



# Robust and accurate LSF location with Laguerre method

Michał Lenarczyk

Institute of Computer Science, Polish Academy of Sciences  
(IPI PAN)

m.lenarczyk@phd.ipipan.waw.pl

## Abstract

A new algorithm for finding line spectral frequencies, LSF, is introduced, based on Laguerre method of root approximation. The method allows to assuredly find all roots one by one without recourse to polynomial deflation, which allows approximation to a high precision. Error bounds can be estimated by approximating from two sides with added margin. An improved variant of Laguerre recursion scheme is proposed to deal with unfavourable starting points, resulting in faster convergence.

**Index Terms:** line spectral frequencies, LSP, ISP, root finding, Laguerre method

## 1. Introduction

Line spectral frequencies (LSF), introduced by Itakura [1], play an important role in speech coding as a representation of linear predictive filters. Due to favourable quantization and interpolation, this representation is now used in most of the more modern LPC-based speech coders. The disadvantage of LSF's as a representation of filters is that they require polynomial root finding which in general is difficult, but in this case it can be eased thanks to certain known properties of these roots.

The first of the methods taking advantage of these properties was due to Soong and Juang [2] who reduced the complex-domain root finding problem to a real-domain problem. The method was later improved by Kabal and Ramachandran [3] by expressing the polynomials of interest in Tchebycheff base. Both methods applied grid search combined with bisection of sign change intervals to approximate the root positions more closely. This approach faces the problem of setting the initial grid fine enough so that no two roots fall in the same interval, in which case the method fails to detect them. Bisection is also a slow method of root approximation and alternative methods have been considered. For example, Rothweiler [4] showed a way to apply Newton's method such that no LSF's are missed, but due to the use of deflation, estimates of successive roots are less reliable, which can be a problem if the polynomial order is high. Most other methods suffer one of these two problems.

In this article, I consider the question of accurate location of LSF's and propose a new algorithm that accomplishes this task fast and with convergence guarantees. The algorithm builds on the reexpression of the problem in Tchebycheff polynomial base but uses Laguerre's method for approximating the roots. With cubic convergence, it is even faster than Newton's method; moreover, no deflation is needed. Derivations for the method are presented in Section 2 and the algorithm introduced in Section 3. The convergence is examined in Section 4. A modification is proposed to avoid the slowdown the method may suffer in case of unfavourable starting point, and experimental results based on real speech data are presented. Section 5 summarizes and concludes the paper.

## 2. LSF in Tchebycheff polynomial base

A sequence of real coefficients  $a_n, n = 0, \dots, N$  obtained in LPC analysis defines a FIR filter of order  $N$  having frequency characteristics given by the  $z$ -transform formula  $A(z) = \sum_{n=0}^N a_n z^{-n}$ , which is a polynomial in  $z^{-1}$ . The corresponding autoregressive synthesis filter  $H(z) = 1/A(z)$  is stable if its poles, or the roots of  $A$ , are within the unit circle  $|z| < 1$ . Line spectral frequencies are the arguments of complex roots of two polynomials derived from  $A$ , which are referred to as line spectrum pair (LSP), in the following decomposition:

$$\begin{cases} F^{(+1)}(z) &= A(z) + z^{-N-1}A(z^{-1}) \\ F^{(-1)}(z) &= A(z) - z^{-N-1}A(z^{-1}). \end{cases} \quad (1)$$

As shown by Schüssler [5] (for a related decomposition that later became known as the immittance spectrum pair, ISP; since LSP can be considered a special case of ISP, the following derivations are applicable to LSF's obtained either way), all roots of  $F^{(\pm 1)}$  are simple, of unit magnitude, and interlacing on the circumference of the unit circle  $|z| = 1$ . The last property means that between every two roots of one polynomial there is a root of the other polynomial. Writing jointly the roots of both polynomials as  $z_n = e^{i\omega_n}$ , where  $\omega_n$  are the line spectral frequencies, we have

$$\omega_0 < \omega_1 < \dots < \omega_n < \omega_{n+1} < \dots \quad (2)$$

where LSF's at odd indices belong to  $F^{(+1)}$  and at even indices to  $F^{(-1)}$ .

There are two constant roots at 1 ( $\omega_0 = 0$ ) in  $F^{(-1)}(z)$  and  $-1$  ( $\omega_N = \pi$ ) in either  $F^{(-1)}(z)$ , if  $N$  is odd, or  $F^{(+1)}(z)$ , if  $N$  is even, which can be factored out to obtain new polynomials  $G^{(\pm 1)}(z)$  having only complex roots in conjugate pairs. From these roots, the original polynomial  $A(z)$  can be reconstructed. To handle the odd and even orders uniformly, we set  $M^{(-1)} = (N - 1)/2$  and  $M^{(+1)} = (N + 1)/2$  if  $N$  is odd, otherwise  $M^{(-1)} = M^{(+1)} = N/2$ . The following symmetry property then holds for the coefficients  $g_n$  of  $G(z) = G^{(-1)}(z)$  or  $G^{(+1)}(z)$ , where  $M$  is respectively  $M^{(-1)}$  or  $M^{(+1)}$ :

$$g_n = g_{2M-n}, \quad n = 0, \dots, 2M \quad (3)$$

leading to the following representation of polynomials:

$$G(z) = z^{-M} \left[ g_M + \sum_{n=1}^M g_{M-n} (z^n + z^{-n}) \right], \quad (4)$$

where the term in brackets is real. Evaluating  $G$  over  $z = e^{i\omega}$  gives a linear phase frequency spectrum:

$$G(e^{i\omega}) = e^{-iM\omega} \left[ g_M + 2 \sum_{n=1}^M g_{M-n} \cos n\omega \right]. \quad (5)$$

We are now interested in finding zeros of the expression in brackets, since they are the roots of  $G$ . To this end, let  $x = \cos \omega = \text{Re}(e^{i\omega})$ , then the amplitude spectrum can be reexpressed by posing

$$\cos n\omega = T_n(x), \quad (6)$$

where  $T_n(x)$  are Tchebycheff polynomials of the first kind, of order  $n$ . These polynomials follow the recurrence formula

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad (7)$$

which together with initial conditions

$$T_0(x) = 1, \quad T_1(x) = x \quad (8)$$

defines Tchebycheff polynomials for arbitrary  $x \in \mathbb{R}$ . Constraints (7-8) allow the values of Tchebycheff polynomials of subsequent orders, evaluated in a particular point  $x$ , to be expressed as a response of a filter defined by the following equation, where the input is the Kronecker impulse  $\delta_n$ :

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) + \delta_n - x\delta_{n-1} \quad (9)$$

Labeling the expression in brackets in (5) as  $f(x)$ , we have

$$\begin{aligned} f(x) &= g_M T_0(x) + 2 \sum_{n=1}^M g_{M-n} T_n(x) = \\ &= -g_M + 2 \sum_{n=0}^M g_{M-n} T_n(x) \end{aligned} \quad (10)$$

Thus, the problem of finding roots on the unit circle becomes that of finding their real projections in interval  $[-1, +1]$ . We recognize the summation in (10) as a convolution of the sequence of coefficients  $g_n, n = 0, \dots, M$  and the sequence of Tchebycheff polynomials of subsequent orders  $T_n$  evaluated at  $x$ . Denoting

$$h_m(x) = \sum_{n=0}^M g_{m-n} T_n(x) = (g * T(x))_m \quad (11)$$

we have in the  $z$  transform  $H(x, \hat{z}) = G(\hat{z}) \cdot T(x, \hat{z})$ , where  $T(x, \hat{z})$  is from (9)

$$T(x, \hat{z}) = \frac{1 - x\hat{z}^{-1}}{1 - 2x\hat{z}^{-1} + \hat{z}^{-2}} \quad (12)$$

and the symbol  $\hat{z}$  is used to differentiate the domain of subsequent approximations of  $f$  in orthogonal Tchebycheff base from the  $z$ -domain with real part  $x$  that corresponds to sampled time domain. Thus  $x$  and  $\hat{z}$  are two independent variables.

We can now see that since the sequence  $T_n(x)$  is the impulse response of the filter given by (12), instead of computing the convolution (11), we can obtain  $h_n(x)$  by using  $g_n$  as input to the filter (12). The polynomial is then evaluated as  $f(x) = 2h_M(x) - g_M$ . Since we only need the last,  $M$ -th output, the numerator of (12) need only be evaluated once in the final step, while the denominator requires  $M$  multiplications and  $2M$  additions/subtractions. This procedure is computationally equivalent to the Clenshaw algorithm applied by Kabal and Ramachandran [3] in their "reverse recursion", but more conveniently expressed as a filtering operation.

At this stage, the authors of [3] evaluated  $f(x)$  over a grid in order to detect sign changes, further dividing intervals of sign

change in a bisection procedure to narrow down the root position, and performed linear interpolation between endpoints as the final step. The grid size was set to 100 points equally spaced between  $[-1, +1]$  at a sampling rate of 8 kHz based on statistical distribution of LSF's of actual speech. With a fixed number of four bisections, this amounts to less than 150 function evaluations for all LSF's. Industrial implementations of the method use grids that are uniform in angular domain and nonuniform on  $x$ , which better corresponds to the distribution of roots, with only 60 points at 8 kHz [6] and 100 points at 16 kHz [7]. More recently, Bäckström et al. [8] found a theoretical lower bound on the minimum separation of roots of LSP polynomials under the assumption on the maximum roots radius which can be considered met due to regularization in LPC computations. The minimum angular LSF spacing at 8 kHz was found to be  $3,3 \cdot 10^{-5}$ , much lower than the value estimated from speech data in [3]. Using this bound to define a grid with guarantee of finding all roots would grow the number of points by two orders of magnitude, making the computational cost exorbitant. Without that, the grid search may occasionally fail to detect all roots, and vocoders employing this method must be able to recover in such cases.

### 3. Laguerre method for LSF

I propose the use of a different approach that is based on Laguerre's original method of root approximation [9]. The method is applicable to polynomials having only real roots, which is the case in the problem considered.

Any point that is not a root always has a certain neighbourhood in which no roots exist. Laguerre gave the formula for the largest margins on either side of the chosen point defining an interval which is guaranteed to not contain any roots:

$$\delta_{\pm} = \frac{Nf}{-f' \pm \sqrt{(N-1)^2(f')^2 - N(N-1)ff''}} \quad (13)$$

where  $f, f', f''$  denote the values of the polynomial and its first and second derivative with respect to  $x$  and the sign of the square root in the denominator should be taken the same as  $f$  to give a negative (left-hand side) margin:  $\delta_{\text{sgn } f} < 0$  and opposite for positive (right-hand side) margin:  $\delta_{-\text{sgn } f} > 0$ . Either of the endpoints of the interval can then be selected as the new starting point. Hence, it is possible to approach a root in the desired direction using the substitution  $x^{(i+1)} = x^{(i)} + \delta_{\pm}(x^{(i)})$  with an appropriate choice for the sign.

Figure 1 shows the shape of  $\delta_{\pm}$  for an example polynomial. In the roots,  $\delta'_{-\text{sgn } f'} = -1$  and  $\delta''_{-\text{sgn } f'} = 0$ , so the function  $x + \delta_{-\text{sgn } f'}(x)$  has inflection points in the roots and is nondecreasing. Thus, the approximated root can never be overshoot.

To obtain the correction given by (13),  $f$  is calculated as described in the previous section and from (12) it follows that the derivatives can be found by filtering  $g_n$  with filters

$$T'(x, \hat{z}) = \frac{dT}{dx}(x, \hat{z}) = \frac{\hat{z}^{-1}(1 - \hat{z}^{-2})}{(1 - 2x\hat{z}^{-1} + \hat{z}^{-2})^2} \quad (14)$$

$$T''(x, \hat{z}) = \frac{dT'}{dx}(x, \hat{z}) = \frac{4\hat{z}^{-2}(1 - \hat{z}^{-2})}{(1 - 2x\hat{z}^{-1} + \hat{z}^{-2})^3} \quad (15)$$

and taking the last output (scaled by 2). This result generalizes to the following formula for the derivative of arbitrary order:

$$T^{(p)}(x, \hat{z}) = 2^{p-1} p! \frac{\hat{z}^{-p}(1 - \hat{z}^{-2})}{(1 - 2x\hat{z}^{-1} + \hat{z}^{-2})^{p+1}}. \quad (16)$$

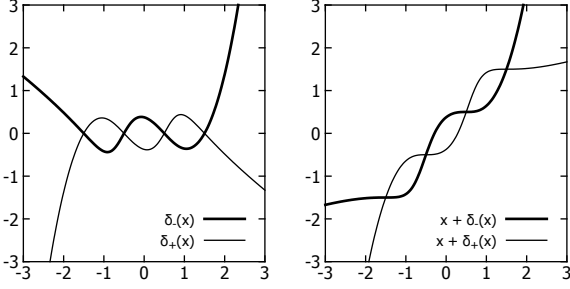


Figure 1: Characteristics of  $\delta_{\pm}(x)$  (left) and  $x + \delta_{\pm}(x)$  (right) for a polynomial with 4 roots at  $-1.5, -0.5, +0.5$  and  $+1.5$ .

Note that this filtering can be efficiently implemented using second-order IIR sections. After passing the sequence  $g_n$  through  $(1 - 2x\hat{z}^{-1} + \hat{z}^{-2})^{-1}$  and evaluating the function, the output sequence is passed again through the same section (with initial state zeroed out) to find the derivative, and so forth. The value of the derivative is obtained by applying the numerator (FIR section) to the filtered sequence which amounts to a difference between coefficients at indices  $M - 1$  and  $M - 3$ . Note that the last coefficient is not used, each subsequent derivative requires filtering of one less coefficient, as reflected by the term  $\hat{z}^{-p}$  in the numerator of (16). The cost of evaluation of subsequent derivative from the byproduct of previous evaluation is thus  $M - p$  multiplications and  $2M - 2p + 1$  additions.

The interlacing property of LSP polynomials makes the problem ideally suited for solving with Laguerre's method: roots are known to be in interval  $]-1, +1[$  so the starting point can be either of its ends. Having converged to a root of one polynomial, it is known to isolate the roots of the other polynomial and can thus be used as a new starting point; no deflation is required.

To formalize the procedure :

1. start at  $x = 1$  and set  $G = G^{(+1)}, M = M^{(+1)}$
2. approach the root from the right until convergence
3. pose  $G = G^{(-1)}$  or  $G^{(+1)}, M = M^{(-1)}$  or  $M^{(+1)}$  on an alternating basis
4. repeat 2 and 3 until all roots are found

## 4. Convergence

### 4.1. Stopping criterion

Since  $\delta_{-\text{sgn } f'}(x) \approx x_0 - x$  in the vicinity of the target root  $x_0$ , the process can be stopped when the magnitude of the correction drops below a threshold of required precision. The problem with this criterion is that  $\delta_{-\text{sgn } f'}(x)$  may also be very small in the vicinity of another root. Iteration starting too close to a previous root would be stopped immediately. The solution to this is an improved variant of the method described below.

### 4.2. Accelerated escape from previous root

Laguerre's method is known to have third-order convergence. It means the number of significant digits roughly triples with each iteration in the vicinity of the root. However, away from the target root, the method slows down. Since  $\delta_{\pm}(x)$  is proportional to  $f(x)$ , the size of the correction step can be arbitrarily small.

To avoid getting trapped in the region of slow convergence close to a previous root, a modification can be introduced that

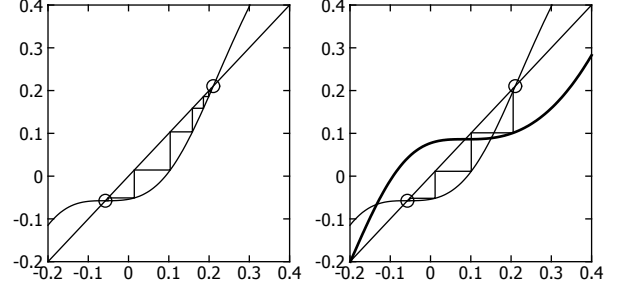


Figure 2: Laguerre iteration in conventional (left) and first-order acceleration (right) variants.

allows to quickly escape from that region. Consider the case of Fig. 2, which was observed in real speech data. With conventional Laguerre method (left), the iteration started from  $x^{(0)} \approx 0.2$  must move inside the wedge formed by  $y = x$  and  $y = x + \delta(x)$ . However,  $f$  always has a minimum or maximum between the two roots, corresponding to a zero of the derivative  $f'$ , which may be easier to approximate, since the initial point is also further away from the previous zero of  $f'$ , resulting in a potentially larger step. This procedure only makes sense when the root of  $f'$  is ahead (in the direction of search) of the current point, i.e. when

$$f f' < 0. \quad (17)$$

This reasoning can be repeated with subsequent derivatives. We begin with Laguerre correction for  $f$  and then, if condition (17) is satisfied, find a correction for  $f'$  calculated with  $f', f'', f'''$  substituted for  $f, f', f''$ . If the result is larger, we take the larger value and proceed to computing the correction for  $f''$ , as long as  $f' f'' < 0$  is met, etc. As can be seen on the plot on the right of Fig. 2, just one step for  $f'$  allowed to escape from the wedge and enter the region of fast convergence.

### 4.3. Precision

When  $x$  varies in close vicinity of the root being approximated,  $x_0$ , the denominator in (13) is practically constant and the value of the correction step becomes directly proportional to the estimate of  $f(x)$  in the numerator. Figure 4.3 shows a badly conditioned case obtained from a synthetically created polynomial, evaluated on a grid with granularity corresponding to the floating point precision at that scale. As can be seen from the plot, there is no unique root position as the evaluation is burdened with error due to rounding. This error cannot be avoided by using fixed point arithmetic because rounding always takes place in IIR filtering. The root finding process can be trapped at any point where the function evaluates to zero, or cycle between two or more points where  $f$  evaluates to nonzero positive and negative values. It is important for stability of the approximation process that the same initial sign be used in all iterations (13) for a given root and not the sign of the current evaluation of  $f$ . Otherwise, if this sign changes, the process will start wandering away in the search of the next root.

Another way to handle this effect is to introduce a safety margin  $\epsilon > 0$ , such that

$$x^{(i+1)} = x^{(i)} + \delta_{\pm}(x^{(i)}) + \epsilon. \quad (18)$$

A value of  $\epsilon$  sufficiently large to prevent the function from changing sign makes it possible to bound roots within intervals by approximating them from the left and from the right and

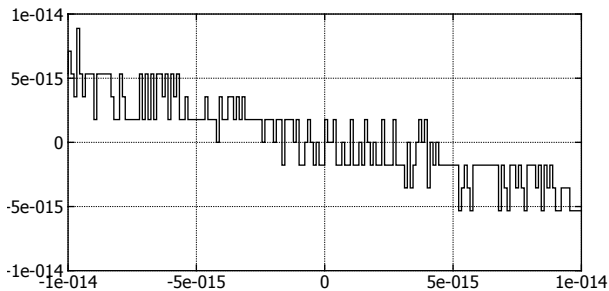


Figure 3: Evaluation of  $f$  in the vicinity of a root  $x_0 \approx 0,826$ , centered at zero.

thus estimate precision. The appropriate margin value depends on the magnitude of errors in the evaluation of  $f(x)$ , which are proportional to the magnitude of coefficients  $g_n$ , and the derivative  $f'(x)$ . It is therefore a parameter dependent on data.

#### 4.4. Experimental results

A large speech data set (*CORPORA* Polish speech corpus) was used for evaluating the performance of the proposed method. The corpus contains speech sampled at 16 kHz which was then LPC coded with order  $N = 15$ . The LSF's have been determined to a high accuracy by approaching roots from the right and from the left using the proposed method. This can be done without changing the algorithm by posing  $a_0, -a_1, a_2, -a_3, \dots$  in place of the original coefficients, yielding negative LSF's in the cosine domain, in reverse order. The mean of the two estimates served as the reference for estimating the accuracy achieved in subsequent steps. In the experiment, double-precision IEEE floating point arithmetic was used with  $\epsilon = 10^{-15}$ , which sets a limit on achievable precision. For comparison, the bisection approach of Kabal and Ramachandran was used with 100 grid intervals (uniform in angular domain) followed by a variable number of bisections and a final linear interpolation.

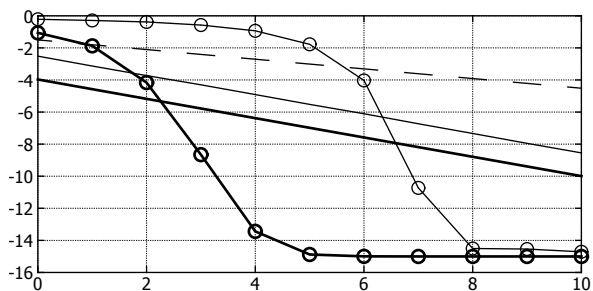


Figure 4: Precision achieved with conventional bisection and the new method for different number of steps. Log deviations from reference in average (bold) and worst case (thin) for bisection and the proposed method (marked with circles). Dashed line shows theoretical precision guarantee given by bisection.

Figure 4 shows the plots of mean precision as a function of the number of steps of Laguerre iteration (bold line with circles) and bisection (bold line without markers). Thin lines show respective worst cases observed in the entire set of over 125000 speech frames, and the dashed line represents the theoretical

iterations	1	2	3	4	5
conventional	0	3,51	19,64	72,53	93,04
accelerated	0	3,51	19,64	72,53	97,03

iterations	6	7	8	9	10
conventional	97,69	99,00	99,49	99,79	99,96
accelerated	99,85	100	100	100	100

Table 1: Percent convergence to within  $3 \cdot 10^{-15}$  in subsequent iterations for conventional Laguerre and the accelerated variant

precision guarantee of bisection. The initial level of precision corresponds to maximum grid size and each bisection step improves it by a factor of 2. Clearly, this upper bound overestimates the actual error both in the average and the worst case, because the effect of interpolation improves at small scales, where nonlinearity vanishes. In contrast, the initial deviation of the proposed algorithm is larger, but on average becomes significantly smaller after only 3 steps and reaches target level within 5 steps. The most difficult case observed required 8 iterations.

It is interesting to know how the accelerated escape affects the convergence of the algorithm. Table 1 shows the percentage of converged roots as a function of the number of iterations for both variants. As read from Figure 4, all cases converged within 8 iterations in accelerated variant and the maximum deviation from reference after 10 steps was less than  $2 \cdot 10^{-15}$ . In contrast, there is a certain percentage of cases where the conventional variant did not converge to the defined level of precision (the observed worst case error was as large as  $10^{-3}$ ). Clearly, the improved method speeds up convergence in difficult cases.

## 5. Discussion

The proposed method gives different kind of guarantees than existing approaches. Similar to Rothweiler's method, it allows to find all LSF's one by one, but since deflation is not needed, they can be tracked to a high degree of precision which the previous method cannot assure. Moreover, using two-sided approximation with margin  $\epsilon$ , a bounding interval can be obtained for each root. Apart from optional  $\epsilon$ , there are no other parameters to tune and the method will always work. In contrast, the method of Kabal and Ramachandran requires adaptation of grid for a given sampling rate and prediction order. It assures a computational bound at a given precision, but without the guarantee of always finding all LSF's, unless an unrealistic grid size is used.

One step of Laguerre's method is computationally more costly than bisection by a factor of roughly 3, but it avoids grid search and when high accuracy is required, its fast convergence justifies the additional expense. While there is no theoretical computational bound, in practice, the process can be stopped after a fixed number of steps. The formula (13) involves square root extraction and division, which is not a problem on machines with floating point arithmetic but may be a challenge in fixed-point DSP applications. The proposed solution seems most appropriate in situations demanding high precision and where nonconvergence is not acceptable.

## 6. Acknowledgements

The study is co-financed by the European Union under the European Social Fund. Project PO KL "Information technologies: Research and their interdisciplinary applications", Agreement UDA-POKL.04.01.01-00-051/10-00.

## 7. References

- [1] F. Itakura, "Line spectrum representation of linear predictor coefficients of speech signals," *The Journal of the Acoustical Society of America*, vol. 57, no. S1, 1975.
- [2] F. Soong and B.-H. Juang, "Line spectrum pair (LSP) and speech data compression," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84.*, vol. 9, Mar 1984, pp. 37–40.
- [3] P. Kabal and R. Ramachandran, "The computation of line spectral frequencies using chebyshev polynomials," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 6, pp. 1419–1426, Dec 1986.
- [4] J. Rothweiler, "A rootfinding algorithm for line spectral frequencies," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 2, Mar 1999, pp. 661–664 vol.2.
- [5] H. Schüssler, "A stability theorem for discrete systems," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 1, pp. 87–89, Feb 1976.
- [6] ETSI EN 300 726, *Digital cellular telecommunications system (Phase 2+); Enhanced Full Rate (EFR) speech transcoding; (GSM 06.60 version 8.0.1 Release 1999)*, European Telecommunications Standards Institute.
- [7] ETSI TS 126 190, *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; Transcoding functions (3GPP TS 26.190 version 6.1.1 Release 6)*, European Telecommunications Standards Institute.
- [8] T. Bäckström, C. Magi, and P. Alku, "Minimum separation of line spectral frequencies," *Signal Processing Letters, IEEE*, vol. 14, no. 2, pp. 145–147, Feb 2007.
- [9] E. Laguerre, "Sur une méthode pour obtenir par approximation les racines d'une équation algébrique qui a toutes ses racines réelles," *Nouvelles annales de mathématiques*, Sér. 2, t. 19, pp. 161–171, 193–202, 1880.