



AALBORG UNIVERSITY
DENMARK

Aalborg Universitet

Leja ordering LSFs for accurate estimation of predictor coefficients

Pedersen, Christian Fischer

Published in:

Proceedings of the International Conference on Spoken Language Processing

Publication date:

2011

Document Version

Early version, also known as pre-print

[Link to publication from Aalborg University](#)

Citation for published version (APA):

Pedersen, C. F. (2011). Leja ordering LSFs for accurate estimation of predictor coefficients. *Proceedings of the International Conference on Spoken Language Processing*. http://www.isca-speech.org/archive/interspeech_2011/i11_2545.html

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Leja ordering LSFs for accurate estimation of predictor coefficients

C.F. Pedersen

Department of Electronic Systems, Aalborg University, Denmark

cfp@es.aau.dk

Abstract

Linear prediction (LP) is the most prevalent method for spectral modelling of speech, and line spectrum pair (LSP) decomposition is the standard method to robustly represent the coefficients of LP models. Specifically, the angles of LSP polynomial roots, i.e. line spectrum frequencies (LSFs), encode exactly the same information as LP coefficients. The conversion of LP coefficients to LSFs and back, has received considerable attention since mid 1970s when LSFs were introduced.

The present paper demonstrates how Leja ordering LSFs reduce amplification of rounding errors when converting LSFs to LP coefficients. The theory behind Leja ordering and the LSFs to LP coefficients conversion is presented. To supplement theory, numerical experiments illustrate the accuracy gain achieved by Leja ordering LSFs prior to conversion. Accuracy is measured as the root mean square deviation between estimated coefficient vectors with and without prior Leja ordering.

Index Terms: Line spectral frequencies, linear prediction coefficients, Leja order

1. Introduction

Linear prediction (LP) is the premier method for spectral modelling of speech. The coefficients representing the LP model are, however, sensitive to quantization errors, i.e. small errors may lead to detrimental distortions in the spectral domain. In the mid 1970s, a method for robust representation of LP coefficients was introduced, cf. [1]. The method, now known as line spectrum pair (LSP) decomposition, decomposes the LP model's denominator polynomial into LSP polynomials with useful inherent properties; see, e.g. [2, 3, 4] for elaborate presentations of LSP polynomial properties. A notable property is that LSP polynomials' roots are all unit-modulus; hence, they can unambiguously be represented by their arguments (frequencies). Unit-modulus roots appear as vertical lines in the spectral domain, thus the term *line spectral frequencies* (LSFs). Compared to LP coefficients, LSFs quantize well and encode exactly the same information. Therefore, LSFs are predominant when parameterizing analysis and synthesis filters in linear predictive coding of, e.g. speech and audio.

By exploiting properties of LSP polynomials' roots, e.g. they occur in complex conjugate pairs and interlace on the unit circle, LSF estimation is done in \mathbb{R}^1 and very efficient techniques exist. See, e.g. [5] for an effective root estimator based on a Chebyshev series formulation of the LSP polynomials. Matlab[®] incorporate functionality to convert LP coefficients to LSFs and back; the functions are `poly2lsf` and `lsf2poly` respectively. Intrinsic Matlab[®] functions are, in this paper, set in typewriter font. Both of these methods are based on work presented in [5].

However, even in well conditioned cases, significant perturbations may occur when computing polynomial coefficients

from roots, e.g. LP coefficients from LSFs. This may come as a surprise, as the procedure appears simple and straightforward, but rounding errors tend to accumulate. The present paper demonstrates how the accumulation of rounding errors can be suppressed by ordering the LSP polynomial roots prior to computing the LP coefficients. The ordering scheme is known as *Leja ordering* due to the Polish mathematician Franciszek Leja, cf. [6]. Leja ordering, which is in time $\mathcal{O}(n^2)$ [7], is not conducted as part of `lsf2poly`. Inspired by [8], a Matlab[®] implementation of Leja ordering has been published in [9]; the implementation is utilized in the present paper. The papers [10, 11] also contain interesting insights into suppression of rounding error accumulation by Leja ordering and Leja sequences.

The results of the current paper show that by introducing the Leja ordering, the root mean square deviation (RMSD) between true and estimated LP coefficient vectors is in the neighborhood of machine epsilon, ε , up to the tested maximum coefficient vector length of 160. In this paper, $\varepsilon \approx 2.22 \cdot 10^{-16}$, i.e. IEEE 754 double precision. These results are significantly better than what is obtained when estimating the coefficients without prior Leja ordering with `lsf2poly`. Especially without Leja ordering, the RMSD increases as a function of LP coefficient vector length, and for lengths beyond 50, the rounding errors accumulate to such an extent as to dominate LP coefficient estimations.

The remainder of this paper is organized as follows. Section 2 introduces the preliminaries of the study, i.e. LSP polynomials, LSP decomposition, LSF, and Leja ordering. Section 3 presents the proposed method, and in section 4 the method is tested with regard to LP coefficient estimation accuracy. Section 5 presents the results of the tests conducted in the numerical experiment, and in the closing section, section 6, the results are discussed along with future perspectives.

2. Preliminaries

This section introduces the preliminaries of the current paper, i.e. LSP polynomials, LSP decomposition, LSF and Leja ordering.

2.1. Line spectrum pair polynomials

Decomposing the denominator polynomial of a LP model into LSP polynomials is often referred to as LSP decomposition. To introduce the decomposition, the following definition is useful.

Definition 1 *Palindromic and anti palindromic polynomial.* A real polynomial, $a(x) = \sum_{n=0}^N a_n x^n$, is palindromic iff $a_n = a_{N-n}$ and anti-palindromic iff $a_n = -a_{N-n}$.

Note the following properties of (anti-) palindromic polynomials; cf. [2, 3, 4] for elaborate presentations of LSP polynomial properties.

Property 1 (Anti-) palindromic polynomials.

1: Every real polynomial that has all of its roots on the unit circle is either palindromic or anti-palindromic.

2: Conversely, not every palindromic or anti-palindromic polynomial has all its roots on the unit circle.

In LSP decomposition, the idea is to define (anti-) palindromic polynomials with all roots on the unit circle, cf. property 1,2.

Definition 2 LSP decomposition.

Any real polynomial, $a(x)$, of order N can be stated as the sum of a palindromic polynomial, $p(x)$, and an anti-palindromic polynomial, $q(x)$:

$$a(x) = \frac{1}{2}(p(x)+q(x)) \quad \text{where} \quad \begin{aligned} p(x) &= a(x) + x^{N+1}a(x^{-1}) \\ q(x) &= a(x) - x^{N+1}a(x^{-1}) \end{aligned}$$

The LSP decomposition is bijective and the polynomials $p(x)$ and $q(x)$ are referred to as LSP polynomials. Notable properties of $a(x)$, $p(x)$, and $q(x)$, proved in [3], are:

Property 2 LSP polynomials.

1: If all the roots of $a(x)$ are inside the unit circle, then all the roots of $p(x)$ and $q(x)$ are interlaced on the unit circle.

2: Conversely, if the roots of two real polynomials of the same order, one palindromic and one anti-palindromic, e.g. $p(x)$ and $q(x)$, are interlaced on the unit circle, then their sum, e.g. $a(x)$, always has all its roots inside the unit circle.

Polynomial $p(x)$ has a real root at -1, and $q(x)$ has a real root at 1; all other roots occur in complex conjugate pairs. A root vector, e.g. of $a(x)$, is denoted by

$$\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_N]^T \in \mathbb{C}^N$$

2.2. Line spectrum frequency

As LSP polynomials' roots lie on the unit circle they can unambiguously be expressed by their arguments, i.e. frequencies. This leads to the following definition:

Definition 3 Line spectrum frequencies.

LSFs are the arguments (frequencies) of LSP polynomials' roots.

Since LSP polynomials' roots occur in complex conjugate pairs, except for the two real roots at ± 1 , it suffices to determine the LSFs on the upper half unit circle, i.e. in the interval $]0; \pi[$. See [5] for an effective root estimator based on a Chebyshev series formulation of the LSP polynomials. A LSF vector, e.g. for the N 'th order polynomial $a(x)$, that leaves room for the two real roots' arguments can be denoted by

$$\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_{N+2}]^T \in \mathbb{R}_+^{N+2}$$

The roots of a polynomial define the polynomial's coefficients up to scaling. Hence, the LSFs define the LSP polynomials, $p(x)$ and $q(x)$, which in turn define the LP model's denominator polynomial, $a(x)$, cf. definition 2. Further, stability of the estimated LP model is ensured as $a(x)$ will have all roots inside the unit circle, cf. property 2. This illustrates the path of conversion from LSFs to LP coefficients.

Roots on the unit circle can unambiguously be represented by their arguments (frequencies) and appear in the spectral domain as vertical lines; hence, the name line spectral frequencies.

2.3. Leja ordering

Leja ordering proves useful when the coefficients of the LP model's denominator polynomial, $a(x)$, are to be determined accurately from the LSFs. In theory, the mapping between LP coefficients and LSFs is bijective up to order and scaling, but in numerical computations, accumulation of rounding errors can become detrimental. In the present paper, Leja ordering is considered as a remedy to alleviate rounding error accumulation in the LSFs to LP coefficients conversion.

Definition 4 Weighted Leja ordering [8]

$$|\lambda_n| \prod_{i=1}^{n-1} |\lambda_n - \lambda_i| = \max_{n \leq l \leq N} |\lambda_l| \prod_{i=1}^{n-1} |\lambda_l - \lambda_i|$$

for $n = 1, 2, \dots, N$.

For the first root, $n = 1$, the equation reduces to

$$|\lambda_1| = \max_{1 \leq l \leq N} |\lambda_l|$$

For the second root, $n = 2$, the equation is

$$|\lambda_2| \cdot |\lambda_2 - \lambda_1| = \max_{2 \leq l \leq N} |\lambda_l| \cdot |\lambda_l - \lambda_1|$$

Example 1 illustrates that Leja ordering is not unique as the maximization may yield more candidates.

Example 1 Leja ordering in \mathbb{R}^1

$$\begin{aligned} \boldsymbol{\lambda}_{\text{in}} &= [1, 2, 3, 4, 5] \\ \lambda_1 &= 5, & \lambda_2 &= 2 \vee 3, & \lambda_3 &= 4 \vee 1, \\ \lambda_4 &= 1 \vee 4, & \lambda_5 &= 3 \vee 2 \\ \underline{\boldsymbol{\lambda}}_{\text{out}} &= [5, 2, 4, 1, 3] \\ \boldsymbol{\lambda}_{\text{out}} &= [5, 3, 1, 4, 2] \end{aligned}$$

Already, implementations of the Leja ordering scheme exist, e.g. a Matlab[®] implementation - inspired by [8] - is published in [9]. The ordering is in time $\mathcal{O}(n^2)$ [7].

3. Proposed method

Table 1 outlines the proposed method and algorithm for estimating LP coefficients from Leja ordered LSFs. The real coefficient vectors for $a(x)$, $p(x)$ and $q(x)$ are denoted by \mathbf{a} , \mathbf{p} and \mathbf{q} respectively. Root vectors are denoted by $\boldsymbol{\lambda}$ and LSF vectors by $\boldsymbol{\theta}$. The LP polynomial, $a(x)$, is of order N .

-
- 1 Form unit-modulus roots $\boldsymbol{\lambda} \in \mathbb{C}^{2N}$ from $\boldsymbol{\theta} \in \mathbb{R}^N$. Complex conjugates included.
 - 2 De-interlace $\boldsymbol{\lambda}$ and form $\boldsymbol{\lambda}_p, \boldsymbol{\lambda}_q \in \mathbb{C}^N$, cf. prop. 2.
 - 3 Leja order $\boldsymbol{\lambda}_p$ and $\boldsymbol{\lambda}_q$, cf. def. 4.
 - 4 Compute coefficients $\mathbf{a}_p, \mathbf{a}_q \in \mathbb{R}^{N+1}$ by expanding

$$\prod_{n=1}^N (x - \lambda_{p,n}) \quad \text{and} \quad \prod_{n=1}^N (x - \lambda_{q,n})$$

- 5 Convolve real roots ± 1 into \mathbf{a}_q and \mathbf{a}_p respectively.
 - 6 Compute $\mathbf{a} = \frac{1}{2}(\mathbf{a}_p + \mathbf{a}_q) \in \mathbb{R}^{N+1}$, cf. def. 2.
-

Table 1: Outline of the proposed method and algorithm.

The significant difference between `lsf2poly` and the proposed method is the Leja ordering, i.e. step 3 in table 1. The ordering is in time $\mathcal{O}(n^2)$ [7].

4. Numerical experiment

4.1. Experiment setup

In this numerical experiment, the proposed method, cf. table 1, is compared to `lsf2poly`. As Leja ordering is applied to reduce rounding error accumulation, the objective of the experiment is to measure potential improvements in accuracy. This is done by evaluating the root mean square deviation (RMSD) between true LP coefficient vectors and vectors estimated with `lsf2poly` and the proposed method.

4.2. Data material

The data material is generated by converting LP coefficient vectors into LSF vectors. The coefficient vectors are randomized and range in length. Theorem 1 is employed in coefficient vector generation to ensure that all roots of the LP model's denominator polynomial, $a(x)$, lie inside the unit circle. Now, the estimations can be compared with the true LP coefficient vectors.

Theorem 1 *Eneström-Kakeya* [12]

$$\text{If } a(x) = \sum_{n=0}^N a_n x^n \text{ with } a_0 \geq a_1 \geq \dots \geq a_N > 0,$$

then all the roots of $a(x)$ lie outside the open unit disk. Conversely, if $a_N \geq a_{N-1} \geq \dots \geq a_0 > 0$, then all the roots of $a(x)$ lie in the closed unit disc.

Minimum phase sequences are generated, i.e. $a_N \geq a_{N-1} \geq \dots \geq a_0 > 0$ all in \mathbb{R} , by reversing the coefficient ordering of $a_N = 1$, $a_{N-1-i} = a_{N-i} + r$, $i \in [0; N-1]$, and making the polynomial monic. The uniform distribution is denoted by \mathcal{U} and $r \sim \mathcal{U}[0, 1]$. Figure 1 exemplifies a coefficient vector and the pertaining LSFs. In the upper panel, the coefficients are ordered in descending powers, i.e. how Matlab[©] orders polynomial coefficients.

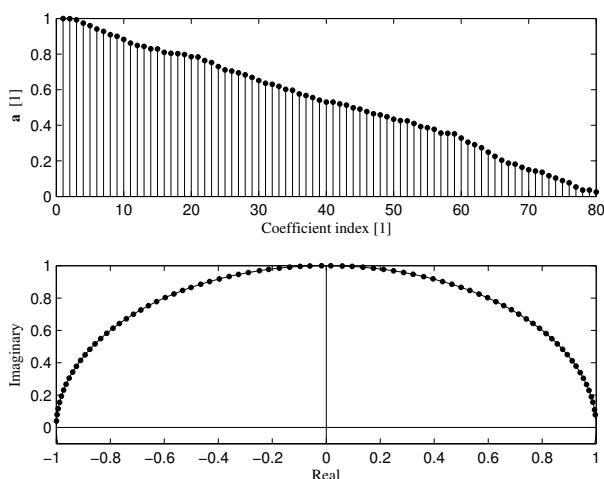


Figure 1: Upper: Coefficient vector, \mathbf{a} , from the data set, $N = 80$. Lower: LSFs computed from the example vector above.

5. Results

In figure 2, a typical example of error between the true and the estimated coefficient vectors is illustrated. The dataset is a single minimum phase sequence of length 80.

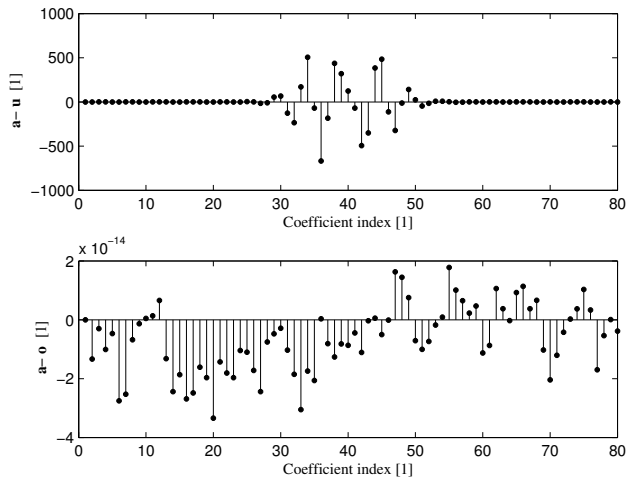


Figure 2: Typical instance of error between true, \mathbf{a} , and estimated coefficient vectors. Upper: Without Leja ordering, \mathbf{u} . Lower: With Leja ordering, \mathbf{o} .

Figure 3 illustrates the RMSD between true and estimated coefficient vectors. The dataset consists of 31 minimum phase sequences that range in length 50-80. The range has been chosen to illustrate the abrupt accumulation of inaccuracies when LSP polynomial roots are not Leja ordered.

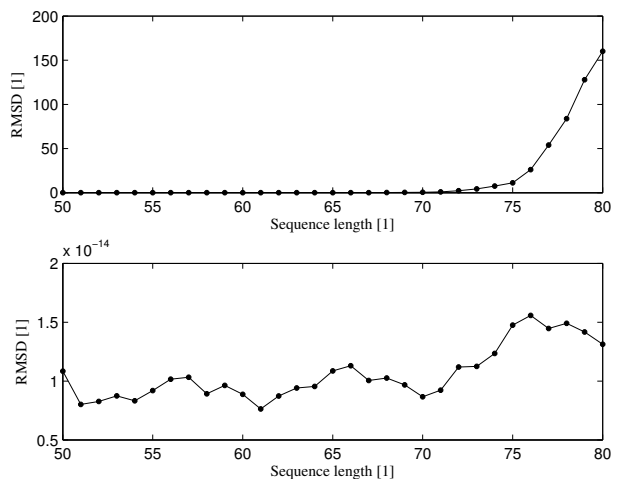


Figure 3: RMSD between true and estimated coefficient vectors. Upper: Without Leja ordering. Lower: With Leja ordering.

Table 2 lists the mean and standard deviation of 14 RMSD populations. For each coefficient sequence length, $N = 50, 60, \dots, 160$, the RMSDs between 50 true and estimated coefficient vectors are obtained. Again, estimations are done with and without Leja ordering.

| | Without Leja ordering Format: $\mu; \sigma$ | With Leja ordering Format: $\mu; \sigma$ |
|-------|--|---|
| N=40 | $2.27 \cdot 10^{-8}; 1.00 \cdot 10^{-8}$ | $4.90 \cdot 10^{-15}; 1.71 \cdot 10^{-15}$ |
| N=60 | $2.08 \cdot 10^{-3}; 9.46 \cdot 10^{-4}$ | $9.62 \cdot 10^{-15}; 3.12 \cdot 10^{-15}$ |
| N=80 | $1.73 \cdot 10^2; 7.04 \cdot 10^1$ | $1.63 \cdot 10^{-14}; 5.03 \cdot 10^{-15}$ |
| N=100 | $1.93 \cdot 10^7; 6.99 \cdot 10^6$ | $2.10 \cdot 10^{-14}; 6.71 \cdot 10^{-15}$ |
| N=120 | $1.70 \cdot 10^{12}; 7.65 \cdot 10^{11}$ | $2.56 \cdot 10^{-14}; 6.92 \cdot 10^{-15}$ |
| N=140 | $1.82 \cdot 10^{17}; 7.97 \cdot 10^{16}$ | $3.20 \cdot 10^{-14}; 1.02 \cdot 10^{-14}$ |
| N=160 | $2.03 \cdot 10^{22}; 7.73 \cdot 10^{21}$ | $4.78 \cdot 10^{-14}; 1.67 \cdot 10^{-14}$ |

Table 2: Population mean and standard deviation of RMSD. The population size is 50 for each N .

6. Discussion

The results express differences in rounding error accumulation with and without Leja ordering the LSP polynomial roots. It is evident, cf. figure 3 and table 2, that the errors accumulate to such an extent as to dominate the LP coefficient estimations when Leja ordering is not applied. Quantization or rounding errors are unavoidable whenever a continuous space is discretized to allow for numerical evaluation. In the present paper, the computations have been done in Matlab[®] using IEEE 754 double precision floating point numbers, i.e. the machine epsilon is $\varepsilon = 2^{-52} \approx 2.22 \cdot 10^{-16}$. That is, between 2^n and 2^{n+1} the numbers are equispaced with increments of 2^{n-52} ; as n increases, the spacing increases. The spacing between 1 and 2, i.e. $n = 0$, yields ε . The rounding procedure is round-to-nearest and round-half-up. Hence, the maximum relative error induced by rounding the result of a single arithmetic operation is $\varepsilon/2$; broadly, the rounding level is about 16 decimal digits.

In speech processing, the order of a linear predictive model is typically 10-12. To fit 4 resonant peaks, i.e. formants, 8 poles are required; a few extra poles may increase modelling accuracy. However, the decrease in prediction error as function of model order is not pronounced beyond order 10-12. From this practical viewpoint, the results in the present paper are mostly of theoretical interest.

7. References

- [1] F. Itakura, "Line spectrum representation of linear predictor coefficients of speech signals," *Journal of the Acoustical Society of America*, vol. 57, pp. S35(A), April 1975.
- [2] Hans W. Schüssler, "A stability theorem for discrete systems," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, pp. 87-89, Feb. 1976.
- [3] Frank K. Soong and Biing-Hwang Juang, "Line spectrum pair (LSP) and speech data compression," in *Acoustics, Speech, and Signal Processing*. IEEE, 1984, vol. 9.
- [4] T. Bäckström and C. Magi, "Properties of line spectrum pair polynomials - A review," *Signal Processing*, vol. 86, no. 11, pp. 3286-3298, February 2006.
- [5] Peter Kabal and Ravi Prakash Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 6, pp. 1419-1426, Dec. 1986.
- [6] F. Leja, "Sur certaines suites liées aux ensembles plan et leur application à la représentation conforme," *Ann. Polon. Math.*, vol. 4:8-13, 1957.
- [7] Stefano De Marchi, "On Leja sequences: Some results and applications," *Elsevier, Applied Mathematics and Computation*, vol. 152, pp. 621-647, 2004.

- [8] Noël M. Nachtigal, Lothar Reichel, and Lloyd N. Trefethen, "A hybrid GMRES algorithm for nonsymmetric linear systems," *SIAM, Journal of Matrix Analysis and Applications*, vol. 13, no. 3, pp. 796-825, July 1992.
- [9] Markus Lang and Bernhard-Christian Frenzel, "A new and efficient program for finding all polynomial roots," Tech. Rep. TR93-08, Rice University ECE, Houston, TX, USA, 1993.
- [10] Lothar Reichel, "The application of Leja points to Richardson iteration and polynomial preconditioning," *Elsevier, Linear algebra and its applications*, vol. 154-156, pp. 389-414, Aug-Oct 1991.
- [11] J. Baglama, D. Calvetti, and L. Reichel, "Fast Leja points," *Electronic Transactions on Numerical Analysis*, vol. 7, pp. 124-140, 1998.
- [12] P. Borwein and T. Erdélyi, *Polynomials and Polynomial Inequalities*, Springer, 1995.