

AN ALGORITHM FOR FAST DIRECT CALCULATION OF QUANTIZED LSP PARAMETERS

S. Grassi, M. Ansoerge, F. Pellandini

Institute of Microtechnology
 University of Neuchâtel, Rue A.-L. Breguet 2, CH-2000 Neuchâtel, Switzerland
 Tel.: ++41 32 718 3432 Fax: ++41 32 718 3402 E-mail : sara.grassi@imt.unine.ch
<http://www-imt.unine.ch>

ABSTRACT

Line Spectrum Pairs (LSP) are commonly used in speech coding for spectral quantization. Usually, the LSPs are first calculated, and then quantized. When scalar quantization is used, computational saving can be achieved by direct calculation of the quantized LSPs, searching zero-crossings on the grid formed by the values of the quantization tables (quantized-domain search).

In a previous publication we have shown that by combining Kabal's LSP calculation method with a quantized-domain search on a 34-bit quantization table, the number of polynomial evaluations is reduced from 150 to 71.

In this paper we consider the use of a binary-tree quantized-domain search to further decrease the complexity. It is explained why this type of search cannot be combined with Kabal's method, and it is shown that Saoudi's LSP calculation method can be easily combined with a binary-tree quantized-domain search, obtaining a fast algorithm for direct quantized LSP calculation. The proposed algorithm reduces computational complexity and improves quantization performance.

1. INTRODUCTION

Speech coding finds application in portable devices such as digital cellular telephones, vocal pagers, and portable multimedia terminals and computers, which require low power consumption and small size. Optimization at the algorithmic level (algorithm choice and simplification) is the key for a low power implementation as it allows savings of orders of magnitude in power consumption. This paper proposes an efficient algorithm to reduce computational complexity in LSP calculation, which is a complex task found in most speech coders.

Line spectrum pair (LSP) representation of linear predictive coefficients (LPC) is widely used in the speech coding domain for spectral quantization and interpolation. In Section 2, the definition of LSP parameters is given and Kabal's method for LSP calculation is briefly explained. This method can be combined with a quantized-domain search to obtain a more efficient algorithm, named "Quantized-search Kabal" (see § 2.2).

To further reduce the complexity of "Quantized-search Kabal" algorithm, we have considered the use of a binary-tree search in the quantized domain. To do this, we would need a test to know with certitude if an LSP lies above or below an arbitrarily selected value of the quantization table, without calculating the actual LSP value. This test is not available in Kabal's LSP calculation method, as the search for a given LSP relies on the calculation of previous LSPs.

Different existing LSP calculation methods were studied and it was found that Saoudi's method, which is explained in Section 3, provides a simple test to know if a given LSP lies

above or below an arbitrarily selected value of the quantization table, and this without calculating the actual LSP value. Saoudi's method was then combined with a binary-tree quantized-domain search, obtaining the fast algorithm for direct quantized LSP calculation presented in this paper.

The proposed algorithm is explained in Section 4. Experimental evaluation and computational complexity comparison are also given in this section.

Finally, conclusions are drawn in Section 5.

2. LSP PARAMETERS

LSP representation of 10-th order LPC coefficients is used in nearly all narrowband speech coder standards, with bit rates of less than 16 kbps [1]. Hereafter, an LPC order of 10 is assumed.

Given the 10-th order LPC analysis filter:

$$A_{10}(z) = 1 + \sum_{k=1}^{10} a_k \cdot z^{-k} \quad (1)$$

The polynomials, $P'_{10}(z)$ and $Q'_{10}(z)$, are given by [2]:

$$\begin{aligned} P_{10}(z) &= A_{10}(z) + z^{-11} A_{10}(z^{-1}) = (1 + z^{-1}) \cdot P'_{10}(z) \\ Q_{10}(z) &= A_{10}(z) - z^{-11} A_{10}(z^{-1}) = (1 - z^{-1}) \cdot Q'_{10}(z) \end{aligned} \quad (2)$$

It can be proved that if the roots of $A_{10}(z)$ are inside the unit circle then the roots of $P'_{10}(z)$ and $Q'_{10}(z)$ lie on the unit circle and are interlaced [2]. Conversely, if the roots of $P'_{10}(z)$ and $Q'_{10}(z)$ lie on the unit circle and are interlaced, then the roots of $A_{10}(z)$ are inside the unit circle. This property is used to ensure stability of the LPC synthesis filter $H_{10}(z) = 1 / A_{10}(z)$ upon quantization.

Given that $P'_{10}(z)$ and $Q'_{10}(z)$ have real coefficients and that their roots lie on the unit circle, $P'_{10}(z)$ and $Q'_{10}(z)$ can be completely specified by the angular positions of their roots in the upper semicircle of the z -plane. These angles are the 10 LSP parameters, denoted as $\{\omega_i\}$. Due to the interlacing property:

$$0 < \omega_1 < \omega_2 < \dots < \omega_{10} < \pi \quad (3)$$

2.1 Kabal's Method for LSP Calculation

The most popular method for LSP calculation is Kabal's method [2]. The 5-th order polynomials $P'_{10}(x)$ and $Q'_{10}(x)$ are obtained by evaluating $P'_{10}(z)$ and $Q'_{10}(z)$ on the unit circle ($z = e^{j\omega}$), and using the mapping $x = \cos(\omega)$. The roots of $P'_{10}(x)$ and $Q'_{10}(x)$ are the LSPs in the "x-domain", denoted as x_i , with $x_i = \cos(\omega_i)$. Thus, from Equation (3):

$$+1 > x_1 > x_2 > \dots > x_{10} > -1 \quad (4)$$

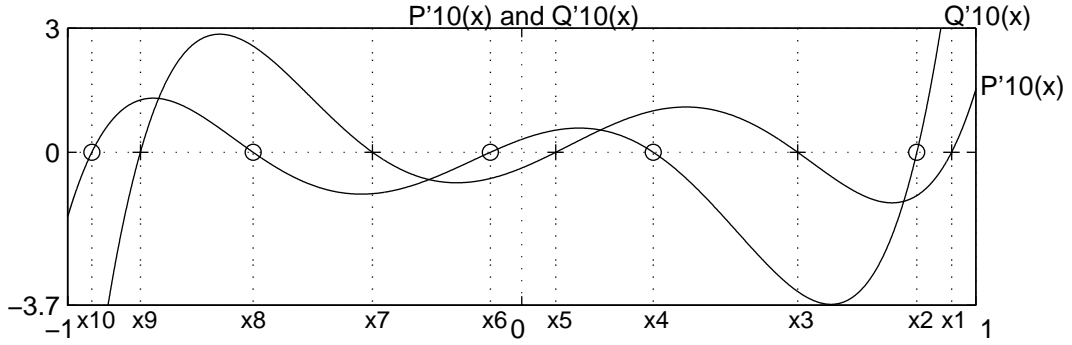


Figure 1: Behavior of the functions $P'_{10}(x)$ and $Q'_{10}(x)$ (x_1 to x_{10} are the LSPs in the “x-domain”, with $x = \cos(\omega)$).

An example of the behavior of the functions $P'_{10}(x)$, and $Q'_{10}(x)$ and their associated LSPs can be observed in Figure 1.

As $P'_{10}(x)$ and $Q'_{10}(x)$ are 5th-order polynomials, their zeros cannot be calculated in a closed form. In the numerical solution proposed by Kabal, the zero crossings are searched starting at $x = +1$, with decrements of $\Delta = 0.02$. Once a zero crossing is found, its position is refined by four successive bisections and a final linear interpolation. The search is done alternatively on $P'_{10}(x)$ and $Q'_{10}(x)$, starting from the position of the last LSP that was found. A maximum of 150 polynomial evaluations is needed. An efficient recursion for polynomial evaluation requiring only 4 multiplications and 9 additions is also proposed in [2].

Note that the calculation of each LSP relies on the calculation of previous LSPs. It can also be observed that for a given value of x , $\gamma \in (-1, +1)$, the evaluation of the polynomial $P'_{10}(x = \gamma)$ or $Q'_{10}(x = \gamma)$ does not give a clue on how many roots of the polynomial lie below or above γ . This precludes the use of Kabal’s method with quantized-domain binary-tree search (see § 4).

2.2 Quantized-search Kabal

In the CELP FS1016 coder [3], the LSPs are first calculated and then quantized using a 34-bit non-uniform scalar quantization. To speed up the calculation and quantization processes, we have proposed in [4] a modified version of Kabal’s algorithm called “Quantized-search Kabal” (“Q.-s. Kabal”), in which the zero crossing search is done directly on the values of the quantization tables. As the actual LSPs are not calculated, two criteria to select the “closest” quantized LSPs are used [4]. The measured quantization performance of “Q.-s. Kabal” is very close to the performance of Kabal’s algorithm followed by quantization (see § 4.1). The maximum number of polynomial evaluations is reduced from 150 to 71 [4].

3. SAUDI’S ALGORITHM

In Saudi’s algorithm [5], two functions are derived from the polynomials $P_{10}(z)$ and $Q_{10}(z)$, which obey a three-term recurrence relation, leading to the following tridiagonal matrices:

$$M_5 = \begin{bmatrix} 2\alpha_1 + \alpha_2 - 2 & 1 & 0 & 0 & 0 \\ \alpha_2\alpha_3 & \alpha_3 + \alpha_4 - 2 & 1 & 0 & 0 \\ 0 & \alpha_4\alpha_5 & \alpha_5 + \alpha_6 - 2 & 1 & 0 \\ 0 & 0 & \alpha_6\alpha_7 & \alpha_7 + \alpha_8 - 2 & 1 \\ 0 & 0 & 0 & \alpha_8\alpha_9 & \alpha_9 + \alpha_{10} - 2 \end{bmatrix}$$

$$M_5^* = \begin{bmatrix} \alpha_2^* - 2 & 1 & 0 & 0 & 0 \\ \alpha_2^*\alpha_3^* & \alpha_3^* + \alpha_4^* - 2 & 1 & 0 & 0 \\ 0 & \alpha_4^*\alpha_5^* & \alpha_5^* + \alpha_6^* - 2 & 1 & 0 \\ 0 & 0 & \alpha_6^*\alpha_7^* & \alpha_7^* + \alpha_8^* - 2 & 1 \\ 0 & 0 & 0 & \alpha_8^*\alpha_9^* & \alpha_9^* + \alpha_{10}^* - 2 \end{bmatrix} \quad (5)$$

the values α_m^* and α_m are obtained by using the antisymmetric split-Levinson recursion [5], given by:

$$P_0^*(z) = 0, \quad P_1^*(z) = 1 - z^{-1}, \quad p_{m,0}^* = 1 \quad \text{for } m \geq 1$$

$$\beta_0^* = 1, \quad \tau_0^* = r_0$$

for $1 \leq m \leq 10$:

$$\tau_m^* = \begin{cases} \sum_{i=0}^t (r_i - r_{m-i}) p_{m,i}^* & \text{for } m = 2t + 1 \\ \sum_{i=0}^{t-1} (r_i - r_{m-i}) p_{m,i}^* + r_t p_{m,t}^* & \text{for } m = 2t \end{cases}$$

$$\alpha_m^* = \frac{\tau_m^*}{\tau_{m-1}^*}, \quad \beta_m^* = 2 - \frac{\alpha_m^*}{\beta_{m-1}^*}, \quad \alpha_m = \beta_m^* (2 - \beta_{m-1}^*)$$

$$P_{m+1}^*(z) = (1 + z^{-1}) P_m^*(z) - \alpha_m^* z^{-1} P_{m-1}^*(z) \quad (6)$$

where the r_i are the autocorrelation coefficients of the speech frame. The eigenvalues of M_5 and M_5^* , denoted as λ_i , correspond to the odd- and even-suffixed LSPs respectively, with $\lambda_i = 2 \cdot \cos(\omega_i)$. Thus, the λ_i are ordered as follows:

$$+2 > \lambda_1 > \lambda_2 > \dots > \lambda_{10} > -2. \quad (7)$$

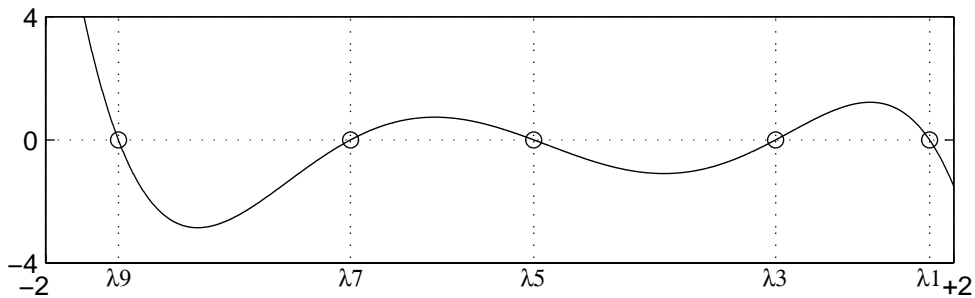


Figure 2.a: Behavior of the function $L_5(x)$ whose roots λ_i correspond to the odd-suffixed LSPs ($\lambda_i = 2\cos(\omega_i)$).

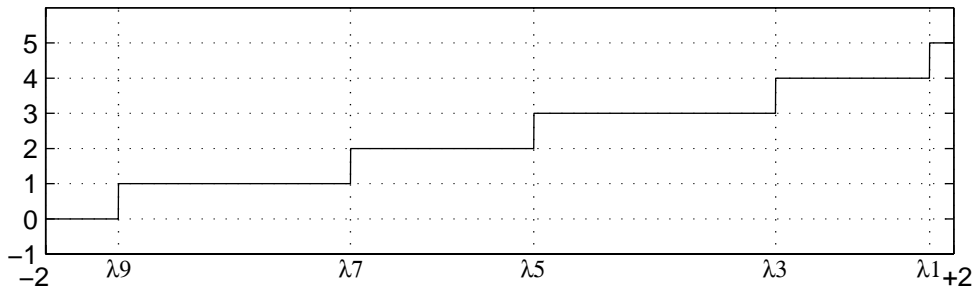


Figure 2.b: Behavior of the function $S_5(x)$, corresponding to the sign changes incurred in the sequence $\{L_0(x), \dots, L_5(x)\}$ when evaluating $L_5(x)$. Note that $S_5(x)$ gives a clear indication of how many roots of $L_5(x)$, lie above or below a given value of x .

The eigenvalues of M_5 and M_5^* are the roots of their characteristic polynomials, $L_5(x)$ and $L_5^*(x)$, which obey the following recursions [6]:

$$\begin{aligned}
 L_0(x) &= 1 \\
 L_1(x) &= (d(0) - x) \\
 L_2(x) &= (d(1) - x)L_1(x) - e(0) \cdot L_0(x) \\
 L_3(x) &= (d(2) - x)L_2(x) - e(1) \cdot L_1(x) \\
 L_4(x) &= (d(3) - x)L_3(x) - e(2) \cdot L_2(x) \\
 L_5(x) &= (d(4) - x)L_4(x) - e(3) \cdot L_3(x)
 \end{aligned} \tag{8}$$

$$\begin{aligned}
 L_0^*(x) &= 1 \\
 L_1^*(x) &= (d^*(0) - x) \\
 L_2^*(x) &= (d^*(1) - x)L_1^*(x) - e^*(0) \cdot L_0^*(x) \\
 L_3^*(x) &= (d^*(2) - x)L_2^*(x) - e^*(1) \cdot L_1^*(x) \\
 L_4^*(x) &= (d^*(3) - x)L_3^*(x) - e^*(2) \cdot L_2^*(x) \\
 L_5^*(x) &= (d^*(4) - x)L_4^*(x) - e^*(3) \cdot L_3^*(x)
 \end{aligned} \tag{9}$$

where $d(k)$ and $d^*(k)$ are respectively the diagonal elements of M_5 and M_5^* , and $e(k)$ and $e^*(k)$ are the elements below the diagonal.

An example of the behavior of the function $L_5(x)$, which is similar to the function $P'_{10}(x)$, is shown in Figure 2.a. The behavior of $L_5^*(x)$ (not showed) is similar to the behavior of $Q'_{10}(x)$.

As the sequence of polynomials $L_n(x)$ is Sturmanian [6], for a given value of $x = \gamma$, the number of sign changes in the numerical sequence $\{L_0(\gamma), \dots, L_5(\gamma)\}$ gives the number of roots of $L_5(x)$ which are smaller than γ . This is seen in Figure 2.b, where the function $S_5(x)$ corresponds to the sign changes incurred in the sequence $\{L_0(x), \dots, L_5(x)\}$ when evaluating $L_5(x)$. Similarly, the

function $S_5^*(x)$ (not showed), corresponds to the sign changes incurred in the sequence $\{L_0^*(x), \dots, L_5^*(x)\}$ when evaluating $L_5^*(x)$. Thus, $S_5(x)$ and $S_5^*(x)$ give a clear indication of how many roots of $L_5(x)$ and $L_5^*(x)$ lie above or below a given value of x . This property is used, together with Equation (7), to calculate each LSP independently, using the bisection method.

Each LSP is calculated over the interval $(-2, +2)$, using eight successive bisections [5]. The recursion of Equation (8) or (9) is evaluated using the mid-value of the interval and the number of sign changes in the obtained sequence is used to know with certitude in which of the two bisected intervals the LSP is located. Thus zero-crossings cannot be missed, and this independently of the speech database [5].

4. THE PROPOSED ALGORITHM

The proposed algorithm can be employed in speech coders that use scalar quantization such as (but not restricted to) the CELP FS1016 [3].

In Section 3 it is seen that Saoudi's method provides a simple test to know with certitude if a given LSP lies above or below any value in the quantization table, without calculating the actual LSP. Thus, this method can be combined with a binary-tree quantized-domain search for fast, direct calculation of the quantized LSPs. Due to the similarity between the bisection method and a binary-tree search, the adaptation of Saoudi's algorithm is straightforward. The obtained algorithm is referred to as "Quantized-search Saoudi" ("Q.-s. Saoudi").

The CELP FS1016 uses 34-bit non-uniform scalar quantization. A different table of (8 or 16) quantization values is used to quantize each of the 10 LSPs. In the proposed algorithm we use a table containing the mid-values of adjacent quantization levels.

Each quantized LSP is searched using its corresponding mid-value quantization table. A test is done, evaluating Equation (8) or (9) and counting the sign changes, to know if the LSP lies in the upper or the lower half of its quantization table. Then, the sub-table containing the LSP is selected, and the test is repeated, to know if the LSP lies in the upper or lower half of this sub-table. The test is done 3 times for an 8-level quantization table, and 4 times for a 16-level table. Thus, the total number of evaluations of Equation (8) or (9) is 34, which corresponds to the number of bits used to quantize all the LSPs.

The ordering property of equation (7) must be preserved upon quantization to have a stable LPC synthesis filter. A table containing, for each quantization level, the first allowed index for the next quantized LSP is used. Once a quantized LSP is found, the ordering property is tested with the help of this table and, if necessary, the LSP index is corrected.

4.1 Experimental Evaluation

Kabal's and Saoudi's algorithms, as well as a high precision method were used to calculate the LSPs, which were then quantized with the 34-bit scalar quantizer of the CELP FS1016. Spectral distortion was measured in all cases [1] using the whole TIMIT database (6300 speech files). The speech files were downsampled to 8 kHz and the LPC vectors were calculated as in the CELP FS1016 [3], using high-pass filtering of the speech input, 30 ms Hamming windowing, autocorrelation method, and 15 Hz bandwidth expansion (see also § 4.1.1).

The resulting average spectral distortion and percentage of outliers (with spectral distortion between 2-4 dB, and greater than 4 dB) are given in Table 1, together with the spectral distortion measured for the "Q.-s. Kabal" and "Q.-s. Saoudi" algorithms.

The results obtained using Kabal's and "Q.-s. Kabal" algorithms are very close to those obtained with the high precision method. The quantization performance is degraded when Saoudi's LSP calculation is used, due to the low precision in the calculated LSPs, as only 8 bisections are used. The precision could be increased by using more bisections at the cost of increased computational complexity [1]. Otherwise, the "Q.-s. Saoudi" algorithm is a cost effective way of improving the performance.

Algorithm	Spectral Distortion (dB)		
	average	% 2-4	% >4
High precision	1.5329	12.3450	0.1888
Kabal	1.5329	12.3453	0.1888
Saoudi	1.6536	19.1166	0.2025
"Q.-s. Kabal"	1.5330	12.3501	0.1895
"Q.-s. Saoudi"	1.5348	12.4318	0.1926

Table 1. Comparison among different methods to calculate quantized LSPs, in terms of spectral distortion.

The performance of "Q.-s. Saoudi" is slightly worse than "Q.-s. Kabal", due to the fact that in "Q.-s. Kabal" a criterion that takes into account the interaction between successive LSPs to minimize distortion is used [4] while in "Q.-s. Saoudi" the interaction between successive LSPs is only taken into account to preserve the ordering property.

4.1.1 A Note on Bandwidth Expansion

A drawback in the utilization of the algorithms of Saoudi in the CELP FS1016 is that the 15 Hz bandwidth expansion cannot be easily applied, as the LPC coefficients are not calculated in the antisymmetric split-Levinson recursion. An effect similar to bandwidth expansion can be obtained with the spectral smoothing technique described in [7], in which the autocorrelation coefficients are multiplied by a Gaussian window. On the other hand the spectral smoothing technique would not give numerically the same results than the bandwidth expansion. Thus, in order to make meaningful comparisons among the different algorithms, the autocorrelation coefficients needed in the antisymmetric split-Levinson recursion were obtained by transformation from the bandwidth expanded LPC coefficients.

4.2 Computational Complexity

The total number of operations required by Kabal's, Saoudi's and "Q.-s. Kabal" as reported in [1] is shown in Table 2, as well as the figures for the "Q.-s. Saoudi" which are obtained by subtracting $46 \times (9 \text{ Add}, 8 \text{ Mult})$ from the figures of Saoudi's algorithm, due to the reduction from 80 to 34 evaluations of Equation (8) or (9). The overhead incurred in the quantization process when using Kabal's and Saoudi's algorithm is not shown in Table 2.

Algorithm	Mult	Add	Div
Kabal	730	1530	20
Saoudi	706	941	20
"Q.-s. Kabal"	394	769	10
"Q.-s. Saoudi"	338	527	20

Table 2. Total number of operations per frame needed to obtain the LSPs, using different LSP calculation algorithms.

The complexity of "Q.-s. Saoudi" is much lower than Saoudi's, however it is not clear if "Q.-s. Saoudi" outperforms "Q.-s. Kabal", depending strongly on the final implementation. An attempt of comparison is made based on the DSP56001 implementation of "Q.-s. Kabal" reported in [1]. We assigned a weight of 31 to divisions, and a weight of one to multiplications and additions, obtaining a complexity figure of 1485 for "Q.-s. Saoudi" and 1473 for "Q.-s. Kabal".

4.3 Extension to other LPC Calculation Methods

In Section 3 it is seen that the recursions of Equation (8) or (9) provide a simple test to know with certitude if a given LSP lies above or below any value in the quantization tables, without calculating the actual LSP. This test is used to do binary-tree search in the quantized domain, obtaining the fast LSP calculation method explained in this section.

Saudi's algorithm does not use a standard LPC calculation method, but instead uses the antisymmetric split-Levinson recursion of Equation (6) to calculate the values of α_m^* and α_m , needed in Equation (8) and (9).

On the other hand, speech coders found in scientific literature and standards do not use the split-Levinson recursion, but different LPC calculation methods, such as Levinson-Durbin [8], Lattice methods [8] and Leroux-Gueguen [9]. All these methods give the reflection coefficients, k_m .

In [10] it is shown that the α_m^* and α_m can also be obtained from the reflection coefficients k_m by using:

$$\begin{aligned} \alpha_1 &= \beta_0^* \beta_1 & \alpha_1^* &= \beta_0 \beta_1^* \\ \alpha_2 &= \beta_1^* \beta_2 & \alpha_2^* &= \beta_1 \beta_2^* \\ &\vdots & & \\ \alpha_m &= \beta_{m-1}^* \beta_m & \alpha_m^* &= \beta_{m-1} \beta_m^* \\ &\vdots & & \\ \alpha_{10} &= \beta_9^* \beta_{10} & \alpha_{10}^* &= \beta_9 \beta_{10}^* \end{aligned} \quad (10)$$

where the β_m and β_m^* are given by:

$$\begin{aligned} \beta_0 &= \beta_0^* = 1 \\ \beta_1 &= 1 + k_1 & \beta_1^* &= 1 - k_1 \\ &\vdots & & \\ \beta_m &= 1 + k_m & \beta_m^* &= 1 - k_m \end{aligned} \quad (11)$$

Thus the fast LSP calculation method proposed in this paper can easily be adapted to any LPC calculation method that gives the reflection coefficients.

5. CONCLUSIONS

We have proposed an algorithm for fast direct calculation of quantized LSPs for application in speech coders that use scalar quantization. The proposed algorithm not only reduces the computation required by Saoudi's method, but also improves the quantization performance. Although the proposed algorithm is comparable in complexity to "Q.-s. Kabal", it has the additional advantages of "intrinsic reliability" (zero crossings cannot be missed) and easier adaptability to different quantization tables and LPC orders.

Future work goes in the direction of combining the proposed algorithm with a further stage of Vector Quantization and include it in a speech coder such as the G.729 [11]. We also would like to explore by simulation the robustness of the proposed algorithm with respect to the use of fixed-point arithmetic.

6. ACKNOWLEDGEMENTS

This work was partially supported by the Swiss Federal Office for Education and Science under Grant OFES C97.0050 (COST 254 project).

We are very grateful to A. Dufaux and L. Besacier for proof-reading this paper.

7. REFERENCES

- [1] S. Grassi, "Optimized Implementation of Speech Processing Algorithms", Ph.D. Thesis, IMT, University of Neuchâtel, Switzerland, 1998.
- [2] P. Kabal and P. Ramachandran, "The Computation of Line Spectral Frequencies Using Chebyshev Polynomials", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. 34, No. 6, pp. 1419-1426, 1986.
- [3] Campbell J. et al., "The DoD 4.8 kbps Standard (Proposed Federal Standard 1016)" in *Advances in Speech Coding*, Atal, Cuperman and Gersho Eds., Kluwer Academic Publishers, Boston, MA, USA, 1991.
- [4] S. Grassi, M. Ansoorge, and F. Pellandini, "Fast LSP Calculation and Quantization with Application to the CELP

FS1016 Speech Coder", *Proc. EUSIPCO'98*, Rhodes, Greece, Vol. III, pp. 1425-1428, 1998.

- [5] S. Saoudi and J. M. Boucher, "A New Efficient Algorithm to Compute the LSP Parameters for Speech Coding", *Signal Processing*, Vol. 28, No. 2, pp. 201-212, Elsevier, 1992.
- [6] F. S. Acton, *Numerical Methods that Work*, Mathematical Association of America, Washington, DC, 1990.
- [7] Y. Tohkura et al., "Spectral Smoothing Technique in Parcor Speech Analysis-synthesis", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. 26, No. 6, pp. 587-596, 1978.
- [8] A. M. Kondozi, *Digital Speech: Coding for Low Bit Rate Communications Systems*, Wiley, Chichester, 1994.
- [9] J. Leroux and C. Gueguen, "A Fixed Point Computation of Partial Correlation Coefficients", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. 25, No. 3, pp. 257-259, 1977.
- [10] P. Delsarte and Y. V. Genin, "The Split Levinson Algorithm", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. 34, No. 3, pp. 470-478, 1986.
- [11] A. Kataoka et al., "LSP and Gain Quantization for the Proposed ITU-T 8 kb/s Speech Coding Standard", *IEEE Workshop on Speech Coding for Telecommunications*, Annapolis, USA, pp. 7-8, 1995.