_1(q_1)h_1(\tau)}{\gamma_{4,0}h_1(q_1)}$	4: $h_2(\tau,\tau) = \dfrac{c_{y(3)}^{(0)}(q,\tau) - \gamma_{4,0}h_2(q,q)h_1(\tau)}{\gamma_{4,0}h_1(q)}$
5: for $h_2(\tau_1,\tau_2)$ use Eq. below with $q = q_1$	5: for $h_2(\tau_1,\tau_2)$ use Eq. below

$$h_2^*(\tau_1,\tau_2) = \frac{1}{2\gamma_{1,1}^2 h_1^2(q)}\left[c_{y(2)}^{(1)}(q-\tau_1, q-\tau_2) - \sum_{i=0}^{\tau_1 - \lfloor \frac{j}{\tau_2} \rfloor} \sum_{j=0}^{\tau_2} \widetilde{h}_2(i,j) h_1(i+q-\tau_1) h_1(j+q-\tau_2) \right]$$

these expressions are evaluated at suitably chosen lags, sparse equations with respect to the Volterra kernels result. To proceed with the Volterra kernel identification algorithms, we distinguish two different cases: $q_1 > q_2$ and $q_1 = q_2 = q$. Appropriate normalization constraints are imposed for each case.

The proposed methods involve four steps carried out in the following sequence:

1. Compute the linear kernel using fourth order cumulants and a q-slice formula
2. Compute the input intensities $\gamma_{4,0}$ and $\gamma_{1,1}$
3. Compute the diagonal elements of the second order kernel using third order cumulants and a q-slice formula
4. Compute the off-diagonal elements of the second order kernel using third order cumulants and linear system solvers.

The above four steps are implemented by the algorithms of Table 1. Note that all relevant expressions are exact and hence the Volterra system is uniquely identifiable.

6 Sparse adaptive ℓ_1-regularized algorithm

In this section, we propose a family of sparse adaptive ℓ_1-regularized algorithms that can be used for sparse parameter estimation. The derived family of sparse adaptive algorithms is based on the Expectation Maximization (EM) framework. Let us start by considering a model that captures the dynamics of the unknown parameter vector $\boldsymbol{h}(n)$ (at time n). A popular technique in the adaptive filtering

Table 2. EM-KALMAN filter for sparse adaptive tracking

Algorithm description
Initialization : $\boldsymbol{h}_0 = \bar{\boldsymbol{h}}_0,\ \boldsymbol{P}_0 = \delta^{-1}\boldsymbol{I}$ with δ =const.
For $n := 1, 2, \ldots$ **do**
1: $\boldsymbol{k}(n) = \dfrac{\boldsymbol{P}(n-1)\boldsymbol{x}^*(n)}{\sigma_\eta^2 + \boldsymbol{x}^T(n)\boldsymbol{P}(n-1)\boldsymbol{x}^*(n)}$
2: $\boldsymbol{\psi}(n) = \boldsymbol{h}(n-1) + \boldsymbol{k}(n)\varepsilon(n)$
3: $\boldsymbol{P}(n) = \boldsymbol{P}(n-1) + r_n\boldsymbol{I} - \boldsymbol{k}(n)\boldsymbol{x}^T(n)\boldsymbol{P}(n-1)$
4: $\boldsymbol{h}(n) = \mathrm{sgn}\left(\boldsymbol{\psi}(n)\right)\Big[\lvert\boldsymbol{\psi}(n)\rvert - \gamma(\sigma^2_{\boldsymbol{\psi}_{n-1}} + r_n)\boldsymbol{I}\Big]_+$
end For

literature is to describe parameter dynamics by the first-order model [12]

$$\boldsymbol{h}(n) = \boldsymbol{h}(n-1) + \boldsymbol{q}_{|\Lambda_0}(n) = \boldsymbol{h}_0 + \sum_{i=1}^{n} \boldsymbol{q}_{|\Lambda_0}(i); \quad \boldsymbol{h}_0 \sim \mathcal{N}(\overline{\boldsymbol{h}}_0, \sigma_0^2 \boldsymbol{I}_{|\Lambda_0}) \tag{4}$$

where Λ_0 denotes the true support set of $\boldsymbol{h}_0$, i.e. the set of the non-zero coefficients. The noise term $\boldsymbol{q}(n)$ is zero outside $_{|\Lambda_0}$ and zero-mean Gaussian inside $_{|\Lambda_0}$ with diagonal covariance matrix $\boldsymbol{R}_{|\Lambda_0}(n) = \mathrm{diag}\left(\sigma^2_{q_1}(n), \ldots, \sigma^2_{q_d}(n)\right)$, where d is the ℓ_0-norm of $\boldsymbol{h}_0$. The variances $\{\sigma^2_{q_i}(n)\}_{i=1}^d$ are in general allowed to vary with time. The stochastic processes $\boldsymbol{\eta}(n)$, $\boldsymbol{q}(n)$ and the random variable $\boldsymbol{h}_0$ are mutually independent.

To apply the Expectation-Maximization method we have to specify the complete and incomplete data. The vector $\boldsymbol{h}(n)$ at time n is taken to represent the complete data vector, whereas $\boldsymbol{y}(n-1)$ accounts for the incomplete data [13]. In this context the conditional density $p(\boldsymbol{h}(n)|\boldsymbol{y}(n-1))$ plays a major role. This density is Gaussian with mean $\boldsymbol{\psi}(n) = \mathbb{E}[\boldsymbol{h}(n)|\boldsymbol{y}(n-1)]$. Under broad conditions the maximizer of the incomplete likelihood is obtained by maximizing the complete likelihood function through successive application of the following two steps:

E-step : computes the conditional expectation

$$Q\big(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(n-1)\big) = \mathbb{E}_{p(\boldsymbol{h}(n)|\boldsymbol{y}(n-1);\widehat{\boldsymbol{\theta}}(n-1))}\left[\log p(\boldsymbol{h}(n);\boldsymbol{\theta})\right]$$

M-step : maximizes the Q-function minus the ℓ_1-penalty with respect to $\boldsymbol{\theta}$

$$\widehat{\boldsymbol{\theta}}(n) = \arg\max_{\boldsymbol{\theta}} \left\{ Q\big(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(n-1)\big) - \gamma \|\boldsymbol{\theta}\|_{\ell_1} \right\}$$

Note that $p(\boldsymbol{h}(n);\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\psi}_n(\boldsymbol{\theta}), \boldsymbol{\Sigma}(n))$ and hence the Q-function takes the form

$$Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}_{n-1}) = \text{const.} + \boldsymbol{\theta}\boldsymbol{\Sigma}^{-1}(n)\boldsymbol{\psi}(n) - \frac{1}{2}\boldsymbol{\theta}^H \boldsymbol{\Sigma}^{-1}(n)\boldsymbol{\theta} \tag{5}$$

where the constant incorporates all terms that do no involve $\boldsymbol{\theta}$ and hence do not affect the maximization.

The parameter $\boldsymbol{\psi}(n)$ is recursively computed by the Kalman filter [12], see Table 2 steps $1-3$, which in the special case of the time-varying random walk model Eq. (4) takes an RLS type appearance. Note that $\varepsilon(n)$, in Table 2, denotes the prediction error given by $\varepsilon(n) = y(n) - \boldsymbol{x}^T(n)\boldsymbol{h}(n-1)$.

Maximization of the Q function leads to the *soft thresholding* function, see Table 2 step 4. This operation shrinks coefficients above the threshold in magnitude value. The complete algorithm is presented in Table 2.

7 Sparse Adaptive Orthogonal Matching Pursuit algorithm

This section converts a powerful greedy scheme developed in [14] into an adaptive algorithm. Greedy algorithms form an essential tool for sparse parameter estimation. However, their inherent batch mode discourages their use in time-varying environments due to significant complexity and storage requirements.

The proposed algorithm relies on three modifications to the CoSaMP structure [14]: the proxy identification, estimation, and error residual update. The error residual is now evaluated by

$$v(n) = y(n) - \boldsymbol{x}^T(n)\boldsymbol{h}(n). \tag{6}$$

The above formula involves the current sample only, in contrast to the CoSaMP scheme which requires all the previous samples. Eq. (6) requires s complex multiplications, whereas the cost of the sample update in the CoSaMP is sn multiplications. A new proxy signal that is more suitable for the adaptive mode, can defined as:

$$\boldsymbol{p}(n) = \sum_{i=1}^{n-1} \boldsymbol{x}^*(i)v(i) \tag{7}$$

and is updated by $\boldsymbol{p}(n) = \boldsymbol{p}(n-1) + \boldsymbol{x}^*(n-1)v(n-1)$. The last modification attacks the estimation step. The vector $\boldsymbol{h}(n)$ is updated by standard adaptive algorithms such as the LMS and RLS.

LMS is one of the most widely used algorithm in adaptive filtering due to its simplicity, robustness and low complexity. Hence, for reasons of simplicity and complexity we focus on the LMS algorithm. At each iteration the current regressor $\boldsymbol{h}(n)$ and the previous estimate $\boldsymbol{w}(n-1)$ are restricted to the instantaneous support originated from the support merging step. The resulting algorithm is presented in Table 3, where $\boldsymbol{h}_{|\Lambda}$ and $\boldsymbol{w}_{|\Lambda}$ denote the sub-vectors corresponding to the index set Λ, $\max(|a|, s)$ returns s indices of the largest elements of a and Λ^c represents the complement of set Λ. The following Theorem establishes the steady state Mean Square Error (MSE) error performance of the SpAdOMP algorithm:

Table 3. SpAdOMP Algorithm

Algorithm description		Complexity
$\boldsymbol{h}(0) = 0, \boldsymbol{w}(0) = 0, \boldsymbol{p}(0) = 0$	{Initiliazation}	
$v(0) = y(0)$	{Initial residual}	
$0 < \lambda \leq 1$	{Forgetting factor}	
$0 < \mu < 2\lambda_{\max}^{-1}$	{Step size}	
For $n := 1, 2, \ldots$ **do**		
1: $\boldsymbol{p}(n) = \lambda \boldsymbol{p}(n-1) + \boldsymbol{x}^*(n-1)v(n-1)$	{Form signal proxy}	M
2: $\Omega = \mathrm{supp}(\boldsymbol{p}_{2s}(n))$	{Identify large components}	M
3: $\Lambda = \Omega \cup \mathrm{supp}(\boldsymbol{h}(n-1))$	{Merge supports}	s
4: $\varepsilon(n) = y(n) - \boldsymbol{x}_{\|\Lambda}^T(n)\boldsymbol{w}_{\|\Lambda}(n-1)$	{Prediction error}	s
5: $\boldsymbol{w}_{\|\Lambda}(n) = \boldsymbol{w}_{\|\Lambda}(n-1) + \mu \boldsymbol{x}_{\|\Lambda}^*(n)\varepsilon(n)$	{LMS iteration}	s
6: $\Lambda_s = \max(\|\boldsymbol{w}_{\|\Lambda}(n)\|, s)$	{Obtain the pruned support}	s
7: $\boldsymbol{h}_{\|\Lambda_s}(n) = \boldsymbol{w}_{\|\Lambda_s}(n), \quad \boldsymbol{h}_{\|\Lambda_s^c}(n) = \mathbf{0}$	{Prune the LMS estimates}	
8: $v(n) = y(n) - \boldsymbol{x}^T(n)\boldsymbol{h}(n)$	{Update error residual}	s
end For		$\mathcal{O}(M)$

Theorem 3 *(SpAdOMP)*[1]*. The proposed algorithm, for large n, produces an s-sparse approximation $\boldsymbol{h}(n)$ that satisfies the following steady-state error bound*

$$\|\boldsymbol{h} - \boldsymbol{h}(n)\|_{\ell_2} \lesssim C_1(n)\|\boldsymbol{\eta}(n)\|_{\ell_2} + C_2(n)\|\boldsymbol{x}_{|\Lambda}(n)\|_{\ell_2}|e_o(n)|,$$

where $e_o(n)$ is the estimation error of the optimum Wiener filter and $C_1(n)$, $C_2(n)$ are constants independent of $\boldsymbol{h}$ (given explicitly in [6]) and are only functions of the restricted isometry constants, λ_{min} (the minimum eigenvalue of the input covariance matrix) and the step-size μ.

8 Blind Identification of sparse channels via the EM algorithm

The purpose of this section is to develop a blind identification algorithm for the estimation of sparse channels, under the assumption that the transmitted symbols are i.i.d. and take values in a finite alphabet set. A batch algorithm for blind channel estimation is reported in [15] using the iterative nature of the Expectation Maximization (EM) algorithm. We propose exploitation of the sparse nature of the channel by regularizing the cost function of the blind EM algorithm via the use of the ℓ_1 norm constraint.

In blind identification, the EM algorithm can be used to iteratively maximize $\log p(\boldsymbol{y}(n); \boldsymbol{\theta})$ (where $\boldsymbol{\theta} = \bar{\boldsymbol{h}}$)), without explicitly computing it. To use the EM algorithm, we consider the observations $\boldsymbol{y}(n)$ as the incomplete data and

[1] Proof is omitted due to space limitations.

Table 4. The Sparse BW Algorithm

Algorithm description	
$\alpha_1(i) = \pi_i b_i(\boldsymbol{y}_1), i := 1, \ldots, M^L, \quad \beta_N(i) = 1, i := 1, \ldots, M^L, \quad \sigma_\eta^2 = 1$	{Initiliazation}
For $\ell := 0, 1, \ldots,$ **do**	
1: $\alpha_{n+1}(j) = \sum_{i=1}^{M^L} \alpha_n(i) p_{ij} b_j(\boldsymbol{y}_{n+1}), \quad n := 1, \ldots, N-1, \quad j := 1, \ldots, M^L$	{Forward Recursion}
2: $\beta_n(i) = \sum_{j=1}^{M^L} \beta_{n+1}(j) p_{ij} b_j(\boldsymbol{y}_{n+1}), \quad n := N-1, \ldots, 1, \quad i := 1, \ldots, M^L$	{Backward Recursion}
3: $\gamma_n(i\|\boldsymbol{\theta}^{(\ell)}) = \dfrac{\alpha_n(i)\beta_n(i)}{\sum_{j=1}^{M^L} \alpha_n(j)\beta_n(j)}, \quad n := 1, \ldots, N, \quad i := 1, \ldots, M^L$	{Posterior probabilities}
4: $\boldsymbol{h}_i^{(\ell+1)} = \dfrac{\operatorname{sgn}(\boldsymbol{r}_i^{(\ell)})}{\boldsymbol{R}_{i,i}^{(\ell)}} \left[\|\boldsymbol{r}_i^{(\ell)}\| - \tau\right]_+$	{Channel estimation}
5: $\sigma_\eta^{(\ell+1)2} = (N+1)^{-1} \sum_{n=1}^{N} \|y_n - \widehat{\boldsymbol{x}}_n^{(\ell)T} \boldsymbol{h}^{(\ell)}\|^2$	{Noise Variance est.}
end For	

$(\boldsymbol{y}(n), \boldsymbol{X}(n))$ as the complete data. The EM algorithm is a two-step iterative procedure which under mild conditions converges to a local maximum. The proposed variant of the EM algorithm for blind sparse channel estimation iterates between the following two steps until convergence is reached:

1) **E-step:** Compute $Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}^{(\ell)}) = \mathbb{E}\big\{ \log p(\boldsymbol{y}(n), \boldsymbol{X}(n)|\boldsymbol{\theta})|\boldsymbol{y}(n); \widehat{\boldsymbol{\theta}}^{(\ell)} \big\}$
2) **M-step:** Solve $\widehat{\boldsymbol{\theta}}^{(\ell+1)} = \arg\max_{\boldsymbol{\theta}} \{Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}^{(\ell)}) - 2\tau\|\boldsymbol{\theta}\|_{\ell_1}\}.$

The E-step is a symbol detector and is carried out by the forward-backward recursions of Table 4 steps 1-2. Maximization of the penalized Q-function with respect to $\boldsymbol{h}$ at the M-step, has a closed form expression to each component of $\boldsymbol{h}^{(\ell+1)}$ and is given by the *soft-thresholding* function, see Table 4 step 4. The method outlined above is summarized in Table 4.

9 Conclusions

This dissertation aimed to develop new methods and algorithms for the estimation of nonlinear communications systems that are modeled by Volterra series. The estimation is achieved either with the help of a training signal or by blind identification methods. Moreover, the estimation performance depends on the pattern of channel (sparse or dense). Thus, the developed methods take into account the pattern of the channel, and simply estimate those parameters that actually contribute to the output

References

1. M. Schetzen, The Volterra and Wiener Theories of Nonlinear Systems, Wiley, 1980.
2. G. Mileounis, P. Koukoulas, N. Kalouptsidis, Input-output identification of nonlinear channels using PSK, QAM and OFDM inputs, Signal Process. (Elsevier) 89 (2009) 1359–1369.
3. G. Mileounis, N. Kalouptsidis, P. Koukoulas, Blind identification of Hammerstein channels using QAM, PSK and OFDM input using higher order cumulants, IEEE Trans. Commun. 57 (12) (2009) 3652–3661.
4. G. Mileounis, N. Kalouptsidis, Blind identification of second order Volterra systems with complex random inputs using higher order cumulants, IEEE Trans. Signal Process. 57 (10) (2009) 4129–4135.
5. N. Kalouptsidis, G. Mileounis, B. Babadi, V. Tarokh, Adaptive algorithms for sparse system identification, accepted in Signal Process. (Elsevier), subject to revisions.
6. G. Mileounis, B. Babadi, N. Kalouptsidis, V. Tarokh, An adaptive greedy algorithm with application to nonlinear communications, IEEE Trans. Signal Process. 58 (6) (2010) 2998–3007.
7. G. Mileounis, N. Kalouptsidis, B. Babadi, V. Tarokh, Blind identification of sparse channels and symbol detection via the EM algorithm, submitted in IEEE Signal Process. Lett.
8. S. Benedetto, E. Biglieri, Principles of Digital Transmission: with wireless applications, Springer, 1999.
9. S. Yasui, Stochastic functional fourier series, Volterra series, and nonlinear system analysis, IEEE Trans. Autom. Control 24 (2) (1979) 230–242.
10. S. Prakriya, D. Hatzinakos, Blind identification of LTI-ZMNL-LTI nonlinear channel models, IEEE Trans. Signal Process. 43 (12) (1995) 3007–3013.
11. N. Kalouptsidis, P. Koukoulas, Blind identification of Volterra-Hammerstein systems, IEEE Trans. Signal Process. 53 (8) (2005) 2777–2787.
12. L. Ljung, General structure of adaptive algorithms: Adaptation and tracking, in Adaptive System Identification and Signal Processing Algorithms, Eds. N. Kalouptsidis and S. Theodoridis, 1993.
13. M. Feder, Statistical signal processing using a class of iterative estimation algorithms, Ph.D. thesis, M.I.T., Cambridge, MA (1987).
14. D. Needell, J. Tropp, CoSaMP: Iterative signal recovery from incomplete and inaccurate samples, Appl. Comput. Harmon. Anal 26 (2009) 301–321.
15. G. Kaleh, R. Vallet, Joint parameter estimation and symbol detection for linear or nonlinear unknown channels, IEEE Trans. Commun. 42 (7) (1994) 2606–2413.