Adaptive Estimation of Zeros of Time-Varying Z-Transforms

Pedersen, Christian Fischer; Andersen, Ove; Dalsgaard, Paul

Published in:
Proceedings of the International Conference on Spoken Language Processing

Publication date:
2011

Document Version
Early version, also known as pre-print

Adaptive Estimation of Zeros of Time-Varying Z-Transforms

C.F. Pedersen, O. Andersen, P. Dalsgaard

Department of Electronic Systems, Aalborg University, Denmark
{cfp,oa,pd}@es.aau.dk

Abstract

In the present paper, a method is proposed for adaptive estimation and tracking of roots of time-varying, complex, and univariate polynomials, e.g. z-transform polynomials that arise from finite signal sequences. The objective with the method is to alleviate the computational burden induced by factorization. The estimation is done by solving a set of linear equations; the number of equations equals the order of the polynomial. To avoid potential drifting of the estimations, it is proposed to verify with Aberth-Ehrlich's factorization method at given intervals.

A numerical experiment supplements theory by estimating roots of time-varying polynomials of different order. As a function of order, the proposed method has a lower run time than Lindsey-Fox and computing eigenvalues of companion matrices. The estimations are quite accurate, but tend to drift slightly in response to increasing coefficient pertubation lengths.
Index Terms: Z-transform zeros, estimation, time-varying

1. Introduction

The computational burden of estimating zeros of time-varying z-transforms of long real-valued signal sequences is rather heavy. The z-transform can be regarded as a real and univariate polynomial where the signal sequence represents the coefficient vector; thus, the zeros are obtained by estimating the roots of the z-transform polynomial. State of the art factorization algorithms like Lindsey-Fox (LF) is in time $\mathcal{O}(n^2)$ and in space $\mathcal{O}(n)$; eigenvalue estimation of a polynomial's companion matrix is in time $\mathcal{O}(n^3)$ and in space $\mathcal{O}(n^2)$ [1]. These methods do not utilize knowledge about prior (coefficient,root)-vector pairs and therefore, re-factorizes for every coefficient vector pertubation. However, the roots of a polynomial are continuous functions of the coefficients of the polynomial [2]. Hence, for well-conditioned polynomials, small coefficient pertubations lead to small root pertubations, and large coefficient pertubations may lead to large root pertubations. Adaptive estimation methods can use this a priori knowledge when adapting a root vector to a perturbed coefficient vector, e.g. an initial factorization is conducted whereupon the entire root vector is updated in response to a coefficient vector pertubation.

Often, factorization algorithms uses deflation and estimates only one root at a time, e.g. the Jenkins-Traub method [3] and LF [1], but this requires that the factorization must be restarted in response to coefficient pertubations which is computationally ineffective in the adaptive case. In the literature, methods that update the entire root vector have already been proposed, e.g. the Durand-Kerner method [4] [5], the Aberth-Ehrlich method [6] [7], inverse iteration [8] [9, chap. 7], and by Gauss-Newton based optimization [8, 10]. However, none of these methods takes as explicit advantage of slowly time-varying root vectors as the method in the present paper. On the other hand, they do not require slowly time-varying coefficient vectors. Also, several algorithms for adaptive root estimation have been proposed, e.g. [11, 8, 12, 13, 10, 14]. Of these, only [14] addresses specifically the situation where coefficients exhibit a relatively slow rate of change. With the slow rate of change requirement it is possible to focus on fast, but locally only, convergent methods as the root pertubations can be assumed small. However, the local convergence constraint may entail drifting or incorrect estimations, if the requirement of slow rate of change is not met. As a novelty, the method proposed in the present paper addresses this problem by verifying the estimations at given intervals; the verification comes at a very modest run time cost.

In the present paper, a method is proposed for adaptive estimation of zeros of time-varying, complex, and univariate polynomials, e.g. z-transform polynomials that arise from finite speech signal sequences. In short, the method is dubbed ARE (Adaptive Root Estimation). The objective with the method is to alleviate the computational burden induced by factorization, e.g. when utilizing the zeros of the z-transform representation in speech signal processing for mixed phase signal separation. The method presented is most accurate when the polynomial coefficients have a relatively slow rate of change as the estimation is locally convergent only. Application wise, this is often the case in analysis of voiced speech where the signal changes relatively slowly compared to the sampling rate. This ratio is due to the underlying physical system, i.e. the speech production apparatus, that varies in relatively slow continuous motions compared to the most common speech signal sampling rates. In contrast, e.g. plosives may entail large coefficient pertubations relative to the sampling rate. The adaptive estimation is supplemented with factorization at given time intervals with Aberth-Ehrlich's method [6] [7] to avoid potential estimation drifting; thus, the estimation error accumulation is bounded. The required frequency of factorizations depend on the coefficients' rate of change; qualifying *relatively slow rate of change* is left to a sequel. Although the proposed method is in time $\mathcal{O}(n^3)$ the results indicate that it outperforms the intrinsic Matlab© function `roots` which outperforms LF for most practically sized input sequence lengths. For longer input sequences, LF gains according to the asymptotic time complexities.

The method proposed in this paper can be applied whenever the method's constraints are respected. Adaptive root estimation of time-varying polynomials may arise for instance in: i) Estimation of arrival direction of acoustic waves by adaptive root-MUSIC (MUltiple SIgnal Classification) where roots represent the angle of arrival [15]. ii) In coding of speech signals by factorization of line spectral pair (LSP) polynomials [16]. iii) In pole/zero-representation of autoregressive moving average (ARMA) processes [13].

Further, the proposed method may be applied in time-invariant cases with empirical polynomials, i.e. polynomials

where some or all coefficients are with limited accuracy and thus may vary in some defined neighborhood. Such coefficient inaccuracies occur due to, e.g. sampling and/or quantization, in almost all practical signal processing applications.

2. Problem formulation

Before posing the problem, some notation is introduced. Quantities denoted by upper case bold face and lower case bold face letters are matrices and vectors respectively; the transpose is denoted by $(\cdot)^T$. Root vectors obtained by numerical factorization, e.g. with Aberth-Ehrlich's method, are considered exact, whereas root vectors obtained by the proposed ARE method are considered estimates. Of course, factorization of high degree polynomials can not be considered mathematically exact, but only exact up to computational precision. The following notation is used to indicate the time of evaluation of an arbitrary time dependent quantity x;

$$x^{(\tau)} = x^{(\tau)}(t) \triangleq x(t+\tau), \quad t, \tau \in \mathbb{Z}_+$$

Now, to the problem. Given a polynomial coefficient vector ordered in ascending powers

$$\mathbf{a}^{(\tau)} = [a_0, a_1, ..., a_{N-1}]^T \in \mathbb{C}^N, \tag{1}$$

the corresponding root vector

$$\boldsymbol{\lambda}^{(\tau)} = [\lambda_1, \lambda_2, ..., \lambda_{N-1}]^T \in \mathbb{C}^{N-1},$$

and $\mathbf{a}^{(\tau+1)}$, the objective is to estimate the root vector

$$\boldsymbol{\lambda}^{(\tau+1)} = \boldsymbol{\lambda}^{(\tau)} + \boldsymbol{\delta} \tag{2}$$

where

$$\boldsymbol{\delta} = [\delta_1, \delta_2, ..., \delta_{N-1}]^T \in \mathbb{C}^{N-1}$$

represents a root pertubation vector. The $\boldsymbol{\delta}$ vector is a finite forward difference in $\boldsymbol{\lambda}$, i.e.

$$\boldsymbol{\delta}^{(\tau)}(\boldsymbol{\lambda}) = \boldsymbol{\delta}^{(\tau)}[\boldsymbol{\lambda}](t) \triangleq \boldsymbol{\lambda}^{(t+\tau)} - \boldsymbol{\lambda}^{(t)}, \quad t, \tau \in \mathbb{Z}_+$$

Unless otherwise stated, $\tau = 1$ and will for brevity be omitted. Table 1 lists the main properties of the proposed ARE method.

1	Rare factorization of the coefficient vector.
2	Takes advantage of small coefficient pertubations.
3	Bounded estimation error accumulation.
4	Tracks entire root vectors.
5	Handles high degree, univariate polynomials.
6	Fast convergence.
7	Restricted to slow rate of change of the coefficients.

Table 1: Main properties of the proposed ARE method.

3. Proposed algorithm and method

3.1. Algorithm outline

In the following sections, the methods needed to realize the algorithm proposed in table 2 are elaborated. Steps 1, 5, and 6 do not need further explanation. Steps 2 and 4 are explained in section 3.4, and step 3 is elaborated in section 3.3. Section 3.2 establishes the relationship between a z-transform's coefficients and zeros.

1	Initialize time indices $t,\ \tau = 0$
2	Factorize $\mathbf{a}^{(\tau)}$ to obtain $\boldsymbol{\lambda}^{(\tau)}$
3	Estimate $\boldsymbol{\delta}$ and $\boldsymbol{\lambda}^{(\tau+1)}$
4	At intervals of τ, factorize $\mathbf{a}^{(\tau+1)}$ to verify $\boldsymbol{\lambda}^{(\tau+1)}$
5	Increment $\tau = \tau + 1$
6	Start over from step 3

Table 2: Outline of the proposed algorithm.

3.2. Zeros of the z-transform

At every time instance, the vectors $\boldsymbol{\lambda}$ and $\mathbf{a}$ are interrelated by the z-transform factorization

$$A(z) = \sum_{n=0}^{N-1} a_n z^{-n} = \frac{a_0 \prod_{m=1}^{N-1}(z - \lambda_m)}{z^{(N-1)}},\ a_0 = 1,\ z \in \mathbb{C} \tag{3}$$

where it is assumed that $\forall\ i \neq j:\ \lambda_i \neq \lambda_j$. In practical cases with voiced speech as generating sequence, this assumption is not restrictive. Turning to matrix notation, (3) can be stated as

$$\mathbf{A}^{(\tau)} = \prod_{n=1}^{N-1} \boldsymbol{\Lambda}_n^{(\tau)} \tag{4}$$

where

$$\mathbf{A}^{(\tau)} = \begin{bmatrix} 1 & & & \\ a_1 & \ddots & & \\ \vdots & \ddots & & \\ a_{N-1} & \dots & a_1 & 1 \end{bmatrix},\ \boldsymbol{\Lambda}_n^{(\tau)} = \begin{bmatrix} 1 & & & \\ -\lambda_n & \ddots & & \\ & \ddots & & \\ & & -\lambda_n & 1 \end{bmatrix}$$

and all the blank entries are zeros. In response to a root pertubation, cf. (2), (4) can be expressed as

$$\mathbf{A}^{(\tau+1)} = \prod_{n=1}^{N-1} (\boldsymbol{\Lambda}_n^{(\tau)} - \boldsymbol{\Delta}_n) \tag{5}$$

where $\boldsymbol{\Delta}_n = \mathbf{L}\delta_n$ and $\mathbf{L}$ is a $N \times N$ lower shift matrix.

3.3. Adaptive estimation of zeros

Not to clutter notation in this section, the superscript, $(\cdot)^{(\tau)}$, on $\boldsymbol{\Lambda}$ is omitted. To estimate $\boldsymbol{\delta}$ in (2), (5) can be restated as

$$\begin{aligned} \mathbf{A}^{(\tau+1)} &\overset{\star}{\approx} \prod_{n=1}^{N-1} \boldsymbol{\Lambda}_n - \sum_{n=1}^{N-1} (\boldsymbol{\Delta}_n \prod_{\substack{j=1 \\ j \neq n}}^{N-1} \boldsymbol{\Lambda}_j) \\ &= \mathbf{A}^{(\tau)} - \sum_{n=1}^{N-1} (\boldsymbol{\Delta}_n \boldsymbol{\Lambda}_n^{-1} \boldsymbol{\Lambda}_n \prod_{\substack{j=1 \\ j \neq n}}^{N-1} \boldsymbol{\Lambda}_j) \\ &= \mathbf{A}^{(\tau)} - \sum_{n=1}^{N-1} \boldsymbol{\Delta}_n \boldsymbol{\Lambda}_n^{-1} \mathbf{A}^{(\tau)} \end{aligned} \tag{6}$$

where the starred approximation is due to ignored second and higher order terms of $\boldsymbol{\Delta}$ [14]. Thus, the approximation error is a function of the root pertubation length, cf. table 1. The reduction is possible since $\boldsymbol{\Lambda}_n$ is invertible, $\det(\boldsymbol{\Lambda}_n) = 1$, and commute under matrix multiplication, $\boldsymbol{\Lambda}_n$ is lower-triangular Toeplitz. The left and right hand sides of (6) are lower-triangular Toeplitz with ones in the diagonal so only the first

column minus the first row needs to be considered. Cf. (1) where $a_0 = 1$, the first column can be expressed as

$$\begin{aligned}\mathbf{a}^{(\tau+1)} &\approx \mathbf{a}^{(\tau)} - \mathbf{L}\sum_{n=1}^{N-1}\delta_n \mathbf{\Lambda}_n^{-1}\mathbf{a}^{(\tau)} \\ &= \mathbf{a}^{(\tau)} - \mathbf{L}\left(\left[\mathbf{\Lambda}_1^{-1}\,\mathbf{a}^{(\tau)}\ \ldots\ \mathbf{\Lambda}_{N-1}^{-1}\,\mathbf{a}^{(\tau)}\right]\boldsymbol{\delta}\right)\end{aligned} \tag{7}$$

where each column vector entry, $\mathbf{\Lambda}_n^{-1}\ \mathbf{a}^{(\tau)}$, is considered a block. If top and bottom row cancellation is denoted by overline and underline respectively, (7) can be formulated as

$$\overline{\mathbf{a}}^{(\tau)} - \overline{\mathbf{a}}^{(\tau+1)} = \left[\underline{\left(\mathbf{\Lambda}_1^{-1}\right)}\,\mathbf{a}^{(\tau)}\ \ldots\ \underline{\left(\mathbf{\Lambda}_{N-1}^{-1}\right)}\,\mathbf{a}^{(\tau)}\right]\boldsymbol{\delta} \tag{8}$$

Note that, as $\mathbf{\Lambda}_n$ is $N \times N$ then

$$\mathbf{\Lambda}_n^{-1} = \begin{bmatrix} 1 & & & & \\ \lambda_n^1 & \ddots & & & \\ \lambda_n^2 & \ddots & & & \\ \vdots & \ddots & & & \\ \lambda_n^{N-1} & \ldots & \lambda_n^2 & \lambda_n^1 & 1 \end{bmatrix}$$

where all the blank entries are zeros. Although $\mathbf{\Lambda}_n^{-1}$ is Toeplitz, the system of linear equations in (8) is not. Toeplitz systems are solvable in time $\mathcal{O}(n^2)$ by, e.g. Levinson-Durbin recursion, whereas the non-Toeplitz case in (8) is in $\mathcal{O}(n^3)$. Hence, the root pertubation vector, $\boldsymbol{\delta}$, for a $N-1$ degree univariate and complex polynomial can be achieved by solving a set of $N-1$ linear equations in $\mathcal{O}(n^3)$.

3.4. Verification of the adaptive estimation

The Aberth-Ehrlich method [6] [7] is a factorization method for simultaneous approximation of all the roots of a univariate and complex polynomial. Given such a polynomial, $p(\mathbf{a}^{(\tau)}, \mathbf{z}) = z^{N-1} + \sum_{n=0}^{N-2} a_n z^n$, with an approximated root vector, $\boldsymbol{\lambda}^{(\tau)}$, the method refines the roots iteratively by:

$$w_k = -\frac{Q}{1 - Q\sum_{\substack{m=1 \\ m\neq k}}^{N-1} 1/(\lambda_k - \lambda_m)}, \quad Q = \frac{p(\mathbf{a}^{(\tau)}, \lambda_k)}{p'(\mathbf{a}^{(\tau)}, \lambda_k)}$$

where $k = 1, \ldots, N-1$ and prime, $(\cdot)'$, is Lagrange's differentiation notation; the method does not require p to be monic. The refined root vector becomes $\boldsymbol{\lambda}_{i+1}^{(\tau)} = \boldsymbol{\lambda}_i^{(\tau)} + \mathbf{w}$, where $\mathbf{w} = [w_1, w_2, ..., w_{N-1}]^T$ and i is the iteration number. The number of iterations needed to reach a given approximation accuracy strongly depends on how close the initial root vector approximation, $\boldsymbol{\lambda}_{i=0}^{(\tau)}$, is to the true root vector, $\boldsymbol{\lambda}_{i=i\_max}^{(\tau)}$. In table 2 step 4, $\boldsymbol{\lambda}_{i=0}^{(\tau+1)} = \boldsymbol{\lambda}^{(\tau+1)}$, and in step 2, $\boldsymbol{\lambda}_{i=0}^{(\tau)}$ is a vector of $N-1$ equispaced points on a circle around origo of the complex plane with a radius centered between the lower and upper Cauchy modulus bounds denoted β_* and β^* respectively. All roots, $\boldsymbol{\lambda}^{(\tau)}$, of the monic polynomial, $p(\mathbf{a}^{(\tau)}, \mathbf{z})$, lie in the annulus [17] [2] [18]:

$$\begin{aligned}|\lambda_n| < \beta^* &= 1 + \max_{0\leq k\leq N-2}\{|a_k|\} \\ |\lambda_n| \geq \beta_* &= (1 + \max_{1\leq k\leq N-1}\{|a_k/a_0|\})^{-1}\end{aligned}$$

where $n = 1, \ldots, N-1$. Cf. [19] for an efficient implementation of the Aberth-Ehrlich method and a general strategy for choosing a good initial approximation.

4. Numerical experiment

4.1. Experiment setup

This experiment indicates the run time and accuracy of a Matlab© implementation of ARE, cf. table 2. The run time of ARE is compared with the run times of the state of the art algorithms: LF and `roots`. Intrinsic Matlab© functions are set in typewriter font. The Matlab© LF implementation is the second version of "lroots", cf. [1]; all diagnostics are turned off to lower run time. The function, `roots`, estimates eigenvalues of a polynomial's companion matrix in time $\mathcal{O}(n^3)$; LF is in time $\mathcal{O}(n^2)$ [1]. Run time is measured with `tic/toc`. Accuracy is measured as the root-mean-square-deviation (RMSD) as a function of absolute polynomial coefficient pertubation lengths. Note that measurements like these are highly computer and implementation dependent and thus only indicative.

4.2. Dataset

The dataset for the accuracy test is divided into subsets:
S01 = $\{\mathbf{a}^{(\tau)}, \mathbf{a}^{(\tau+1)}, \boldsymbol{\lambda}^{(\tau)} : \tau \in [0; 10],\ N = 3\}$,
S02 = $\{\mathbf{a}^{(\tau)}, \mathbf{a}^{(\tau+1)}, \boldsymbol{\lambda}^{(\tau)} : \tau \in [0; 10],\ N = 4\}$,
⋮
S79 = $\{\mathbf{a}^{(\tau)}, \mathbf{a}^{(\tau+1)}, \boldsymbol{\lambda}^{(\tau)} : \tau \in [0; 10],\ N = 81\}$
where
$a_n^{(\tau=0)} \sim \mathcal{U}]0, 1[$, $a_n^{(\tau=1)} = a_n^{(\tau-1)} + \tau\alpha$, ..., $a_n^{(\tau=10)} = a_n^{(\tau-1)} + \tau\alpha$. The continuous uniform distribution is denoted by $\mathcal{U}$, and the coefficient pertubation by α; $\alpha = 0.001$. The dataset for the run time test is similar, only $N = [270; 800]$ to focus on the progressed tendencies. Run times are reported as averages across τ.

5. Results

The proposed ARE runs in time $\mathcal{O}(n^3)$ as systems of linear equations with no apparent exploitable structure need to be solved, cf. (8). Fig. 1 illustrates the run time ratios `roots`/ARE and LF/ARE as functions of polynomial degree. At the cross over point, LF starts to gain from `roots`.

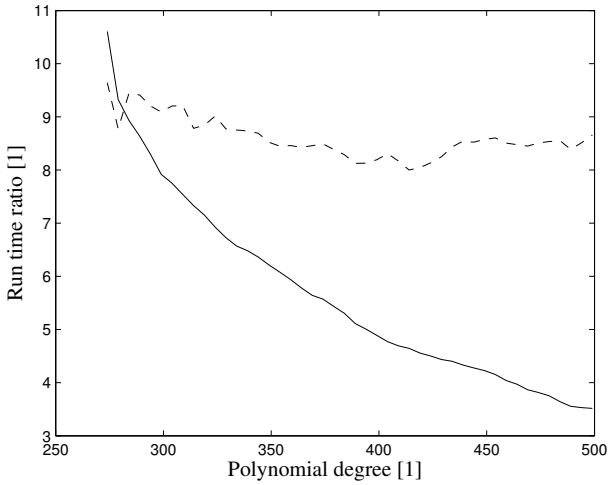

Figure 1: Run time ratios as functions of polynomial degree. Dashed: `roots`/ARE. Solid: LF/ARE.

Fig. 2 illustrates the RMSD between root vectors estimated with `roots` and ARE as a function of pertubation length.

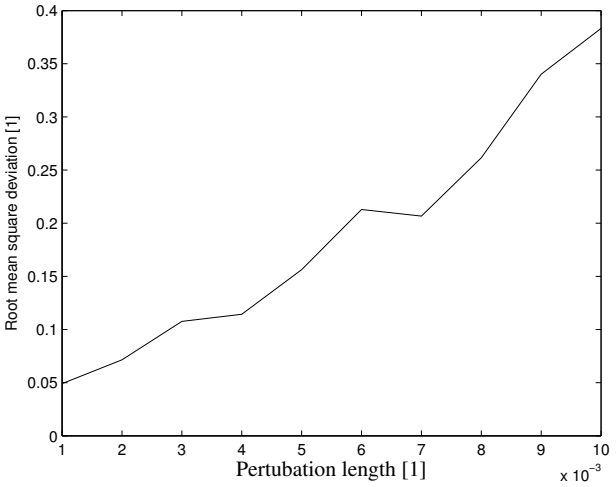

Figure 2: RMSD between results of `roots` and ARE.

6. Discussion

A method for adaptive estimation of the roots of time-varying, univariate, and complex polynomials, e.g. zeros of the z-transform, has been presented. The estimation is done by solving a set of linear equations; the number of equations equals the polynomial degree. To avoid potential drifting of the adaptive estimations, it has been proposed to verify the estimations with Aberth-Ehrlich's factorization method at given time intervals. The potential estimation error of the adaptive method depends on the root pertubation length, cf. (6), and the condition of the matrix to be inverted when solving the set of linear equations, cf. (8).

As `roots` and ARE are in time $\mathcal{O}(n^3)$ their run times will tend to n^3 asymptotically. However, the test case illustrated in fig. 1 indicate that ARE outperforms `roots` for most practically sized input sequence lengths. The polynomial degree range is chosen to illustrate the LF/`roots` cross over point and the progressed run time tendencies. Note that run time measurements like these are computer and implementation dependent and thus only indicative. In [1], a run time comparison between `roots` and LF has been performed; `roots` showed a lower run time for $n \lessapprox 180$. Again, this is highly computer and implementation dependent.

As expected, cf. (6), the ARE estimations tend to drift as a function of polynomial coefficient pertubation length, cf. fig. 2. The roots of a polynomial are continuous functions of the coefficients of the polynomial; hence, for well-conditioned polynomials, small coefficient pertubations lead to small root pertubations. Likewise, large coefficient pertubations lead to long root trajectories and thus, most likely, to large root pertubations.

In a sequel, it would be interesting to devise a dynamic heuristic on when to apply the Aberth-Ehrlich verification instead of verifying at regular intervals. This would address the Achilles' heel of ARE: potential estimation inaccuracies.

7. References

[1] Gary A. Sitton, C. Sidney Burrus, James W. Fox, and Sven Treitel, "Factoring very-high-degree-polynomials," *IEEE Signal Processing Magazine*, vol. 20, no. 6, pp. 27–42, November 2003.

[2] Q.I. Rahman and G. Schmeisser, *Analytic Theory of Polynomials*, Oxford University Press, 2002.

[3] M.A. Jenkins and J.F. Traub, "A three-stage variable-shift iteration for polynomial zeros and its relation to generalized rayleigh iteration," *Numerische Mathematik*, vol. 14, no. 3, pp. 252–263, February 1970.

[4] E. Durand, *Solutions Numériques des Équations Algébriques*, Masson, Paris, 1960.

[5] Immo O. Kerner, "Ein gesamtschrittverfahren zur berechnung der nullstellen von polynomen," *Numerische Mathematik*, vol. 8, no. 3, pp. 290–294, May 1966.

[6] O. Aberth, "Iteration methods for finding all zeros of a polynomial simultaneously," *Mathematics of Computation*, vol. 27, no. 122, pp. 339344, April 1973.

[7] L.W. Ehrlich, "A modified newton method for polynomials," *Communications of the ACM*, vol. 10, no. 2, pp. 107–108, February 1967.

[8] David Starer and Arye Nehorai, "Polynomial factorization algorithms for adaptive root estimation," in *Int. Conf. on Acoustics, Speech, and Signal Processing*, Glasgow, UK, May 1989, IEEE, vol. 2, pp. 1158–1161.

[9] Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 3 edition, 1996.

[10] David Starer and Arye Nehorai, "Adaptive polynomial factorization by coefficient matching," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 527–530, February 1991.

[11] S. Orfanidis and L. Vail, "Zero-tracking adaptive filters," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 6, pp. 1566–1572, December 1986.

[12] J.F. Yang and M. Kaveh, "Adaptive algorithms for tracking roots of spectral polynomials," in *International Conference on Acoustics, Speech, and Signal Processing*, Glasgow, UK, May 1989, IEEE, vol. 2, pp. 1162–1165.

[13] Arye Nehorai and David Starer, "Adaptive pole estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 5, pp. 825–838, May 1990.

[14] David Starer and Arye Nehorai, "High-order polynomial root tracking algorithm," in *International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, CA, USA, March 1992, IEEE, vol. 4, pp. 465–468.

[15] A.J. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *ICASSP*. IEEE, April 1983, vol. 8, pp. 336–339.

[16] T. Bäckström and C. Magi, "Properties of line spectrum pair polynomials - a review," *Signal Processing*, vol. 86, no. 11, pp. 3286–3298, February 2006.

[17] A.L. Cauchy, *Exercises de Mathématiqués*, vol. Oeuvres 2, Paris, 1829.

[18] Christian Fischer Pedersen, Ove Andersen, and Paul Dalsgaard, "ZZT-domain immiscibility of the opening and closing phases of the LF GFM under frame length variations," in *Interspeech*, Brighton, UK, September 2009, ISCA.

[19] Dario Andrea Bini, "Numerical computation of polynomial zeros by means of aberth's method," *Numerical Algorithms*, vol. 13, no. 2, pp. 179–200, September 1995.