EFFICIENT CALCULATION OF LINE SPECTRAL FREQUENCIES BASED ON NEW METHOD FOR SOLUTION OF TRANSCENDENTAL EQUATIONS

V. Semenov, A. Kalyuzhny, A.Kovtonyuk

"DELTA" S.P.E.
27 Academician Krymsky St., Kiev, 03142 Ukraine
e-mail: aleko@deltacorp.net

ABSTRACT

A novel effective method of calculation of line spectral frequencies (LSF) is proposed. The method is based on developed algorithm of full numerical solution of transcendental equations which do not have multiple roots. This algorithm is composed of two parts: effective location of intervals containing single zero and successive refinement of root value by one of standard rootfinding procedures. Different modifications of proposed LSF calculation method are verified on real speech signals. In oppose to majority of existing LSF computation algorithms, proposed method provides arbitrary high accuracy, guarantees stability of corresponding autoregressive filter and does not require any a priori information about LSF location. It is shown that developed method can be applied in real-time applications.

1. INTRODUCTION

The most of modern speech compression methods are based on autoregressive (AR) model of speech generation [1]. According to this model, speech signal $s(n)$ is modeled as the result of passing of excitation $w(n)$ through all-pole filter with coefficients $a_k, k = 1,..., p$:

$$s(n) = -\sum_{k=1}^{p} a_k s(n-k) + w(n) . \qquad (1)$$

The order of AR model p is chosen, as a rule, from 8 to 20.

As is well known [2-4], direct quantization of AR coefficients is inapplicable, because it leads to instability of synthesis filter. The most of contemporary speech coders use transformation of AR coefficients to the set of line spectral frequencies (LSF), which code spectral information more efficiently than other alternative transmission parameters.

At the same time, existing methods of LSF computation make a compromise between amount and accuracy of calculation. Acceptable speed of processing is often achieved at the cost of introduction of inaccuracies in LSF computation. This may lead, in particular, to instability of corresponding synthesis filters and cause the loss of quality and intelligibility of reconstructed speech [3, 4]. The probability of such failures is especially high if AR models of relatively high order are used.

Taking mentioned difficulties into account, the goal of this work is the development of LSF computation method, which is more efficient in comparison with existing techniques. The conceptual framework of proposed technique is a universal method of solution of transcendental equations, which do not have multiple roots. The resulting algorithm provides arbitrary high accuracy of computations and guarantees stability of corresponding AR filter. In oppose to many existing methods, it does not need any a priori information on LSF location. Besides, accuracies of computation of different frequencies do not depend on each other and can be easily varied. Experiments with speech signals show that proposed method can be applied in real-time applications.

2. LINE SPECTRAL FREQUENCIES

Let's briefly discuss existing methodology of transformation of AR coefficients to LSF. At first, whitening minimum-phase filter $A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k}$ ($p = 2M$) is converted into following symmetric and antisymmetric filters:

$$\begin{cases} F_s(z) = A(z) + z^{-p-1} A(z^{-1}) \\ F_a(z) = A(z) - z^{-p-1} A(z^{-1}). \end{cases} \qquad (2)$$

Removing of roots ± 1 leads to the following polynomials:

$$G_m(z) = \sum_{k=0}^{p} g_k^{(m)} z^{-k}, \quad m = 1,2\,(g_0^{(m)} = g_p^{(m)} = 1). \qquad (3)$$

All the roots of $G_1(z)$ and $G_2(z)$ lie on a unit circle, i.e. have a form $z_k = e^{i\omega_k}$. Line spectral frequencies are defined by root angles which lie in the range $(0;\pi)$. The significant feature of LSF is their ordering property [2-4]: frequencies $\omega_1^{(1)},\omega_2^{(1)},...,\omega_M^{(1)}$ corresponding to $G_1(z)$ are interleaved with frequencies $\omega_1^{(2)},\omega_2^{(2)},...,\omega_M^{(2)}$ corresponding to $G_2(z)$:

$$\omega_1^{(1)} < \omega_1^{(2)} < \omega_2^{(1)} < \omega_2^{(2)} < ... < \omega_M^{(1)} < \omega_M^{(2)} . \qquad (4)$$

It worth to note that LSF ordering property is a necessary and sufficient condition for the stability of synthesis filter $1/A(z)$.

Substitution $z + z^{-1} = 2\cos(\omega)$ allows to transform symmetric equations (3) into following trigonometric equations for LSF:

$$r_0^{(m)} + \sum_{k=1}^{M} r_k^{(m)} \cos k\omega = 0 \qquad (m = 1,2). \qquad (5)$$

Existing methods for the solution of problem (5) use evaluation of function $R(e^{i\omega}) = r_0 + \sum_{k=1}^{M} r_k \cos k\omega$ on a reasonably large grid of points to obtain initial estimates of root locations [5]. The main disadvantage of such methods is that necessary size of grid cannot be predicted beforehand since the roots of function $R(e^{i\omega})$ can be arbitrarily close to each other. From the other hand, smaller step of grid will take much more computation. Besides, this approach requires prior storage or large calculation of trigonometric functions, which makes it unsuitable for real-time applications.

That is why transcendental equations (5) are traditionally transformed to the polynomial equations by substitution $x = \cos(\omega)$ (or $x = 2\cos(\omega)$) [2-4]. The values for multiple argument ($\cos(n\omega)$) are expressed with the help of Chebyshev polynomials $T_n(x) = \cos(n\omega) = \cos(n \arccos(x))$. This allows to obtain following typical polynomial equations:

$$\sum_{k=0}^{M} r_k'^{(m)} x^{M-k} = 0 \qquad (m = 1,2). \qquad (6)$$

The roots of equations (6) keep the ordering property, are real and lie in the range (-1;1). After the roots are found, they are mapped to ω-domain by nonlinear transformation $\omega = \arccos(x)$.

Methods, based on such approach to determination of LSF, can be divided into two categories. The typical representative of the first subgroup is the method of Kabal and Ramachandran [2]. As in the case of equations (5), the single-root intervals are determined by evaluation of polynomial function on a large predefined grid of points. The main disadvantage of this approach is that for every AR order the step of grid δ must be estimated on a large speech database. At that, if two roots of one of functions (6) will occur close than δ, this method will become inapplicable (and the smaller value of δ will take more computation).

Another subgroup of methods [3, 4] exploit consequent reduction of degree by deflation of equations (6). However the use of deflation may increasingly lead to worse accuracy of each subsequent zero. This becomes an especially serious problem when working with high-order polynomials [5].

So, existing methods of LSF computation have many weak points. To overcome the mentioned difficulties, in the next section the universal method of solution of transcendental equations is proposed. After that, application of method to LSF problem is demonstrated. Although proposed method can be applied immediately to the direct solution of equations (5), in this work we limit ourselves to consideration of equations (6).

3. PROPOSED ALGORITHM AND ITS APPLICATION TO LSF PROBLEM

In this section universal method of solution of transcendental equations $f(x) = 0$ is proposed. The only its limitation is the continuous differentiability of function $f(x)$ and the absence of multiple roots. After that, application of method to LSF problem is demonstrated.

3.1. Preliminaries

So, we are interested in determination of all roots of equation $f(x) = 0$ on interval $X = [a,b]$. Assume, that constraint for absolute value of j-th derivative is known:

$$\sup_{x \in [a,b]} \left| f^{(j)}(x) \right| \le M_j. \qquad (7)$$

Consider some examples connected with LSF problem.

1) The constraint of order j for function $f(x) = r_0 + \sum_{k=1}^{M} r_k \cos kx$ in equation (5):

$$M_j = \sum_{k=0}^{M} k^j \mid r_k \mid .$$

2) The constraint of order j for function $f(x) = \sum_{k=0}^{M} r_k' x^{M-k}$, in equation (6) on interval (-1;1):

$$M_j = \sum_{k=0}^{M-j} (M-k)(M-k-1)...(M-k-j+1) \mid r_k' \mid . \qquad (8)$$

Consider two statements, which can be proved by expansion of $f(x)$ in Taylor series.

Lemma 1. If interval $[c,d] \subset X$ for some j has a property

$$f[0.5(c+d)] + \sum_{k=1}^{j-1} \frac{\left| f^{(k)}[0.5(c+d)] \right| (d-c)^k}{2^k k!} + \frac{M_j (d-c)^j}{2^j j!} < 0$$ or

$$f[0.5(c+d)] - \sum_{k=1}^{j-1} \frac{\left| f^{(k)}[0.5(c+d)] \right| (d-c)^k}{2^k k!} - \frac{M_j (d-c)^j}{2^j j!} > 0,$$ then

function $f(x)$ does not have roots on $[c,d]$.

Lemma 2. If interval $[c,d] \subset X$ for some j has a property

$$f'[0.5(c+d)] + \sum_{k=2}^{j-1} \frac{\left| f^{(k)}[0.5(c+d)] \right| (d-c)^{k-1}}{2^{k-1} (k-1)!} + \frac{M_j (d-c)^{j-1}}{2^{j-1} (j-1)!} < 0$$ or

$$f'[0.5(c+d)] - \sum_{k=2}^{j-1} \frac{\left| f^{(k)}[0.5(c+d)] \right| (d-c)^{k-1}}{2^{k-1} (k-1)!} - \frac{M_j (d-c)^{j-1}}{2^{j-1} (j-1)!} > 0,$$

then derivative $f'(x)$ does not have roots on $[c,d]$.

With the help of these lemmas we can analyze behavior of $f(x)$ and $f'(x)$ on arbitrary interval $[c,d]$.

For function $f(x)$ two situations are possible:

1) $f(x)$ does not have roots on $[c,d]$,
2) the situation with the constancy of sign of function $f(x)$ is uncertain.

In a similar way, there are two possible situations for derivative $f'(x)$:

1) $f'(x)$ does not have roots on $[c,d]$, i.e. $f(x)$ is monotonic on $[c,d]$,
2) the situation is uncertain.

In following we will refer to number j, appearing in stated above lemmas, as to approximation order. This parameter may be different for $f(x)$ (where $j \geq 1$) and for $f'(x)$ (where $j \geq 2$). The choice of this number depends on a specific character of a problem and, as will be shown further, crucially influences on the computational burden of algorithm. In this work we will limit ourselves to the case of equal approximation orders for $f(x)$ and $f'(x)$.

3.2. Description of algorithm

Consider the initial interval $[a,b]$. Let's analyze behavior of $f(x)$ and $f'(x)$ on $[a,b]$ with the help of lemmas, stated in previous section. Following situations are possible.

a) $f(x)$ does not have roots on $[a,b]$.

b) Situation for $f(x)$ is uncertain, but $f'(x)$ does not have roots on $[a,b]$, i.e. $f(x)$ is monotonic on $[a,b]$. Then the presence of root on this interval can be verified by the sign of product $f(a)f(b)$. If $f(a)f(b) > 0$, then there are no roots on $[a,b]$. Otherwise, $f(x)$ has one root on this interval and the value of root can be determined by any standard method of root refinement, e.g. Newton's or Newton-Raphson method.

c) If the situation is uncertain both for $f(x)$ and $f'(x)$, then interval $[a,b]$ is divided on subintervals $[a, 0.5(a+b)]$, $[0.5(a+b), b]$, these subintervals must be analyzed in a similar way and so on.

Due to assumption about the absence of multiple roots (that's true for most of practical situations and, in particular, for LSF problem), on some stage of recursive division there will be no intervals, where the situation is uncertain both for function and its derivative. So, recursive division will be finished, all single-root intervals will be identified and all roots will be found.

Note, that above mentioned methods of root refinement exploit division by derivative $f'(x)$. In our case this operation is completely "safe", because proposed algorithm guarantees sign constancy of derivative on each of extracted intervals. Besides, on each of extracted single-root intervals derivatives of orders from 1 to $j-1$ in middle point are available (see the conditions of stated above lemmas). It allows to choose efficient initial value for root refinement procedure.

The key characteristic of given algorithm, indicating the speed of its work, is a number of recursive divisions of initial interval $[a,b]$. This parameter essentially depends on the chosen approximation orders.

3.3. Application of proposed algorithm to the computation of LSF

As can be seen from the description of algorithm, accuracies of obtained zeros do not depend on each other and can be easily varied. It means that application of proposed algorithm to equations (5) or (6) guarantees both arbitrary high accuracy of LSF calculation and monotony of resulting LSF (i.e. stability of corresponding AR filter).

Resulting procedure of LSF calculation has following form. Parameters $a_k, k = 1,...,p$ are transformed [2] to coefficients $r_k'^{(2)},\ k = 0,...,M$ of equation (6), corresponding to polynomial $F_a(z)$. For the solution of this equation on interval $[-1;1]$ constraints (8) are used. After the definition of intervals, containing one root of function $f(x) = \sum_{k=0}^{M} r_k'^{(2)} x^{M-k}$, exact values of roots are determined by Newton's method.

Obtained roots $x_1, x_3, ..., x_{p-1}$ correspond to LSF $\omega_p, \omega_{p-2}, ..., \omega_2$. To compute the remaining frequencies coefficients of equation (6), corresponding to $F_s(z)$ are calculated. Due to the LSF ordering property (4) remaining roots $x_2, ..., x_{p-2}, x_p$ belong to intervals $[x_1;\ x_3], ..., [x_{p-3}; x_{p-1}], [x_{p-1}; 1]$ respectively. Their values are also determined by Newton's method. At last, roots of equations (6) are mapped to LSF by transformation $\omega = \arccos(x)$.

It worth to note, that described localization of LSF with the help of $F_a(z)$ polynomial was found to be more computationally efficient in comparison with $F_s(z)$ polynomial. It can be explained by different distribution characters of LSF, corresponding to $F_a(z)$ and $F_s(z)$ [2,3].

4. EXPERIMENTS

In this section proposed LSF calculation method is applied to the processing of real speech signals. Different modifications of method were evaluated for even AR orders from 8 to 20. Speech data of 6 speakers (4 male and 2 female) with total length of 30 minutes were used in our experiments. All utterances were digitized with sampling frequency 8000 Hz. Procedure for every AR model order was as follows. On every time interval of 20 ms (160 samples) AR coefficients were computed by autocorrelation method [1]. Then obtained AR coefficients were transformed to equivalent sets of LSF. At that attention was paid to average number of recursive divisions of initial interval (table 1) and to average number of arithmetic operations per second (table 2). The convergence criterion was based on the value of function: $|f(x)| < 10^{-6}$. For every AR order p approximation orders from 3 to $p/2$ were considered (derivatives of order, higher than $p/2$, are equal to zero).

Table 1. Number of recursive divisions for different approximation orders

	p=8	*p=10*	*p=12*	*p=14*	*p=16*	*p=18*	*p=20*
Appr. order							
3	9.05	14.51	23.58	37.29	54.73	83.86	122.16
4	8.63	9.97	15.28	18.58	26.54	35.59	47.15
5		9.78	13.11	15.61	19.12	23.05	28.99
6			13.00	14.80	17.88	20.60	22.17
7				14.77	17.70	20.08	21.72
8					17.69	20.00	21.62
9						20.00	21.62
10							21.62

Table 2. Number of arithmetic operations for different approximation orders

	p=8	*p=10*	*p=12*	*p=14*	*p=16*	*p=18*	*p=20*
Appr. order							
3	**45530**	86520	152200	257430	405080	657380	997360
4	47190	**71940**	122260	170320	259810	375640	527510
5		74510	**115980**	**159650**	218820	292200	393190
6			119860	160480	**217940**	**283220**	**339130**
7				165340	224070	287960	348060
8					230030	295910	358560
9						302700	368060
10							375390

Table 1 shows that increasing of approximation order leads to reduction of recursive divisions. This fact can be explained by more close approximation of function by Taylor series. However, after some optimal approximation order calculations described in lemmas 1 and 2 become too complicated and it causes increasing of computational expenses. As can be seen from table 2, optimal approximation order monotonically increases with the growing of p . Optimal amount of elementary operations per second varies from 45530 (at *p*=8) to 339130 (at *p*=20).

For the largest AR model order $p = 20$ (and corresponding optimal approximation order) maximum detected number of recursive divisions was only 29. So, developed algorithm has no time delays and is characterized by low computational expenses. These results indicate that proposed method can be easily implemented in real-time processing systems.

Now, let's compare the efficiency of proposed algorithm with method of Kabal and Ramachandran [2] since it is most widely used in speech compression algorithms. The case of AR order $p = 10$ will be considered. The necessary step of grid for Kabal and Ramachandran method was found to be $\delta = 0.02$.

For the objectivity of comparison it is necessary to use convergence criterion based on uncertainty of root position: $| x_k - x_{k-1} | < \varepsilon$, where x_{k-1} , x_k - approximations of root value, obtained on successive iterations. Two requirements of accuracy are considered: $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-6}$. Table 3 shows average number of arithmetic operations per second for the method of Kabal and Ramachandran and proposed method. Since Kabal and Ramachandran used root refinement by bisection method, we also consider combination of their technique with Newton's method. Computational expenses for the localization of single-root intervals are also considered.

Table 3. Comparison of proposed algorithm with method of Kabal and Ramachandran

Method	**Localization**	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-6}$
Proposed method	**39750**	**64550**	**76550**
K.-R. method	61050	107500	177000
K.-R. + Newton method	61050	74980	87420

According to table 3, advantage of proposed method over Kabal-Ramachandran's is about 35% of operations during the localization of roots. Also gains of 40% and 57% are obtained for accuracies $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-6}$ respectively. There is also an advantage over the combination of Kabal-Ramachandran algorithm with Newton's method. However, for practical implementation in real-time systems it is necessary to estimate maximum number of operations per one speech frame (in this context method of Kabal and Ramachandran was always superior over other existing methods, since its computational expenses are constant). That is why we determined maximum numbers of operations of proposed method for the localization of roots and their computation with accuracies $\varepsilon = 10^{-3}$ and 10^{-6} . These extreme values were found to be 1206, 1798 and 2200 respectively and they are lower than corresponding quantities for the method of Kabal and Ramachandran (1221, 2150 and 3540).

Generally, a lot of practical problems require solution of transcendental or polynomial equations. Most of existing methods use predefined uniform grids of points, in which the behavior of function is analyzed. The principal advantage of proposed approach is that it exploits nonuniform grid, which is formed adaptively for the analyzed function.

5. CONCLUSIONS

A novel method for the computation of line spectral frequencies was proposed. For this purpose universal algorithm of solution of transcendental equations which do not have multiple roots was developed. The main parameters of proposed algorithm are the approximation orders of function and its derivative. Different modifications of this algorithm were applied to the solution of LSF problem. It was shown that resulting method provides arbitrary high accuracy, guarantees stability of corresponding autoregressive filter and does not require any a priori information about LSF location. Experiments with real speech signals indicated that proposed method does not have time delays and has low computational expenses. Advantage of proposed technique over widely used method of Kabal and Ramachandran was shown.

6. REFERENCES

[1] L. R. Rabiner and R. Schafer, *Digital processing of speech signals*, Englewood Cliffs, NJ: Prentice Hall, 1978.

[2] P. Kabal and R. P. Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials", IEEE Trans. ASSP, vol. 28, pp. 562-574, 1980.

[3] C.-H. Wu and J.-H. Chen, "A novel two-level method for the computation of LSP frequencies using a decimation-in-degree algorithm", IEEE Trans. Speech Audio Processing, vol. 5, pp. 106-115, 1997.

[4] J. Rothweiler, "A rootfinding algorithm on line spectral frequencies", Proc. ICASSP-99, vol.2, pp. 661-664, 1999.

[5] C.E. Schmidt and L. R. Rabiner, "A study of techniques for finding the zeros of linear phase FIR digital filters", IEEE Trans. ASSP, vol. 25, pp. 96-98, 1977.