FAST FORMANT ESTIMATION BY COMPLEX ANALYSIS OF LPC COEFFICIENTS

Juan-Luis García Zapata[1], Juan Carlos Díaz Martín[2], Pedro Gómez Vilda[3]

[1]Departamento de Matemáticas, [2]Departamento de Informática, Universidad de Extremadura.
[3]Departamento de Arquitectura y Sistemas Informáticos, Universidad Politecnica de Madrid.
{jgzapata, juancarl}@unex.es
pedro@pino.datsi.fi.upm.es

ABSTRACT

This paper describes a selective root finding method based in residue theory. It can find the poles of the LPC model of the speech signal close to the unit circle, without spend computations with the lesser significant internal poles. This feature makes it faster than generic root-finding methods, if the founded poles complain certain specifications. With a high order LPC model, the selected poles are in better correspondence with the formants than the local maxima of the spectral envelope. Experimental results are showed.

1. INTRODUCTION

The bands of resonance in the spectrum of the speech signal (known as formants) are important features in a compact and significant representation of this signal. They are used, among others applications, to estimate vocal track shapes, speech synthesis and speech coding ([1]). In Automatic speech recognition (ASR), the estimated formant parameters (central frequency and bandwidth) can be components of the acoustic vector or template. These parameters can be directly used (as "raw data" in [1] terminology) as in [2, 3], or used after a formant tracking process [4, 5]. In ASR applications the relative position of formants is the main parameter in vowel classification, and the formant trajectories are tied with articulatory points, as revised in [4].

For the formant extraction, the more frequently reported techniques are based in the linear prediction analysis (LPC) [1]. With this all-pole model of the speech signal, three approximations to formant detection are used. The spectrum of the LPC model is a smoothed approximation of the signal power spectrum, and then maxima of the model spectrum match the central frequency of formants [6]. Other approximation is to choose (by adaptive or statistics means) a set of values for formants positions and bandwidths, in order to match as much as possible the energy distribution of the signal spectrum [7, 2]. Finally, other approximation is the analysis of the transfer function of the LPC model not restricted to unit circumference ("off-axis"). The poles in the inner circle are the sources of the spectral distribution [8, 9].

Our interest in formant extraction comes from the development of IVORY, a methodology for speech recognition in adverse environments [10]. In this system, the input signal is adaptively filtered to cancel noise, before the recognition stage. As a byproduct of this filtering, the LPC polynomial of order 32 (inverse of transfer function) is obtained. The information about formants extracted from this polynomial can be used in later recognition stages. The above cited methods of extraction, spectral maxima and statistical matching, cannot be applied safely to this high order model due to effects of merged formants and spurious peaks [11].

On other hand, the poles of the transfer function (the roots of the LPC polynomial) with influence in the spectral shape are the four or five ones closest to the unit circle. But with a general purpose root finder, all the 31 roots are founded (and it is needed to work out with high precision [12]). It is not possible to restrict the search to specific regions of the complex plane. This makes the root method has high computational load, and precludes its use.

In this paper is described a method able to select a zone of the plane, and to focus the calculus inside this area. It is based in complex analysis, following [9, 13]. In the next section the theoretical basis are revised. In section 3 features of implementation are commented. The computational requirements are determined in section 4, and section 5 shows experimental results in a DSP system.

2. COMPLEX ANALYSIS AND ROOT FINDING

The LPC polynomial $A(z)$ can be viewed as an holomorphic function from z-plane to w-plane. A mathematical result which relates the ceros and poles of a function with local characteristics is the residue theorem [13]. In the especial case of a polynomial, this theorem assures that the number of roots inside an area of z-plane bordered by a closed curve Γ, equals the number of loops surrounding the origin, in w-plane, of the transformed curve $A(\Gamma)$. This transformed curve is the evaluation of LPC polynomial in each point of the curve Γ.

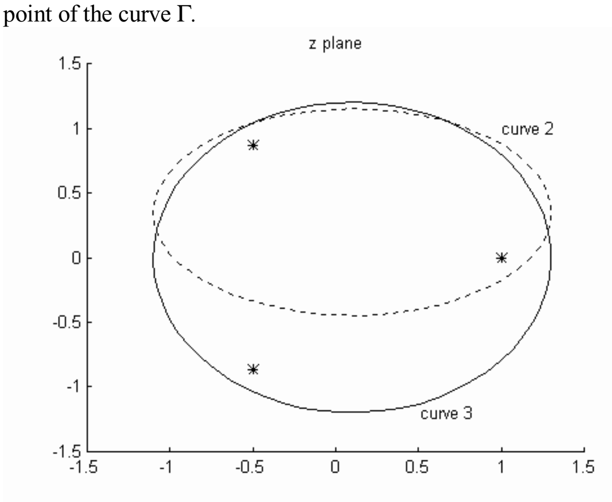

Figure 1: Two examples of Γ curves

In a graphic example, Figure 1 shows two closed curves. Let us consider the polynomial $w = A(z) = z^3 - 1$, which have the three marked points as roots. Curve 2 surrounds two roots and curve 3 the three ones.

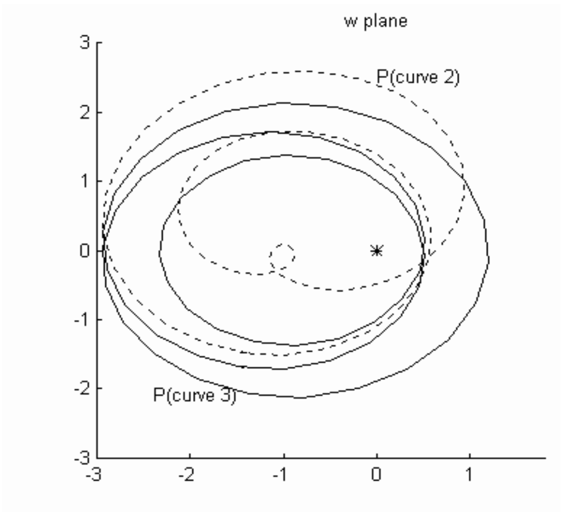

Figure 2: The transformed curves by $w = z^3 - 1$ have winding number of two or three.

This relationship allows us to know if a root is present inside certain area. If this zone is subdivided in smaller regions, a new application of the theorem restrict the area were the root can be situated, then increasing the precision of the root's estimation.

This is in contrast with standard methods of root finding, as Newton-Bairstow, whose iterations spread over complex plane. The convergence of iteration methods only is restricted to an area if the initial point is placed in a narrow neighborhood of a root. The theorem of residues can be used to avoid this spread.

3. IMPLEMENTACION

The discrete model of a curve in complex plane is a linked list of complex points, outlining the continuous shape. So the vertexes of a polygonal $\Gamma^\wedge$ is stored as approximation of the curve Γ in z-plane. The distance between consecutive points of this polygonal $\Gamma^\wedge$ are determined by a parameter R. To increase the resolution in the representation of Γ curve is enough to set this parameter to a lesser value.

The numerical calculation of the number of windings around the origin in w-plane, of the curve $A(\Gamma)$ is performed as described by [13]. The w-plane is divided in eight sectors, such as each quadrant is composed of two of these octagonal wedges. These sectors are numbered from 1 to 8 counterclockwise, with first and eighth wedge placed, respectively, over and under the positive real semi axis, as shown in figure 3.

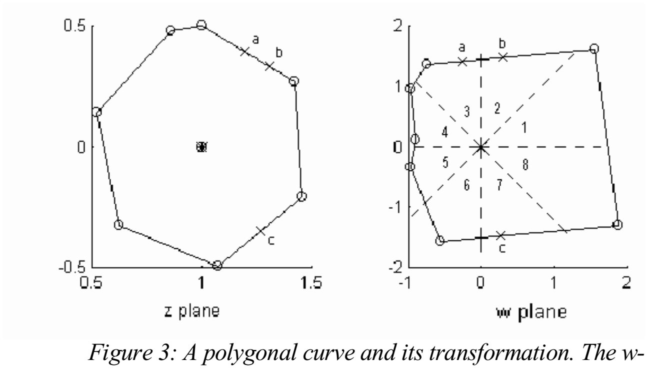

Figure 3: A polygonal curve and its transformation. The w-plane shows the octagonal wedges numbered

The number of twists of the curve $A(\Gamma)$ must be calculated from the list of vertexes of transformed polygonal $A(\Gamma^\wedge)$. These vertexes in w-plane are the evaluation of the LPC polynomial $A(z)$ in each vertex of the z-plane polygonal $\Gamma^\wedge$. Therefore, following the list of vertexes in w-plane, a point in the eighth wedge preceding one belonging to first one, is equivalent to a cross of the continuous curve $A(\Gamma)$ over positive real axis, and hence corresponding to a loop around the w-origin. It should be noted that, in Complex Analysis, the curve orientation are taken counterclockwise. Then, the winding number can be evaluated by the number of steps from eighth wedge to first one (the steps from first to eighth must be counted negatively).

For the accuracy of this vertex-following calculation, it is required that consecutive vertexes of $A(\Gamma^\wedge)$ should be in consecutive wedges, to allow the loops of the continuous curve $A(\Gamma)$ be properly represented. To fulfill this requirement, the polygonal $A(\Gamma^\wedge)$ is scanned, prior to loop counting. If two consecutive vertexes are not in consecutive wedges, an interpolation point is inserted in z-plane (following the shape of the original curve Γ), and then transformed to the w-plane. This insertion increases the resolution of $\Gamma^\wedge$ and $A(\Gamma^\wedge)$. The insertion is performed as many times as necessary to complain the requirement of consecutive wedges in the curve $A(\Gamma^\wedge)$. Figure 3 shows this process of interpolation. Ten points are necessary to follow a path of consecutive wedges in w-plane. The seven initial points of $\Gamma^\wedge$ are marked with O. The three interpolated points are marked with X. The image of the first interpolated point *a* not cover the wedge gap, and then point *b* is inserted.

There is a situation where the residue theorem is not correct. If the curve Γ contain a root in border (known as a "singular contour"), the transformed curve $A(\Gamma)$ contains the origin of w-plane, and the notion of "loop around origin" loose its sense. In our use of the theorem, the curves frequently contain roots or have roots situated very nearby. In the model of the octagonal wedges, the curve $A(\Gamma)$ cross over origin, and points of this curve at both sides of the origin belongs to not consecutive wedges. Therefore, the insertion process described above cannot reach a finish.

To detect this situation of singular contours, in the implementation there are a check of the inserted points in z-plane. If they are placed at a distance below a parameter prefixed Q, a root in border (or at distance lesser than Q of border) is detected. The value of Q should be small compared with the size of contour and is related with the computational load of the procedure, as explained in next section. In the case a singular contour is detected, it is altered to make possible the application of the procedure. A detour is taken around the zone which contains the root (as seen in figure 4). The modified curve has the root situated at distance greater than Q, hence is a non singular contour, and the above loop counting process can be applied.

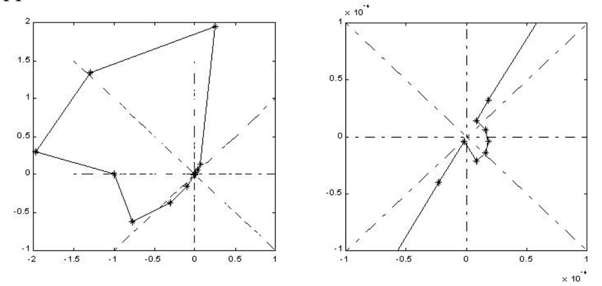

Figure 4: Two images of w-plane. The curve has a root in border, and when the insertion process reach a limit Q, a detour is taken. An enlargement is showed in the second image

Another complexity of implementation comes from the division of the area surrounded by Γ in smaller sub areas to make recursive use of the procedure. The resultant areas must have lesser diameter, and this splitting must be carefully done. The method finally chosen is to divide the area in four parts, cutting across two lines, vertical a horizontal, intersecting in an interior point. To ensure that the resultant sub areas are lesser in diameter than the initial area, this interior point must be near the geometric center. To compute this center point is difficult in general shapes. The center of the hexagon circumscribed to initial area is taken as interior point to make the vertical and horizontal scissions, and can be demonstrated that the sub areas have a diameter halved than the initial one.

If a root is placed inside initial area, one of the new traced borders (vertical or horizontal) can cross over a root, generating a singular contour. The detour process above is performed in such manner that the root belong to only one of the sub areas, and a distance greater than Q of the other ones. This fact is used in the computational analysis.

4. COMPUTATIONAL REQUIREMENTS

To evaluate the computational load of this procedure, the main operation is determination of $A(z_0)$ for a complex z_0, that is, the polynomial evaluation (PE). A polynomial of degree 31, as the inverse of the order 32 LPC transfer function, can be evaluated in 17 multiplications with a preconditioning technique [14]. Each of these 17 complex multiplications can be performed with three float operations; hence a PE is roughly equivalent to 41 floats operations.

4.1 Roots inside a single curve

We estimate firstly the number of PE needed to determine the roots inside a curve Γ. Each vertex of the polygonal Γ^ have to be evaluated, so the number of EP is the number of vertexes on the polygonal Γ^, after the process of insertion of interpolated vertexes described in section 3.

The vertexes of Γ^ previous to the insertion process are related with the parameter R, the resolution of the polygonal Γ^ approximating the continuous curve Γ. If the perimeter length of Γ is denoted by *Per(Γ)*, the number of vertexes of Γ^ is *Per(Γ)*/R prior the insertion process.

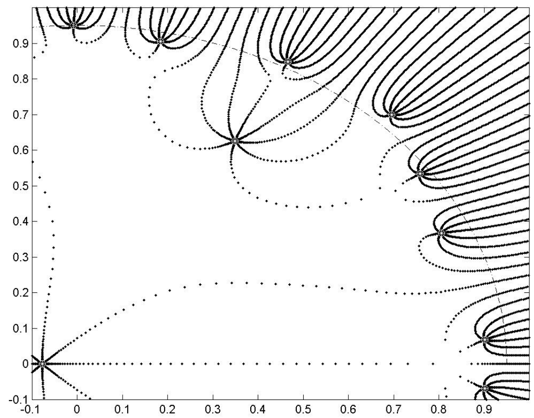

Figure 5: z-plane, with borders between points whose image is in different wedges. A circumference of radius 0.95 is outlined

To give an upper bound of the number of points needed in the insertion process, consider figure 5. It shows a portion of z-plane, with the inverse images (or preimages by the inverse map of *A(z)*) of the limits between octagonal wedges. All the points in a region delimited by this punctured lines, have an image (in w-plane) belonging to the same octagonal wedge. The points where the eight lines meeting, are the roots of the polynomial. These lines (as the roots) are in unknown locations, but are a guide to the deduction of the number of insertions.

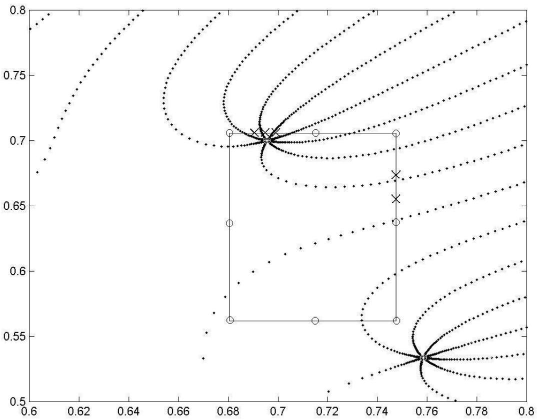

Figure 6: In the upper side of curve Γ, three points are inserted because of the proximity of the root to the border

In figure 6, an enlargement of previous figure is shown, with a curve Γ^ of quadrilateral shape with eight initial vertexes, marked with O. The vertexes due to the process of insertion are marked with X. At the end of the process of insertion there are thirteen vertexes. Let us call *limits points* to the intersections of the punctured lines with the contour Γ (not marked in the figure). The number of vertexes needed by the insertion process can be deduced from the distance of this limits points, in the following manner.

The relation between the punctured lines and the inserted vertexes comes from the fact that, between two consecutive vertexes of the polygonal, only one limit point is allowed. This can be seen because, in the w-plane, between two consecutive vertexes of the transformed polygonal, only a wedge boundary is allowed.

The insertion of interpolation vertexes is performed until the limits points are separated by these vertexes. An upper bound to the number of insertions can be derived from the minimal distance between limits points in the curve Γ. Let be *D_l* this minimal distance between limits points, and recall than R is the distance between vertexes before the insertion process. In general, *D_l* is lesser than R, and several vertexes must be inserted to separate the two limits points located at distance *D_l*. In the worst case, it will be needed n insertions, with n such that $R/2^n < D\_l$. That is $n = \log_2(R/D\_l)$.

To estimate the distance *D_l*, it should be noted that is related with the distance of the root to the curve Γ. While closest to the border is placed the root, lesser is the value of *D_l*. This distance cannot be arbitrarily small, because inthat case the curve Γ must be a singular contour. With the alteration of the curve with a detour, in case of near roots, described in section 3, the distance of any root to the border is greater than the parameter Q. Then, the distance *D_l* cannot be lesser than this value. For this reason, an estimation of the lower possible value of *D_l* is the parameter Q, if this is sufficiently small compared with the size of Γ.
To conclude, each pair of limits point requires n insertions, and each root inside Γ gives eight limits points. Then the number of

vertex insertions required is 8 nr $\log_2$(R/Q) where nr is the number of roots inside Γ.

The total number of EP so is:

$$\frac{Per(\Gamma)}{R} + 8 \cdot nr \cdot \log_2\left(\frac{R}{Q}\right)$$

For example, with concrete values of the parameters used in the implementation, Γ has a perimeter of 0.5, with a resolution R = 0.035, and a Q value of 0.002. The cost of identify the number of roots inside Γ is as most 0.5/0.035 + 8·log2(0.035/0'002) = 15 + 32 = 48 polynomial evaluations for one root, plus others 32 evaluations for each additional one.

4.2 Cost of recursive calls

Let us assume that there are only a root inside the initial curve Γ. With the procedure of division, four regions are produced, each of half diameter than initial one. Then every part is checked, looking for that one which contains the root. Then division is again applied to this part, and so on, until the diameter fall under the accuracy required to the roots.

We denote with *cost*(Γ) the number of EP needed to calculate the number of roots inside Γ, estimated in previous subsection. The cost of the four parts resulting from a division is similar, as they have the same diameter and perimeter. The total cost is then *cost*(Γ) + 4 $cost(\Gamma_1)$ + 4 $cost(\Gamma_2)$ + ... + 4 $cost(\Gamma_p)$ where Γ_i are the curves that outcome from the successive divisions, and p is the number of applications of division.

To show the limitations imposed by this cost, we contrast it with specific values of the parameters. The roots of LPC corresponding with formants of bandwidth lesser than 400 Hz (with a sampling frequency of 22050) have radius (complex module) verifying (-22050/π)·Ln(*module*) < 400 [1]. That is radius > 0.95. The accuracy in frequency of the formants is settle on 10 Hz, which is 0.003 units in z-plane.

The procedure of roots estimation is applied to the circular corona defined by points of radius r with $0.95<r<1$ (outlined in figure 5). The initial perimeter is approximately 1.7, and the diameter 0.4, then the value of p, The number of divisions, is 7 = $\log_2$(0.4/0.003). In these conditions, the above cost formula gives us 1200 EP. This is too much when compared with a generic root-finder (for example Newton-Bairstow method requires approximately 900 EP in similar conditions [12]).

Notwithstanding, with a prescribed accuracy of 0.06 units in z-plane, four calls to division are enough. This gives us a total cost of 690 EP, improving the generic method. This value of $p = 4$ subdivisions is the maximum value (in these conditions) for which the proposed method make lesser EP than generic ones.

5. NUMERICAL RESULTS

The described procedure (with the complex data operations and the preconditioning technique for evaluation of polynomials) has been implemented over a Texas Instruments DSK C6711 system. The audio signal input is LPC modeled with the autocorrelation method, and then the formants frequency and bandwidths is extracted by the described procedure. All the computations (LPC estimation + root-finding) was performed in a fraction of 0.75 of real time. The 60% of this computational power is due to the LPC estimation. This is well suited to the system where the LPC coefficients are obtained from a previous task filtering [10].

REFERENCES

[1] J. D. Markel, A. Gray, *Linear Prediction of Speech.* Springer-Verlag, Berlin, 1976.

[2] L. Welling and H. Ney, "Formant estimation for speech recognition," *IEEE Trans. On Speech and Audio Processing*, vol. 6, no. 1, pp. 36–48, Jan. 1998.

[3] P. N. Garner, and W. J. Holmes, "On the robust incorporation of formant features into hidden Markov models for automatic speech recognition," in *Proc. ICASSP'98*, Seattle, Washington, USA, May 1998, pp. 1-4.

[4] P. Schmid, and E. Barnard, "Robust, n-best formant tracking," in *Proc. EUROSPEECH'95*, Madrid, Spain, Sept 1995, pp. 737-740.

[5] W. J. Holmes and M. J. Russell, "Probabilistic-trajectory segmental HMMs," *Computer Speech & Language*, vol. 13, no. 1, pp. 3–37, Jan. 1999.

[6] L. Rabiner, B. Juang, *Fundamentals of Speech Recognition,* Prentice Hall, New Jersey, 1993

[7] J. P. Olive, "Automatic formant tracking by a Newton-Rapshon Technique," *Journal of the Acoustic Society of America,* vol. 50, no. 2, pp. 661-670, 1971

[8] G. Duncan and M. Jack, "Formant estimation algorithm base don pole focusing offering improved noise tolerante and feature resolution," *IEE Proceedings,* vol. 135, Pt. F, no. 1, pp 18-32, Feb. 1988

[9] R. Snell and F. Milinazzo, "Formant location from LPC análisis data," *IEEE Trans. On Speech and Audio Processing*, vol. 1, no. 2, Apr. 1993

[10] P. Gómez, R. Martínez, A. Álvarez et al, "A DSP-based modular architecture for noisecancellation and speech recognition" *Proc. of the 1008 IEEE International Symposioum on Circuits and Systems,* Monterey, USA, Jun. 1998 pp V.178-181.

[11] S. McCandless, "An algorithm for automatic formant extraction using linear prediction spectra," *IEEE Trans. on Acoustics, Speech and Signal Processing,* vol. ASSP-22, no. 2, Apr. 1974

[12] W. Press, *Numerical recipes in C. Second edition.* Cambridge University Press, Cambridge, 1992.

[13] P. Henrici, *Applied and computacional Complex Analysis*, *vol. 1,* Wiley, New Cork, 1974

[14] D. Knuth, *The art of computer programming, vol. 2.* Addison-Wesley, Massachussets, 1981.