CASCADE RECURSIVE LEAST SQUARES WITH SUBSECTION ADAPTATION FOR AR PARAMETER ESTIMATION

Gaguk Zakaria[1, 2] *and A. A. (Louis) Beex*[2]

[1] Hughes Network Systems, Germantown, MD 20876 USA
[2] Systems Group – DSP Research Laboratory
The Bradley Department of Electrical and Computer Engineering, VIRGINIA TECH
Blacksburg, VA 24061 USA

ABSTRACT

We propose the adaptive cascade recursive least squares (CRLS-SA) algorithm for the estimation of linear prediction, or AR model, coefficients. The CRLS-SA algorithm features low computational complexity since each section is adapted independently from the other sections. It is shown here that the CRLS-SA algorithm can yield AR coefficient estimates closer to the true values, for some known signals, than the widely used autocorrelation method. CRLS-SA converges faster to the true values of the model, which is critically important for estimation from short data records. While the computational effort of CRLS-SA is a factor of 3 to 4 higher than that for the autocorrelation method, the improvement in performance yields a viable alternative for a number of applications.

I. INTRODUCTION

Today's need for low bit rate speech coding is very high. There are several different technologies used to develop low bit rate speech coding. The analysis-by-synthesis approach is the most popular, and results in the code excited linear prediction (CELP) coder [8].

One of the most important parameter sets in low bit rate speech coding is the set of linear prediction (LP) coefficients, which represents an auto-regressive (AR) model of the vocal tract. The LP coefficients are then quantized and transmitted. Note that usually the LP coefficients are not transmitted directly, rather, they are converted to another parameter set, such as the line spectral pair (LSP).

The number of speech samples used in the computation of LP coefficients is in the range of 40 to 240 samples, representing a 5 to 30 ms speech segment sampled at 8 kHz [1, 4, 9]. The most popular technique for computing the LP coefficients is the autocorrelation method, because it always yields a minimum-phase model and can be computed efficiently using the Levinson algorithm. The question is how good a result the autocorrelation method yields, especially for the shorter records.

Unfortunately, the true LP coefficients of a speech signal are not known. Therefore, a reasonable way to evaluate the performance of LP coefficient estimators is to use known signals with spectral content resembling that of speech. We compare the autocorrelation method and the adaptive cascade recursive least squares algorithm with subsection adaptation (CRLS-SA), for estimating LP coefficients.

II. THE CRLS-SA ALGORITHM

The cascade adaptive filter algorithm requires finding the gradient of the error with respect to the coefficients of each section [2, 5]. The gradient of the *k*-th section, $\underline{\varphi}_k(n)$, is defined as:

$$\underline{\varphi}_k(n) = \frac{\delta e(n)}{\delta \underline{w}_k(n)} \tag{1}$$

where $e(n)$ is the prediction error filter output from the final section, and $\underline{w}_k(n)$ is the tap-weight vector of the *k*-th section of the prediction error filter in second-order-section cascade form. The gradient of the *k*-th section can be computed efficiently by filtering the output $e(n)$ with $W_k^{-1}(z)$ [5, 7], as shown in Figure 1 for an order of four.

Unlike in the direct form RLS algorithm, the autocorrelation matrix of the cascade RLS (CRLS) algorithm is defined as the autocorrelation of the gradient

$$\underline{\varphi}(n) = [\underline{\varphi}_1^T(n), \underline{\varphi}_2^T(n), \ldots, \underline{\varphi}_{N/2}^T(n)]^T \tag{2}$$

where *N* is the order of the filter and each section is a second-order filter. Hence, the CRLS-SA algorithm still needs to compute the inverse of *NxN* autocorrelation matrices.

For the linear prediction case, the computation of $\underline{\varphi}_k(n)$ amounts to computing the output after removing all poles except the *k*-th poles. For applications where the poles are well separated, the gradient of each section is

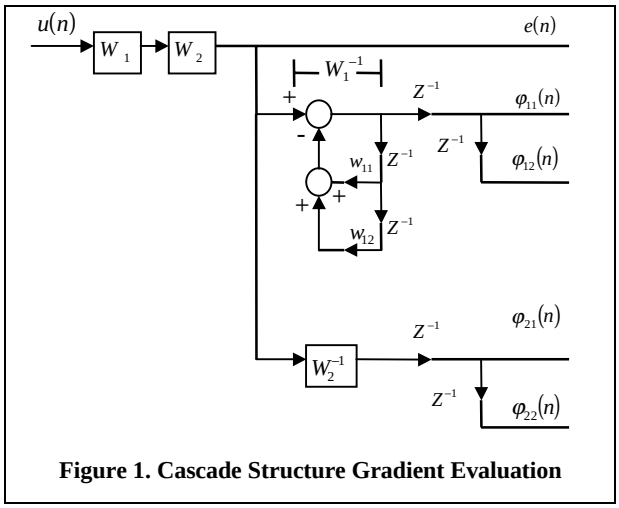

Figure 1. Cascade Structure Gradient Evaluation

nearly orthogonal to the gradient of the other sections because they correspond to different poles, or, in the frequency domain, the poles dominate in different frequency bands. As a result, the autocorrelation matrix *R* will be in the form of a nearly block-diagonal matrix, where the diagonal blocks R_k are 2*x*2 matrices representing the 2*x*2 autocorrelation matrix of the *k*-th section. Hence, we will assume that we can adapt the tap-weight of each section independently; that is, we need only compute the inverses of a number of 2*x*2 R_k matrices. We therefore termed the procedure the cascade RLS with subsection adaptation (CRLS-SA) algorithm. The tap-weights of the *k*-th section, $\underline{w}_k(n)$, are adapted as follows:

$$\underline{w}_k(n) = R_k^{-1}(n)\underline{p}_k(n) \tag{3}$$

where $\underline{p}_k(n)$ is the cross-correlation between the input and the desired signal. Equation (3) can be simplified using the *matrix-inversion lemma* [3]. This leads to the CRLS-SA algorithm which, for each section *k*, consists of the following steps:

$$\varphi_{k,1}(n) = W_k^{-1}(z)\{e_{N/2}(n)\} \tag{4.a}$$

$$\varphi_{k,2}(n) = \varphi_{k,1}(n-1) \tag{4.b}$$

$$\underline{\varphi}_k(n) = [\varphi_{k,1}(n)\varphi_{k,2}(n)]^T \tag{4.c}$$

$$\underline{\kappa}_k(n) = \frac{\lambda P_k(n-1)\underline{\varphi}_k(n)}{1+\lambda\underline{\varphi}_k^H(n)P_k(n-1)\underline{\varphi}_k(n)} \tag{5}$$

$$P_k(n) = \lambda^{-1}P_k(n-1) - \lambda^{-1}\underline{\kappa}_k(n)\underline{\varphi}_k(n)P_k(n-1) \tag{6}$$

$$e_k(n) = d_k(n) - \underline{\hat{w}}_k^H(n-1)\underline{u}_k(n) \tag{7}$$

$$\underline{\hat{w}}_k(n) = \underline{\hat{w}}_k(n-1) + \underline{\kappa}_k(n)e_k(n) \tag{8}$$

where $e_{N/2}(n)$ is the final output of the prediction error filter, $e_k(n)$ is the prediction error output of the *k*-th section, $\underline{u}_k(n)$ is the input to the *k*-th section, and $d_k(n)$ is the desired signal, which equals $u_k(n)$. The computational effort required by CRLS-SA is approximately 10**L***N*/2, where *L* is the length of the data record and *N* is the order of the filter.

III. AUTOCORRELATION METHOD

The autocorrelation method of AR estimation, or the *Yule-Walker* approach [6], solves the equation:

$$R\underline{w} = p \tag{9}$$

where *R* is an *NxN* biased autocorrelation matrix estimate, *p* is an *Nx*1 vector of autocorrelation lags from 1 to *N*, *N* is the order of the filter, and $\underline{w}$ is the vector of AR coefficients. Equation (9) can be solved efficiently using the Levinson recursion. For an *N*-th order filter and an *L* point data record, the autocorrelation method requires computational effort of about (*N*+1)**L* to compute the autocorrelation for lags zero through *N*, and $O(N^2)$ for the Levinson recursion.

IV. PERFORMANCE COMPARISON

In speech coding applications, the number of data points used to estimate the 10-th order LP coefficients is in the range of 40 to 160 samples. The speech segment is then considered to be wide-sense stationary over that range. To compare the performance of the widely used autocorrelation AR estimator and the CRLS-SA algorithm, we use two sets of 10-th order AR processes and estimate their parameters using these two approaches. The 10-th order AR processes are intended to model the formants of voiced speech as well as unvoiced speech. The lengths *L* of the data records used for simulation are 40, 80, and 160 samples. Only the data records of length 160 are used in the generation of the figures. The performance for all the data sets is shown in the tables.

Case I uses an AR process with poles not too close to the unit circle, as shown in Figure 2. Also shown are the estimated poles obtained from the autocorrelation method and the CRLS-SA algorithm. We see that most of the poles estimated by the CRLS-SA algorithm, denoted by 'o', are closer to the original poles, denoted by 'x', than the poles estimated by the autocorrelation method, denoted by '*'.

The spectral densities of the original process and the two estimates are shown in Figure 3. We note that the spectral estimate from the CRLS-SA algorithm is closer to

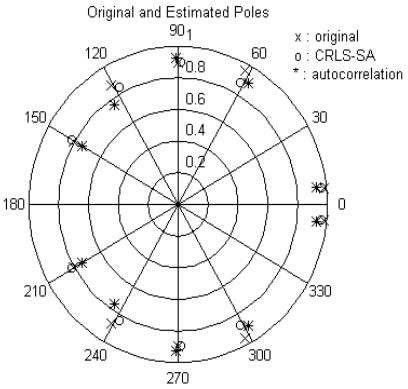

Figure 2. Original and Estimated Poles for Case I.

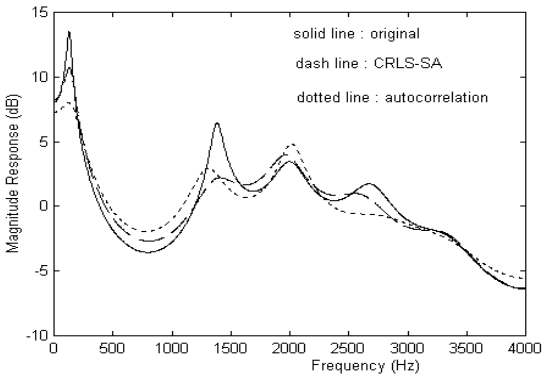

Figure 3. Spectral Densities for Case I.

the original spectrum than the spectrum estimated by the autocorrelation method. Furthermore, the distance from the coefficients estimated by the CRLS-SA algorithm to the original coefficients is much smaller than that same distance measure for the autocorrelation method's estimates, as shown in Table 1. The distance measure is defined as the base 10 logarithm of the sum of squared differences between the true and estimated LP coefficients.

Table 1. LP Coefficient Distance (in dB) for Case I.

# samples method	40	80	160
CRLS-SA	-6.30	-10.71	-11.43
Autocorrelation	-4.5	-6.79	-6.89

Case II uses a more narrowband process, for which the original and estimated poles are shown in Figure 4. The corresponding original and estimated spectral densities are shown in Figure 5. The distance measures between the estimated and original coefficients are shown in Table 2.

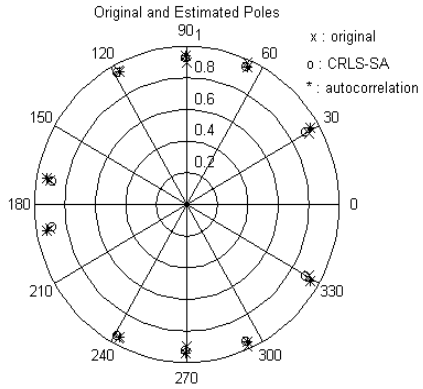

Figure 4. Original and Estimated Poles for Case II.

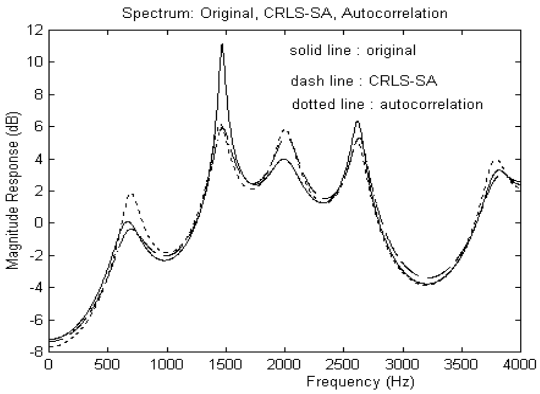

Figure 5. Spectral Densities for Case II.

Table 2. LP Coefficient Distance (in dB) for Case II.

# samples method	40	80	160
CRLS-SA	-5.44	-6.93	-16.05
Autocorrelation	-1.46	-4.72	-11.29

Again, we see that the CRLS-SA algorithm outperforms the autocorrelation method. We offer the following as likely explanation for these results. The autocorrelation method's AR estimate is based on the autocorrelation of the input signal. In practice, the true autocorrelation is unknown, so that it is estimated using the temporal autocorrelation under the assumption that the unobserved data equals zero. It turns out that for a short data record, these estimates are relatively poor, especially for narrowband processes [6].

An RLS-based adaptive algorithm on the other hand, the CRLS-SA algorithm in this case, makes no assumptions about the statistical properties of the input data. Instead, it tries to find a best fit to the input data by minimizing the sum of the squared errors.

To give an indication (not to be confused with an absolute measure) of the computational effort required by the CRLS-SA algorithm in comparison with that for the autocorrelation method, Table 3 shows the number of Matlab flops for the two methods, for each of the data lengths used in the experiments above. The system order *N*, was 10 in all cases.

Table 3. Computational Effort (in Matlab flops).

# samples method	40	80	160
CRLS-SA	6200	12400	24800
Autocorrelation	2119	3559	6439

Note that the computational effort for CRLS-SA is strictly linear with the data length *L*, while for the autocorrelation method that is not quite the case. The important result here is that CRLS-SA requires 3 to 4 times the computational effort of the autocorrelation method, for practical record lengths and orders.

V. CONCLUSION

The CRLS-SA algorithm has been proposed for the estimation of LP coefficients from short data records, in particular those data lengths used in speech coding. CRLS-SA produces better performance than the autocorrelation method, which is widely used in low bit-rate speech coding. Considering that the computational effort of the CRLS-SA algorithm is reasonably low, about 3-4 times that for the autocorrelation method, CRLS-SA can be considered a viable alternative for applications using short record lengths.

VI. REFERENCES

1. J. R. Deller Jr., J. G. Proakis, J. H. L. Hansen, Discrete-time Processing of Speech Signals, New York, Macmillan, 1993.
2. U. Forssen, *Analysis of Adaptive FIR Filters in Cascade Form*, IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing, vol. 41, no.6, pp. 392-401, June 1994.
3. S. Haykin, Adaptive Filter Theory, Englewood Cliffs, Prentice Hall, 1991.
4. ITU-T, Coding of Speech at 8 Kbps Using Conjugate-Structure Algebraic-Code-Excited Linear-Predictive (CS-ACELP) Coding, Draft Recommendation G.729, June 1995.
5. L. B. Jackson and S. L. Wood, *Linear Prediction in Cascade Form*, *IEEE Transactions on ASSP*, vol. ASSP-26, pp. 519-528, Dec. 1978.
6. S. M. Kay, Modern Spectral Estimation, Prentice Hall, 1988.
7. B. D. Rao, *Adaptive IIR Filtering Using Cascade Structures*, International Symposium on Circuits and Systems, pp. 194-197, 1993.
8. M. R. Schroeder and B. S. Atal, *Code-excited Linear Prediction (CELP): High-quality speech at very low bit rates*, ICASSP, pp. 25.1.1-25.1.4, 1985.
9. K. Swaminathan et. al, *A Robust Low Rate Voice Coding for Wireless Communications*, IEEE Workshop on Speech Coding, pp. 75-76, September 1997.