A Robust Total Least Mean Square Algorithm For Nonlinear Adaptive Filter

Ruixuan Wei
School of Electronic and Information Engineering
Xi'an Jiaotong University
Xi'an 710049, P.R. China
rxwei@chinaren.com

Chongzhao Han, Lanzhen Liu
School of Electronic and Information Engineering
Xi'an Jiaotong University
Xi'an 710049, P.R. China
czhan@xjtu.edu.cn

Abstract - The robust nonlinear adaptive filtering problem based on Volterra model is researched when the input and output observation data are both corrupted by noise in this paper. On the basis of minimizing Volterra total mean square error (VTMSE), a robust total least mean square adaptive filtering algorithm for nonlinear Volterra filter is proposed. The performance analysis demonstrates that the robust performance of the presented algorithm is nicer than other existed algorithms. And simulation results have also shown the prominent advantages of the presented algorithm, it can’t only permit to use larger learning factor, but its convergence precision is remarkably higher than other algorithms under higher noise environments.

Keywords: Nonlinear filter, Volterra series, total least mean squares, robust adaptive filter.

1 Introduction

It is well known that nonlinear adaptive filter has play an important role in tracking fusion system[2], communication system[1], and control system etc. Because Volterra series can completely describe the input and output transfer characteristic of a large type of nonlinear system, the Volterra adaptive filter has been widely studied in recent years. Usually, the input and output data sampled from a practical system are all corrupted by noise. Hence, it is necessary to find a efficient robust adaptive filter algorithm, and its convergence performance should be good. The traditional Volterra adaptive filter algorithm is built up based on minimizing the mean squares error [5]. Because there is an underlying assumption that the noise is imposed only and totally in the output observation data, the performance of this kind of algorithm becomes worse when it is applied to a practical filter problem. Although MCA algorithm for the total least squares (TLS) solution can provide better filter effects for this problem [4], but its robust performance will become worse for larger learning factor or under higher noise environments.

On the basis of the idea of minimum total mean square error of nonlinear Volterra filter, by modifying the gradient of Volterra total mean squares error (VTMSE), a robust total least mean square (RTLMS) adaptive filtering algorithm for nonlinear Volterra filter is presented in the paper. The algorithm can efficiently decrease the bad effect of the noise, and more importantly, it permits to use a larger learning factor, meanwhile, it still keeps fine convergence performance, that is very magnetic for the practical applications.

The paper is organized as follows: the section 2 is devoted for ordinary Volterra adaptive filter algorithm. The robust Volterra total least mean squares (VTLMS) algorithm is derived in section 3, meanwhile, its performance is analyzed. The simulation results are shown in section 4. Finally, the section 5 is the conclusion.

2 Volterra adaptive filter algorithm

2.1 Volterra LMS adaptive filter

For SISO time-invariant casual nonlinear system, its output \( y(t) \) can be written as the Volterra series in time domain as following [6]:

\[
y(t) = \sum_{n=1}^{\infty} y_n(t),
\]

where \( y_n(t) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \cdots, \tau_n) \prod_{i=1}^{n} u(t-\tau_i) d\tau_i, n \in \mathbb{N} \), and \( \mathbb{N} \) is the natural number set. \( y_n(t) \) is the output of the \( n \)-th order Volterra series, and \( h_n(\tau_1, \cdots, \tau_n) \) is the \( n \)-th order.
Volterra time-domain kernel or the \textit{n}th-order impulse response function.

In practical filter problem, the discrete truncated form of Volterra series is used as following:

\[
y(k) = \sum_{n=1}^{N} \sum_{m_1=0}^{M} \cdots \sum_{m_n=0}^{M} h_n(m_1, m_2, \cdots, m_n) \prod_{i=1}^{n} u(k - m_i),
\]

where \( M \) is a memory length of the Volterra kernel, and \( N \) is the maximum order of the truncated Volterra series.

The Volterra LMS adaptive filter is just online to adjust the kernel coefficients of Volterra filter by a recursive algorithm to minimize the mean square error of the filter.

Consider that the input and output observation data are both corrupted by noise, and write the input and output at \( k \) as \( u(k) = u(k) + n_i(k) \) and \( d(k) = d(k) + n_o(k) \), where \( n_i(k) \) and \( n_o(k) \) are the additive noise. Define the input observation vector of Volterra filter at \( k \) as:

\[
\tilde{X}_v(k) = \begin{bmatrix} x_1^T(k) & x_2^T(k) & \cdots & x_N^T(k) \end{bmatrix},
\]

where \( x_i(k) \) is the \( i \)th order input observation vector of the Volterra filter, for example:

\[
x_i^T(k) = [u_1^2, u_1u_2, \cdots, u_1u_{m-1}, u_2^2, \cdots, u_{m-1}^2],
\]

where \( u_{m-1} = \tilde{u}(k - m + 1) \). Note that, as usual, the Volterra kernel is assumed to be symmetric without loss of generality[7]. Corresponding define the \( i \)th order kernel vector, for example:

\[
h_v = [h_1(0,0) \cdots h_2(0,M-1) h_2(1,1) \cdots h_2(M-1,M-1)].
\]

And then define the Volterra kernel vector as:

\[
H_v = \begin{bmatrix} h_1^T \ h_2^T \ \cdots \ h_N^T \end{bmatrix}.
\]

Then, the Volterra LMS adaptive filter algorithm can be described as: [5]

\[
H_v(k+1) = H_v(k) + \mu \varepsilon(k) \tilde{X}_v(k),
\]

where \( \varepsilon(k) = \tilde{d}(k) - y(k) \) is error, \( \mu \) is the learning factor which controls the stability and rate of convergence of the adaptive algorithm.

The Volterra LMS adaptive filter algorithm has been extensively applied in the nonlinear adaptive filter. However, there is an underlying assumption that the input signal is known exactly, in practical situation it sometimes may be impossible to avoid noises when the input signal is measured. Due to the presence of noises in both input and output of the analyzed system, the Volterra LMS algorithm can only obtain the suboptimal solutions.

### 2.2 Volterra TLS adaptive filter

In order to improve the performance of the adaptive filter for this kind of practical problem, some adaptive methods based on optimal total least squares (TLS) technique has been proposed. By using Minor Components Analysis (MCA), a modified anti-Hebbian learning rule for TLS solution can be obtained [4]. Similar to Volterra LMS method, by describing the Volterra filter as a pseudo-linear one, the Volterra MCA adaptive filter algorithm can be built as follow:

\[
W(k+1) = W(k) - \mu \varepsilon(k) [Z(k) - \varepsilon(k)W(k)]
\]

where \( W = [H_v^T, -1]^T \) is the Volterra augmented kernel vector, \( Z(k) = [\tilde{X}_v(k), \tilde{d}(k)\tilde{d}^T(k)]^T \) is called as the Volterra augmented observation vector, and \( \varepsilon(k) = Z^T(k)W(k) \). \( \mu \) is the learning factor which controls the stability and rate of convergence of the adaptive algorithm.

However, the filtering performance of the Volterra MCA algorithm will become worse when a larger learning factor is used or signal-noise-rate (SNR) is lower. So a more effective adaptive filter algorithm will be proposed based on minimizing the Volterra total mean squared error.

### 3 Robust Volterra total least mean squares (VTLMSE) algorithm

#### 3.1 The derivation of robust VTLMS algorithm

Define the total error at \( k \) as \( \varepsilon(k) = \frac{\varepsilon(k)}{\sqrt{W^T(k)W(k)}} \). Then refer to the Widrow’s idea [8], the Volterra total mean squared error (VTMSE) can be obtained by taking the expected value of \( \varepsilon^2(k) \).

\[
VTMSE = E[\varepsilon^2(k)] = E\left( \frac{\varepsilon^2(k)}{W^T(k)W(k)} \right)
\]
where \( R = E[Z(k)Z^T(k)] \). Now the adaptive filter is just to adjust the kernel coefficients of Volterra filter to minimize the VTMSE. It is called Volterra total least mean squares (VTLMS) filter. According to (9), it can be known that the solution of VTLMS problem is the normalized eigenvector associated with the smallest eigenvalue of the input autocorrelation matrix \( R \).

Now using the method of steepest descent solves the VTLMS problem. To develop the adaptive algorithm, an estimated gradient should be obtained by using \( \hat{e}(k) \) as the estimated value of \( E\{\hat{e}^2(k)\} \) and differentiating it. And then modifying this estimated gradient as:

\[
\tilde{\nabla}E\{\hat{e}^2(k)\} = \frac{2\hat{e}(k)[Z(k)W^T(k)W(k)-\hat{e}(k-l)W(k)]}{[W^T(k)W(k)]^2}.
\]

So the robust VTLMS algorithm is an implementation of steepest descent using above modified estimated gradient:

\[
W(k+1) = W(k) - \frac{\mu \hat{e}(k)[Z(k)W^T(k)W(k)-\hat{e}(k-l)W(k)]}{[W^T(k)W(k)]^2}. \tag{11}
\]

### 3.2 The performance analysis of the proposed algorithm

Now analyze the robust performance of the presented algorithm by comparing with Volterra MCA algorithm.

Assume that the noise \( n_i(k) \) and \( n_c(k) \) are both independent stationary additive white noise, then denote

\[
Z(k) = \begin{bmatrix} X_i(k) \\ d(k) \end{bmatrix} = \begin{bmatrix} X_i(k) \\ n_i(k) \end{bmatrix} + \begin{bmatrix} n_c(k) \end{bmatrix} = Z(k) + \Delta Z(k). \tag{12}
\]

For the sake of simplicity, assume the Volterra augmented kernel vector at \( k \) has converged to true value \( W^* \), so \( Z^T(k)W(k) = 0 \). And then:

\[
\hat{e}(k) = Z^T(k)W(k) + \Delta Z^T(k)W(k) = \Delta Z^T(k)W(k). \tag{13}
\]

So the expected value of the weight gap can be obtained from (11), (12) and (13) as:

\[
\Delta W(k+1) = W(k+1) - W(k) = -\mu E\left[\Delta Z^T(k)W(k)\Delta Z(k)\right]. \tag{14}
\]

While for the Volterra MCA algorithm (8), it is:

\[
\Delta W(k+1) = -\mu E\{\Delta Z^T(k)W(k)\Delta Z(k)\} + \mu E\{W(k)W^T(k)\Delta Z(k)\Delta Z^T(k)W(k)\}. \tag{15}
\]

Comparing (14) with (15), it is clear that the robust performance of the presented algorithm is nicer than the Volterra MCA adaptive filter algorithm.

### 3.3 The convergence of robust VTLMS algorithm

Assume \( Z(t) \) is stationary, and \( Z(t) \) is not correlated with \( W(t) \), and following the reasoning of Xu [4], (11) can be approximated by the following differential equation:

\[
dW(t) = \left[-RW(t)W^T(t)W(t)+W^T(t)R,W(t-1)W(t)\right] \tag{16}
\]

where \( R = E[Z(k)Z^T(k)] \), \( R_1 = E[Z(k)Z^T(k-l)] \). The asymptotic property of (16) approximates that of (11), and the asymptotic property of (16) can be insured by the following theorem.

**Theorem:** Let \( R \) be a semipositive definite matrix, \( \lambda_{\ell+1} \) and \( c_{\ell+1} \) are respectively it’s smallest eigenvalue and corresponding normalized eigenvector with the non-zero last component. If \( \hat{W}(0)c_{\ell+1} \neq 0 \), then \( \lim_{t \to \infty} W(t) = \alpha_{\ell+1}(t)c_{\ell+1} \), where \( \alpha_{\ell+1}(t) \) is a scalar function. I.e. \( W(t) \) tends to be in the direction of \( c_{\ell+1} \) asymptotically as \( t \to \infty \).

**Proof:** Denote \( L+1 \) eigenvalues of \( R \) by \( \lambda_{\ell+1}, \lambda_{\ell+2}, \ldots, \lambda_{\ell+1} \), where \( \lambda_{\ell+1} \) is the smallest eigenvalue, and denote a set of corresponding normalized eigenvectors by \( c_1, c_2, \ldots, c_{\ell+1} \). So \( R \) and \( W(t) \) can be written as:

\[
R = \sum_{i=1}^{\ell+1} \lambda_i c_i c_i^T,
\]

\[
W(t) = \sum_{i=1}^{\ell+1} \alpha_i(t) c_i.
\]

And then it can be obtained:
Therefore, \( \frac{dW(t)}{dt} = \sum_{i=1}^{L} \frac{d\alpha_i(t)}{dt} c_i \),

\[
\frac{d\alpha_i(t)}{dt} = \frac{\lambda_i \|W(t)\|^2 + W(t)R_i W(t-1) \alpha_i(t) c_i}{\|W(t)\|^2}.
\]

Because of \( \alpha_i(t) = W^T(t) c_i \) and \( W^T(0) c_{L+1} \neq 0 \), so \( \alpha_{L+1}(t) \neq 0(\forall t \geq 0) \). Define

\[
\eta_i(t) = \frac{\alpha_i(t)}{\alpha_{L+1}(t)}.
\]

And then the following differential equation can be obtained:

\[
\frac{d\eta_i(t)}{dt} = \frac{\alpha_{L+1}(t) d\alpha_i(t)/dt - \alpha_i(t) d\alpha_{L+1}(t)/dt}{\alpha_{L+1}(t)}
\]

\[
= \frac{\lambda_{L+1} - \lambda_i}{\|W(t)\|^2} \eta_i(t).
\]

Because \( \lambda_{L+1} \) is the smallest eigenvalue, the differential system described by (19) will asymptotically converge. Therefore, \( \lim_{t \to \infty} \eta_i(t) = 0 \) (i = 1, ..., L). Because of \( \alpha_{L+1}(t) \neq 0 \) for \( \forall t \geq 0 \), then \( \lim_{t \to \infty} \alpha_i(t) = 0 \) (i = 1, ..., L).

So our conclusion can be obtained:

\[
\lim_{t \to \infty} W(t) = \lim_{t \to \infty} \left[ \sum_{i=1}^{L} \alpha_i(t) c_i \right] = \alpha_{L+1}(t) c_{L+1}.
\]

This completes the proof of the theorem.

4 Simulation

In the simulations, three algorithms are used for nonlinear adaptive filter, i.e. the traditional Volterra LMS adaptive filter algorithm [5], the Volterra MCA adaptive filter algorithm [4] and the presented robust Volterra total least mean squares (VRTLMS) adaptive filter algorithm. Their convergence performances are compared under different extent noise environment and by using different learning factor. The nonlinear system is given as following:

\[
y(k) = -0.64u(k) + u(k-2) + 0.9u^2(k) + u^2(k-1)
\]

Define the learning error of the Volterra kernel vector as:

\[
\text{Error}(k) = 20 \log \left| \mathbf{H}_0 - \mathbf{H}(k) \right|,
\]

where \( \mathbf{H}_0 = [-0.64 \ 0 \ 1 \ 0.9 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0] \) is the true Volterra kernel vector of the analyzed nonlinear system, \( \mathbf{H}(k) \) is estimated Volterra kernel vector.

Assume that the SNR (signal-noise rate) of the input signal is equal to the SNR of the output signal. The additive noise is independent zero-mean white noise. And

\[
\text{SNR} = 10 \log \left( \frac{E[\|y(k)\|^2]}{E[\|\mathbf{n}_i(k)\|^2]} \right),
\]

where \( y(k) \) is the output signal vector, \( \mathbf{n}_i(k) \) is the interference of the output.

Fig 1 and Fig 2 show respectively the learning curves for SNR=10dB and SNR=20dB with \( \mu = 0.05 \). These two learning curves indicate that the presented VRTLMS nonlinear adaptive filter algorithm can keep well convergence performance when a larger learning factor is used or SNR is lower, while the Volterra LMS and MCA algorithms cannot. Moreover, Fig 3 and Fig 4 have shown respectively the convergence curves of \( h_1(2) \) and \( h_2(1) \) at SNR=20dB and these convergence curves at SNR=10dB are given in Fig 5 and Fig 6. Because the Volterra LMS and MCA algorithms diverge severely, their corresponding curves are not plotted. Fig 7 and Fig 8 show respectively the convergence curves of \( h_1(2) \) and \( h_2(1) \) for SNR=20dB and \( \mu = 0.005 \). This set of curves illustrates the convergence performance of these algorithms, when a smaller learning factor has been used, and SNR is higher.

The simulation results have shown that the presented robust Volterra total least mean squares algorithm for nonlinear adaptive filter has excellent robust convergence performance, and it permits to use a larger learning factor, that is very fine for the practical applications.

5 Conclusion

On the basis of minimizing VTMSE, the paper proposes a robust adaptive filter algorithm for nonlinear Volterra filter with the corrupted input and output signal. The Volterra series is used to describe this kind nonlinear system. The analysis and simulation results have shown that this algorithm is more efficient to a practical nonlinear filter.
Reference


