Second-order Volterra Filter based on DFP Technique

Shulin Bai1, Jiadong Xu1 and Yumei Zhang2

1School of Electronics and Information, Northwestern Polytechnical University, Xi’an, Shaanxi, 710072, China
2Key Laboratory of Modern Teaching Technology and School of Computer Science, Shaanxi Normal University, Xi’an, Shaanxi, 710062, China

Abstract. There have some problems caused by selection of improper parameters when LMS, NLMS or RLS algorithms are used to evaluate coefficients of a second-order Volterra filter. We propose a Davidon-Fletcher-Powell-based second-order Volterra filter (DFP-SOVF). The proposed filter is based on a posteriori error assumption and has a variable convergence factor. We give the recursive inverse auto-correlation matrix formulation and present computational complexity analysis. Then applying the proposed DFP-SOVF filter to single step predictions for pure Lorenz chaotic series, prediction results show that the DFP-SOVF filter can guarantee its convergence and stability and there have not divergence problems when using LMS and NLMS algorithms.

1. Introduction

Volterra filter has been widely studied in system identification, echo cancellation and nonlinear time series prediction [1-9]. This is because Volterra filter can usually employ both linear and nonlinear factors. References [1-2] give predictions of chaotic series using an adaptive Volterra filter. Reference [3] presents a SOVF filter using a modified NLMS algorithm and gives its echo cancellation application in communication systems. References [4-8] present parameter identification applications for original Volterra filter or some approximate Volterra filter structures by using LMS, NLMS or RLS adaptive algorithms. However, convergence speed of LMS algorithm has relationship with correlation property [3] of the input signals, and be dependent with the input signal spectrum [5, 6, 12] which is determined by the condition number of the inverse auto-correlation matrix of the input signal. RLS algorithm [6, 13] gives a new solution to overcome these problems. RLS algorithm has become an alternative approach for LMS algorithm, especially when there needs a rapid convergence speed. NLMS algorithm provides some improvements of convergence rate but requires more computational complexity. Also, there will always be problem of reasonable parameters selection when LMS, NLMS or RLS algorithms are used. Instability of the algorithm happens when improper parameter is selected. This phenomenon results in difficulty increment of Volterra application.

Based on the above statements, we present an adaptive Davidon-Fletcher-Powell-based second-order Volterra filter (DFP-SOVF), which uses a priori error assumption and has a variable convergence factor. We also give the recursive inverse auto-correlation matrix formulation and present computational complexity analysis. Then, we perform simulations of single step prediction for pure Lorenz chaotic series using the proposed filter. At last, we compare single step predictions results when using the proposed filter with LMS-SOVF filter and NLMS-SOVF filter, respectively.
2. Volterra filter

Reference [4] defines a truncated second-order Volterra filter

\[ \hat{y}(n) = h_0 + \sum_{i=0}^{n-1} h_i(n) x(n-i) + \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} h_{ij}(n) x(n-i) x(n-j), \]

where \( x(n) \) is input signal and \( \hat{y}(n) \) is prediction output signal at time instant \( n \). \( h_0 \) is a constant (here take \( h_0 = 0 \) for simplicity) and \( m \) is memory length of the Volterra filter. \( h_i(n) \) and \( h_{ij}(n) \) are linear and quadratic filter coefficients at time instant \( n \) to be updated. Let

\[ H(n-1) = [h_1(0; n-1), h_1(1; n-1), \ldots, h_1(m-1; n-1), h_2(0,0; n-1), h_2(0,1; n-1), \ldots, h_2(m-1(m-1); n-1)]^T, \]

\[ X(n) = [x(n), x(n-1), \ldots, x(n-(m-1)), x'(n), x(n-1), \ldots, x^2(n-(m-1))]^T, \]

where \([\cdot]^T \) denotes vector transpose. Equation (1) can be rewritten by the vector form

\[ \hat{y}(n) = H^T(n-1) X(n). \] (2)

Take \( e(n) \) and \( y(n) \) be a priori error signal and desired output, respectively. At time instant \( n \), square error can be represented

\[ e^2(n) = (y(n) - \hat{y}(n))^2 \] (3)

For the truncated Volterra filter given by (2), iterative equation for \( H(n) \) using classic LMS algorithm is

\[ H(n) = H(n-1) - \mu N \epsilon^2(n) \frac{\partial \epsilon^2(n)}{\partial H} = H(n-1) + 2 \mu \epsilon(n) X(n), \] (4)

where \( \mu \) represents a convergence factor that controls stability and convergence speed of the classic LMS algorithm. We must take \( 0 < \mu < \frac{1}{\lambda_{\text{max}}} \) to ensure algorithm convergence, where \( \lambda_{\text{max}} \) is the largest autocorrelation matrix eigenvalue [5, 6] of the input signal vector \( X(n) \). As a result, when the largest eigenvalue takes significant large, \( \mu \) needs take a relatively small value. This will eventually result in very slow convergence speed.

3. DFP-SOVF model

Newton method [14] is usually used to improve convergence speed in classic LMS algorithm. Newton method is performed by left multiplication of the inverse input signal auto-correlation matrix so as to change search direction. In this paper, DFP algorithm, which is a Quasi Newton method, is used to update the SOVF filter coefficient of in (4) and get

\[ H(n) = H(n-1) + 2 \mu \epsilon(n) \hat{R}^{-1}(n-1) X(n) \] (5)

where \( \hat{R}^{-1}(n-1) \) denotes inverse auto-correlation matrix estimate of \( X(n-1) \), namely,

\[ \hat{R}^{-1}(n-1) = (X(n-1)X^T(n-1))^{-1}. \] (6)

Also, consider appropriate convergence factor selection in LMS algorithm, and introduce a variable factor in the Volterra filter coefficient update in (5). Equation (5) can be rewritten as

\[ H(n) = H(n-1) + 2 \mu(n) \epsilon(n) \hat{R}^{-1}(n-1) X(n). \] (7a)

We then simplify notations of (7a)

\[ D_{n-1} = \hat{R}^{-1}(n-1), \] (7b)

\[ X_n = X(n), \] (7c)

\[ H_n = H(n), \] (7d)

\[ \mu_n = \mu(n), \] (7e)

\[ e_n = \epsilon(n). \] (7f)

Rewrite (7a) as
\[ H_n = H_{n-1} + 2\mu_n e_n D_{n-1} X_n. \]  

(8)

Recursive inverse auto-correlation matrix formulation using DFP algorithm [14] is

\[ D_n = D_{n-1} + \frac{p_{n-1} p_{n-1}^T \cdot D_{n-1} q_{n-1} q_{n-1}^T D_{n-1}}{q_{n-1}^T D_{n-1} q_{n-1}}, \]  

(9a)

where

\[ p_{n-1} = H_n - H_{n-1}, \]  

(9b)

\[ q_{n-1} = \hat{V}_n - \hat{V}_{n-1} = 2X_n X_n^T p_{n-1}. \]  

(9c)

Derivation of (9c) is given below,

\[ \hat{V}_n = -2e_{n+1} X_{n+1}. \]

However, \( e_{n+1} \) is unknown at the \( n \) iterative. A posteriori error is then introduced,

\[ e(n) = y_n - H_n^T X_n = y_n - X_n^T H_n. \]  

(9d)

Redefine \( \hat{V}_n = -2e(n) X_n \), and obtain

\[ q_{n-1} = \hat{V}_n - \hat{V}_{n-1} = -2e(n) X_n + 2e(n) X_n = -2((y_n - X_n^T H_n) - (y_n - X_n^T H_n)) X_n = 2X_n X_n^T p_{n-1}. \]

Substitute (9c) for (9a) and get the recursive estimate formulation of the inverse auto-correlation matrix

\[ D_n = D_{n-1} + \frac{p_{n-1} p_{n-1}^T}{2p_{n-1}^T X_n} \cdot \frac{D_{n-1} X_n X_n^T D_{n-1}}{X_n^T D_{n-1} X_n}. \]  

(10)

Derivation of the \( \mu_n \) is given below.

Take derivative of square error of (9d) with \( \mu_n \) and get

\[ \frac{\partial e^2(n)}{\partial \mu_n} = 2[y_n - H_n^T X_n] \cdot [-2X_n^T D_{n-1} X_n] \cdot e_n. \]  

(11)

Let (11) equal to 0 and obtain

\[ y_n - H_n^T X_n = 0. \]  

(12)

Substitute (8) for (12) and obtain

\[ y_n - H_n^T X_n = y_n - (H_{n-1} + 2\mu_n e_n D_{n-1}) X_n = e_n - 2\mu_n X_n^T D_{n-1} X_n = 0. \]  

(13)

From (13) obtain

\[ \mu_n = \frac{1}{2X_n^T D_{n-1} X_n}. \]  

(14)

Substitute (14) into (8) and obtain

\[ H_n = H_{n-1} + 2\mu_n e_n D_{n-1} X_n = H_{n-1} + \frac{e_n D_{n-1} X_n}{X_n^T D_{n-1} X_n}. \]  

(15)

From (15) get

\[ H_n - H_{n-1} = \frac{e_n D_{n-1} X_n}{X_n^T D_{n-1} X_n}. \]  

(16)

So, substitute (16) into (9b) and get

\[ p_{n-1} = H_n - H_{n-1} = \frac{e_n D_{n-1} X_n}{X_n^T D_{n-1} X_n}. \]  

(17)

Substitute (17) into (10) and get

\[ D_n = D_{n-1} + \frac{D_{n-1} X_n X_n^T D_{n-1}}{X_n^T D_{n-1} X_n} \cdot \left( \frac{1}{2X_n^T D_{n-1} X_n} - 1 \right) = D_{n-1} + \frac{D_{n-1} X_n X_n^T D_{n-1}}{X_n^T D_{n-1} X_n} (\mu_n - 1). \]  

(18)

4. Computational Complexity Analysis of the DFP-SOVF Filter
Table 1 gives computational complexity analysis of the proposed DFP-SOVF filter. Note that we here only consider multiplication and division operations, whereas don’t consider addition and subtraction operations. Here we give calculations example of equation (c) to show computational complexity. In equation (c), \( \hat{R}^{-1}(n-1)X(n) \) includes \( M^2 \) multiplication operations, where \( M \) is the \( X(n) \) dimension and \( M = m(m+3)/2 \). \( X^T(n)\hat{R}^{-1}(n-1)X(n) \) and \( \mu(n) = 1/(2X^T(n)\hat{R}^{-1}(n-1)X(n)) \) need \( M^2 + M \) and \( M^2 + M + 1 \) multiplication operations, respectively. Total calculations of (c) amount to \( M^2 + M + 1 \).

Computational complexity analysis from Table 1 illustrates that total calculation operations of the proposed DFP-SOVF filter is \( 2M^2 + 3M + 5 \). Substitute \( M \) with \( m(m+3)/2 \) and get \( (2m^4 + 12m^3 + 21m^2 + 9m + 20)/4 \), i.e., the DFP-SOVF filter computational complexity is \( O(m^4) \). This conclusion gives a result that the proposed filter has a considerable computational complexity with one when the SOVF filter uses RLS algorithm as an adaptive algorithm [6].

| Relational expression | Dimension | Amount of calculations |
|-----------------------|-----------|------------------------|
| (a) \( \hat{y}(n) = H^T(n-1)X(n) \) | 1 \times 1 | \( M \) |
| (b) \( e(n) = y(n) - \hat{y}(n) \) | 1 \times 1 | 0 |
| (c) \( \mu(n) = 1 \times 1 \) | \( 2X^T(n)\hat{R}^{-1}(n-1)X(n) \) | \( M^2 + M + 1 \) |
| (d) \( H(n) = H(n-1) + 2\mu(n)e(n)\hat{R}^{-1}(n-1)X(n)M \) \times 1 | 1 \times 1 | \( M + 2 \) |
| (e) \( \hat{R}^{-1}(n-1)X(n)X^T(n)\hat{R}^{-1}(n-1)(\mu(n)-1) \) | \( M \times M \) | \( M^2 + 2 \) |

5. Simulations

5.1 Prediction for Lorenz series

Lorenz chaotic series is obtained by directly iterating according to initial conditional values. For Lorenz series we take step size and initial condition as 0.01 and [-1 0 1], respectively. In experiment, we compute 3000 data points series employing fourth-order Runge-Kutta integration technique. The first 2000 data points are as training and the last 1000 data as test. Mathematical formulation of Lorenz chaotic series [2] is

\[
\begin{align*}
\dot{x} &= -\sigma(x-y), \\
y &= -xz + rx - y, \quad \sigma = 10, r = 34, b = 8/3, \\
\dot{z} &= xy - bz.
\end{align*}
\]

Figure 1.Single step prediction errors for \( x \) component of Lorenz system by using DFP-SOVF model at different memory lengths.
We perform several situations in this paper, in which memory length of the proposed filter is chosen from 3 to 8. We obtain Mean Square Error (MSE) is 0.033%, 0.12%, 0.20%, 0.32%, 0.59% and 0.27%, which shows that best prediction performance can be obtained using MSE as evaluation criterion when memory length is 3. Prediction errors for single step using DFP-SOVF to Lorenz chaotic $x$ component are shown in Figure 1, in which vertical axis is in logarithmic scale.

5.2 Comparison of DFP-SOVF with LMS-SOVF Algorithms

Prediction errors for single step using LMS algorithm [6] to the SOVF filter in Lorenz chaotic $x$ component are shown in Figure 2, in which $\mu$ is selected for 0.01, 0.05 and 0.1. This figure also shows prediction results of DFP algorithm. Note that memory length is 3. MSEs are 3.52%, 2.14%, 1.50% and 0.033%. We find in experiments that LMS-SOVF algorithm has divergence as $\mu$ is 0.5. Simulation results show that the proposed DFP-SOVF filter is evidently better than LMS-SOVF filter. Moreover, the problem of selecting convergence factor does not occur in the proposed filter.

![Figure 2. Prediction errors for single step in Lorenz $x$ component using LMS-SOVF and DFP-SOVF filters as memory length is 3, respectively.](image1)

![Figure 3. Prediction errors for single step in Lorenz $x$ component using NLMS-SOVF and DFP-SOVF filters as memory length is 3, respectively.](image2)

5.3 Comparison of DFP-SOVF with NLMS-SOVF Algorithms

Prediction errors for single step using NLMS algorithm [6] to the SOVF filter in Lorenz chaotic $x$ component are shown in Figure 3, in which $\mu$ is selected for 0.01, 0.05 and 0.1. This figure also shows prediction results of DFP algorithm. Note that memory length is 3. MSEs are 1.32%, 0.24%, 0.18% and 0.033%. When NLMS algorithm [2, 6] is employed, an auxiliary constant is 0.001. Results show that DFP-SOVF filter is evidently better than NLMS-SOVF filter. Moreover, the problem of how to select convergence factor does not occur in the proposed filter.

6. Conclusions

This paper studies a novel DFP-SOVF filter, which is based on a posteriori error assumption and has a variable convergence factor. We also present the recursive estimate formulation of inverse auto-correlation matrix for the proposed DFP-SOVF filter. Computational complexity analysis is also illustrated and show that DFP algorithm is comparable with RLS. Simulations, which employ DFP-SOVF to predictions of single step in Lorenz series, show better performance than LMS-SOVF and NLMS-SOVF algorithms. Also, DFP filter overcomes some parameter selection difficulties caused by LMS and NLMS and prevent divergence problems from inappropriate parameter selection. Meanwhile, DFP filter has no problem of selecting initial parameter and can guarantee convergence and stability.
Acknowledgments
This work reported in this paper was supported by the National Natural Science Foundation of China (Grant Nos. 11502133, 11172342, 11372167), the Key Science and Technology Innovation Team in Shaanxi Province, China (Grant No. 2014KTC-18).

References
[1] Sabry-Rizk, M., Zgallai, W. (2000) Novel Volterra predictor based on state-space equilibrium of nonlinear single or multi-fractal signals. Proceedings of SPIE, 4116: 322-333.
[2] Zhang, J. S., Xiao, X. C. (2001) A reduced parameter second-order Volterra filter with application to nonlinear adaptive prediction of chaotic time series. Acta Phys Sin., 50: 1248-1254.
[3] Guerin, A., Faucon, G., Le Bouquin-Jeannes, R. (2003) Nonlinear acoustic echo cancellation based on Volterra filters. IEEE Trans Speech Audio Proc., 11: 672-683.
[4] Sigrist, Z., Grivel, E., Alcoverro, B. (2012) Estimating second-order Volterra system parameters from noisy measurements based on an LMS variant or an errors-in-variables method. Signal Proc., 92: 1010-1020.
[5] Zhang, H. J., Han, C. Z. (2004) Simplified Volterra series identification method based on variable step size block least mean square filter. Journal of Xi’an Jiao Tong University, 38: 583-586.
[6] Mathews, V. J. (1991) Adaptive polynomial filters. IEEE Signal Proc Mag., 8: 10-26.
[7] Mathews, V. J. (1996) Adaptive Volterra filters using orthogonal structures. IEEE Signal Proc Let., 3: 307-309.
[8] Mathews, V. J. (1995) Orthogonalization of correlated Gaussian signals for Volterra system identification. IEEE Signal Proc Let., 2: 188-190.
[9] Taikang, N., Shiming, A. N. (1994) A comparison of linear and nonlinear modeling of the hippocampal EEG. Proceedings of IEEE EMBS, 1: 221-222.
[10] Lang, Z. Q., Billings, S. A., Yue, R. (2007) Output frequency response function of nonlinear Volterra systems. Automatica, 43: 805-816.
[11] Aguirre, L. A., Billings, S. A. (1995) Identification of models for chaotic systems from noisy data: implications for performance and nonlinear filtering. Physica D, 85: 239-258.
[12] Widrow, B., McCool, J. M., Larimore, M. G. (1976) Stationary and nonstationary learning characteristics of the LMS adaptive filter. Proc IEEE, 64: 1151-1162.
[13] Bazaraa, M. S., Sherali, H. D., Shetty, C. M. (2006) Nonlinear Programming: Theory and Algorithms, 3rd ed. Wiley, Hoboken.
[14] De Campos, M. L. R., Antoniou, A. (1997) A new quasi-Newton adaptive filtering algorithm. IEEE Trans Circuits and Systems-II, 44: 924-934.