An Improved Approximate QR-LS Based Second-Order Volterra Filter

Yi Zhou, S. C. Chan and *K. L. Ho*

Department of Electrical and Electronic Engineering,
The University of Hong Kong, Pokfulam Road, Hong Kong.
Email : {yizhou, scchan, klho}@eee.hku.hk

Abstract: This paper proposes a new transform-domain approximate QR Least-Squares-based (TA-QR-LS) algorithm for adaptive Volterra filtering (AVF). It improves the approximate QR Least-Squares (A-QR-LS) algorithm for multichannel adaptive filtering by introducing a unitary transformation to decorrelate the input signal vector so as to achieve better convergence and tracking performances. Further, the Givens rotation is used instead of the Householder transformation to reduce the arithmetic complexity. Simulation results show that the proposed algorithm has much better initial convergence and steady state performance than the LMS-based algorithm. The fast RLS AVF algorithm [6] was found to exhibit superior steady state performance when the forgetting factor is chosen to be 0.995, but the tracking performance of the TA-QR-LS algorithm was found to be considerably better.

I INTRODUCTION

The Volterra filter (VF) is a simple representation of nonlinear systems, which can be used to model a large class of nonlinear engineering systems. For instances, it has been applied to the modeling of nonlinear dynamical systems, interference and noise reduction, and echo cancellation for communication problems, etc. The input and output relation of a Volterra filter has the following form [1]:

$$y(n) = h_0 + \sum_{m_1=0}^{\infty} h_1(m_1)x(n-m_1) + \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} h_2(m_1,m_2)x(n-m_1)x(n-m_2) + \ldots + \sum_{m_1=0}^{\infty}\cdots\sum_{m_p=0}^{\infty} h_p(m_1,m_2,\ldots,m_p)x(n-m_1)x(n-m_2)\ldots x(n-m_p) + \ldots \quad (1)$$

where $y(n)$ and $x(n)$ are respectively the output and input of the system. $h_p(m_1,m_2,\ldots,m_p)$ represents the p-th order Volterra kernel of this system, which can be considered to be symmetric [1],[2]. However, the large number of filter coefficients of this expansion limits its wide application. Therefore, it is usually truncated to a low-order Volterra filter such as the following 2nd order system:

$$y(n) = \sum_{m_1=0}^{M-1} h_1(m_1)x(n-m_1) + \sum_{m_1=0}^{M-1}\sum_{m_2=m_1}^{M-1} h_2(m_1,m_2)x(n-m_1)x(n-m_2) \quad (2)$$

which has M-1delay elements. (2) can be written more compactly in matrix form as

$$y(n) = \mathbf{x}_M^T(n)\boldsymbol{\theta}_m, \quad (3)$$

where
$$\boldsymbol{\theta}_m = [h_1(0),\ldots,h_1(M-1),h_2(0,0),\ldots h_2(0,M-1), h_2(1,1),\ldots,h_2(M-1,M-1)]^T \quad (4)$$

$$\mathbf{x}_M(n) = [x(n),\ldots,x(n-M+1),x^2(n), \ldots,x(n)x(n-M+1),x^2(n-1),\ldots x^2(n-M+1)]^T; \quad (5)$$

where $\boldsymbol{\theta}_m$ and $\mathbf{x}_M(n)$ are respectively the coefficient vector and input signal vector. A very important advantage of the VF is that the output is still a linear function of the system parameters $\boldsymbol{\theta}_m$. Therefore, a treasure of adaptive filtering algorithms such as the well-known least mean squares (LMS) or recursive least squares (RLS) algorithms can be applied. The LMS AVF [3] has a low arithmetic complexity of $O(N)$, where N is the number of parameters to be estimated. However, its convergence speed is significantly affected by the eigenvalue spread of the input autocorrelation matrix. Alternative algorithms for improving the convergence speed were proposed [4,5]. A fast transversal algorithm for RLS AVF was also proposed by Lee and Mathews [6], which has a fast convergence speed and a computational complexity of $O(M^3)$, where M-1 represents the memory order of the VF. A numerically more stable RLS AVF based on QR-decomposition was proposed by Syed and Mathews [7] with a complexity of $O(M^3)$. Other AVF algorithms include the recent work by Glentis *et al.* [8].

In this paper, we present a new AVF based on the transform domain approximate QR-LS algorithm (TA-QR-LS). It has an arithmetic complexity of $O(N)$. (For second order VF, the complexity is $O(M^2)$.) Simulation results show that it has a much faster convergence speed and better numerical stability than the LMS AVF. Although the convergence speed and steady state error are slightly inferior to the more complex RLS algorithm, the TA-QR-LS AVF has a considerably better tracking performance than the RLS-based algorithms. The paper is organized as follows: the new TA-QR-LS AVF algorithm is described in Section II. Experimental results and comparison to other algorithms are given in section III. Finally, conclusions are drawn in Section IV.

II. TA-QR-LS 2nd ORDER AVF

The objective of the adaptive Volterra filtering algorithm is to determine the filter coefficients of the VF given the input $x(n)$ and output $d(n)$ (the desired signal) of the nonlinear system to be identified. The estimation error at time instant n is thus given by

$$e(n) = d(n) - \mathbf{x}_N^T(n)\boldsymbol{\theta}_n \quad (6)$$

If the least squares criterion is used as a performance measure, the following time-averaged squared magnitude error $\xi(n)$ is minimized:

$$\xi(n) = \sum_{j=0}^{n} \lambda^{n-j}|e(j)|^2 = \sum_{j=0}^{n} \lambda^{n-j}\left|d(j) - \mathbf{x}_N^T(j)\boldsymbol{\theta}_n\right|^2 \quad (7)$$

where the constant λ is the forgetting factor with a value between 0 and 1. (6) can be written in matrix form as

$$\boldsymbol{e}(n) = \boldsymbol{d}(n) - \boldsymbol{X}_N(n)\boldsymbol{\theta}_n \quad (8)$$

where
$$\boldsymbol{d}(n) = [d(0),d(1),\ldots,d(n)]^T, \quad (9)$$

$$\boldsymbol{X}_N(n) = [\boldsymbol{x}_N(0),\boldsymbol{x}_N(1),\ldots,\boldsymbol{x}_N(n)]^T \quad (10)$$

$$\boldsymbol{x}_N(n) = [x_1(n),x_2(n),\ldots,x_N(n)]^T.$$

$\boldsymbol{X}_N(n)$ and $\boldsymbol{x}_N(n)$ are respectively the data matrix and the received signal vector. Then, the least squares objective function $\xi(n)$ in (7) becomes

0-7803-7997-7/03/$17.00 © 2003 IEEE

$$\xi(n) = e^H(n)W^2(n)e(n) = \| W(n)e(n) \|^2, \quad (11)$$

where $W(n) = diag(\sqrt{\lambda^n}, \sqrt{\lambda^{n-1}}, ..., \sqrt{\lambda}, 1)$. Like the conventional adaptive transversal filter, the least squares problem can be solved by performing the QR decomposition (QRD) on the matrix $\mathbf{W}(n)\mathbf{X}_N(n)$. Then

$$Q(n)W(n)e(n) = \begin{bmatrix} \hat{d}_N(n) - \Re_N(n)\theta_n \\ c_{n+1-N} \end{bmatrix}, \quad (12)$$

where $Q(n)$ is a $(n+1)\times(n+1)$ unitary matrix and $\Re_N(n)$ is an $(N \times N)$ upper triangular matrix, and $\mathbf{Q}(n)\mathbf{W}(n)\mathbf{d}(n) = [\hat{\mathbf{d}}_N^T(n) \quad \mathbf{c}_{n+1-N}^T]^T$. Since $Q(n)$ is a unitary matrix, the square of the Euclidean norm on the left hand side in (12) is equal to $\xi(n)$ while the norm on the right hand side of (12) reaches its minimum value when θ_n is chosen as

$$\Re_N(n)\theta_n = \hat{d}_N(n). \quad (13)$$

To derive a recursive LS algorithm based on the QRD, let

$$D_N(n-1) = W(n-1)[X_N(n-1) \quad d_N(n-1)]$$

be the augmented data matrix. Suppose that the QRD of $W(n-1)X_N(n-1)$ has been computed. Then, we have:

$$D_N^*(n-1) = Q(n-1)D_N(n-1) = \begin{bmatrix} \Re_N(n-1) & \hat{d}_N(n-1) \\ 0 & c_{n-N} \end{bmatrix}.$$

Given the new data vector $\psi_n = [x_N^T(n), d(n)]^T$, then we have

$$D_N(n) = W(n)[X_N(n) \quad d(n)] = \begin{bmatrix} \sqrt{\lambda}D_N(n-1) \\ \psi_n^T \end{bmatrix}. \quad (14)$$

Premultiply (14) by the following augmented matrix

$$Q'(n) = \begin{bmatrix} Q(n-1) & 0 \\ 0^T & 1 \end{bmatrix},$$

one gets $Q'(n)D_N(n) = \begin{bmatrix} \sqrt{\lambda}\Re_N(n-1) & \sqrt{\lambda}\hat{d}_N(n-1) \\ 0 & \sqrt{\lambda}c_{n-N} \\ x_N^T(n) & d(n) \end{bmatrix}$. (15)

where $\Re_N(n-1)$ is an upper triangular matrix. The new QRD can be computed by zeroing out $x_N^T(n)$ by a series of transformations such as Householder reflections or Givens rotations. Let $Q^{(i)}(n)$ be the transformation used to zero out the element $x_i(n)$ at the i-th stage. After performing N stages of transformations, the first N elements in the last row of (15) become zero

$$Q^{(N)}(n)\cdots Q^{(1)}(n)Q'(n)D_N(n) = \begin{bmatrix} \Re_N(n) & \hat{d}_N(n) \\ 0 & c_{n-N} \\ 0^T & d^{N+1}(n) \end{bmatrix}.$$

Although it is possible to recursively update the triangular matrix with $O(N)$ arithmetic operations, direct back solving of the optimal parameter vector θ^o still requires about $O(N^2)$ arithmetic complexity. Here, we employ an improved approximate QR-LS (A-QR-LS) algorithm proposed by Liu [9]. The A-QR-LS algorithm combines the updating and the back solving processes together using the Householder transformation and a special structure of the upper triangular matrix. This yields a very efficient algorithm requiring N square roots, $17N$ multiplications and $9N$ additions. The basic idea is to approximate the upper triangular matrix $\Re(n-1)$ by a diagonal matrix D_{n-1}. This yields the following equation for solving the new estimate $\hat{\theta}_n$:

$$\Phi_n\theta_n = b_n, \quad (16)$$

where

$$\Phi_n = \begin{bmatrix} wD_{n-1} \\ x_n^T \end{bmatrix}, \; b_n = \begin{bmatrix} wD_{n-1}\hat{\theta}_{n-1} \\ d(n) \end{bmatrix},$$

$$D_{n-1} = diag\{r_{1,1}(n-1), \cdots, r_{N,N}(n-1)\}.$$

(16) can be solved by computing the QRD of Φ_n using say the Householder transformation. However, because the matrix D_{n-1} in Φ_n is a diagonal matrix, the system can be solved using the QRD in order $O(N)$ arithmetic complexity. More precisely, we can construct the appended matrix $D_N(n) = [\Phi_n, b_n]$ as follows [9]

$$D_N(n) = \begin{bmatrix} wr_{1,1}(n-1) & & & & -ws_1(n-1) \\ & wr_{2,2}(n-1) & & O & -ws_2(n-1) \\ & O & \ddots & & \vdots \\ & & & wr_{N,N}(n-1) & -ws_N(n-1) \\ x_1(n) & x_2(n) & \cdots & x_N(n) & d(n) \end{bmatrix} \quad (17)$$

By zeroing out the elements $x_i(n)$ successively from i=1 to N using the Householder transformation, we obtain an upper triangular matrix, which can be solved using back substitution for the parameter θ_n at the *n-th* iteration. This yields the algorithm for solving θ_n in [9]. In fact, it can be shown in [10] that this algorithm is actually a variable step size LMS algorithm with individual normalization of the elements in the input signal vector. Thus, its convergence speed is sensitive to the eigenvalue spread of the input autocorrelation matrix. To remedy this problem, we propose to transform the input signal vector by some unitary or orthogonal matrix to approximately decorrelate the input signal vector before applying the A-QR-LS algorithm. This considerably improves the convergence speed. Further, the algorithm converges in the mean if $0 < w < 1$. Moreover, we find that Givens rotation is more efficient than the Householder transformation when signal vectors are processed one at a time. Besides, Givens rotations enable us to develop a "square-root free" version of the resulting Givens-based $O(N)$ approximate QR-LS algorithm, similar to the classical work in [11]. Consequently, the complexity of our algorithm can be greatly reduced. Let's consider the triangularzation of the following $(N+1)\times(N+1)$ matrix F having the same structure as in (17):

$$F = \begin{bmatrix} f_{1,1} & & & & f_{1,N+1} \\ & f_{2,2} & & O & f_{2,N+1} \\ & O & \ddots & & \vdots \\ & & & f_{N,N} & f_{N,N+1} \\ f_{N+1,1} & f_{N+1,2} & \cdots & f_{N+1,N} & f_{N+1,N+1} \end{bmatrix}$$

Owing to the special structure of matrix F, only two rows, the *i-th* row and the $(N+1)$-th row, are changed when applying Givens rotation to it during the *i-th* iteration, i.e.

$$f_{i,j}^{(k)} = \begin{cases} f_{i,j} & \text{if } k \le i \\ f_{i,j}^* & \text{otherwise} \end{cases}, \quad (18)$$

where $f^*_{i,j}$ denotes the element of the upper triangular matrix transformed from matrix $\boldsymbol{F}$. At the i-th stage, a Givens rotation matrix is applied to the i-th and the last rows to zero out the element $f_{N+1,i}$. Since its preceding coefficients in the first i columns are all zero, we only need to perform N-i+1 such rotations. The operation on the i-th and the last rows is illustrated as follows

$$\begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 & f_{i,i} & 0 & \cdots & 0 & f_{i,N+1} \\ 0 & \cdots & 0 & f^{(i)}_{N+1,i} & f^{(i)}_{N+1,i+1} & \cdots & f^{(i)}_{N+1,N} & f^{(i)}_{N+1,N+1} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \cdots & 0 & f^*_{i,i} & f^*_{i,i+1} & \cdots & f^*_{i,N} & f^*_{i,N+1} \\ 0 & \cdots & 0 & 0 & f^{(i+1)}_{N+1,i+1} & \cdots & f^{(i+1)}_{N+1,N} & f^{(i+1)}_{N+1,N+1} \end{bmatrix}. \quad (19)$$

where

$$c_i = \frac{f_{i,i}}{\sqrt{f_{i,i}^2 + (f^{(i)}_{N+1,i})^2}} = f_{i,i} / f^*_{i,i}\ ;\ s_i = \frac{f^{(i)}_{N+1,i}}{\sqrt{f_{i,i}^2 + (f^{(i)}_{N+1,i})^2}} = f^{(i)}_{N+1,i} / f^*_{i,i}\ ;$$

$$f^*_{i,i} = \sqrt{f_{i,i}^2 + (f^{(i)}_{N+1,i})^2}\ ;$$

$$f^*_{i,i+j} = s_i f^{(i)}_{N+1,i+j}\ ;\ f^{(i+1)}_{N+1,i+j} = c_i f^{(i)}_{N+1,i+j}\ ; j=1,\ldots,N\text{-}i;$$

$$f^*_{i,N+1} = c_i f_{i,N+1} + s_i f^{(i)}_{N+1,N+1}\ ;\ f^{(i+1)}_{N+1,N+1}(n) = c_i f^{(i)}_{N+1,N+1} - s_i f_{i,N+1}\ .$$

Then, the following approximate QR-LS algorithm based on Givens rotations is obtained:

Upper Triangular Algorithm (20)

$\pi_0 = 1$
For $i = 1,2,\cdots,N$ *Loop*
  $\alpha_i = \sqrt{(f_{i,i})^2 + (\pi_{i-1} f_{N+1,i})^2}$
  $c = f_{i,i}/\alpha_i$, $\rho = -\pi_{i-1} f_{N+1,i}/\alpha_i$
  $\pi_i = \pi_{i-1} c$, $\delta_i = -\rho\pi_{i-1}$
  $f^*_{i,i} = \alpha_i$
  $f^*_{i,N+1} = c f_{i,N+1} - \rho f^{(i)}_{N+1,N+1}$
  $f^{(i+1)}_{N+1,N+1} = \rho f_{i,N+1} + c f^{(i)}_{N+1,N+1}$
  For $j = i+1, i+2,\cdots,N$ *Loop*
    $f^*_{i,j} = \delta_i f_{N+1,j}$
  End of Loop j
End of Loop i

Backsolving Algorithm

$\gamma_N = 0, s_N = -f^*_{N,N+1}$
$\theta_n(N) = s_N / f^*_{N,N}$
For $i = N-1, N-2,\cdots,1$ *Loop*
  $\gamma_i = \gamma_{i+1} + f_{N+1,i+1}\theta_n(i+1)$
  $s_i = -f^*_{i,N+1} - \delta_i \gamma_i$
  $\theta_n(i) = s_i / f^*_{i,i}$
End of Loop i

where $\gamma_i = \sum_{j=i+1}^{N} f_{N+1,j}\theta_n(j)$. In the TA-QR-LS algorithm, the input signal vector is transformed by the discrete cosine transform (DCT) to decorrelate the input signal. The resulting algorithm has a computational complexity of $O(N) + C(DCT(N))$, where N is the number of parameters to be estimated, and $C(DCT(N))$ is the complexity of the DCT, which depends on the transform length. If N is a power of two, then $C(DCT(N))$ is of the order $O(N \log_e N)$.

III SIMULATION RESULTS

In this section, the performance of the proposed TA-QR-LS-based second-order adaptive Volterra filter is evaluated by computer simulations. It is compared with the Fast RLS algorithm in [6] which has an arithmetic complexity of $O(3M^3 + 16.5M^2)$. All the following results are obtained by averaging over 100 independent runs.

A. Stationary environment

The second-order Volterra filter to be identified is the one used in [6] and is shown as follows:

$$y(n) = -0.78x(n) - 1.48x(n-1) + 1.39x(n-2) + 0.04x(n-3) + 0.54x^2(n)$$
$$+ 3.72x(n)x(n-1) + 1.86x(n)x(n-2) - 0.76x(n)x(n-3) - 1.62x^2(n-1)$$
$$+ 0.76x(n-1)x(n-2) - 0.12x(n-1)x(n-3) + 1.41x^2(n-2)$$
$$- 1.52x(n-2)x(n-3) - 0.13x^2(n-3) + v(n),$$

where $x(n)$, $y(n)$, and $v(n)$ represent respectively the system input, output and measurement noise. The norms of the linear and quadratic coefficient error vectors in [6] are used as the performance measure.

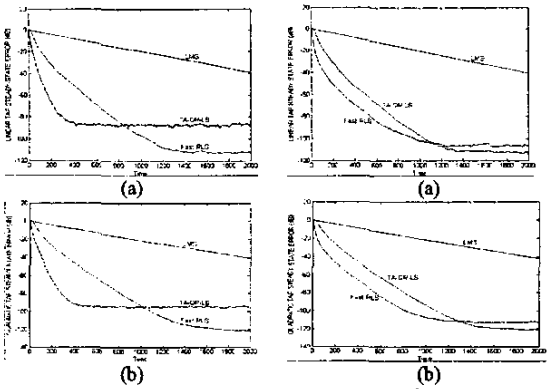
(1) **White input**: $x(n)$ is generated as a zero mean white Gaussian process with unit variance. The measurement noise is also a zero mean white Gaussian noise process with a smaller variance of σ^2=0.01. The simulation results are shown in Fig. 1. The TA-QR-LS-based algorithm and the Fast RLS proposed in [6] have a much faster convergence speed than the LMS-based algorithm, with the former slightly faster than the other. On the other hand, although both the TA-QR-LS and Fast RLS algorithms possess satisfactory steady-state error, the Fast RLS has a smaller steady state error than that of the TA-QR-LS, which is to be expected. The steady-state error of the TA-QR-LS algorithm can be decreased by increasing the forgetting factor w. Fig.2 shows the simulation result when w is increased from 0.95 to 0.99. Therefore, there is trade-off between convergence speed and steady state error in the TA-QR-LS algorithm.

(2) **Colored input**: $x(n)$ is obtained by processing a zero mean white Gaussian series with variance 0.05 through a linear time-invariant filter with impulse response as given in [6]: $\boldsymbol{h}$ = [0.9045; 1; 0.9045]^T. The results are shown in Figures 3 and 4. Though the TA-QR-LS algorithm has a faster initial convergence speed, the Fast RLS has a superior steady-state performance than the TA-QR-LS algorithm and the LMS algorithm, especially for coefficients of quadratic kernels. Figures 5 and 6 show the performance of the fast RLS algorithm for different values of forgetting factor λ from 0.9955 to 0.98, 0.97, and so. It can be seen that the TA-QR-LS algorithm with a forgetting factor w equal to 0.99 has a similar steady-state error as the Fast RLS algorithm with a forgetting factor value of λ = 0.98 to 0.97.

B. Tracking ability

The first coefficient $a(k)$ of the system model used in the previous experiment is chosen to be a time-varying sinusoid:
$a(k) = 2\sin(2\pi fk)$, with f=0.01, 0.001, and 0.0001Hz. Simulation results for colored input having the same setting in A(2) are presented in Figures7 to 10. It can be seen that the TA-QR-LS algorithm performs significantly better than the Fast RLS algorithm for all the frequencies f of the sinusoids being tested. It can also be observed that the performance of the Fast RLS algorithm can be

fairly improved by decreasing the forgetting factor λ, at the expense of a higher steady-state-error, as demonstrated earlier.

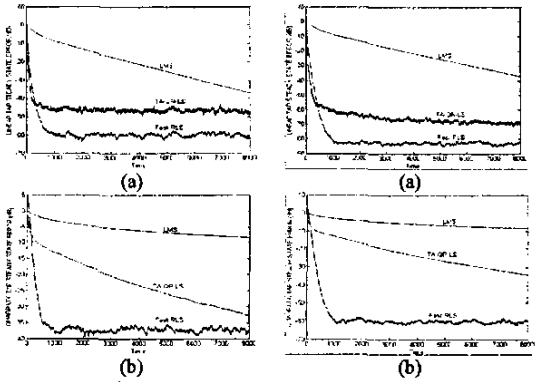


(a) (a)

(b) (b)

Fig. 1. Fig. 2.

Fig.1. White input. λ=0.9955; μ=0.001; w=0.95, $\sigma^2 = 0.01$. (a) $\|V_L(n)\|$; (b) $\|Vq(n)\|$. Fig.2. White input. λ=0.9955; μ=0.001; w=0.99; $\sigma^2 = 0.01$. (a) $\|V_L(n)\|$; (b) $\|Vq(n)\|$.

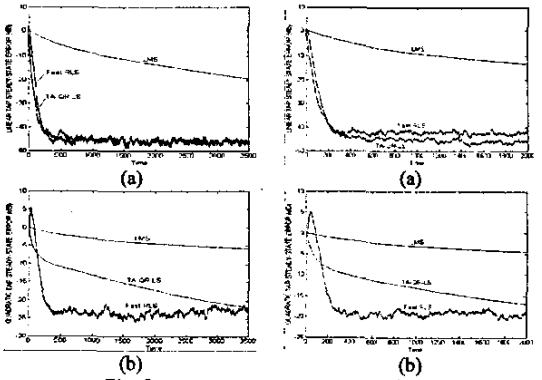


(a) (a)

(b) (b)

Fig. 3. Fig. 4.

Fig.3 Colored input. λ=0.9955; μ=0.01; w=0.99; SNR=10dB. (a) $\|V_L(n)\|$; (b) $\|Vq(n)\|$. Fig.4 Colored input. λ=0.9955; μ=0.01; w=0.99; SNR=20dB. (a) $\|V_L(n)\|$; (b) $\|Vq(n)\|$.



(a) (a)

(b) (b)

Fig. 5. Fig. 6.

Fig.5 Colored input. λ=0.98; μ=0.01; w=0.99; SNR=10dB. (a) $\|V_L(n)\|$; (b) $\|Vq(n)\|$. Fig.6. Colored input. λ=0.97; μ=0.01; w=0.99; SNR=10dB. (a) $\|V_L(n)\|$; (b) $\|Vq(n)\|$.

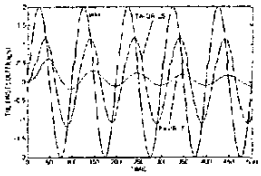


Fig. 7

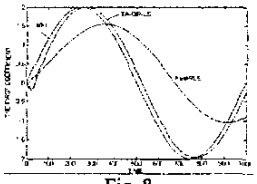


Fig. 8

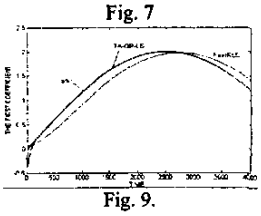


Fig. 9.

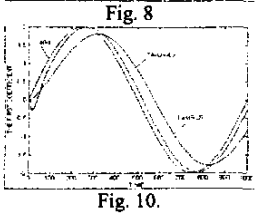


Fig. 10.

Fig.7 λ=0.9955; w=0.95; $f = 0.0001$. Fig.8 λ=0.9955; w=0.95; $f = 0.001$. Fig.9 λ=0.9955; w=0.95; $f = 0.01$. Fig.10 λ=0.98; w=0.95; $f = 0.001$.

IV CONCLUSION

A new transform-domain approximate QR-LS-based (TA-QR-LS) adaptive volterra filtering algorithm is presented. It improves the approximate QR-based LS (A-QR-LS) algorithm by introducing a unitary transformation for decorrelating the input signal vector to achieve better convergence and tracking performance. Further, the Givens rotation is used instead of the Householder transformation to reduce the arithmetic complexity. Simulation results show that the proposed algorithm has much better initial convergence and steady state performance than the LMS algorithm. The fast RLS AVF algorithm exhibits superior steady state performance when the forgetting factor is chosen to be 0.995, but the tracking performance of the TA-QR-LS algorithm was found to be considerably better.

REFERENCES

[1] W. J. Rugh, *Nonlinear System Theory. The Volterra-Wiener Approach*, Johns Hopkins University Press, Baltimore, Maryland, 1981.

[2] M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems*, John Wiley & sons, Inc., New York, 1980.

[3] T. Koh, and E. J. Powers, "Second-order Volterra filtering and its application to nonlinear system identification", *IEEE Trans. Acoust. Speech and Signal Proc.*, Vol. ASSP-33, pp. 1445-1455, Dec. 1985.

[4] Beaufays. F. : "Transform-domain adaptive filters: an analytical approach", *IEEE Trans. Signal Processing*, SP-43, pp. 422-431, 1995.

[5] D. W. Griffith, "Partially decoupled Volterra filters: formulation and LMS adaptation", *IEEE Trans. Signal Processing*, Vol. 45, No.6, 1485-1494, June, 1997.

[6] J. Lee and V. J. Mathews, "A fast recursive least squares adaptive second order Volterra filter and its performance analysis," *IEEE Trans. Signal Processing*, vol. 41, pp. 1087-1102, Mar. 1993.

[7] M. Syed and V. J. Mathews, "QR-decomposition based algorithms for adaptive Volterra filering," *IEEE Trans. Circuits Syst.* II, vol. 41, pp. 202-214, Mar. 1994.

[8] G. A. Glentis, P. Koukoulas and N. Kalouptsidis, "Efficient Algorithms for Volterra System Identification", *IEEE Trans. Signal Processing*, vol. 47, No. 11, pp. 3042-3057, Nov. 1999.

[9] Z. S. Liu, "QR methods of O(*N*) complexity in adaptive parameter estimation," *IEEE Trans. Signal Processing*, vol. 43, pp. 720-729. 1995.

[10] S. C. Chan and X. X. Yang, "Improved approximate QR-LS algorithms for adaptive filtering," accepted for publication in *IEEE Trans. CAS-II.*

[11] W. M. Gentleman and H. T. Kung, " Matrix triangularization by systolic array," in *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 298, pp. 298-303, 1981.