A STABLE AND EFFICIENT DSP IMPLEMENTATION OF A LSL ALGORITHM FOR ACOUSTIC ECHO CANCELLING

André H.C. Carezia, Phillip M.S. Burt, Max Gerken, Maria D. Miranda[†]*, Magno T.M. da Silva*

Laboratory of Communications and Signals
Dept. of Telecommunications and Control Engineering
Polytechnic School/University of São Paulo, Brazil
<http://www.lcs.poli.usp.br> {acarezia,phillip,mgk,maria,magno}@lcs.poli.usp.br
[†] Universidade Presbiteriana Mackenzie, Brazil
<http://www.mackenzie.com.br> mdm@mackenzie.com.br

ABSTRACT

In this paper we present an optimized DSP implementation of a modified error-feedback lattice least-square (EF-LSL) adaptive filtering algorithm. Simple measures that provide numerical stability for poor persistent excitation are also proposed. As a result of the optimization and the stability measures, an efficient and stable implementation of a fast algorithm of the RLS family was attained. We present the results of an acoustic echo cancelling experiment performed with the implemented algorithm. With a 40 MIPS SHARC DSP, up to 290 adaptive filter coefficients can be used. This represents an effective alternative to algorithms of the LMS family, while still retaining the good convergence properties of the RLS family.

1. INTRODUCTION

Acoustic Echo Cancellation (AEC) is a challenging application for adaptive filtering due to the required on-line processing through high order adaptive filters and to poor excitation characteristics of voice signals [1]. The convergence and consistent parameter estimation properties of Recursive Least Squares (RLS) algorithms make them interesting alternatives for this application. It is well known that in comparison with Least Mean Squares (LMS) algorithms the trade off for the improved performance of fast RLS algorithms is an increase in computational complexity and possible numerical instability. Moreover, the high order of the adaptive filter and poor persistent excitation make numerical instability show up more easily, stressing the importance of numerically stable algorithms. However, high implementation cost hinders the use of numerically stable fast RLS algorithms as the backward stable QR-LSL versions [2, 3].

In the described scenario, Least Squares Lattice (LSL) algorithms represent possible alternatives due to their known numerical robustness. Particularly, the *a priori* error feedback LSL algorithm (EF-LSL) proposed by Ling, Manolakis and Proakis [7] could be a good choice. It is certainly among the most numerically accurate LSL algorithms, comparing favorably with the *a priori* or *a posteriori* QR-LSL algorithms [3]. Though not sharing some theoretical properties with the QR-LSL algorithms that guarantee stable error propagation [2, 3], the *a priori* EF-LSL algorithm is computationally less complex and, as it will be shown, still presents numerically stable behavior when properly implemented.

In Section II we briefly present a modified error-feedback LSL algorithm based on the *a priori* error feedback algorithm [7, 8]. We also propose a simple numerical convention that ensures numerical stability for poor persistent excitations. Divisions may be implemented using denominators with reduced wordlength. Thus, lookup tables may be used to implement inversion of denominators, removing one of the reasons why LSL adaptive filters are usually not used in high-order real-time applications like AEC, i.e. the high computational load of divisions. These properties make the presented algorithm a good option for stringent high-order real-time adaptive filtering applications like AEC.

In Section III we present an efficient implementation of the modified EF-LSL algorithm for the floating point SHARC DSP. The implementation followed a thorough optimization procedure, which allowed the paralellism of this processor to be fully exploited. As a result, a 290 coefficient (for a 8 kHz sampling rate) EF-LSL acoustic echo canceller was made possible.

2. MODIFIED ERROR-FEEDBACK LSL ALGORITHM

The conventional *a priori* EF-LSL algorithm can be found for example in [8, p.633]. It deals with the following variables: ($\xi^f_m(n)$, $\eta_m(n)$, $\Gamma^f_m(n)$) and ($\xi^b_{m-1}(n)$, $\psi_m(n)$, $\Gamma^b_m(n)$) that represent respectively energies, *a priori* prediction errors and reflection coefficients of the forward and backward predictions, ($\gamma_m(n)$, $\alpha_m(n)$, $\kappa^d_m(n)$) that are respectively conversion factors, *a priori* estimation errors and regression coefficients and the forgetting factor λ. The subscript m indicates the corresponding order. In its original form, the computational complexity of this algorithm is $18M$ multiplications, $9M$ additions and $4M$ divisions, where M is the number of adaptive filter coefficients.

To reduce the computational complexity of this algorithm we introduced normalized *a posteriori* prediction errors that are related to the *a priori* prediction errors as follows [4]:

$$\begin{aligned}\overline{f}_{m-1}(n) &= f_{m-1}(n)/\xi^f_{m-1}(n) \\ &= \gamma_{m-1}(n-1)\eta_{m-1}(n)/\xi^f_{m-1}(n), \\ \overline{b}_{m-1}(n) &= b_{m-1}(n)/\xi^b_{m-1}(n) \\ &= \gamma_{m-1}(n)\psi_{m-1}(n)/\xi^b_{m-1}(n).\end{aligned}$$

This leads to the algorithm presented in Table 1 [6], which requires $13M$ multiplications, $9M$ additions and $2M$ divisions. It can be seen that equations are divided in four groups, (1, 2, 3, 4, 5), (6, 7, 8), (9, 10) and (11, 12, 13). Any equation in one given group uses only results from the preceding groups, which is a desirable feature when implementing the algorithm in a DSP. It is worth noting

that the error feedback mechanism is maintained in the modified algorithm, and therefore numerical accuracy is not affected.

Initialization ($n = 0$):	
For $m = 1$ to M do	
$\{\xi^f_{m-1}(n) = \xi^b_{m-1}(n) = \lambda\zeta;$	
$\Gamma^f_m(n) = \Gamma^b_m(n) = \kappa^d_m(n) = \psi_{m-1}(n) = 0\}$	
For $n = 1, 2, 3, ...$ do $\{$	
For $m = 0$ do $\{\gamma_m(n) = 1;$	
$\eta_m(n) = u(n); \psi_m(n) = u(n); \alpha_m(n) = d(n)\ \}$	
For $m = 1$ to M do { 1) to 13) below }}	
1)	$\eta_m(n) = \eta_{m-1}(n) - \Gamma^f_m(n\text{–}1)\psi_{m-1}(n\text{–}1)$ [j] [a] [b] [c]
2)	$f_{m-1}(n) = \gamma_{m-1}(n\text{–}1)\eta_{m-1}(n)$ [k] [d]
3)	$\psi_m(n) = \psi_{m-1}(n\text{–}1) - \Gamma^b_m(n\text{–}1)\eta_{m-1}(n)$ [l] [e]
4)	$b_{m-1}(n) = \gamma_{m-1}(n)\psi_{m-1}(n)$ [m] [f] [g]
5)	$\alpha_m(n) = \alpha_{m-1}(n) - \psi_{m-1}(n)\kappa^d_m(n\text{–}1)$ [n] [h] [i]
6)	$\Gamma^f_m(n) = \Gamma^f_m(n\text{–}1) + \bar{b}_{m-1}(n\text{–}1)\eta_m(n)$ [s] [o]
7)	$\xi^f_{m-1}(n) = \lambda\ \xi^f_{m-1}(n\text{–}1) + f_{m-1}(n)\eta_{m-1}(n) + \zeta$ [t] [p] [q]
8)	$\xi^b_{m-1}(n) = \lambda\xi^b_{m-1}(n\text{–}1) + b_{m-1}(n)\psi_{m-1}(n) + \zeta$ [u] [r]
9)	$\bar{f}_{m-1}(n) = f_{m-1}(n)/\xi^f_{m-1}(n)$ [v]
10)	$\bar{b}_{m-1}(n) = b_{m-1}(n)/\xi^b_{m-1}(n)$ [w]
11)	$\Gamma^b_m(n) = \Gamma^b_m(n\text{–}1) + \bar{f}_{m-1}(n)\psi_m(n)$ [x]
12)	$\kappa^d_m(n) = \kappa^d_m(n\text{–}1) + \bar{b}_{m-1}(n)\alpha_m(n)$ [y]
13)	$\gamma_m(n) = \gamma_{m-1}(n) - \bar{b}_{m-1}(n)b_{m-1}(n)$ [z]

Table 1. EF-LSL Algorithm variables with corresponding identificators.

To guarantee robust numerical behavior even for poor persistent excitation it is necessary to avoid divisions by values close to zero in Equations (9,10) of Table 1. To this end it is sufficient to add a positive constant ζ_o to the denominators of these equations. Alternatively, as shown in Table 1, a constant $\zeta = \zeta_o(1-\lambda)$ can be added to Equations (7,8) of Table 1. Considering that the input signal satisfies $-2^{k/2} \leq u(n) \leq 2^{k/2}$, with k as low as possible [5], extensive simulations have shown that $\zeta = 2^{k-b}(1-\lambda)$ is sufficient (though not strictly necessary) to assure numerical stability. Here, b is the mantissa wordlength to which the energies are quantized. This simple numerical convention was used with excellent results in several simulations. It was also verified that divisions may be implemented using denominators with reduced mantissa wordlength (for example, 8 bits), without producing instability.

Considering that numerical instability problems show up more easily when low forgetting factors are used, we performed acoustic echo cancellation experiments with more than one thousand coefficients and with forgetting factors as low as $\lambda = 0.1$. In these cases the algorithm remained numerically stable and did not break down, although it did not produce useful results due to the low λ values that lead to large estimation errors.

3. DSP IMPLEMENTATION

The EF-LSL algorithm was implemented in assembly language in a floating point Analog Devices 21061 DSP (SHARC). This DSP has sixteen 40-bit internal registers, which are used for paralell additions and multiplications. In the following, these registers are denoted by F0, F1,...., F15 (”F ” standing for ”floating point ”). For indirect memory adressing 16 pointers are used, which are denoted by I0, I1,..., I15. Two simultaneous transfers between registers and memory can be made, as long as one transfer uses a pointer in I0-I7 and the Data Memory (DM) bus and the other transfer uses a pointer in I8-I15 and the Program Memory (PM) bus. A reduced wordlength division can be carried out in two machine cycles: an 8-bit tabulated inversion and a multiplication.

3.1. Optimization

The particular organization of the steps of the modified EF-LSL algorithm presented in Table 1 is a suitable starting point for obtaining an efficient DSP implementation, as the steps within each group can be carried out independently. Performing then a careful allocation of the required operations to the processing resources, the 13 steps of the algorithm can be fitted, initially, into 19 machine cycles. The contents of the DSP registers before each of these 19 cycles are shown in Table 2. Table 3 shows the variables that are assigned to each pointer and Table 4 shows the transfers between memory and registers carried out in each cycle. The number of required machine cycles can still be reduced to 17 by transferring steps 18 and 19 in Tables 2 and 4 to the beginning of the following iteration and step 0 in Table 4 to the end of the previous iteration. This involves making adjustments in steps 12 and 13 in Tables 2 and 4. The core of the final algorithm is shown (in assembly language [9]) in Table 5. With the 40 MIPS SHARC DSP that was employed, it is possible to use up to $M = 290$ coefficients for a 8 kHz sampling rate.

3.2. Numerical aspects

Using the definitions of Section II, in the DSP implementation $k = 30$ (due to a 16-bit A/D) and $b = 8$ (the mantissa length to which the prediction energies are quantized). This would lead to $\zeta = 2^{22}(1-\lambda) \approx 2^{12}$ (for $\lambda = 0.999$). However, considering the high value of λ that is used, a smaller value of $\zeta = 2^7$ was adopted. As an extra safeguard against instability, the absolute value of the conversion factor γ_m was used in the implementation. For the SHARC DSP this did not represent any additional computational load. It was also necessary to saturate the prediction errors in order to avoid indefinite results (NaN), which is obtained activating the saturation mode in the DSP.

3.3. An acoustic echo cancellation experiment

Using a measured impulse response of 1024 samples (at a 8 kHz sampling rate) an echo signal was produced from a recorded voice signal. The first 512 samples of the measured impulse response are shown in Figure 1. The voice signal and the echo signal $d(n)$

Table 2: Contents of DSP registers previously to each cycle or result after each cycle

	Multiplier Unit						Accumulator Unit						
	X			Y			X			Y			
Cycle	f0	f1	f2	f4	f5	f6	f8	f9	f10	f12	f13	f14	f15
1	b			c									
2	a			d						bc			
3	a			ad=k	e		a			bc			
4	a		a-bc=j	k				c			ae		
5	f				g	c-ae=l	ak						ζ
6	fg=m	i			g		ak+ζ						
7	m				g			h				gi	
8		h-gi=n	j		o			gm	b				
9			p		q				b	jo			
10			p		r		ak+ζ		b+jo=s		pq		
11							ak+ζ+pq=t					pr	
12			inv(t)	k				gm				pr	
13			k/t=v			l		gm+pr=u					ζ
14				u			vl	u+ζ		e			
15	m			inv(u+ζ)			vl			e			
16		n		m/(u+ζ)=w			e+vl=x			i			
17	m			w				nw		i			
18								i+nw=y	f	mw			
19									f-mw=z				

Table 3: Assignment of pointers to variables.

Pointer	Vector size	Identifiers
I0	2(M+1)	c,l,g
I1	M+1	a,j
I2	1	h,n
I3	M	o,w
I4	1	p
I8	M	b,s
I9	2(M+1)	d,f,z
I10	M	e,x
I11	M	i,y
I14	M	q,t
I15	M	r,u

Table 4: Transfers between registers and memory

Cycle	DM bus	PM bus
0	(I0) → f4	(I8) → f0
1	(I1) → f0	(I9) → f4
2	(I1) → f8	(I10) → f5
3	(I0) → f9	
4	(I0) → f5	(I9) → f0
5	f2 → (I1)	(I11) → f1
6	(I2) → f9	
7	(I3) → f5	(I8) → f10
8	(I4) → f2	(I14) → f5
9	f6 → (I0)	(I15) → f5
10	f1 → (I2)	f10 → (I8)
11		f8 → (I14)
12		
13		(I10) → f12
14		f9 → (I15)
15		(I11) → f12
16	f4 → (I3)	f8 → (I10)
17		(I9) → f10
18		f9 → (I11)
19		f10 → (I9)

Table 5: Core of the assembly program. The values of the modifiers are:
$m0 = m8 = M + 1$, $m1 = m9 = M + 2$, $m5 = m13 = 0$, $m6 = m14 = 1$ and $m15 = -1$

1)	flt:	f12 = f0*f4,	f10 = f10-f12,	f0 = dm(i1,m5),	f4 = pm(i9,m8);
2)		f4 = f0*f4,	f10 = abs f10,	f8 = dm(i1,m6),	f5 = pm(i10,m13);
3)		f13 = f0*f5,	f2 = f8-f12,	f9 = dm(i0,m0),	pm(i9,m13) = f10;
4)		f8 = f0*f4,	f6 = f9-f13,	f5 = dm(i0,m6),	f0 = pm(i9,m13);
5)		f0 = f0*f5,	f8 = f8+f15,	dm(i1,m5) = f2,	f1 = pm(i11,m15);
6)		f14 = f1*f5,		f9 = dm(i2,m5),	pm(i11,m14) = f3;
7)		f9 = f0*f5,	f1 = f9-f14,	f5 = dm(i3,m5),	f10 = pm(i8,m13);
8)		f12 = f2*f5,		f2 = dm(i4,m5),	f5 = pm(i14,m13);
9)		f13 = f2*f5,	f10 = f10+f12,	dm(i0,m0) = f6,	f5 = pm(i15,m13);
10)		f14 = f2*f5,	f8 = f8+f13,	dm(i2,m5) = f1,	pm(i8,m14) = f10;
11)			f2 = recips f8,		pm(i14,m14) = f8;
12)		f2 = f2*f4,	f9 = f9+f14,		f10 = pm(i9,m9);
13)		f8 = f2*f6,	f9 = f9+f15,		f12 = pm(i10,m13);
14)			f4 = recips f9,		pm(i15,m14) = f9;
15)		f4 = f0*f4,	f8 = f8+f12,		f12 = pm(i11,m14);
16)		f9 = f1*f4,		dm(i3,m6) = f4,	pm(i10,m14) = f8;
17)	flt0:	f12 = f0*f4,	f3 = f9+f12,	f4 = dm(i0,m5),	f0 = pm(i8,m13);

were then fed to the DSP system and error $\alpha_M(n)$ was recorded. A forgetting factor $\lambda = 0.999$ was used. The instantaneous echo return loss enhancement (ERLE) given by

$$\text{ERLE} = 10\log[\overline{d^2(n)}/\overline{\alpha_M^2(n)}]$$

was calculated. Figure 2 shows 2s of the echo signal prior to cancellation and the obtained ERLE. Each ERLE value shown is the mean of 64 consecutive measurements. We observe that the ERLE range is approximately 30-40 dB, a good result considering the order of the echo canceller, the quantization of denominators of Equations (9,10) of Table 1 and the usual noise levels in hands-free mobile telephony environments. No noise was superposed to the echo signal in order to show the stable behaviour of the echo canceller. It can be seen that the algorithm remains stable, without the need for additional control measures, even when the voice signal decays to almost zero.

Figure 3 is similar to Figure 2 but considering an echo canceller with 1024 coefficients. In this case the described algorithm was implemented by a C routine embedded in MatLab and using the same numerical conventions as the DSP. It can be seen that the obtained ERLE range is improved due to the higher number of coefficients and that algorithm still works well. It should be observed that in this case the maximum attainable ERLE is limited due to mantissa's quantization before division. Similar results have been obtained for higher orders and longer echo impulse responses.

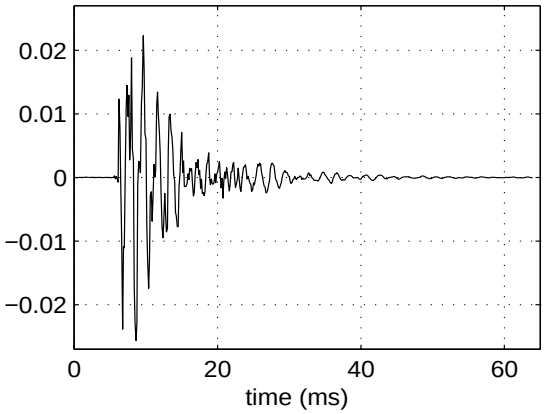

Fig. 1. Measured impulse response (512 samples).

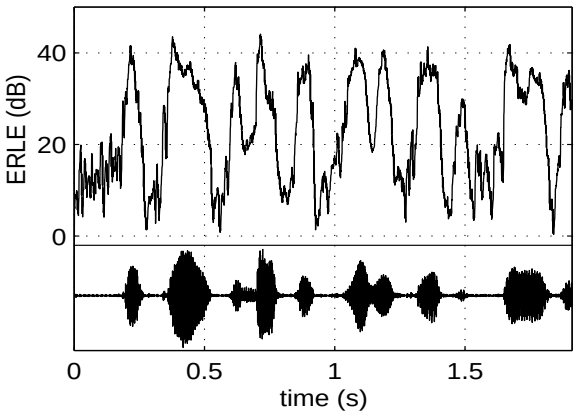

Fig. 2. Result of AEC experiment using ADSP-21061 (290 coefficients). Bottom trace: echo signal prior to cancellation.

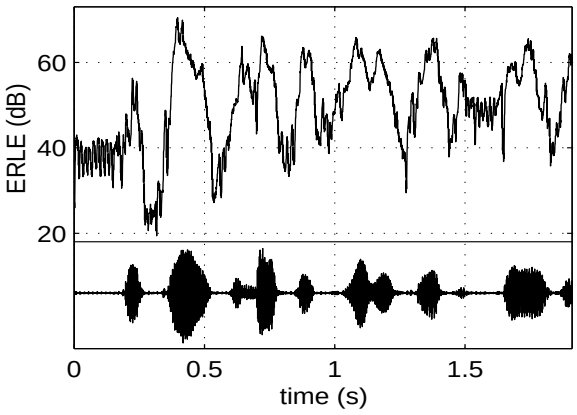

Fig. 3. Result of AEC experiment using MatLab (1024 coefficients). Bottom trace: echo signal prior to cancellation.

4. REFERENCES

[1] A. Gilloire, E. Moulines, D. Slock, P. Duhamel: "State of the Art in Acoustic Echo Cancellation", in A.R. Figueiras-Vidal (Ed.), *Digital Signal Processing in Telecommunications*, Springer, 1996, pp.45-91.

[2] P.A. Regalia: "Numerical properties of a QR-based fast least squares algorithm", *IEEE Trans. on Signal Processing*, June 1993, vol.41, no.6, pp.2096-2109.

[3] M.D. Miranda, M. Gerken: "Performance of the a priori and a posteriori QR-LSL algorithms in a limited precision environment", *Proceedings ICASSP97*, April 1997, Munique, pp.2337-2340.

[4] M. Miranda, M. Gerken: "Optimized implementation of the a priori LSL algorithm with error feedback" (in Portuguese), Proceedings of the *15th Brazilian Telecommunications Symposium*, Recife, 1997, pp. 280-284.

[5] M. Miranda, M. Gerken, M.T.M. da Silva: "Efficient implementation of error-feedback LSL algorithm", *Electronics Letters*, 1999, pp. 1308-1309.

[6] A. H. C. Carezia, M. Gerken, P. M. S. Burt, M. T. M. da Silva: "Efficient Implementation of a LSL Algorithm for Acoustic Echo Cancelling in Automobiles" (in Portuguese), Proceedings of the *17th Brazilian Telecommunications Symposium*, Vila Velha, 1999, pp. 372-377.

[7] F. Ling, D. Manolakis, J.G. Proakis: "Numerically robust least-squares lattice-ladder algorithms with direct updating of the reflection coefficients", *IEEE Trans. on ASSP*, August 1986, vol.34, no.4, pp.837-845.

[8] S. Haykin: *Adaptive filter theory*, Prentice Hall Int., 2nd Ed., 1991.

[9] ADSP-2106x (SHARC) User´s Manual, 2nd Ed., Analog Devices Inc., July 1996.