IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 5, Ver. II (Sep. - Oct. 2016), PP 64-73
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Non Linear Prediction Based Speech Coder using FTRLS Algorithm

Mithun M S[1], Ancy S Anselam[2]
[1]*Dept. of Electronics and Communication Engineering MBCET, Trivandrum, India*
[2]*Dept. of Electronics and Communication Engineering MBCET, Trivandrum, India*

Abstract: *The purpose of this contribution is to present a new approach for the prediction of speech signals that is appropriate to speech coding. Linear predictive coding is probably the most frequently used technique in speech signal processing. Since its neglects nonlinear effects during the speech production process, some adaptive techniques are used to make the system nonlinear to perform better than linear techniques, enabling better performance in a number of speech processing applications. Also adaptive techniques with nonlinear prediction of speech based on truncatedVolterra series is used. The present contribution is the Conventional Recursive Least-Squares (CRLS) Algorithm as the adaptive technique used. Here the proposed adaptive technique is the fast transversal recursive least-squares (FTRLS) algorithms, which are very attractive due to their reduced computational complexity. The paper also examines the approach in relation to speech signal representation for coding and compression. The method outlined in this contribution offers significant advantages compared to the standard prediction methods. In this paper we implemented the speech coder G723.1 by using nonlinear prediction coefficients.*

Index Terms: *Nonlinear speech processing, Pitch period, Prediction, Volterra series, Adaptive Filters, RLS, least-squares.*

I. Introduction

Models based on linear prediction have been used for several years in various areas of speech signal processing, such as coding, synthesis, speakerand speech recognition. It is usually assumed that during the speech production a power source (i.e. lungs) initiates a sound and the environment (i.e. vocal tract) shapes or "filters"[1]. Linear prediction model assumes that the excitation source and the vocal tract act independently, which enables separation of the source and filter in the model (i.e. source-filter model). This neglects the interdependence of the excitation source and the vocal tract. So that this assumptions are not valid for all the cases. Therefore, adaptive techniques that account for the nonlinear generation process of the speech signal should outperform linear techniques, enabling better performance in a number of speech processing applications.

Many number of methods are deals with nonlinear speech prediction are reported in the literature. The Volterra series model is most widely used in nonlinear adaptive filtering. This paper deals with short term and long-term prediction (LTP) of speech based on truncattingVolterra series. The exponential increase in the number of coefficients and it results in the intrication in the algorithm for Volterra series [2], the model proposed here will be limited to the third order.

From the standpoint of performance, it is widely known that the Recursive LeastSquares (RLS) [3] algorithm offers fast convergence and good error performance. This surety makes the algorithm heavly useful for adaptive noise cancelation. Unfortunately, the computational power of devices today increases, so it difficult to utilize the algorithms like CRLS for real-time signal processing.This is high due to computational complexity of the usual RLS algorithm. The computational time of the algorithm scales with $O(M^2)$, where M is the filter order.
In this paper we examine the Fast Transversal RLS (FTRLS) filter. The filter is designed provide the least squares solution for adaptive filtering problem in a way that mulitiplies with O(M), which makes it a more viable candidate for real-time applications.The FTRLS algorithms can be derived by solving the forward and backward linear prediction problems.the FTRLS algorithms require only time-recursive equations. The FTRLS algorithm can also be considered a fast version of an algorithm to update the transversal filter for the solution of the RLS problem.

II. Fast-Transversal Recursive Least Squares

The Fast Transversal RLS (FT-RLS) filter is intentional to provide the solution to the filtering problem with performance with same as the standard recursive least-squares (RLS) algorithm like CRLS and LMS. Because when the input signal and/or the intented signal are nonstationary, the best values of the coefficients are time variant. That means the autocorrelation matrix and/or the cross-correlation vector are time variant. For example, typically in a system identification application the autocorrelation matrix is time invariant and the cross-

correlation matrix is time variant, because in this case the designer can choose the input signal. This is the main reason for the drawback of CRLS algorithm.

In addition, the FT-RLS filter contributes this solution with less computational trouble which scales linearly with the filter order. This makes it a very good solution in real-time noise cancellation applications. The FT-RLS development is based on the derivation of lattice-based least-squares filters, but has the structure of four transversal filters working together to compute update quantities reducing the computational complexity. The relations derived for the backward and forward prediction in the lattice-based algorithms [5]can be used to derive the FTRLS algorithms. The resulting algorithms have computational complexity of order M making them especially attractive for practical implementation.

When compared to thelattice-based algorithms, the computational difficulty of the FTRLS algorithms is lower due to the absence of order-updating equations. In particular, FTRLS algorithms typically require 7N to11N multiplications and divisions per output sample, as compared to 14N to 29N for the LRLS algorithms. Therefore, FTRLS algorithms are considered the fastest implementation solutions of the RLS problem than the CRLS .The four transversal filters [3] used for forming the update equations are:

1) *Forward Prediction*: The forward prediction transversal filter contributes the forward filter weights in such a manner that minimizes the prediction error (in the least squares ways) of the next input sample based on the previous input samples. This filter also finds the prediction error of estimated signal using both a priori and a posterior filter weights, in addition to the minimum weighted least squares error for this forward prediction.

2) *Backward Prediction*: The backward prediction transversal filter computes backward filter weights in such a manner that minimizes the prediction error (in the least squares sense) of the u(n−M) sample using the vector input $u_b(n) = [u(n)u(n-1)\cdots u(n-M+1)]^T$. This filter will also compute the successive prediction error of estimation using both a priori and a posterior filter weights, in addition to the minimum weighted least squares error for this backward predictions.

3) *Conversion Factor*: The gain computation transversal filter is used to recursively find a gain vector which is used to updating the forward, backward and joint process estimation filter weights. As such the filter also contributes recursive computation of the factors relating a priori and a posteriori error quantities.

4) *Joint-Process Estimation*: The joint-process estimation transversal filter computes filter weights in such a way that the error between the estimated signal and the intented input signal (d(n)) is minimized. It is the joint process estimation weights that are equivalent to filter weights in other adaptive filtering algorithms.

The structural diagram of the FT-RLS using the subcomponent transversal filters is shown in Figure 1.

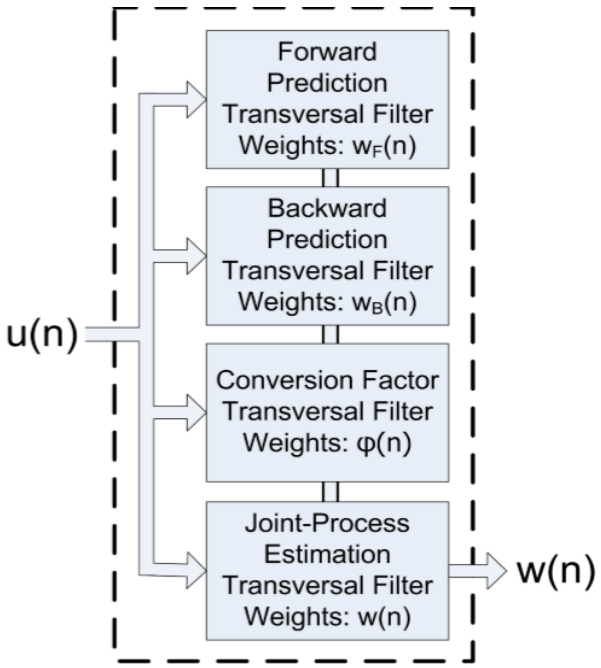


Fig. 1: Fast Transversal RLS Block Diagram

III. Non-Linear Prediction Using Volterra Series

The **Volterra series** is a model for non-linear behavior which is homogeneous to the Taylor series. It differs from the Taylor series in its capability to capture 'memory' effects. The Taylor series can be used for approximating the response of a nonlinear system to a given input if the output of the system depends upon strictly on the input at that particular time. In the Volterra series the output of the nonlinear system depends on the input to the system at all other times. The Volterra series is a model that is frequently used in system identification and approximation of weakly nonlinear systems. It can be regarded as a generalized Taylor series of a function with memory.

A discrete-time Volterra series with infinite memory has the expression [1]-[2]

$$y(n) = h_0 + \sum_{k=0}^{\infty} h_1(k) \cdot x(n-k) + \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} h_2(i,j) \cdot x(n-i) \cdot x(n-j) + \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\sum_{k=0}^{\infty} h_3(i,j,k) x(n-i) x(n-j) x(n-k) + \ldots \quad (1)$$

where x(n) is the input signal, y(n) is the output of the model, h_0 is the bias coefficient, $h_1$is the linear coefficients, h_2 the quadratic coefficients, h_3 the cubic coefficients etc.

The model is attractive since the expansion is a linear combination of nonlinear functions of the input signal. As a consequence, the adaptation algorithms valid in linear case can be extended to Volterra series. Both Least Mean Squares (LMS) and Recursive Least Squares (RLS) are used in practice to identify unknown parameters. In this paper we are using the Fast Transversal recursive least-squares (FTRLS) algorithm. Since the use of models based on Volterra series can lead to significant increase in complexity, so the number of model coefficients increases exponentially with the order of nonlinearities. That is the main reason why they are truncated to low nonlinearity orders (usually 2^{nd} or 3^{rd} orders).

The modeling of signals by Volterra series involve a minimization of the error, as in the method of least squares, and it requires excessive computational resources because the number of coefficients to be determined increases exponentially with the model's degree of nonlinearity and the Volterra filter length. This is the reason why the Volterra models are usually limited to low-order kernels[1]. A nonlinear predictor based on Volterra series estimates a current signal value by a linear combination of past signal values, and additionally by a linear combinations of products of past signal values. Hence, the predictor is nonlinear in the signal values, but linear in the filter coefficients. As a consequence, the adaptation algorithms valid in the linear case can be extended to Volterra filters. Without loss of generality, the system will be treated with the first and second-order kernels only. Then, the prediction error is given by[4]-[5]

$$\begin{aligned} e(n) &= x(n) - \hat{x}(n) \\ &= x(n) \\ &\quad - \left(\sum_{k=1}^{p} h_1(k) x(n-k) \right. \\ &\quad \left. + \sum_{i=1}^{p}\sum_{j=1}^{p} h_2(i,j) x(n-i) x(n-j) \right) \end{aligned} \quad (2)$$

IV. Second Order Volterra Short-Term Speech Predictiors

A nonlinear predictor based on Volterra series estimates a current signal value by a linear combination of past signal values, and additionally by a linear combinations of products of past signal values Without loss of generality the system is treated with the first and second order kernels only. Then, the predicted signal is given by

$$\hat{x}(n) = \sum_{k=1}^{p} h_1(k) \cdot x(n-k) + \sum_{i=1}^{m}\sum_{j=1}^{m} h_2(i,j) \cdot x(n-i) \cdot x(n-j) \quad (3)$$

where $\hat{x}$ (n) is the estimation of x(n). The nonlinear filter consists of a linear part of a prediction order p with coefficients $h_1(k)$ and a quadratic part of a prediction order m with coefficients $h_2(i,j)$. The symmetry of the coefficients is assumed, so $h_2(i,j)=h_2(j,k)$. In this case, the overall number of coefficients for the second order Voltera predictor equals

$$n_c = p + \frac{m \cdot (m+1)}{2} \tag{4}$$

The RLS algorithm, which is often used in linear case, can be easily adapted for the nonlinear second order Volterra model. Recursive Least Squares (RLS) are used in practice to identify unknown parameters.. The complete FTRLS algorithm can be represented in the following form [9]

Initialization

$$\mathbf{w}_f(-1,N) = \mathbf{w}_b(-1,N) = \mathbf{w}(-1,N) = \mathbf{0}$$
$$\hat{\boldsymbol{\phi}}(-1,N) = \mathbf{0},\ \gamma(-1,N) = 1$$
$$\xi^d_{b_{\min}}(-1,N) = \xi^d_{f_{\min}}(-1,N) = \epsilon \text{ (a small positive constant)}$$

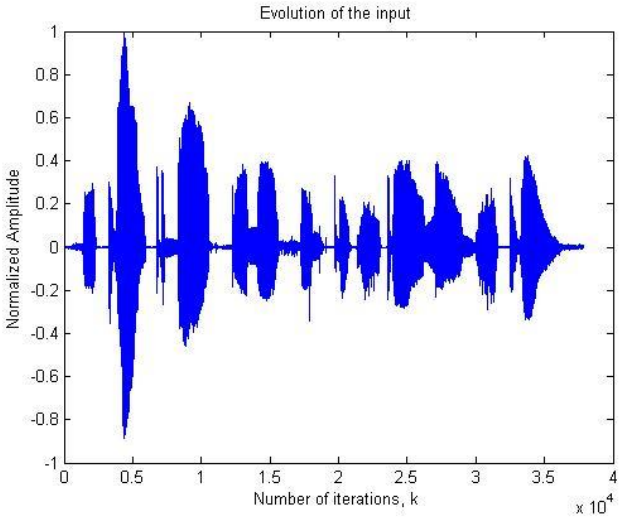
Prediction Part

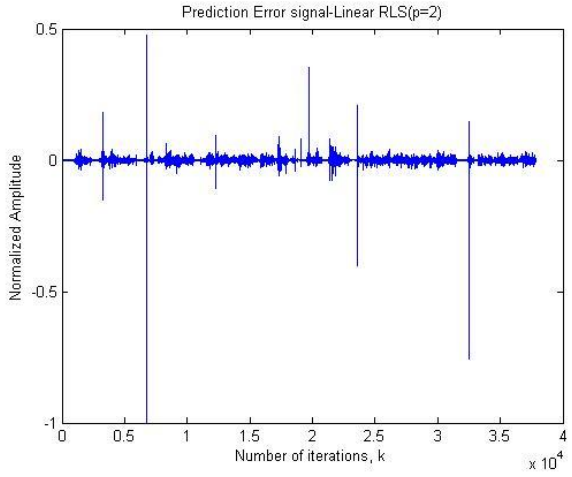
Do for each $k \geq 0$,

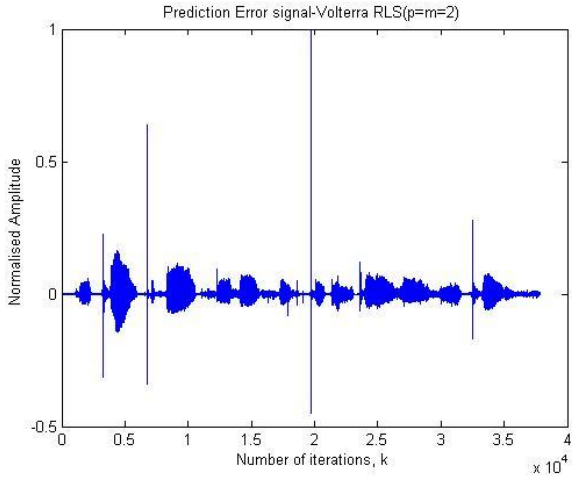
$$e_f(k,N) = \mathbf{x}^T(k,N+1)\begin{bmatrix} 1 \\ -\mathbf{w}_f(k-1,N) \end{bmatrix}$$
$$\varepsilon_f(k,N) = e_f(k,N)\gamma(k-1,N)$$
$$\xi^d_{f_{\min}}(k,N) = \lambda\xi^d_{f_{\min}}(k-1,N) + e_f(k,N)\varepsilon_f(k,N)$$
$$\mathbf{w}_f(k,N) = \mathbf{w}_f(k-1,N) + \hat{\boldsymbol{\phi}}(k-1,N)\varepsilon_f(k,N)$$
$$\hat{\boldsymbol{\phi}}(k,N+1) = \begin{bmatrix} 0 \\ \hat{\boldsymbol{\phi}}(k-1,N) \end{bmatrix} + \frac{1}{\lambda\xi^d_{f_{\min}}(k-1,N)}\begin{bmatrix} 1 \\ -\mathbf{w}_f(k-1,N) \end{bmatrix} e_f(k,N)$$
$$\gamma(k,N+1) = \frac{\lambda\xi^d_{f_{\min}}(k-1,N)}{\xi^d_{f_{\min}}(k,N)}\gamma(k-1,N)$$
$$e_b(k,N) = \lambda\xi^d_{b_{\min}}(k-1,N)\hat{\boldsymbol{\phi}}_{N+1}(k,N+1)$$
$$\gamma^{-1}(k,N) = \gamma^{-1}(k,N+1) - \hat{\boldsymbol{\phi}}_{N+1}(k,N+1)e_b(k,N)$$
$$\varepsilon_b(k,N) = e_b(k,N)\gamma(k,N)$$
$$\xi^d_{b_{\min}}(k,N) = \lambda\xi^d_{b_{\min}}(k-1,N) + \varepsilon_b(k,N)e_b(k,N)$$
$$\begin{bmatrix} \hat{\boldsymbol{\phi}}(k,N) \\ 0 \end{bmatrix} = \hat{\boldsymbol{\phi}}(k,N+1) - \hat{\boldsymbol{\phi}}_{N+1}(k,N+1)\begin{bmatrix} -\mathbf{w}_b(k-1,N) \\ 1 \end{bmatrix}$$
$$\mathbf{w}_b(k,N) = \mathbf{w}_b(k-1,N) + \hat{\boldsymbol{\phi}}(k,N)\varepsilon_b(k,N)$$

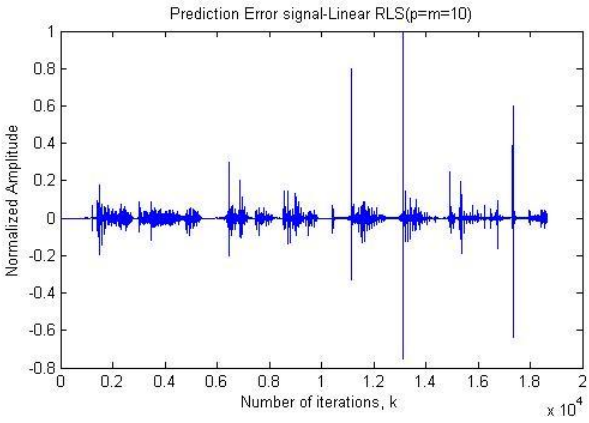
Joint-Process Estimation

$$e(k,N) = d(k) - \mathbf{w}^T(k-1,N)\mathbf{x}(k,N)$$
$$\varepsilon(k,N) = e(k,N)\gamma(k,N)$$
$$\mathbf{w}(k,N) = \mathbf{w}(k-1,N) + \hat{\boldsymbol{\phi}}(k,N)\varepsilon(k,N)$$









Type	Prediction gain [dB]	Number of coefficients
Linear FTRLS (p=2)	10.2	2
Non Linear FTRLS (p=m=2)	10.5	5
Non Linear FTRLS (p=m=10)	22.8	65
Non Linear FTRLS (p=10,m=4)	16.2	20

Table 1: Prediction Gains and Number Of Coefficients For Linear RLS And Different Volterra RLS Algorithms Using G723.1 Coder Using Ftrls Algorithm

V. Volterra Based Long-Term Speech Prediction

A. *Nonlinear Long-Term Linear Prediction*

Short-term linear predictor is able to eliminate only correlations between successive samples. It is clearly known that the prediction order must be high enough to include at least one pitch period, in order to model a voiced signal adequately. However, this is not acceptable for most practical implementations due to large delay and increased complexity. Hence, it is necessary to design long-term predictor (LTP) that is able to remove far-sample redundancies due to the presence of a pitch excitation. Contrary to the general belief that the pitch complexes are long-term dependent, and it is possible to reduce them using a short-term nonlinear predictor. The price was, as expected, increased complexity and number of coefficients. The standard solution to this problem in linear prediction is the use of a model with two linear predictors, short-term and long-term, connected in cascade.

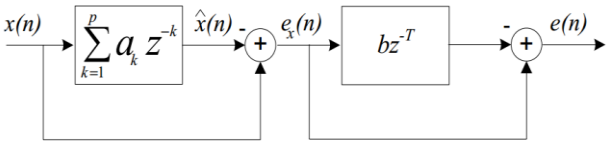


Fig 2: The cascade connection of short-term and long-term linear predictors

A long term linear predictor in such a realization targets correlation between samples one or multiple pitch periods apart, which is the result of periodic vibrations of vocal cords during the voiced speech generation process. This solution is used in a number of speech coders (CELP, RPE-LTP etc.). Long-term linear predictor has low prediction order (usually 1 to 3), hence the transfer function can be given by

$$H(z) = \frac{1}{1 - \sum_{k=0}^{p-1} b_k \cdot z^{-(T+k)}} \tag{5}$$

The most common case is the one-tap long-term predictor; then the optimal coefficient b (long-term gain or pitch gain) can be determined as

$$b = \frac{\sum_{n=1}^{M} e_x(n) \cdot e_x(n-T)}{\sum_{n=1}^{M} e_x{}^2(n-T)} \tag{6}$$

where e_x is the signal at the output of the short-term predictor and Tis the pitch period.

B. Second-Order Long-Term Volterra Prediction

Long-term prediction is an efficient scheme where correlation of the speech signal is modeled by two predictors. The short-term predictor is in charge of correlation between nearby samples, while the long-term predictor targets correlation between samples one or multiple pitch periods apart. It is well-known that the prediction order must be high enough to include at least one pitch period, in order to model a voiced signal adequately. The problem is evident when the prediction error is examined: a lack of fit is indicated by the remaining periodic component. In order to overcome the problem with the growing number of coefficients and eliminate pitch complexes, we propose the non linear long-term predictor based on the modified quadratic Volterra filter, that predicts the current signal sample from a past sample that is one or more pitch periods apart. The method can be defined as a one-tap predictor; that is, prediction is based on one single sample from the distant past.

The cascade connection of short-term linear predictor and long-term second-order Volterra predictor is shown in Fig 3.

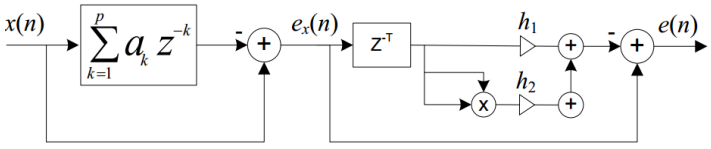


Fig.3: The cascade connection of short-term linear predictor and long-term second-order Volterra predictor

The corresponding predicted error signal $\hat{e}_x$ is determined as

$$\hat{e}_x(n) = h_1 \cdot e_x(n-T) + h_2 \cdot e_x{}^2(n-T) \quad (7)$$

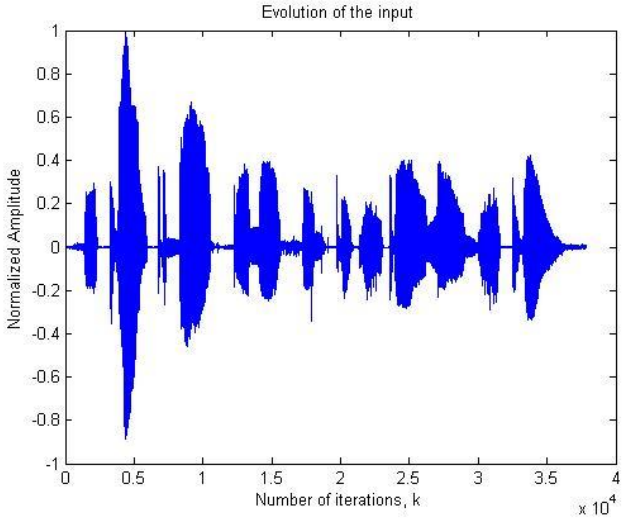
The adaptation algorithm starts with minimization of the criterion function

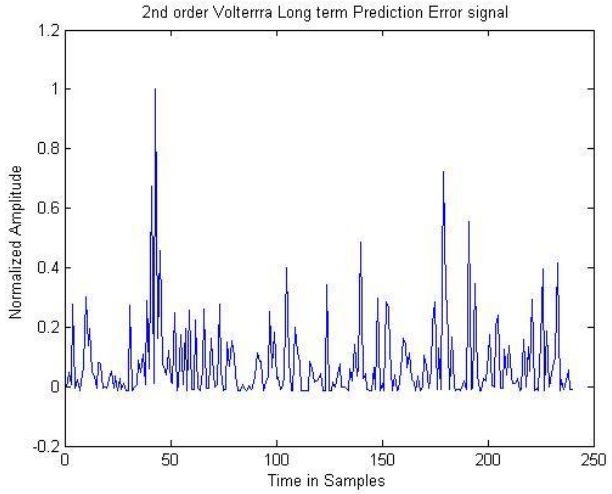
$$J(n) = \sum_n e^2(n) = \sum_n \left(e_x(n) - \hat{e}_x(n)\right)^2 \quad (8)$$

Differentiating (8) with respect to h_1 and h_2 and equating to zero one obtains optimal coefficients which minimize mean square error [1]

$$h_1 = \frac{\sum_n e_x{}^4(n-T) \cdot \sum_n e_x(n) \cdot e_x(n-T) - \sum_n e_x{}^3(n-T) \cdot \sum_n e_x(n) \cdot e_x{}^2(n-T)}{\sum_n e_x{}^4(n-T) \cdot \sum_n e_x{}^2(n-T) - \left(\sum_n e_x{}^3(n-T)\right)^2}$$

$$h_2 = \frac{\sum_n e_x{}^2(n-T) \cdot \sum_n e_x(n) \cdot e_x{}^2(n-T) - \sum_n e_x{}^3(n-T) \cdot \sum_n e_x(n) \cdot e_x(n-T)}{\sum_n e_x{}^4(n-T) \cdot \sum_n e_x{}^2(n-T) - \left(\sum_n e_x{}^3(n-T)\right)^2} \quad (9)$$





C. Third-Order Long-Term Volterra Prediction

Since the number of coefficients is not a critical issue in Volterra long-term prediction, the third-order predictor can be used as well in cascade with a short-term linear predictor, as shown in Fig 3.

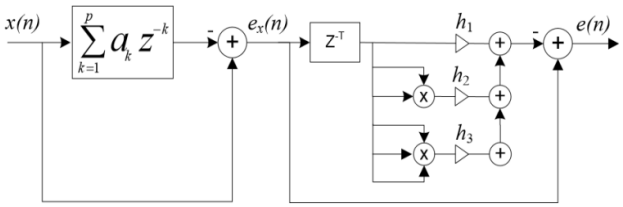


Fig 4: Short-term linear predictor connected in cascade with a long-term third-order Volterra predictor

In this case the number of coefficients is increased only by two compared to linear LTP. The corresponding predicted sample equals

$$\hat{e}_x(n) = h_1 \cdot e_x(n-T) + h_2 \cdot {e_x}^2(n-T) + h_3 \cdot {e_x}^3(n-T) \tag{10}$$

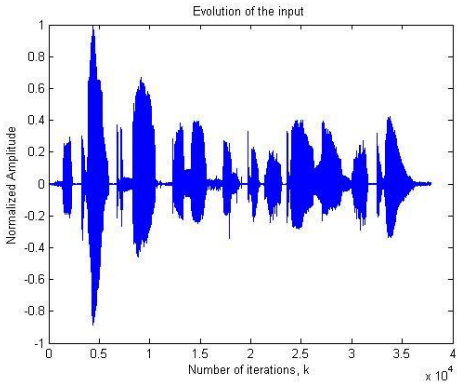
Minimizing the sum of squared errors J closed form expressions for the unknown coefficients are obtained

$$J = \sum_n \left(e_x(n) - \hat{e}_x(n)\right)^2 \tag{11}$$

$$h_1 = \frac{q_2 \cdot q_4 \cdot q_7 - q_3 \cdot q_4 \cdot q_5 + q_3^2 \cdot q_6 - q_3 \cdot q_7 \cdot q_8 - q_2 \cdot q_5 \cdot q_6 + q_5^2 \cdot q_8}{q_3^3 + q_2^2 \cdot q_7 + q_1 \cdot q_5^2 - q_1 \cdot q_3 \cdot q_7 - 2 \cdot q_2 \cdot q_3 \cdot q_5}$$

$$h_2 = \frac{q_1 \cdot q_5 \cdot q_6 - q_3 \cdot q_5 \cdot q_8 - q_2 \cdot q_3 \cdot q_6 + q_2 \cdot q_7 \cdot q_8 - q_1 \cdot q_4 \cdot q_7 + q_3^2 \cdot q_4}{q_3^3 + q_2^2 \cdot q_7 + q_1 \cdot q_5^2 - q_1 \cdot q_3 \cdot q_7 - 2 \cdot q_2 \cdot q_3 \cdot q_5}$$

$$h_3 = \frac{q_1 \cdot q_4 \cdot q_5 - q_2 \cdot q_3 \cdot q_4 + q_2^2 \cdot q_6 - q_2 \cdot q_5 \cdot q_8 - q_1 \cdot q_3 \cdot q_6 + q_3^2 \cdot q_8}{q_3^3 + q_2^2 \cdot q_7 + q_1 \cdot q_5^2 - q_1 \cdot q_3 \cdot q_7 - 2 \cdot q_2 \cdot q_3 \cdot q_5} \tag{12}$$



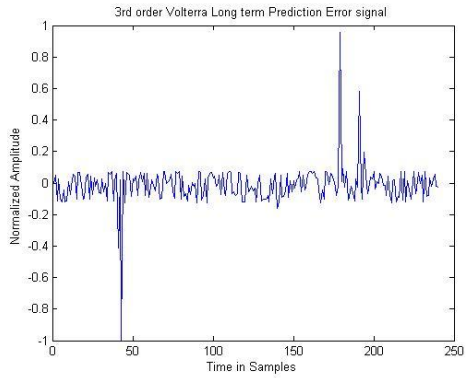
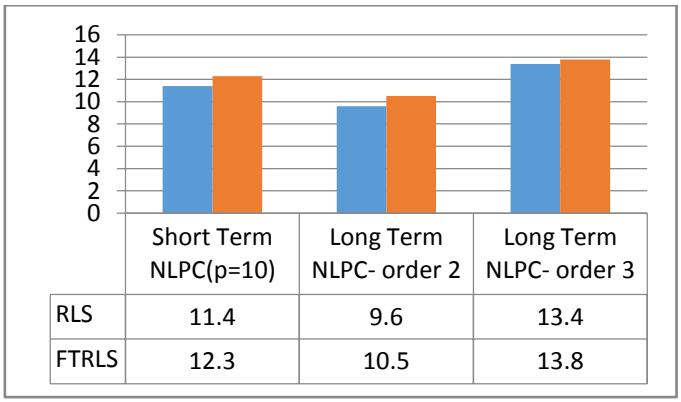


Table 2: Prediction Gains For A Nonlinear Short-Term Predictor Connected In Cascade With Linear Long-Term Predictor And Second-Order Volterra Long-Term Predictor, And Then Third-Oder Long Term Preictor Using Ftrls Algorithm.

Type	Prediction gain [dB]
Short Term NLPC(p=10)	12.3
Long Term NLPC- order 2	10.5
Long Term NLPC- order 3	13.8



	Short Term NLPC(p=10)	Long Term NLPC- order 2	Long Term NLPC- order 3
RLS	11.4	9.6	13.4
FTRLS	12.3	10.5	13.8

Table 3: Comparison of Prediction Gains In Non Linear Predictive Coder Using RLS And Ftrls Algorithm

VI. Nonlinear Speech Prediction With Frame/Subframe Structure

It can be shown that the effectiveness of the presented long-term predictor on removing long-term correlation is limited. The parameters of the long-term predictor need to be updated more frequently than the parameters of the short-term predictor [1]-[2]. That is, it loses its effectiveness when the time interval used for estimation becomes too long, which is due to the dynamic nature of the pitch period. It is known from the linear prediction theory that long term prediction, due to the dynamic nature of the pitch period, is inefficient when interval of adaptation is too long. The parameters of the long-term predictor need to be updated more frequently than the parameters of the short-term predictor. A model can be introduced where each interval of adaptation (frame) can be divided into several intervals of equal length known as subframes. While short term prediction is performed once per frame, long-term predictor coefficients are determined for each subframe within the frame [1].

The frame/subframe structure is used, where short-term linear prediction is applied to frames with the length of 240 samples. The frame is divided into four intervals of equal length, known as subframes. Second order Volterra LTP analysis is then applied to each subframe separately.

Language	**Filename**	**RLS (time in sec)**	**FT-RLS (time in sec)**
English	m_eng.wav	44.341	34.121
	f_eng.wav	40.783	34.553
	mf_conv_eng.wav	43.802	40.982
Kids	m_kid.wav	53.403	43.896
	f_kid.wav	42.985	33.522

Table 3: Time Consumption For Non Linear Predictive Coder Using Rls And Ftrls Algorithm

VII. Conclusion

The main aim of this work was to evaluate the time consuming performance of the nonlinear speech prediction using volterra series with FT-RLS algorithm.In this paper, an overview of the FT-RLS algorithm was provided. After through simulations, it is clear that the FT-RLS algorithm is a highly suitable solution for adaptive filtering applications where a large filter order is required without sacrificing the performance offered by the standard RLS algorithm. The closed form expressions for the second-order and the third-order long-term Volterra predictor coefficients are derived in the paper. Since the one-tap predictors are used; that is, prediction is based on one single sample from the distant past, number of coefficientsis only minimally increased compared to the linear LTP.

References

[1]. Vladimir Despotović and Zoran Perić, "Design of Nonlinear Predictors for Adaptive Predictive Coding of Speech Signals," in 21st Telecommunications forum TELFOR 2013 Serbia, Belgrade, November 26-28, 2013.

[2]. V. Despotovic, N. Goertz and Z. Peric, "Nonlinear long-term prediction of speech based on truncated Volterra series," IEEE Transactions on Audio, Speech and Language Processing, vol. 20, no. 3, pp. 1069-1073, 2012.

[3]. Dan J. Dechene, "Fast Transversal Recursive least-squares (FT-RLS) Algorithm," in Department of Electrical and Computer Engineering University of Western Ontario.

[4]. S. Haykin, Adaptive Filter Theory, 4th ed. Pearson Education, 2002. [2] P. S. R. Diniz, Adaptive Filtering: Algorithms and Practical Implementation. Kluwer Academic Publishers, 1997.

[5]. V. Despotovic, N. Goertz and Z. Peric, "Low-Order VolterraLongTerm Predictors," in Proc. 10th ITG Symposium on Speech Communication, Braunschweig, Germany, Sept. 2012, pp. 26-28.

[6]. Vladimir Despotović, Norbert Görtz,and Zoran Perić, "Improved Non-Linear Long-Term Predictors based on Volterra Filters," Vienna University of Technology, Institute of Telecommunications, Gußhausstr. 25-29, 1040 Vienna

[7]. Schnell and A. Lacroix, "Estimation of Speech Features of Glottal Excitation by Nonlinear Prediction," in Proc. ISCA ITRW Non-Linear Speech Processing (NOLISP 2007), Paris, France, 2007, pp. 116-119

[8]. Gh. Alipoor and M. H. Savoji, "Speech Coding Using Non-linear Prediction Based on Volterra Series Expansion," in Proc. 11th International Conference Speech and Computer (SPECOM 2006), St. Petersburg, Russia, June 2006, pp. 367-370

[9]. "Adaptive_Filtering_Algorithms_and_Practical_Implementation" by Paulo S.R. Diniz Federal University of Rio de Janeiro Rio de Janeiro Brazil

[10]. G. Kubin, "Nonlinear processing of speech", in Speech coding and synthesis, Chapter 16, W. B. Kleijn & K. K. Paliwal, Ed.: Elsevier, 1995.

[11]. J.CioffiandT.Kailath, "Fast Recursive-Least Squares,TransversalFilters for Adaptive Filtering," IEEE Trans. Acoust., Speech, Signal Processing, vol. 32, pp. 304–337, 1984.

[12]. D. T. M. Slock and T. Kailath, "Numerically Stable Fast Transversal Filters for Recursive Least Squares Adaptive Filtering," IEEE Trans. on Signal Processing, vol. 39, pp. 92–113, Jan 1991.

[13]. J. Thyssen, H. Nielsen and S. Hansen, "Non-linear short-term prediction in speech coding," in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Adelaide, Australia, 1994, pp. 185-188.