Published in Proceedings of Signal Processing Workshop (SPW), 61-66, 1999
which should be used for any reference to this work

LSP CALCULATION METHODS FOR APPLICATION TO SPEECH CODING

S. Grassi, M. Ansorge, F. Pellandini

Institute of Microtechnology, University of Neuchâtel
Rue A.-L. Breguet 2, 2000 Neuchâtel, Switzerland
Phone: +41 32 7183432; Fax: +41 32 7183402; Email: sara.grassi@imt.unine.ch

ABSTRACT

Line spectrum pair (LSP) parameters are commonly used in speech coding for quantization of the speech spectral envelope. Unfortunately, the high computational complexity in the calculation of the LSP is a drawback for both real-time implementation and application in low-power portable devices. In this paper, some techniques for reducing computational complexity of the LSP calculation are given. The use of these techniques results in three novel LSP calculation algorithms which are explained and evaluated from the point of view of accuracy and computational complexity.

1. INTRODUCTION

The purpose of a speech coder is to compress speech signals, employing as few bits as possible in their digital representation. The term "speech coding" is commonly used to refer to the coding of telephone bandwidth speech (300-3400 Hz) sampled at 8 kHz.

Typical applications of speech coding are in telecommunications, voice storage systems, personal communications systems, and multimedia for personal computing, where voice storage is becoming a standard feature. All these applications require real time implementation. Additionally, speech coding finds application in portable devices such as digital cellular telephones, vocal pagers, and portable multimedia terminals and computers, which also 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. In this paper we focus on techniques for reducing computational complexity of LSP calculation, which is a computationally intensive task found in most speech coders.

The definition of the LSPs and their utilization in speech coding are discussed in Section 2. An overview of existing LSP calculation methods is given in section 3.

Different techniques to reduce computational complexity of LSP calculation are explained in Section 4. These techniques result in three novel LSP calculation algorithms which, in Section 5 and 6, are evaluated from the point of view of accuracy and computational complexity. Conclusions and further work are given in Section 7.

2. LPC ANALYSIS AND LSP REPRESENTATION

Linear predictive coding (LPC) is an accurate and economic representation of the speech spectral envelope which is widely used in speech coding and other speech processing areas such as speech synthesis and voice recognition.

Line spectrum pair (LSP) parameters are very popular in the domain of speech coding. They have a one to one correspondence with the LPC coefficients and allow more efficient encoding of the spectral information.

2.1 Use of LSP Representation in Speech Coding

LSP representation of 10-th order LPC coefficients is used in nearly all narrowband speech coding standards, with bit rates of less than 16 kbps, such as [1]:

- The ITU-T G.729 CS-ACELP coder, at 8 kbps.
- The ITU-T G.723.1, dual rate speech coder for multimedia, at 5.3 /6.3 kbps.
- The GSM 6.60, enhanced full rate, at 12.2 kbps.
- The new GSM Adaptive Multi Rate (AMR) coder.
- The TIA IS-96, North-American standard for CDMA cellular telephony, variable rate QCELP.
- The TIA IS-641, enhanced full rate coder for North-American TDMA cellular telephony, at 7.4 kbps.
- The Japanese half-rate personal digital cellular standard.
- The US DoD FS1016 CELP at 4.8 kbps.
- The US DoD FS1017 MELP, at 2.4 kbps.

Additionally, nearly all the CELP coders found in recent publications as well as some parametric coders use LSP representation of 10-th order LPC. Hereafter, an LPC order of 10 is assumed.

2.2 Definition of LSP Parameters

Given the 10-th order LPC analysis filter:

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

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

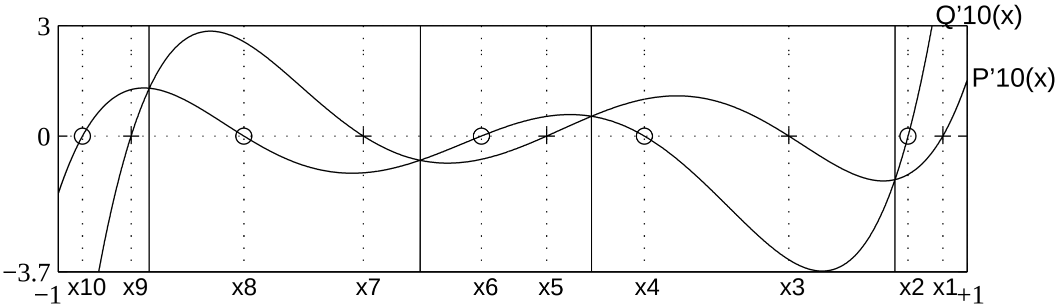

Figure 1.a: Behavior of the functions $P'_{10}(x)$ and $Q'_{10}(x)$ (x1 to x10 are the LSPs in the "x-domain", with $x = \cos(\omega)$).

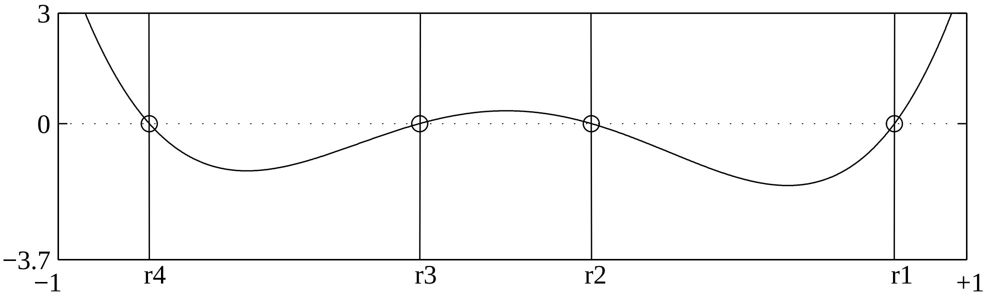

Figure 1.b: Behavior of the function $D_{10}(x)$ (r1 to r4 are the roots of $D_{10}(x)$). Note that the roots of $D_{10}(x)$ divide the interval (−1,+1) into five sections, containing each only one zero-crossing of $P'_{10}(x)$ and one zero crossing of $Q'_{10}(x)$.

$$P_{10}(z) = A_{10}(z) + z^{-11}A_{10}(z^{-1}) = (1+z^{-1}) \cdot P'_{10}(z)$$
$$Q_{10}(z) = A_{10}(z) - z^{-11}A_{10}(z^{-1}) = (1-z^{-1}) \cdot Q'_{10}(z) \qquad (2)$$

The polynomials $P'_{10}(z)$ and $Q'_{10}(z)$ are symmetrical. 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 $\{\omega_i\}$. The odd-suffixed LSPs correspond to roots of $P'_{10}(z)$ while the even-suffixed LSPs correspond to roots of $Q'_{10}(z)$. Due to the interlacing property:

$$0 < \omega_1 < \omega_2 < \ldots < \omega_{10} < \pi \qquad (3)$$

3. CALCULATION OF THE LSP PARAMETERS

The calculation of LSP parameters from LPC coefficients is a computationally intensive task, as it involves the resolution of polynomials by numerical root search. A survey of existing algorithms for LSP calculation was done [1] and it was found that the algorithms of Kabal [2] and Saoudi [3] are the most promising for efficient real time implementation. These methods are briefly explained in this Section. We also proposed three novel efficient algorithms referred to as "Mixed-LSP", "Quantized-search Kabal" and "Quantized-search Saoudi" (see § 4). A brief explanation of other methods, such as Chan's method and methods based on discrete Fourier or cosine transform, can be found in [1].

3.1 Kabal's Algorithm

The most popular method for LSP calculation is Kabal's algorithm [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):

$$+1 > x_1 > x_2 > \ldots > x_{10} > -1 \qquad (4)$$

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.a.

As $P'_{10}(x)$ and $Q'_{10}(x)$ are 5^{th}-order polynomials, their zeros cannot be calculated in a closed form. In the numerical solution proposed by Kabal in [2], 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 § 4.3).

3.2 Saoudi's Algorithm

In Saoudi's algorithm [3], 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:

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

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

the values α_m^* and α_m are obtained by using the antisymmetric split-Levinson recursion [3], given by:

$$\begin{aligned} &P_0^*(z) = 0, \quad P_1^*(z) = 1 - z^{-1}, \quad p_{m,0}^* = 1 \quad \text{for} \quad m \ge 1 \\ &\beta_0^* = 1, \quad \tau_0^* = r_0 \\ &\text{for} \quad 1 \le m \le 10: \\ &\tau_m^* = \begin{cases} \sum_{i=0}^{t} (r_i - r_{m-i})\, p_{m,i}^* & \text{for} \quad m = 2t+1 \\ \sum_{i=0}^{t-1} (r_i - r_{m-i})\, p_{m,i}^* + r_t\, p_{m,t}^* & \text{for} \quad m = 2t \end{cases} \\ &\alpha_m^* = \frac{\tau_m^*}{\tau_{m-1}^*}, \quad \beta_m^* = 2 - \frac{\alpha_m^*}{\beta_{m-1}^*}, \quad \alpha_m = \beta_m^*(2 - \beta_{m-1}^*) \\ &P_{m+1}^*(z) = (1 + z^{-1})\, P_m^*(z) - \alpha_m^*\, z^{-1} P_{m-1}^*(z) \end{aligned} \tag{6}$$

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

$$+2 > \lambda_1 > \lambda_2 > \ldots > \lambda_{10} > -2 \tag{7}$$

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 [3]:

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

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

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

As the sequence of polynomials $L_n(x)$ is Sturmanian [4], for a given value of $x = \gamma$, the number of sign changes in the numerical sequence $\{L_0(\gamma),\ldots,L_5(\gamma)\}$ 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 $\{L_0(x),\ldots, L_5(x)\}$ when evaluating $L_5(x)$. Similarly, the function $S_5^*(x)$ (not showed), corresponds to the sign changes incurred in the sequence $\{L_0^*(x),\ldots, L_5^*(x)\}$ when evaluating $L_5^*(x)$.

It is observed that $S_5(x)$ 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 [3]. 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 [3].

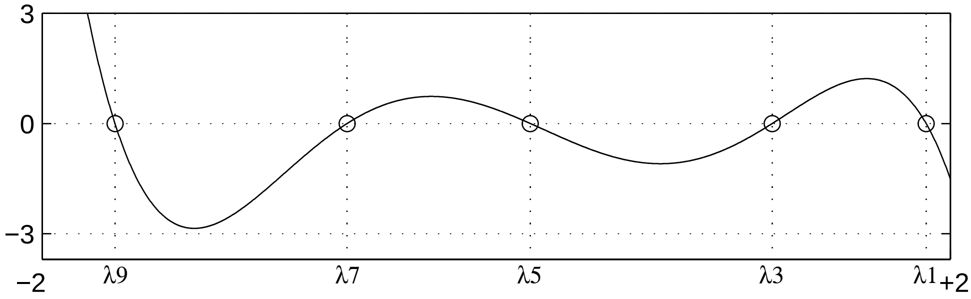

Figure 2.a: Behavior of the function $L_5(x)$ whose roots λi correspond to the odd-suffixed LSPs ($\lambda i = 2\cos(\omega i)$).

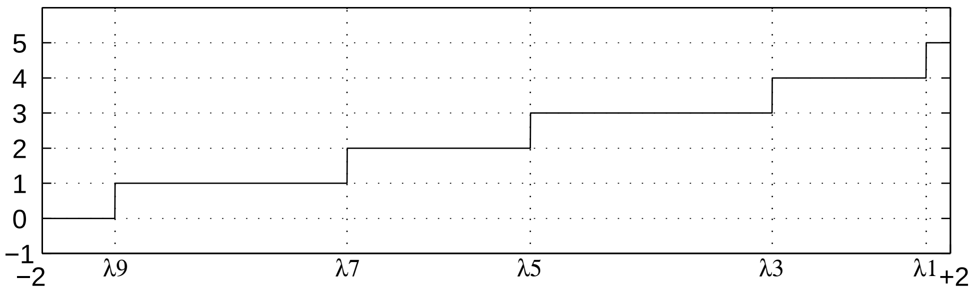

Figure 2.b: Behavior of the function $S_5(x)$, corresponding to the sign changes incurred in the sequence $\{L_0(x),\ldots, L_5(x)\}$ when evaluating $L_5(x)$. Note that $S_5(x)$ gives a clear indication of how many roots of $L_5(x)$, lie above or below a given value of x.

4. THE PROPOSED ALGORITHMS

4.1 "Mixed-LSP"

The precision of the LSPs obtained with Kabal's method is higher than required by speech coding applications, but the number of bisections cannot be decreased, or the size of the grid increased, without compromising the zero crossing search.

In [5], it is shown that five intervals, containing each only one zero crossing of $P'_{10}(x)$ and one zero crossing of $Q'_{10}(x)$, can be calculated, as the roots of a 4-th order polynomial, $D_{10}(x)$. The behavior of this polynomial is observed in Figure 1.b. The roots of $D_{10}(x)$ divide the interval (–1,+1) into five sections, containing each only one zero-crossing of $P'_{10}(x)$ and one zero crossing of $Q'_{10}(x)$. This fact is used to avoid the zero crossing search, allowing a trade-off between LSP precision and computational complexity. The resulting algorithm is called Mixed-LSP algorithm, and needs a total of 60 polynomial evaluations. These evaluations are done using Kabal's efficient polynomial evaluation.

The calculation and sorting of the roots of $D_{10}(x)$ was carefully optimized [5], and finally needs the following operations: 20 multiplications, 34 add/sub, 2 divisions and 5 square roots, as well as 3 comparison/swapping operations. It was found that Mixed-LSP algorithm needs 33 % less MIPS than Kabal's algorithm on a DSP56001 implementation [1].

The proposed Mixed-LSP algorithm is computationally less expensive but also less accurate than Kabal's method [1]. On the other hand, the accuracy of the Mixed-LSP algorithm is sufficient for speech coding applications using the 34-bit quantizer of the CELP. The Mixed-LSP algorithm can be used not only with the scalar quantization of the CELP FS1016, but also with other scalar quantization schemes, as well as vector quantization.

4.2 Quantized-search Kabal

The LSPs obtained with Mixed-LSP, Kabal's or Saoudi's algorithms are not quantized. If these methods are used in the CELP FS1016, the LSPs are first calculated, and then quantized using the 34-bit non-uniform scalar quantization [6]. The algorithm denoted as "Quantized-search Kabal" is a modified version of Kabal's algorithm, in which the zero crossings search is done on a grid formed with the values of the quantization tables [7].

As the actual LSPs are not calculated, two new criteria to select the quantized LSPs which are closer to the actual LSPs are proposed [7]. These criteria take into account the interaction between successive LSPs. The efficiency and reliability of this algorithm are improved by using the interlacing property of the LSPs and knowledge of the direction of the sign-change at every zero-crossing.

The quantization performance of "Quantized-search Kabal" is very close to the performance of Kabal's algorithm followed by quantization (see § 5). The maximum number of polynomial evaluations is reduced to 71, resulting in a saving of 66 % MIPS on a DSP56001 implementation [1].

4.3 Binary-tree quantized-domain search

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. Such a test is available in Saoudi's method (see § 3.2). Thus, this method can be easily combined with a binary-tree quantized-domain search for fast, direct calculation of the quantized LSPs [8].

4.3.1 "Quantized-search Saoudi"

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 either 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.

4.3.2 Extension to other LPC Calculation Methods

Saoudi's algorithm can be easily adapted to binary-tree quantized domain search, but it has the drawback that it uses the antisymmetric split-Levinson instead of LPC calculation. On the other hand, speech coders found in scientific literature and standards use different LPC calculation methods [9], such as Levinson-Durbin, Lattice methods and Leroux-Gueguen. All these methods give the reflection coefficients, k_m.

It can be shown that the α_m^* and α_m needed for the recursion of Equation (8) and (9) can also be obtained from the reflection coefficients k_m by using:

$$\begin{aligned} \alpha_1 &= (1+k_1) & \alpha_1^* &= (1-k_1) \\ \alpha_2 &= (1+k_2)(1-k_1) & \alpha_2^* &= (1-k_2)(1+k_1) \\ &\vdots & &\vdots \\ \alpha_{10} &= (1+k_{10})(1-k_9) & \alpha_{10}^* &= (1-k_{10})(1+k_9) \end{aligned} \tag{10}$$

Thus the binary-tree quantized-domain LSP calculation proposed in this section can be easily adapted to any LPC calculation method that gives the reflection coefficients.

5. 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. 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.

	Spectral Distortion (dB)		
Algorithm	*average*	*% 2-4*	*% >4*
High precision	1.5329	12.3450	0.1888
Kabal	1.5329	12.3453	0.1888
Mixed-LSP	1.5331	12.3631	0.1885
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 results obtained using Kabal's, Mixed-LSP and "Q.-s. Kabal" algorithms are very close to those obtained with the high precision method. Thus, although the Mixed-LSP method is less accurate than Kabal's method [5], it is sufficient for speech coding applications using the 34-bit scalar quantizer of the CELP FS1016. 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. 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 [7] while in "Q.-s. Saoudi" the interaction between successive LSPs is only taken into account to preserve the ordering property.

6. 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*(9 Add, 8 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*	*Sqrt*
Kabal	730	1530	20	-
Saoudi	706	941	20	-
Mixed-LSP	390	764	22	5
"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".

7. CONCLUSIONS AND FURTHER WORK

In this paper we have presented several techniques for reducing computational complexity in LSP calculation methods, with application to speech coding. These techniques result in three novel efficient algorithms referred to as Mixed-LSP, "Q.-s. Kabal" and "Q.-s. Saoudi".

The Mixed-LSP algorithm can be used not only with scalar quantization but also with vector quantization, however its utilization is limited to an LPC order of 10. Nevertheless, an LPC order of 10 is used in nearly all the standard and emerging low bit rate narrowband speech coders.

"Q.-s. Kabal" algorithm is more efficient than Mixed-LSP and Kabal's, but is tied to the utilization of the 34-bits non-uniform scalar quantizer of the CELP FS1016.

"Q.-s. Saoudi" 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 "Q.-s. Saoudi" with a further stage of Vector Quantization and include it in a speech coder such as the G.729 [10]. We also would like to explore by simulation the robustness of "Q.-s. Saoudi" algorithm with respect to the use of fixed-point arithmetic.

8. ACKNOWLEDGEMENTS

We acknowledge the financial support given by the Swiss Federal Office for Education and Science under Grant OFES C97.0050 (COST 254 project).

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

9. 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. ASSP., Vol. 34, No. 6, pp. 1419-1426, 1986.

[3] S. Saoudi and J. Boucher, "A New Efficient Algorithm to Compute the LSP Parameters for Speech Coding", Signal Processing, Vol. 28, No. 2, pp. 201-212, Elsevier, 1992.

[4] F. S. Acton, *Numerical Methods that Work*, Mathematical Association of America, Washington, DC, 1990.

[5] S. Grassi, A. Dufaux, M. Ansorge, and F. Pellandini, "Efficient Algorithm to Compute LSP Parameters from 10-th order LPC Coefficients", Proc. ICASSP'97, Vol. 3, pp. 1707-1710, 1997.

[6] "Federal Standard 1016, Telecommunications: Analog to Digital Conversion of Radio Voice by 4,800 bit/second Code Excited Linear Prediction (CELP)", National Communications Systems, Office of Technology and Standards, Washington, DC20305-2010, 1991.

[7] S. Grassi, M. Ansorge, 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.

[8] S. Grassi, M. Ansorge, F. Pellandini, "An Algorithm for Fast Direct Calculation of Quantized LSP Parameters", to be presented at the Fifth Bayona Workshop on Emerging Technologies in Telecommunications, Bayona, Spain, Sep. 6-8, 1999.

[9] A. M. Kondoz, *Digital Speech: Coding for Low Bit Rate Communication Systems,* Wiley, Chichester, 1994.

[10] A. Kataoka et al., "LSP and Gain Quantization for the Proposed ITU-T 8 kb/s Speech Coding Standard", *IEEE Speech Coding Workshop,* pp. 7-8, 1995.