Steady-State Analysis of the Maximum Correntropy Volterra Filter with Application to Nonlinear Channel Equalization

Lu Lu*†, Haiquan Zhao* and Benoit Champagne†

* School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China
Email: lulu19900303@gmail.com, hqzhao_swjtu@126.com

† Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada
Email: benoit.champagne@mcgill.ca

Abstract—As a well-established adaptation criterion, the maximum correntropy criterion (MCC) has received increased attention due to its robustness against outliers. In this paper, a new complex maximum correntropy criterion Volterra filter (Volterra-CMCC) that does not need any a priori information about the noise statistical characteristics, is proposed based on the recursive scheme. We study the steady-state excess mean-square-error (EMSE) behavior of the Volterra-CMCC algorithm by using the energy conservation relation and Taylor series expansion approach. Then, the proposed algorithm is applied to the nonlinear channel equalization problem, where the channel is contaminated by impulsive noise. The results obtained from simulation study establish the effectiveness of this new Voltera-CMC equalizer.

I. Introduction

Over the past decade, several improvements based on the l_2-norm have been developed to enhance the performance of the recursive least square (RLS) algorithm [1], [2]. Nevertheless, the l_2-norm type algorithms may fail to work when the second-order moments of additive noise are infinite [3]. This problem has been extensively investigated for the case of physical channels, such as underwater acoustic communication channel, VHF radio, and portable radio [4]–[6]. These studies have shown that the additive noise in these scenarios can be well modeled by the α-stable distribution [4]. The α-stable distribution has no finite second-order or higher order moments. Therefore, the performance of the RLS-based algorithms deteriorate in the presence of α-stable noise.

For high data rate communication systems, the communication quality can suffer severely from intersymbol interference (ISI), nonlinear distortions, and noise [7]. To compensate for distortions and achieve reliable signal transmission (i.e., low bit error rate (BER)), equalizers have been widely applied in communication systems. In recent years, various nonlinear equalization techniques have been reported [8]–[10]. In [9] and [10], a kernel least mean square (LMS) algorithm based on Wirtingers calculus and an augmented complex kernel LMS based on widely linear estimation were proposed for the nonlinear channel equalization problem. Note that the above-mentioned kernel methods can be regarded as a generalization of minimum mean square error (MSE) criterion. For non-

Gaussian signals, the use of the MSE criterion is sub-optimal as it does not take into account the effects of higher-order moments.

Recently, the maximum correntropy criterion (MCC) has received considerable attention in the signal processing literature [11]. The MCC provides a measure of local similarity between two random variables and as such, shares a close relationship with M-estimation [12]. Due to its simplicity and low sensitivity to outlier noise samples, it has been widely used recently as an underlying criterion in the design of robust adaptive filtering schemes for various applications [13], [14].

In this paper, a new complex-valued recursive Volterra filter, called Volterra-CMCC, is proposed by recursively minimizing the MCC function. Moreover, the steady-state performance of the Volterra-CMCC algorithm is analyzed. The proposed algorithm is useful for nonlinear channel equalization, especially when the signals contain large outliers or are contaminated by impulsive noises. The improved performance of Volterra-CMCC algorithm is confirmed in the context of nonlinear channel equalization with α-stable noise.
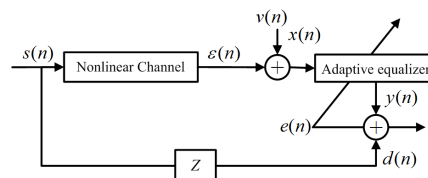
II. Problem Formulation

Fig. 1. Block diagram of an equalizer in digital transmission system.

Fig. 1 shows a discrete-time model of a communication system, where transmission occurs over a nonlinear channel and an adaptive equalizer is employed ar the receiver. The error signal of adaptive equalizer can be defined as

$$
\begin{aligned}
e(n) &= d(n) - y(n) \\
&= s(n - Z) - y(n)
\end{aligned}
\tag{1}
$$

where $s(n)$ is the transmitted signal (i.e., known training sequence), $d(n) = s(n - Z)$ is the desired or reference signal

driving the adaptation, Z is the transmission delay related to the communication channel, and $y(n)$ is the equalizer output.

Nonlinear distortions considered in the channel during the process of transmission are labeled as 'Nonlinear channel'. The nonlinear function is:

$$\varepsilon(n) = \xi\{s(n)\} \qquad (2)$$

where $\varepsilon(n)$ denotes the channel output.

The input to the equalizer (corrupted signal) is defined as

$$x(n) = \varepsilon(n) + v(n) \qquad (3)$$

where $v(n)$ is the additive noise.

Considering the computational cost of the Volterra filter, the second-order Volterra (SOV) expansion [15] is used in this paper. The expanded input vector $\boldsymbol{x}(n)$ of the SOV filter is expressed as

$$\boldsymbol{x}(n) = [\underbrace{x(n), x(n-1), ..., x(n-1+M)}_{linear\ kernels},$$
$$\underbrace{x^2(n), x(n)x(n-1), ..., x^2(n-1+M)}_{quadratic\ kernels}]^T \qquad (4)$$

where M denotes the length of the linear kernel. The length of the second-order Volterra filter L can be calculated by $L = M(M+3)/2$. Thus input signal $\boldsymbol{x}(n)$ is passed into the adaptive equalizer, which reduces the effect of channel and recovers transmitted signals according to the received signal samples. One important challenge comes from channel equalization of impulsive noise. These impulsive noise processes can be modeled as α-stable noise process which has no closed form probability density function (PDF) expression. The characteristic function of symmetric α-stable ($S\alpha S$) distribution, which is the Fourier transform of the PDF, can be described by [4]

$$\varphi(t) = \exp\{-\gamma|t|^\alpha\} \qquad (5)$$

where $0 < \alpha < 2$ denotes the characteristic exponent, and γ denotes the *dispersion* parameter whose behavior is similar to the variance of the Gaussian distribution. The small α indicates the highly impulsive noise. Specially, if $\alpha = 2$, (5) becomes the characteristic function of the Gaussian distribution.

III. Derivation of the Volterra-CMCC algorithm

Let D and Y be two random variables with the same dimensions, the measure of correntropy is defined as [12]

$$V(D,Y) = E\{\kappa(D,Y)\} = \int \kappa(d,y)\mathrm{d}\Re_{D,Y}(d,y) \qquad (6)$$

where $\kappa(\cdot,\cdot)$ denotes a *shift-invariant Mercer kernel*, and $\Re_{D,Y}(d,y)$ stands for the joint distribution function of (d,y). In this work, the kernel function in correntropy is chosen as the following Gaussian kernel

$$\kappa(D,Y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{|D-Y|^2}{2\sigma^2}\right\} \qquad (7)$$

where σ stands for the kernel size. The cost function of the MCC-based algorithms can be defined as follows [12]

$$J(n) = E\left\{\exp\left(-\frac{|e(n)|^2}{2\sigma^2}\right)\right\} \qquad (8)$$

where $E\{\cdot\}$ stands for expectation. The MCC is a measure of the similarity between two random variables, whose value is mainly decided by the kernel function. Fig. 2 illustrates the MCC cost function $\kappa(d,y)$ in the joint space of d and y. For Gaussian kernel, the MCC divides space in three different regions. When two points are close, the MCC behaves like an l_2-norm distance in the Euclidean region. When two points are getting apart, the MCC behaves like an l_1-norm and eventually like an l_0-norm when two points are further apart [12]. This property demonstrates robust performances of correntropy for outliers. To achieve improved performance, the RLS-based algorithms are usually preferred. Therefore, the cost function of Volterra-CMCC can be defined as follows

$$\begin{aligned} \Upsilon(n) &= \sum_{i=1}^{n} \lambda^{n-i}\kappa(d(i),y(i,n)) \\ &= \frac{1}{\sqrt{2\pi}\sigma}\sum_{i=1}^{n}\lambda^{n-i}\exp\left\{-\frac{e(i,n)e(i,n)^*}{2\sigma^2}\right\} \end{aligned} \qquad (9)$$

where $0 \ll \lambda < 1$ denotes the forgetting factor, and * is the conjugate operation. The output signal $y(i,n)$ and the error signal $e(i,n)$ can be expressed as

$$y(i,n) = \boldsymbol{w}^H(n)\boldsymbol{x}(i), \qquad (10)$$

$$e(i,n) = d(i) - \boldsymbol{w}^H(n)\boldsymbol{x}(i) \qquad (11)$$

where $\boldsymbol{w}(n)$ denotes the weight vector, and superscript H denotes Hermitian operator (conjugate transpose). Taking the gradient of $\Upsilon(n)$ with respect to the weight vector $\boldsymbol{w}(n)$, we obtain
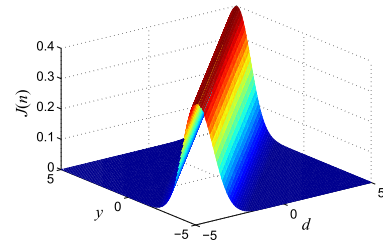

Fig. 2. MCC cost function in the joint space.

$$\begin{aligned} \frac{\partial \Upsilon(n)}{\partial \boldsymbol{w}^*(n)} &= -\frac{1}{2\sigma^2} \cdot \\ &\frac{1}{\sqrt{2\pi}\sigma}\sum_{i=1}^{n}\lambda^{n-i}\exp\left\{-\frac{e(i,n)e(i,n)^*}{2\sigma^2}\right\}\frac{\partial e(i,n)e^*(i,n)}{\partial \boldsymbol{w}*(n)}. \end{aligned} \qquad (12)$$

Letting (12) be zero, one gets

$$\sum_{i=1}^{n}\lambda^{n-i}\psi(i,n)\boldsymbol{x}(i)\boldsymbol{x}^H(i)\boldsymbol{w}(n) = \sum_{i=1}^{n}\lambda^{n-i}\psi(i,n)\boldsymbol{x}(i)d^*(i) \qquad (13)$$

where the weighting factors $\psi(i,n)$ is expressed as

$$\psi(i,n) = \exp\left\{-\frac{e(i,n)e^*(i,n)}{2\sigma^2}\right\}. \qquad (14)$$

Then, the expression of $\boldsymbol{w}(n)$ is obtained as follows:

$$\boldsymbol{w}(n) = \boldsymbol{F}(n)\boldsymbol{\pi}(n) = \boldsymbol{R}^{-1}(n)\boldsymbol{\pi}(n) \qquad (15)$$

where $\boldsymbol{F}(n) = \boldsymbol{R}^{-1}(n)$, $\boldsymbol{R}(n) = \sum_{i=1}^{n} \lambda^{n-i} \psi(i,n) \boldsymbol{x}(i) \boldsymbol{x}^H(i)$ and $\boldsymbol{\pi}(n) = \sum_{i=1}^{n} \lambda^{n-i} \psi(i,n) d^*(i) \boldsymbol{x}(i)$. For $\psi(i,n) = 1$, the algorithm becomes the RLS algorithm. When $\psi(i,n) \neq 1$, $\boldsymbol{R}(n)$ and $\boldsymbol{\pi}(n)$ are the weighted autocorrelation matrix and the weighted cross-correlation vector of the optimal weights via $\psi(i,n)$. We need to recalculate (15) at each iteration. To avoid this inconvenience, a sliding window method is proposed in [16]. However, the algorithm carries the main drawback of sliding-window strategy: the algorithm needs to keep in memory all previous samples within a window. To overcome this drawback, $\boldsymbol{R}(n)$ and $\boldsymbol{\pi}(n)$ are updated by recursive expression as follows:

$$\boldsymbol{R}(n) \approx \sum_{i=1}^{n} \lambda^{n-i} \psi(i,i) \boldsymbol{x}(i) \boldsymbol{x}^H(i)$$
$$= \lambda \boldsymbol{R}(n-1) + \psi(n,n) \boldsymbol{x}(n) \boldsymbol{x}^H(n), \quad (16)$$

$$\boldsymbol{\pi}(n) \approx \sum_{i=1}^{n} \lambda^{n-i} \psi(i,i) d^*(i) \boldsymbol{x}(i)$$
$$= \lambda \boldsymbol{\pi}(n-1) + \psi(n,n) \boldsymbol{x}(n) d^*(n). \quad (17)$$

An important point in (15)-(17) needs to be highlighted. Note from (15-17) that the adaptation is similar to the Wiener solution [14], which required matrix inverse operation. Hence, (15) demands heavy computational costs and is seldom used in practice. By using matrix inversion lemma [17], $\boldsymbol{F}(n)$ can be expressed as

$$\boldsymbol{F}(n) = \lambda^{-1} \boldsymbol{F}(n-1) - \lambda^{-1} \boldsymbol{\Phi}(n) \boldsymbol{x}^H(n) \boldsymbol{F}(n-1) \quad (18)$$

where $\boldsymbol{F}(0) = \rho^{-1} \boldsymbol{I}$, ρ is a small positive number, and the gain factor is defined by $\boldsymbol{\Phi}(n) = \frac{\psi(n,n) \boldsymbol{F}(n-1) \boldsymbol{x}(n)}{\lambda + \psi(n,n) \boldsymbol{x}^H(n) \boldsymbol{F}(n-1) \boldsymbol{x}(n)}$. Thus, $\boldsymbol{w}(n)$ can be updated as

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \boldsymbol{\Phi}(n)[d^*(n) - \boldsymbol{x}^H(n) \boldsymbol{w}(n-1)]. \quad (19)$$

Remark 1: Note that (14) and (19) demonstrate an implicit relationship between $\boldsymbol{w}(n)$ and $\psi(n,n)$ that cannot be solved in one step. Hence, the algorithm requires an iterative approximation to the solution, where $\psi(n,n)$ is calculated by using $\boldsymbol{w}(n-1)$, and the new value for $\boldsymbol{w}(n)$ is obtained via the value of $\psi(n,n)$.

Remark 2: The proposed algorithm is nearly blind since it does not require any priori information on the noise characteristics, and it can be implemented using only σ and λ.

IV. STEADY-STATE PERFORMANCE ANALYSIS

In this section, we analyze the steady-state performance of the Volterra-CMCC algorithm. Since the output of the Volterra depends linearly on the coefficients of the filter itself, we can use the method in [17], [18] for studying the steady-state excess MSE (EMSE) of Volterra-CMCC algorithm.

Consider the desired response arising from the model

$$d(n) = \boldsymbol{w}_o^H \boldsymbol{x}(n) + v(n) \quad (20)$$

where $\boldsymbol{w}_o$ is a vector containing the optimal coefficient values.

Define the weight deviation vector as:

$$\boldsymbol{\Omega}(n) = \boldsymbol{w}_o - \boldsymbol{w}(n). \quad (21)$$

Then, some assumptions are given as

1) The input signal $\boldsymbol{x}(n)$ is independent and identically distributed (i.i.d.) with zero-mean and is approximately independent of the a priori excess errors $e_a(n) = \boldsymbol{x}^H(n) \boldsymbol{\Omega}(n-1)$.

2) The noise signal $v(n)$ is i.i.d. with zero-mean and variance σ_v^2.

3) $v(n)$ and $\boldsymbol{x}(n)$ are mutually independent.

The adaptation of the weight deviation vector of the Volterra-CMCC algorithm can be given by

$$\boldsymbol{\Omega}(n) = \boldsymbol{\Omega}(n-1) - \frac{\psi(n,n) \boldsymbol{F}(n-1) \boldsymbol{x}(n)}{\lambda + \psi(n,n) \boldsymbol{x}^H(n) \boldsymbol{F}(n-1) \boldsymbol{x}(n)}$$
$$\cdot [d^*(n) - \boldsymbol{x}^H(n) \boldsymbol{w}(n-1)]. \quad (22)$$

Considering (18), and applying the matrix inversion formula, we have

$$\boldsymbol{F}^{-1}(n) = \lambda^{n+1} \rho \boldsymbol{I} + \sum_{i=0}^{n} \lambda^{n-i} \psi(i,i) \boldsymbol{x}^H(i) \boldsymbol{x}(i). \quad (23)$$

Then, (22) can be rewritten as

$$\boldsymbol{\Omega}(n) = \boldsymbol{\Omega}(n-1) - \vartheta(n) \boldsymbol{F}(n) \boldsymbol{x}^H(n) e(n) \quad (24)$$

where $\vartheta(n) = \frac{1}{\frac{\lambda}{\psi(n,n)} + \boldsymbol{x}^H(n) \boldsymbol{F}(n-1) \boldsymbol{x}(n)}$. When the exponential term is expanded with a second-order Taylor series, we obtain $\vartheta(n) = \frac{1}{\lambda [2\sigma^2 - |e(n)|^2]^{-1}/(2\sigma^2)^{-1} + \boldsymbol{x}^H(n) \boldsymbol{F}(n-1) \boldsymbol{x}(n)}$. In steady-state, we define the a posterior error $e_p(n) = \boldsymbol{x}^H(n) \boldsymbol{\Omega}(n)$. Multiplying both sides of (24) by $\boldsymbol{x}(n)$, we obtain the relationship between the *a priori* and the *a posteriori* estimation errors

$$e_p(n) = e_a(n) - \|\boldsymbol{x}(n)\|^2_{\boldsymbol{F}(n)\vartheta(n)} e(n) \quad (25)$$

where $\|\boldsymbol{x}\|^2_{\boldsymbol{\Lambda}} = \boldsymbol{x}^* \boldsymbol{\Lambda} \boldsymbol{x}$ stands for the squared-weighted Euclidean norm of a vector. When $\boldsymbol{x}(n) \neq 0$, the energy conservation relation (ECR) expression [17] for Volterra-CMCC algorithm can be given as

$$\boldsymbol{\Omega}(n) + \frac{\vartheta(n) \boldsymbol{F}(n) \boldsymbol{x}^H(n)}{\|\boldsymbol{x}(n)\|^2_{\vartheta(n) \boldsymbol{F}(n)}} e_a(n) =$$
$$\boldsymbol{\Omega}(n-1) + \frac{\vartheta(n) \boldsymbol{F}(n) \boldsymbol{x}^H(n)}{\|\boldsymbol{x}(n)\|^2_{\vartheta(n) \boldsymbol{F}(n)}} e_p(n). \quad (26)$$

Combining (24) and (25), and using $\vartheta^{-1}(n) \boldsymbol{F}^{-1}(n)$ as a weighting matrix for the squared-weighted Euclidean norm of a vector, we obtain

$$\|\boldsymbol{\Omega}(n)\|^2_{\vartheta^{-1}(n) \boldsymbol{F}^{-1}(n)} + \frac{|e_a(n)|^2}{\|\boldsymbol{x}(n)\|^2_{\vartheta(n) \boldsymbol{F}(n)}}$$
$$= \|\boldsymbol{\Omega}(n-1)\|^2_{\vartheta^{-1}(n) \boldsymbol{F}^{-1}(n)} + \frac{|e_p(n)|^2}{\|\boldsymbol{x}(n)\|^2_{\vartheta(n) \boldsymbol{F}(n)}}. \quad (27)$$

Suppose the Volterra-CMCC algorithm converges. Therefore in the steady state when $n \to \infty$ the optimum (minimum) EMSE can be obtained. Under this assumption the following approximation can be made:

$$E\left\{ \|\boldsymbol{\Omega}(n)\|^2_{\vartheta^{-1}(n) \boldsymbol{F}^{-1}(n)} \right\} \approx E\left\{ \|\boldsymbol{\Omega}(n-1)\|^2_{\vartheta^{-1}(n) \boldsymbol{F}^{-1}(n)} \right\} \quad (28)$$

Taking expectations of both sides of (27), and substituting (28) into (27) results in

$$
\mathrm{E}\left\{\frac{|e_a(n)|^2}{\|\boldsymbol{x}(n)\|^2_{\vartheta(n)\boldsymbol{F}(n)}}\right\} = \mathrm{E}\left\{\frac{|e_p(n)|^2}{\|\boldsymbol{x}(n)\|^2_{\vartheta(n)\boldsymbol{F}(n)}}\right\}. \tag{29}
$$

Now, substituting (25) into (29), we obtain that at steady state

$$
\mathrm{E}\left\{\|\boldsymbol{x}(n)\|^2_{\vartheta(\infty)\boldsymbol{F}(\infty)}\,|e(\infty)|^2\right\} = 2\mathrm{Re}\left(\mathrm{E}\left\{e_a^*(\infty)e(\infty)\right\}\right) \tag{30}
$$

where $\mathrm{Re}\{x\}$ denotes the real part of x. Consider $e(n) = e_a(n) + v(n)$, (22) can be expressed as

$$
\sigma_v^2\mathrm{E}\left\{\|\boldsymbol{x}(n)\|^2_{\vartheta(\infty)\boldsymbol{F}(\infty)}\right\} + \mathrm{E}\left\{\|\boldsymbol{x}(n)\|^2_{\vartheta(\infty)\boldsymbol{F}(\infty)}\,|e_a(\infty)|^2\right\}
$$
$$
= 2\mathrm{E}\left\{|e_a(\infty)|^2\right\}. \tag{31}
$$

Assume that $\|\boldsymbol{x}(n)\|^2_{\vartheta(\infty)\boldsymbol{F}(\infty)}$ is independent of $|e_a(\infty)|^2$ at steady-state, we have

$$
\sigma_v^2\mathrm{E}\left\{\|\boldsymbol{x}(n)\|^2_{\vartheta(\infty)\boldsymbol{F}(\infty)}\right\} + \mathrm{E}\left\{\|\boldsymbol{x}(n)\|^2_{\vartheta(\infty)\boldsymbol{F}(\infty)}\right\}
$$
$$
\cdot\mathrm{E}\left\{|e_a(\infty)|^2\right\} = 2\mathrm{E}\left\{|e_a(\infty)|^2\right\}. \tag{32}
$$

Inserting (32) into (30), we obtain

$$
\varsigma = \frac{\theta\sigma_v^2}{2-\theta} \tag{33}
$$

where $\varsigma \triangleq \mathrm{E}\left\{|e_a(\infty)|^2\right\}$ and $\theta \triangleq \mathrm{E}\left\{\|\boldsymbol{x}(n)\|^2_{\vartheta(\infty)\boldsymbol{F}(\infty)}\right\} = \mathrm{E}\left\{\frac{(2\sigma^2)^{-1}Tr(\boldsymbol{x}(n)\boldsymbol{x}^H(n)\boldsymbol{F}(\infty))}{(2\sigma^2-|e(\infty)|^2)^{-1}\lambda+(2\sigma^2)^{-1}Tr(\boldsymbol{x}(n)\boldsymbol{x}^H(n)\boldsymbol{F}(\infty))}\right\}$.

Recalling from (23), the steady-state mean value of $\boldsymbol{F}^{-1}(n)$ is obtained as $\boldsymbol{\mathcal{F}}^{-1} \triangleq \lim_{n\to\infty}\mathrm{E}\left\{\boldsymbol{F}^{-1}(n)\right\} = \frac{\mathrm{E}\{\psi(n,n)\}\boldsymbol{Q}(n)}{1-\lambda}$. The notation $\boldsymbol{Q}(n)$ stands for the covariance matrix $\boldsymbol{Q}(n) = \mathrm{E}\{\boldsymbol{x}(n)\boldsymbol{x}^H(n)\}$. According to the approximation in [17] and using the second-order Taylor series, the following approximation can be made:

$$
\mathrm{E}\left\{\boldsymbol{F}(\infty)\right\} \approx \left(\mathrm{E}\left\{\boldsymbol{F}^{-1}(\infty)\right\}\right)^{-1} = \boldsymbol{\mathcal{F}} = \frac{(1-\lambda)\boldsymbol{Q}^{-1}(\infty)}{\mathrm{E}\left\{\exp\left(-\frac{|e(\infty)|^2}{2\sigma^2}\right)\right\}}
$$
$$
\approx \frac{\mathrm{E}\{(2\sigma^2-|e(\infty)|^2)^{-1}\}(1-\lambda)\boldsymbol{Q}^{-1}(\infty)}{(2\sigma^2)^{-1}}. \tag{34}
$$

For $0 \ll \lambda < 1$, we have $|e_a(n)| \ll |v(n)|$ in the steady-state. Thus, substituting (34) and into (33), we have

$$
\varsigma = \frac{\sigma_v^2(1-\lambda)L\eta}{2-(1-\lambda)L\eta} \tag{35}
$$

where $\eta = \mathrm{E}\left[\frac{1}{\lambda\mathrm{E}\left((2\sigma^2-|v(n)|^2)^{-1}\right)(2\sigma^2-|v(n)|^2)+(1-\lambda)L}\right]$. Eq. (35) is difficult to further simplify. The theoretical results contain a random variable $v(n)$, but after the expect operation, we can obtain an exact value. Furthermore, (35) is also applicable to the analysis of linear CMCC algorithm.

Here, we compare the computational complexity of the Volterra-RLS algorithm and the proposed algorithm, as shown in Table I. It is well known that the RLS algorithm requires $4M^2 + 12M - 1$ real additions and $4M^2 + 16M + 1$ real multiplications for complex-valued signals. By using SOV expansion, the Volterra-RLS algorithm requires $M^2 + M$

TABLE I
SUMMARY OF THE COMPUTATIONAL COMPLEXITY.

Operation	Volterra-RLS	Volterra-CMCC
Real additions	$5M^2 + 13M - 1$	$5M^2 + 17M + 1$
Real multiplications	$6M^2 + 18M + 1$	$6M^2 + 26M + 7$
Other operations	0	1 exponential operation

real additions and $2M^2 + 2M$ real multiplications for computing $\boldsymbol{x}(n)$. Therefore, the Volterra-RLS algorithm requires $5M^2 + 13M - 1$ real additions and $6M^2 + 18M + 1$ real multiplications in total. The proposed algorithm moderately increases the computational complexity.
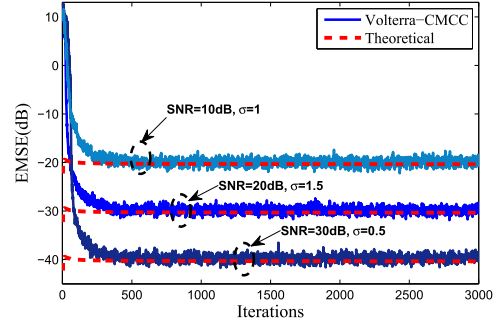

Fig. 3. Theoretical and simulation learning curves for Volterra-CMCC algorithm (λ=0.99).

V. SIMULATION RESULTS

In this section, we conduct a series of simulations to evaluate the performance of the Volterra-CMCC algorithm. In the first part, we provide the theoretical validation of the proposed algorithm. In the second part, we use simulations to illustrate the effectiveness of the Volterra-CMCC algorithm for nonlinear channel equalization and compare it with the Volterra-RLS algorithm. All the simulation results shown in this section are calculated over 100 independent runs.

A. Verification of EMSE

First, we perform simulation verification for the EMSE formulas in context of nonlinear system identification for the Volterra-CMCC algorithm. The input signal is the complex white Gaussian signal (CWGN) with zero-mean and unit variance, and the noise signal is the CWGN. Fig. 3 illustrates a comparison between the theoretical and simulation results for Volterra-CMCC algorithm with signal-to-noise ratio (SNR)=$[10, 20, 30]$dB. The unknown plant is a second-order nonlinear system, which is presented by [19]. As shown in Fig. 3, the theoretical results show a good match with the simulations. In addition, the simulation and theoretical EMSE values with different Volterra systems and noise environments are given in Table II. The number of iterations used is 3000. As can be seen, the simulation results agree quite well with the theoretical results.

TABLE II
EMSE FOR THE VOLTERRA-CMCC ALGORITHM (λ=0.99).

Noise power	Kernel size	Filter order	EMSE	
			Theory	Simulation
0.01	1	14 [18]	−31.46dB	−31.77dB
0.1	0.5	14 [18]	−21.97dB	−22.26dB
0.001	1.5	14 [15]	−41.30dB	−41.72dB
0.01	1	14 [15]	−31.48dB	−31.77dB

B. Application to nonlinear channel equalization

In second example, the performance of the Volterra-CMCC algorithnm is examined in the context of nonlinear channel equalization. To achieve good nonlinear modeling capabilities, the SOV filter is used [15]. To compare the BER performance of the Volterra-RLS and Volterra-CMCC algorithms fairly, we change the forgetting factor so that they have the similar convergence rate.

The SNR is defined as [15]

$$\text{SNR} = \frac{\sigma_x^2}{\gamma} \tag{36}$$

where σ_x^2 denotes the power of the input vector.

The transmitted signal and channel output has the following relationship:

$$x(n) = s(n) - 0.5s(n-1) + 0.001s^2(n) + 0.002s^2(n-1) + 0.007s(n)s(n-1). \tag{37}$$

The transmitted message is a 2−PAM signal. The characteristic exponent α of the noise is set to 1.8. The length of the SOV filter is set to $L = 14$ with the memory size of linear kernels $M = 4$ and the memory size of quadratic kernels 10. The BER performance of the algorithms in α-stable noise is illustrated in Fig. 4. For all the equalizers the BER decreases with SNR for nonlinear models. Again, the proposed algorithm outperforms the Volterra-RLS equalizer.
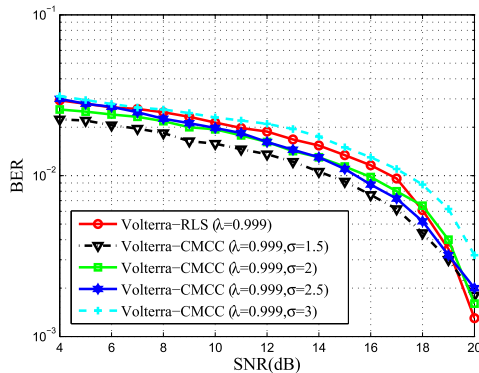

Fig. 4. BER performance of the equalizers.

VI. CONCLUSION

Based on the MCC, a new Volterra-CMCC algorithm, not requiring any a priori information, is proposed along with a Gaussian kernel for solving the nonlinear channel equalization problem. The MCC, which has been proven to be a robust and efficient optimization criterion for outliers, is used to improve the performance of equalizer. Moreover, we study the steady-state behavior of the Volterra-CMCC algorithm. As compared to the MSE-based criterion, the proposed algorithm achieves superior performance. Simulation results verified the efficiency of the proposed algorithm.

REFERENCES

[1] Z. Liu and C. Li, "Recursive least-squares for censored regression," IEEE Trans. Signal Process., vol. 65, no. 6, pp. 1565–1579, 2017.
[2] X. Hong, J. Gao, and S. Chen, "Zero attracting recursive least squares algorithms," IEEE Trans. Veh. Technol., vol. 66, no. 1, pp. 213–221, 2017.
[3] A. Navia-Vazquez and J. Arenas-Garcia, "Combination of recursive least p-norm algorithms for robust adaptive filtering in alpha-stable noise," IEEE Trans. Signal Process., vol. 60, no. 3, pp. 1478–1482, 2012.
[4] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: Stable processes and their applications," Proc. IEEE, vol. 81, no. 7, pp. 986–1010, 1993.
[5] D. Middleton, "Non-Gaussian noise models in signal processing for telecommunications: new methods an results for class A and class B noise models," IEEE Trans. Inform. Theory, vol. 45, no. 4, pp. 1129–1149, 1999.
[6] K. Pelekanakis and M. Chitre, "Adaptive sparse channel estimation under symmetric alpha-stable noise," IEEE Trans. Wireless Commun., vol. 13, no. 6, pp. 3183–3195, 2014.
[7] A. F. Molisch, Wireless Communications. John Wiley & Sons, 2012, vol. 34.
[8] K. Burse, R. N. Yadav, and S. Shrivastava, "Channel equalization using neural networks: A review," IEEE Trans. Syst. Man Cybern., vol. 40, no. 3, pp. 352–357, 2010.
[9] P. Bouboulis and S. Theodoridis, "Extension of Wirtinger's calculus to reproducing kernel Hilbert spaces and the complex kernel LMS," IEEE Trans. Signal Process., vol. 59, no. 3, pp. 964–978, 2011.
[10] P. Bouboulis, S. Theodoridis, and M. Mavroforakis, "The augmented complex kernel LMS," IEEE Trans. Signal Process., vol. 60, no. 9, pp. 4962–4967, 2012.
[11] W. Liu, P. P. Pokharel, and J. C. Príncipe, "Correntropy: Properties and applications in non-Gaussian signal processing," IEEE Trans. Signal Process., vol. 55, no. 11, pp. 5286–5298, 2007.
[12] B. Chen and J. C. Principe, "Maximum correntropy estimation is a smoothed MAP estimation," IEEE Signal Process. Lett., vol. 19, no. 8, pp. 491–494, 2012.
[13] L. Lu and H. Zhao, "Active impulsive noise control using maximum correntropy with adaptive kernel size," Mech. Syst. Signal Process., vol. 87, pp. 180–191, 2017.
[14] J. P. Guimaraes, A. I. Fontes, J. B. Rego, A. d. M. Martins, and J. C. Príncipe, "Complex correntropy: Probabilistic interpretation and application to complex-valued data," IEEE Signal Process. Lett., vol. 24, no. 1, pp. 42–45, 2017.
[15] B. Weng and K. E. Barner, "Nonlinear system identification in impulsive environments," IEEE Trans. Signal Process., vol. 53, no. 7, pp. 2588–2594, 2005.
[16] M. Belge and E. L. Miller, "A sliding window RLS-like adaptive algorithm for filtering alpha-stable noise," IEEE Signal Process. Lett., vol. 7, no. 4, pp. 86–89, 2000.
[17] A. H. Sayed, Fundamentals of Adaptive Filtering. John Wiley & Sons, 2003.
[18] J. Lee and V. J. Mathews, "A fast recursive least squares adaptive second order Volterra filter and its performance analysis," IEEE Trans. Signal Process., vol. 41, no. 3, pp. 1087–1102, 1993.
[19] S. Kalluri and G. R. Arce, "A general class of nonlinear normalized adaptive filtering algorithms," IEEE Trans. Signal Process., vol. 47, no. 8, pp. 2262–2272, 1999.