Performance Analysis of Third-Order Nonlinear Wiener Adaptive Systems

Shue-Lee Chang and Tokunbo Ogunfunmi

Department of Electrical Engineering
Santa Clara University
Santa Clara, CA 95053
E-mail: schang@scudc.scu.edu, tokunbo@matthew.scu.edu

ABSTRACT

This paper presents a detailed performance analysis of third-order nonlinear adaptive systems based on the Wiener model. In earlier work, we proposed the discrete Wiener model for adaptive filtering applications for any order. However, we had focused mainly on first and second-order nonlinear systems in our previous analysis. Now, we present new results on the analysis of third order systems. All the results can be extended to higher-order systems. The Wiener model has many advantages over other models such as the Volterra model. These advantages include less number of coefficients and faster convergence. The Wiener model performs a complete orthogonalization procedure to the truncated Volterra series and this allows us to use linear adaptive filtering algorithms like the LMS to calculate all the coefficients efficiently. Unlike the Gram-Schmidt procedure, this orthogonalization method is based on the nonlinear discrete Wiener model. It contains three sections: a single-input multi-output linear with memory section, a multi-input, multi-output nonlinear no-memory section and a multi-input, single-output amplification and summary section. Computer simulation results are also presented to verify the theoretical performance analysis results.

1. INTRODUCTION

The nonlinear discrete-time Wiener model, which is based on orthogonal polynomial series derived from the Volterra series. The particular polynomials to be used are determined by the characteristics of the input signal that we are required to model. For Gaussian, white input, Hermite polynomial is chosen [1][2]. This Wiener model gives us a good eigenvalue spread of autocorrelation matrix which implies that faster convergence speed can be achieved. However, to calculate all the coefficients in the model sometimes is very difficult. This problem can easily be solved by using adaptive algorithm. The simplest way is to apply LMS (Least Mean Square) algorithm which has been applied to linear filtering problems because of its simplicity and low computational complexity. Another alternative way is to apply the RLS (Recursive Least Square) adaptive algorithm. However, the RLS algorithms are either computationally intensive or over-parameterized [3].

Different from the most previous works which are mostly based on Volterra model [4] and Gram-Schmidt procedure [5], this paper presents LMS adaptive filtering method which is based on the discrete nonlinear Wiener model. Similar concept as [6] but without using DFT, the main object of this model is to expand third-order truncated Volterra series by using some other orthogonal polynomial functions.. Theoretically, any M-sample memory third-order truncated Volterra series can have a three-channel, third-order nonlinear discrete Wiener model representation. For practical application, the efficient delay-line structure is developed in this paper. From the performance analysis, we find that the autocorrelation matrix of adaptive filter input vector can have much smaller eigenvalue spread than Volterra model. It is also interesting to note that , by using the nonlinear Wiener model, the linear adaptive filtering properties can be still preserved.

The rest of this paper is organized as follows. Sections 2 presents the Wiener model for the third-order Volterra system. Section 3 contains the analysis of LMS adaptive algorithm. The computer simulation examples are examined in Section 4. For simplicity and without loss of generality, we only consider the real input data case. Finally, the conclusions are given in Section 5.

2. NONLINEAR WIENER MODEL

The Volterra filter is based on the Volterra series, where each additive term is the output of a polynomial functional that can be thought of as a multidimensional convolution. The M-sample memory third-order Volterra filter output is the combination of various of these functionals which can be written as

$$\begin{aligned} y(n) = h_0 &+ \sum_{k_1=0}^{M-1} h_1(k_1)x(n-k_1) \\ &+ \sum_{k_1=0}^{M-1}\sum_{k_2=0}^{M-1} h_2(k_1,k_2)x(n-k_1)x(n-k_2) \qquad (1) \\ &+ \sum_{k_1=0}^{M-1}\sum_{k_2=0}^{M-1}\sum_{k_3=0}^{M-1} h_3(k_1,k_2,k_3)x(n-k_1)x(n-k_2)x(n-k_3) \end{aligned}$$

where $\{h_j(k_1,..., k_j), 0\leq j\leq 3\}$ is the set of third-order Volterra kernel coefficients. Assume that the kernels are symmetric, i.e.; $h_j(k_1, ..., k_j)$ is unchanged for any of j! permutation of indices k_1 , ..., k_j. One can think of a Volterra series expansion as a Taylor series with memory.

We can express the output y(n) by the equivalent third-order discrete nonlinear Wiener model which is [1]

$$y(n) = \sum_{i=0}^{3}\sum_{\alpha_i} c_{\alpha_i}^{(i)}\,\tilde{Q}_{\alpha_i}^{(i)}(z) \qquad (2)$$

The superscript i indicates the ith degree $\tilde{Q}$-polynomial, α_i= perm(n_0, n_1, ..., n_i) which can be obtained by the permutation of the elements of n_0, n_1, ..., n_i and z = z_{n_n} (n). The $\tilde{Q}$-polynomial is defined as

$$\tilde{Q}_{\alpha_i}^{(i)}(z) = \prod_{i=1}^{r} H_{k_i}[z_{n_i}(n)] \qquad (3)$$

where $\sum_{i=1}^{r} k_i$ = i and H_{k_i}(z) means Hermite polynomial. If z is the zero mean Gaussian signals with variance σ_x^2, the first four Hermite polynomials are $H_0(z)$ =1, $H_1(z)$ = z, $H_2(z) = z^2-\sigma_x^2$ and $H_3(z) = z^3 - 3z\sigma_x^2$ and $\tilde{Q}$-polynomial satisfies E{$\tilde{Q}_a^{(i)}(z)$} = 0 and has the orthogonal property

$$E\{\tilde{Q}_a^{(p)}(z)\tilde{Q}_b^{(q)}(z)\} = \text{const.}\,\delta(a-b)\delta(p-q) \qquad (4)$$

where E{ } is statistical mean and δ is Dirac delta function. To realize (2), we can select orthonormal bases as a delay line. The block diagram is shown in Fig.1:

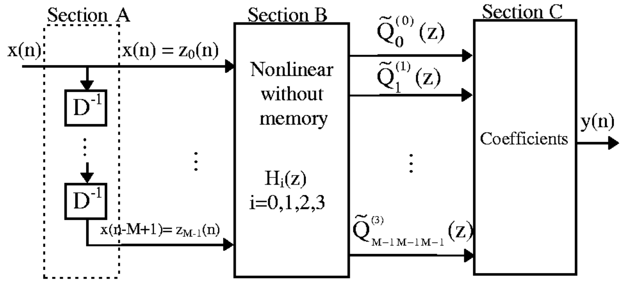


Fig.1 Delay line structure of thir-order nonlineadicretes Wiener model

3. ADAPTATION AND PERFORMANCE ANALYSIS

To explore the adaptive nonlinear Wiener filtering algorithm, consider a system identification application shown in Fig.2:

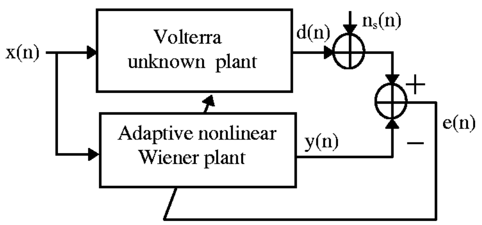


Fig.2 System identification model

For Gaussian white input x(n) and Gaussian white plant noise $n_s(n)$, it is very suitable to apply nonlinear Wiener model in adaptive plant block. It is because the $\tilde{Q}$-polynomial has perfect orthogonality that allows us to perform LMS algorithm with a reasonably good convergence rate. To develop it, we need to write (2) in a matrix form

$$y(n) = \mathbf{S}_{\tilde{Q}}^{-1}\tilde{\mathbf{Q}}^{T}(n)\mathbf{C}(n) \qquad (5)$$

where the superscript ^{T} means transpose, $\tilde{\mathbf{Q}}(n) = [1, \tilde{Q}_0^{(1)}, ..., \tilde{Q}_{M-1M-1M-1}^{(3)}]^T$ and $\mathbf{C}(n) = [w_0(n), w_1(n), ..., w_L(n)]^T$ are vectors with L+1 length. $\mathbf{S}_{\tilde{Q}}^{-1}$ is a scale matrix which can make $\mathbf{S}_{\tilde{Q}}^{-1}\mathbf{R}_{\tilde{Q}\tilde{Q}}(n)\ \mathbf{S}_{\tilde{Q}}^{-1}$ become an identity matrix, where $\mathbf{R}_{\tilde{Q}\tilde{Q}}$ = E{$\tilde{\mathbf{Q}}(n)\tilde{\mathbf{Q}}^{T}(n)$}. The coefficients can be updated according to

$$\mathbf{C}(n+1) = \mathbf{C}(n) + 2\mu\, e(n)\, \mathbf{S}_{\tilde{Q}}^{-1}\tilde{\mathbf{Q}}(n) \qquad (6)$$

where μ is the step size and measurement error e(n)=d(n) -y(n). The detailed performance analysis of LMS algorithm of Fig.2 is shown as follows. For the M-sample memory truncated third-order Volterra system, the plant input vector is

$$\begin{aligned}\mathbf{X}(n) = [\ & x(n), x(n-1), ..., x(n-M+1), x^2(n),... x^2(n-M+1),\\ & x(n)x(n-1), ..., x^3(n), ... x^3(n-M+1), x^2(n)x(n-1), ...,\\ & x(n-M-1)x(n-M)x(n-M+1)]^T \qquad (7)\end{aligned}$$

Assume the unknown plant has $\mathbf{C}^* = [w_0^*, w_1^*, ..., w_L^*]^T$ weight vector of length L+1. For Volterra model, the output d(n) is equal to

$$d(n) = \mathbf{C}^{*T}\mathbf{Q}(n) = \mathbf{C}^{*T}[1, \mathbf{X}^T(n)]^T = w_0^* + \mathbf{W}^{*T}\mathbf{X}(n) \qquad (8)$$

where $\mathbf{Q}(n) = [1, \mathbf{X}^T(n)]^T$ and $\mathbf{W}^* = [w_1^*, ..., w_L^*]^T$. Consider third-order discrete nonlinear Wiener structure as in Fig.1, in section A, there are M-1 delay elements. H_0, H_1, H_2 and H_3 are applied in section B. The input vector of section C is

$$\begin{aligned}\tilde{\mathbf{X}}(n) = \mathbf{S}_{\tilde{X}}^{-1}\ & [x(n), x(n-1), ..., x(n-M+1), x^2(n)-\sigma_x^2, ...,\\ & x^2(n-M+1)-\sigma_x^2, ..., x(n)x(n-1), ..., x^3(n)-3\sigma_x^2x(n),..., \quad (9)\\ & x^3(n-M+1)-3\sigma_x^2x(n-M+1), ..., (x^2(n)-\sigma_x^2)x(n-1), ...,\\ & x(n-M-1)x(n-M)x(n-M+1)]^T\end{aligned}$$

The output of adapted plant y(n) is

$$y(n) = \mathbf{C}^T(n)\,\tilde{\mathbf{Q}}(n) = \mathbf{C}^T(n)[1, \tilde{\mathbf{X}}^T(n)]^T = w_0 + \mathbf{W}^T(n)\mathbf{S}_{\tilde{X}}^{-1}\tilde{\mathbf{X}}(n) (10)$$

where $\mathbf{C}(n) = [w_0(n), \mathbf{W}^T(n)]^T$, $\mathbf{W}(n) = [w_1(n), ...,w_L(n)]^T$ and $\tilde{\mathbf{Q}}(n) = [1, \tilde{\mathbf{X}}^T(n)]^T$. $\mathbf{C}(n)$ is the coefficient vector in section C. These coefficients are adapted to the proper values, then the model of adaptive plant will match exactly where the same model of the unknown plant. The error signal can be written as

$$\begin{aligned}e(n) &= d(n) - y(n) + n_s(n)\\ &= n_s(n) + w_{0*} - w_0(n) + \mathbf{W}^{*T}\mathbf{X}(n) - \mathbf{W}^T(n)\,\mathbf{S}_{\tilde{X}}^{-1}\tilde{\mathbf{X}}(n) \qquad (11)\end{aligned}$$

Then, the mean square error ξ(n)= E{$e^2(n)$} can be obtained by expanding (11).

$$\begin{aligned}\xi(n) = \sigma_{n_s}^2 &+ w_0^{*2} + w_0^2(n) - 2w_0^*w_0(n)\\ &+ 2w_0^*E\{\mathbf{X}^T(n)\}\mathbf{W}^* - 2w_0(n)E\{\mathbf{X}^T(n)\}\mathbf{W}^* + \mathbf{W}^{*T}\mathbf{R}_{XX}\mathbf{W}^*\\ &+ \mathbf{W}^T(n)\,\mathbf{S}_{\tilde{X}}^{-1}\mathbf{R}_{\tilde{X}\tilde{X}}\mathbf{S}_{\tilde{X}}^{-1} - 2\mathbf{W}^T(n)\,\mathbf{S}_{\tilde{X}}^{-1}\mathbf{R}_{\tilde{X}X}\mathbf{W}^* \qquad (12)\end{aligned}$$

where $\sigma_{n_s}^2 = E\{n_s^2(n)\}$, $\mathbf{R}_{\mathbf{XX}}=E\{\mathbf{X}(n)\mathbf{X}^T(n)\}$, $\mathbf{R}_{\tilde{\mathbf{X}}\tilde{\mathbf{X}}}= E\{\tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)\}$ and $\mathbf{R}_{\tilde{\mathbf{X}}\mathbf{X}}= E\{\tilde{\mathbf{X}}(n)\mathbf{X}^T(n)\}$. To minimize (12), we need to take derivative respective to $\mathbf{C}(n) = [w_0(n), \mathbf{W}^T(n)]^T$, which is

$$\nabla\xi(n) = \mathbf{0} \qquad (13)$$

where ∇ means the gradient operator and $\mathbf{0}$ is a zero column vector. Then, the optimal solutions can be obtained

$$w_{0/optm}= w_0^* + E\{\mathbf{X}^T(n)\}\mathbf{W}^* \qquad (14a)$$

$$\mathbf{W}_{optm}= \mathbf{S}_{\tilde{\mathbf{X}}}\,\mathbf{R}_{\tilde{\mathbf{X}}\tilde{\mathbf{X}}}^{-1}\,\mathbf{R}_{\tilde{\mathbf{X}}\mathbf{X}}\,\mathbf{W}^* \qquad (14b)$$

Substitute (14a) and (14b) into (12), the minimum mean square error is

$$\xi_{min}= E\{n_s^2(n)\} = \sigma_{n_s}^2 \qquad (15)$$

It is interesting to note that the unknown plant is Volterra model and the adapted plant is nonlinear Wiener model which both are not linear system, but from the derivations, we can see that (14a), (14b) and (15) have similar forms as LMS algorithm for linear system. To derive step size range, we need to consider the instantaneous version of (12). Define ε(n) as the error power $e^2(n)$ which can be obtained

$$\begin{aligned}\varepsilon(n) &= [n_s(n) + w_0^* + w_0(n)]^2 + \mathbf{W}^{*T}\mathbf{X}(n)\mathbf{X}^T(n)\,\mathbf{W}^*\\ &+ \mathbf{W}^T(n)\,\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\,\tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)\,\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\mathbf{W}(n) + 2n_s(n)\,\mathbf{X}^T(n)\mathbf{W}^*\\ &+ 2w_0^*\,\mathbf{X}^T(n)\mathbf{W}^* - 2w_0(n)\mathbf{X}^T(n)\mathbf{W}^* - 2n_s(n)\,)\tilde{\mathbf{X}}^T(n)\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\mathbf{W}(n)\\ &- 2w_0^*\,\tilde{\mathbf{X}}^T(n)\,\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\mathbf{W}(n) - 2w_0(n)\,)\tilde{\mathbf{X}}^T(n)\,\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\mathbf{W}(n)\\ &- 2\mathbf{W}^T(n)\,\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\,\tilde{\mathbf{X}}(n)\,\mathbf{X}^T(n)\mathbf{W}^* \qquad (16)\end{aligned}$$

Take derivative of (16), we can obtain

$$\nabla\varepsilon(n) = -2e(n) - 2\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\,\tilde{\mathbf{X}}(n)e(n) \qquad (17)$$

Based on steepest descent method, the weight is updated by [7]

$$\mathbf{C}(n+1) = -2e(n)\mathbf{S}_{\tilde{\mathbf{Q}}}^{-1}\tilde{\mathbf{Q}}(n+1) \qquad (18)$$

This is similar but different from LMS algorithm for linear systems. The alternative form of (18) is

$$\mathbf{C}(n+1) = (\,\mathbf{I} - 2\mu\,\mathbf{S}_{\tilde{\mathbf{Q}}}^{-1}\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}}\mathbf{S}_{\tilde{\mathbf{Q}}}^{-1})\mathbf{C}(n) + 2\mu\,\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}}\mathbf{C}^* \qquad (19)$$

where $\mathbf{I}$ is an identity matrix. Define the weight error vector

$$\mathbf{V}(n) = \mathbf{S}_{\tilde{\mathbf{Q}}}^{-1}\mathbf{C}(n) - \mathbf{C}^* \qquad (20)$$

Substitute (20) into (19), we can have

$$\mathbf{V}(n+1) = (\,\mathbf{I} - 2\,\mu\,\mathbf{S}_{\tilde{\mathbf{Q}}}^{-2}\,\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}})\mathbf{V}(n) \qquad (21)$$

Note $(\mathbf{S}_{\tilde{\mathbf{Q}}}^{-2}\,\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}})\,\mathbf{P} = \mathbf{PD}$ where $\mathbf{P}$ and $\mathbf{D}$ are eigenvector square matrix and eigenvalue square matrix of $\mathbf{S}_{\tilde{\mathbf{Q}}}^{-2}\,\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}}$ respectively. With $\mathbf{V}(n) = \mathbf{P}\mathbf{V}'(n)$, then

$$\mathbf{V}'(n) = (\,\mathbf{I} - 2\mu\,\mathbf{D})^n\mathbf{V}'(0) \qquad (22)$$

For convergence, the range of μ is $0 < \mu < 1/\lambda_{max}$ or

$$0< \mu < 1/\mathrm{tr}[\mathbf{S}_{\tilde{\mathbf{Q}}}^{-2}\,\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}}] < 1/\lambda_{max} \qquad (23)$$

where λ_{max} is the maximum eigenvalue of $\mathbf{S}_{\tilde{\mathbf{Q}}}^{-2}\,\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}}$.

To derive the misadjustment, we need to consider the steady state condition. The steady state of (12) can be expressed as

$$\begin{aligned}\hat{\xi}(n) &= E\{[n_s(n) + w_0^* - \hat{w}_o(n)]^2\} + \mathbf{W}^{*T}\,E\{\mathbf{X}(n)\mathbf{X}^T(n)\}\mathbf{W}^*\\ &+ \hat{\mathbf{W}}^T\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\,E\{\tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)\}\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\hat{\mathbf{W}}(n) + 2w_0^*E\{\mathbf{X}^T(n)\}\mathbf{W}^*\\ &- 2\,\hat{w}_0(n)\,E\{\mathbf{X}^T(n)\}\mathbf{W}^* - 2\hat{\mathbf{W}}^T(n)\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\,E\{\tilde{\mathbf{X}}(n)\mathbf{X}^T(n)\}\mathbf{W}^* \qquad (24)\end{aligned}$$

where the header ^ means steady state. In steady state, from (14b) we can assume that $\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\hat{\mathbf{W}}(n) \approx \mathbf{R}_{\tilde{\mathbf{X}}\tilde{\mathbf{X}}}^{-1}\mathbf{R}_{\tilde{\mathbf{X}}\mathbf{X}}\mathbf{W}^*$, and substitute this formula into (24), expand, rearrange and simplify we can obtain

$$\begin{aligned}\hat{\xi}(n) &\approx \xi_{min} +\\ &[(\mathbf{R}_{\tilde{\mathbf{X}}\mathbf{X}}^{-1}\mathbf{R}_{\tilde{\mathbf{X}}\mathbf{X}}\mathbf{W}^*)^T - \mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\hat{\mathbf{W}}^T(n)]\mathbf{R}_{\tilde{\mathbf{X}}\tilde{\mathbf{X}}}[\mathbf{R}_{\tilde{\mathbf{X}}\tilde{\mathbf{X}}}^{-1}\mathbf{R}_{\tilde{\mathbf{X}}\mathbf{X}}\mathbf{W}^* - \mathbf{S}_{\tilde{\mathbf{X}}}^{-1}\hat{\mathbf{W}}(n)]\\ &\approx \xi_{min} + \tilde{\mathbf{V}}^T\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}}\tilde{\mathbf{V}} \qquad (25)\end{aligned}$$

where $\tilde{\mathbf{V}}(n) = \mathbf{S}_{\tilde{\mathbf{Q}}}^{-1}\hat{\mathbf{C}}^T(n) - \mathbf{C}^*$. Define the excess mean-square error as

$$\begin{aligned}\text{excess MSE} &= E\{\hat{\xi}(n) - \xi_{min}\} = \tilde{\mathbf{V}}^T(n)\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}}\cdot\tilde{\mathbf{V}}(n)\\ &= E\{\tilde{\mathbf{V}}^T(n)\tilde{\mathbf{P}}^T\tilde{\mathbf{D}}\,\tilde{\mathbf{P}}\,\tilde{\mathbf{V}}(n)\}\\ &= \tilde{\mathbf{V}}'^T(n)\tilde{\mathbf{D}}\tilde{\mathbf{V}}'(n) \qquad (26)\end{aligned}$$

where $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{D}}$ are eigenvector square matrix and eigenvalue square matrix of $\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}}$ respectively. Define $\tilde{\mathbf{V}}(n) = \tilde{\mathbf{P}}\mathbf{V}'(n)$, then Consider

$$\hat{\nabla}\,e(n) = \nabla\,e(n) + \mathbf{N}(n) \qquad (27)$$

where $\mathbf{N}(n)$ is defined as a gradient estimation noise vector [7]. In steady state, $\nabla\,e(n) \approx 0$, then

$$\hat{\nabla}\,e(n) = \mathbf{N}(n) = -2\,e(n)\mathbf{S}_{\tilde{\mathbf{Q}}}^{-1}\tilde{\mathbf{Q}}(n) \qquad (28)$$

There are three steps to find $E\{\tilde{\mathbf{V}}'(n)\,\tilde{\mathbf{V}}'^T(n)\} = \mathrm{cov}\{\tilde{\mathbf{V}}'(n)\}$. Assume $e(n)$ and $\tilde{\mathbf{Q}}(n)$ are independent. First, find the covariance of $\mathbf{N}(n)$ which is

$$\mathrm{cov}\{\mathbf{N}(n)\} \approx 4\xi_{min}\,\mathbf{S}_{\tilde{\mathbf{Q}}}^{-1}\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}}\mathbf{S}_{\tilde{\mathbf{Q}}}^{-1} \qquad (29)$$

Second, define $\tilde{\mathbf{N}}'(n) = \tilde{\mathbf{P}}^{-1}\mathbf{N}(n)$, then

$$\mathrm{cov}\{\tilde{\mathbf{N}}'(n)\} = 4\xi_{min}\,\mathbf{S}_{\tilde{\mathbf{Q}}}^{-1}\tilde{\mathbf{D}}\,\mathbf{S}_{\tilde{\mathbf{Q}}}^{-1} \qquad (30)$$

Third, use (29), (30) and (21), we can have [8]

$$\mathrm{cov}\{\tilde{\mathbf{V}}'(n)\} \approx \mu\xi_{min}\mathbf{S}_{\tilde{\mathbf{Q}}}^{-2} \qquad (31)$$

Finally, the excess mean square error is

$$\text{excess MSE} = E\{\tilde{\mathbf{V}}'^T(n)\tilde{\mathbf{D}}\tilde{\mathbf{V}}'(n)\} \approx \mu\xi_{min}\mathrm{tr}[\mathbf{S}_{\tilde{\mathbf{Q}}}^{-2}\,\mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}}] \qquad (32)$$

The misadjustment now is defined as [7]

$$\text{MISADJ= excess } \text{MSE}/\xi_{\min} \approx \mu\, \text{tr}[\mathbf{S}_{\tilde{\mathbf{Q}}}^{-2}\, \mathbf{R}_{\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}}] \qquad (33)$$

For time constant analysis, let us rewrite (21) in scalar form

$$v'(n) = (1 - 2\mu\lambda_n)^n\, v'(0) \qquad (34)$$

The term $(1-2\mu\lambda_n)^n$ can be approximated by $(e^{1/\tau_n})^n$. If τ_n is large, then

$$1 - 2\mu\lambda_n \approx e^{1/\tau_n} \approx 1/\lambda_n \qquad (35)$$

The time constant is found as

$$\tau_n \approx 1/2\mu\lambda_n \qquad (36)$$

All the derivations above can be extended to higher-order without difficulty.

4. SIMULATION

To evaluate the adaptive nonlinear Wiener filter, let us consider that the unknown plant contains a 3-sample memory third-order Volterra filter which has following input-output relationship:

$$\begin{aligned} y(n) = &-0.78x(n)-1.48x(n-1)+1.39x(n-2)+0.54x^2(n)-1.62x^2(n-1) \\ &+1.41x^2(x-2) +3.72x(n)x(n-1) +1.86x(n)x(n-2) \\ &+0.76x(n-1)x(n-2)+0.04x^3(n)-0.76x^3(n-1)+0.5x^3(n-2) \\ &-0.12x^2(n)x(n-1)-1.52x^2(n)x(n-2)-0.13x^2(n-1)x(n) \\ &+0.15x^2(n-1)x(n-2)-0.23x^2(n-2)x(n)-0.75x^2(n-2)x(n-1) \\ &+0.33x(n)x(n-1)x(n-2) \end{aligned} \qquad (37)$$

We can use delay-line structure as in Fig.1 of 2 delay elements in section A. H_0, H_1, H_2 and H_3 are used in Section B. There are 20 coefficients in Section C. Properly select scale matrix which can let autocorrelation matrix become an identity matrix. With SNR = -40 db, the simulation results of ensemble average for the 50 independent runs for both Wiener model and Volterra model are shown in Fig.3. For μ = 0.0005, we see that Wiener model adaptive filter has much better performance than Volterra model which does not converge to the right values. For μ =0.003, the Wiener model adaptive filter is still very stable. But the Volterra model becomes unstable and fails to identify this third-order Volterra system. This is because all $\tilde{Q}$-polynomials in nonlinear Wiener model are orthogonal which reduces the eigenvalue spread and improves the nonlinear LMS adaptive filter performance dramatically. For an ensemble average over 50 runs, the misadjustment of Wiener model of μ =0.0005 is equal to 0.012 which is close to the theoretical value 0.01. To examine the eigenvalue spread characteristics, we need to evaluate the eigenvalue of autocorrelation matrix. We can find that for Volterra model, the eigenvalue spread is equal to 64.136 (compared with the theoretical value 63.779). For Wiener model, the eigenvalue spread is 1.12 (compared with the theoretical value 1). The eigenvalue spread is reduced about 64 times. The autocorrelation matrix of Wiener model is shown in Fig.4.

5. CONCLUSIONS

Different from most previously published Volterra model structure, this paper presented LMS adaptive filtering algorithm based on nonlinear discrete Wiener model for third-order Volterra system identification. For Gaussian white input, the $\tilde{Q}$-functional can characterize the nonlinear system behaviors. In practical application, , it allows us to have small eigenvalue spread and good convergence rate which are confirmed by the performance analysis and computer simulations. Future work will involve application to least-squares and fast adaptive algorithms.

Fig.3 Learning curve of Volterra model and Wiener model

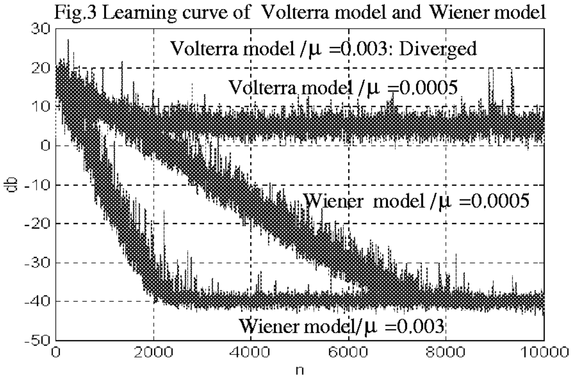
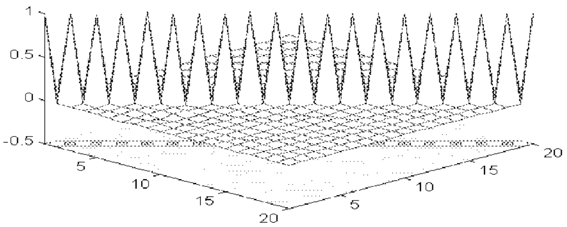


Fig.4 Autocorrelation matrix of Wiener model



6. REFERENCES

[1] M. Schetzen, *Volterra and Wiener Theory of the Nonlinear Systems*, New York; Wiley, 1980.

[2] Charlie W. Therrien and Atalla Hashad,"The discrete Wiener model for Representation of NonGaussian Stochastic Processes," *The 27th Asilomar Conf. Record on Signal, Systems & Computers*, Nov. 1993, pp.451-455.

[3] V. John Mathews, "Adaptive Polynomial Filters," *IEEE Signal Processing Mag.*, vol. 8, no. 3, July 1991, pp.10- 26.

[4] M. A., Syed and V. J. Matthews, "Lattice algorithms for recursive least squares adaptive second-order Volterra filtering," *IEEE Trans.Cir. syst.-II,* vol. 41, Mar., 1994.

[5] V. John Mathews, "Adaptive Volterra Filters Using Orthogonal Structures," *IEEE Signal Processing Letters*, vol. 3, no. 12, Dec. 1996, pp.307-309.

[6] Shue-Lee Chang and Tokunbo Ogunfunmi, " An improved nonlinear adaptive filter and PDF estimator using Discrete Fourier Transform," *Proc. of 8th IEEE DSP Workshop*, August, 1998, pp.112-115.

[7] Bernard Widrow and Samuel D. Stearns, *Adaptive signal processing*, Prentice-Hall, Englewood N.J., 2nd ed., 1985.

[8] Shue-Lee Chang and Tokunbo Ogunfunmi, "Performance analysis of nonlinear adaptive filter based on LMS algorithm," *The 31st Asilomar Conf. Record on Signal, Systems & Computers*, Nov. 1997, pp.107-110.