Computing the Poles of Autoregressive Models from the Reflection Coefficients[1]

Gregory S. Ammar[2] Daniela Calvetti[3] Lothar Reichel[4]

Abstract

A new approach to the computation of the poles of a stable autoregressive system from the reflection coefficients is proposed. Equivalently, we compute the zeros of Szegő polynomials from the associated Schur parameters. The numerical method utilizes an efficient algorithm for computing the (unimodular) zeros of a unitary Hessenberg matrix; this step can be regarded as the computation of the poles of an associated lossless system. These eigenvalues are then used as starting points for a continuation procedure for finding the zeros of the desired polynomial. The procedure is efficient and parallelizable, and may therefore be suitable for real-time applications.

1. Introduction

Autoregressive models are of fundamental importance in time series analysis and discrete-time control. For example, in linear prediction, one is often given the autocorrelation matrix of a wide sense stationary real-valued signal $\{x_j\}_{j=-\infty}^{\infty}$. This matrix is a real symmetric positive definite Toeplitz matrix

$$M_{n+1} = [\mu_{j-k}]_{j,k=0}^{n}.$$

For each positive integer $p \leq n$, the time series can be modeled as the response of an autoregressive (AR) process of order p driven by white noise; i.e.,

$$x_j = w_j - \sum_{k=1}^{p} \alpha_k x_{j-k}, \tag{1}$$

where $\{w_j\}_{j=-\infty}^{\infty}$ is a white noise process. The original time series is approximated by the AR(p) process $\{\hat{x}_j\}_{j=-\infty}^{\infty}$, given by

$$\hat{x}_j = -\sum_{k=1}^{p} \alpha_k x_{j-k}.$$

The coefficients $\{\alpha_k\}_{k=1}^{p}$ that minimize the mean square prediction error are determined by solving the *Yule-Walker equations* of order p,

$$M_p \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix} = - \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_{p+1} \end{bmatrix}. \tag{2}$$

[1] To appear in the Proceedings of the Thirty-First Annual Allerton Conference on Communication, Control, and Computing.

[2] Department of Mathematical Sciences, Northern Illinois University, DeKalb, IL 60115. E-mail: ammar@math.niu.edu

[3] Department of Pure and Applied Mathematics, Stevens Institute of Technology, Hoboken, NJ 07030. E-mail: calvetti@mathrs1.stevens-tech.edu.

[4] Department of Mathematics and Computer Science, Kent State University, Kent, OH 44242. E-Mail: reichel@mcs.kent.edu. Research supported in part by the National Science Foundation under grants DMS-9002884 and DMS-9205531.

In the z-transform domain, the filter (1) takes the form

$$X(z) = \frac{1}{A_p(z)} W(z),$$

where

$$A_p(z) = \sum_{k=0}^{p} \alpha_k z^{-k} \qquad (\alpha_0 \equiv 1).$$

Since the transfer function $(A_p(z))^{-1}$ has no (finite) zeros, AR processes are also referred to as *all-pole filters*. The above assumptions guarantee that the process (1) is *stable*; that is, the zeros of $A_p(z)$ are all of modulus less than one. See, for example, [17, Chapter 6].

The entries of the Toeplitz matrix M_{n+1} can be regarded as the moments of a distribution function on the unit circle in the complex plane. These moments can be used to construct the associated inner product $< \cdot, \cdot >$ on polynomials of degree at most n by the linear extension of the identity $< \lambda^j, \lambda^k > := \mu_{j-k}$. The monic polynomials $\psi_j(\lambda)$ that are orthogonal with respect to this measure are known as Szegő polynomials in this context. They are also called Levinson polynomials and backward predictor polynomials in the engineering literature.

Szegő polynomials can be recursively computed using the well-known Levinson-Durbin algorithm. This algorithm is equivalent with the use of the recursions formulas introduced by Szegő (see, e.g., [21, 15]) to construct the coefficients of the Szegő polynomials in the power basis. In particular, the Szegő recursions for constructing the monic polynomials ψ_j are given by

$$\begin{aligned} \psi_{j+1}(\lambda) &= \lambda\psi_j(\lambda) + \gamma_{j+1}\tilde{\psi}_j(\lambda), \\ \tilde{\psi}_{j+1}(\lambda) &= \lambda\gamma_{j+1}\psi_j(\lambda) + \tilde{\psi}_j(\lambda), \end{aligned}$$

where $\psi_0(\lambda) = \tilde{\psi}_0(\lambda) = 1$, and

$$\gamma_{j+1} = - < 1, \lambda\psi_j > / < \psi_j, \psi_j > .$$

The recursion coefficients γ_j are known as *Schur parameters*, and also as *Szegő parameters, reflection coefficients*, and *partial correlation coefficients.*

It follows from the Szegő recursions that the auxiliary polynomials $\tilde{\psi}_j(\lambda)$ satisfy $\tilde{\psi}_j(\lambda) = \lambda^j \psi_j(\lambda^{-1})$. These are known as *forward predictor polynomials* in linear prediction. In fact, $\tilde{\psi}_p(\lambda) = A_p(\lambda^{-1})$ for $1 \leq p \leq n$. Consequently, the zeros of $A_p(\lambda)$ and $\psi_p(\lambda)$ coincide.

The coefficients of the Szegő polynomial ψ_n in the power basis provide the *recursive realization* of the autoregressive model of order n. On the other hand, the Schur parameters provide the realization of the AR model as a lattice filter. The latter realization has several advantages over the former. For example, the filter is stable if and only if each γ_j has modulus less than one. This condition can be easily enforced, even if the parameters are corrupted by noise. In contrast, it is not clear how one can enforce the stability of a filter in recursive form.

The zeros of the Szegő polynomial ψ_n provide another parameterization of the AR model. Moreover, the knowledge of the poles is often desired. However, the difficulties in reliably and efficiently computing of the zeros of the polynomial ψ_n often discourages their use in practice.

The most straightforward approach to computing the zeros of the polynomials ψ_n is to use a polynomial root-finder; see, e.g., [20, Sections 5.5-5.10] for a description of a few methods. For several of these methods it is important that the zeros be determined in an appropriate order; otherwise severe loss of accuracy in the computed roots may result. Moreover, in order to obtain globally and quickly converging methods, several iterative techniques may have to be employed. These considerations can make software for reliable and quickly converging rootfinders fairly complicated. Consequently, the problem of finding zeros of a polynomial should be replaced by the problem of computing eigenvalues of a matrix when possible, see, e.g., the the discussion in [20, Section 5.5].

The most obvious way to replace the rootfinding problem with an eigenvalue problem is to consider the Frobenius matrix (companion matrix) associated with the power basis coefficients of the polynomial ψ_n. The Schur parameters determine another matrix whose eigenvalues are the roots of ψ_n. This is the upper Hessenberg matrix

$$H_n = H(\gamma_1, \ldots, \gamma_n) = G_1(\gamma_1)G_2(\gamma_2)...G_{n-1}(\gamma_{n-1})\tilde{G}_n(\gamma_n), \tag{3}$$

where $G_j(\gamma_j)$ is the (orthogonal) Givens matrix of order n that operates in the $(j, j+1)$ coordinate plane,

$$G_j(\gamma_j) = \begin{bmatrix} I_{j-1} & & & \\ & -\gamma_j & \sigma_j & \\ & \sigma_j & \gamma_j & \\ & & & I_{n-j-1} \end{bmatrix}, \qquad 1 \leq j < n.$$

Here I_k denotes the $k \times k$ identity matrix. The matrix $\tilde{G}_n(\gamma_n)$ is the truncated Givens matrix

$$\tilde{G}_n(\gamma_n) = \begin{bmatrix} I_{n-1} & \\ & -\gamma_n \end{bmatrix}. \tag{4}$$

The connection between Szegő polynomials and Hessenberg matrices of the form (3) was uncovered by Gragg [12], who showed that the eigenvalues of H_n are the zeros of ψ_n. More precisely,

$$\psi_n(z) = \det[zI_n - H_n]. \tag{5}$$

If we use the QR-algorithm, see [10, Chapter 7] or [6], for computing the eigenvalues of H_n, then $O(n^3)$ arithmetic operations are required. It is the purpose of this paper to outline a new algorithm that requires only $O(n^2)$ arithmetic operations, and, moreover, has a structure that lends itself well to parallel computation. Our method is based on the observation that the matrix H_n is a rank-one modification of an orthogonal upper Hessenberg matrix.

2. Orthogonal and Unitary Hessenberg Eigenproblems

Orthogonal upper Hessenberg matrices form an important special case of matrices in the form (3).

Proposition 1 *The following statements are equivalent:*

- *The autocorrelation matrix M_{n+1} is nonnegative definite and of rank n.*
- *The Schur parameters satisfy $|\gamma_j| < 1$ for $1 \leq j < n$, and $|\gamma_n| = 1$.*
- *The zeros of $\psi_n(\lambda)$ are pairwise distinct and of unit modulus.*
- *H_n is an orthogonal Hessenberg matrix.*

The eigenproblem for orthogonal, and more generally, unitary Hessenberg matrices, is particularly nice. These are normal matrices with distinct eigenvalues, so numerical complications due to multiple eigenvalues and nontrivial Jordan structure is avoided. Moreover, several efficient ($O(n^2)$) algorithms have been presented for solving eigenproblems for orthogonal and unitary Hessenberg matrices. These include the unitary Hessenberg QR algorithm [11], an algorithm for the orthogonal Hessenberg eigenproblem using two half-size SVDs [3], divide-and-conquer methods [13, 14, 5], an approach based on an associated generalized eigenproblem [7], and a bisection method [8].

Eigenproblems for orthogonal and unitary Hessenberg matrices are useful in frequency estimation procedures that are based on finding the poles of lossless AR models such as Pisarenko's method and the Composite Sinusoidal Modeling method [4, 19].

In contrast, the eigenproblem for H_n when $|\gamma_n| < 1$ has a less tractable structure. The eigenvalues are of modulus less than one, but they can be of arbitrary algebraic multiplicity (although the geometric multiplicity of each eigenvalue is equal to one). Moreover, the algorithms for unitary Hessenberg matrices do not extend to the "subunitary" matrices H_n. For example, the QR algorithm does not readily extend because an orthogonal similarity transformation of H_n results in a Hessenberg matrix that cannot, in general, be represented in the form (3).

We now propose the following approach to efficiently compute the eigenvalues of $H_n = H(\gamma_1, \ldots, \gamma_n)$ with each $|\gamma_j| < 1$. Define the function

$$f(\lambda, t) = \det[\lambda I - H_n(t)], \tag{6}$$

where

$$H_n(t) = H(\gamma_1, \ldots, \gamma_{n-1}, \alpha(t))$$

and

$$\alpha(t) = (1 - t)\beta + t\gamma_n$$

with $|\beta| = 1$. Then the zeros of $f(\lambda, 0)$ are the eigenvalues of an orthogonal Hessenberg matrix, which can be quickly computed. These eigenvalues are then used as starting points for a continuation method, in which we track the zeros of $f(\lambda, t)$ as t varies from 0 to 1. The endpoints of these paths are then the eigenvalues of H_n.

The parameter β is chosen so that $H_n(0)$ is the closest orthogonal Hessenberg matrix to $H_n(1)$ in any unitarily invariant norm. In particular, we choose [19]

$$\beta = \begin{cases} \gamma_n / |\gamma_n| & \text{if } \gamma_n \neq 0, \\ 1 & \text{if } \gamma_n = 0 \end{cases} .$$

3. Predictor-Corrector Methods for Path Following

Our approach to the design of continuation methods for following the eigenvalues paths is based on the algorithm presented by Li and Zeng [18] and on ideas given by Allgower and Georg [1, 2].

Having determined the eigenvalues $\lambda_j(0)$, $1 \leq j \leq n$, of $H_n(0)$, we apply a continuation method to track the eigenvalues of $H_n(t)$ as t increases in order to determine the spectrum of $H_n(1)$. This method is based on the identity

$$\det[\lambda_j(t)I_n - H_n(t)] = 0, \qquad 0 \leq t \leq 1, \qquad 1 \leq j \leq n, \tag{7}$$

which, in view of (6), yields

$$0 = \frac{d}{dt} f(\lambda_j(t), t) = f_\lambda(\lambda_j(t), t)\frac{d\lambda_j(t)}{dt} + f_t(\lambda_j(t), t) \tag{8}$$

for $1 \leq j \leq n$.

Thus, if $f(\lambda, t)$ has only a simple zero at $\lambda_j(t)$, we obtain the differential equation

$$\frac{d}{dt}\lambda_j(t) = \frac{-f_t(\lambda_j(t), t)}{f_\lambda(\lambda_j(t), t)}, \qquad 1 \leq j \leq n. \tag{9}$$

Moreover, $f(\lambda, t)$, as well as f_λ and f_t, can be efficiently evaluated in $O(n)$ arithmetic operations using the Szegő recursions.

The eigenvalues $\lambda_j(t)$, $1 \leq j \leq n$, of $H_n(t)$ are continuous functions of t. Moreover, each $\lambda_j(t)$ is a piecewise analytic function of t, for $0 \leq t \leq 1$, whose only singularities are algebraic; see [16, Chapter 2]. A discussion on properties of the map $t \rightarrow \lambda_j(t)$ can also be found in [18]. The purpose of the continuation method is to track the eigenvalue paths

$$\{\lambda_j(t),\ 0 \leq t \leq 1\}, \qquad 1 \leq j \leq n,$$

in order to determine the eigenvalues $\lambda_j(1)$ of $H_n = H_n(1)$ from the eigenvalues $\lambda_j(0)$ of $H_n(0)$. Toward this end, we apply a scheme described by Deuflhard and Hohmann [9, Section 4.4.2] that uses Euler's method as a predictor and Newton's method as a corrector. This scheme is closely related to the methods described in [1, Chapter 6].

Assume for the moment that $\lambda_j(t)$ is a simple eigenvalue of $H_n(t)$ for $0 \leq t \leq 1$, and let $\lambda_j(t_k)$ be explicitly known for a parameter value t_k such that $0 \leq t_k < 1$. Let h be a positive stepsize such that $t_{k+1} = t_k + h \leq 1$. An approximation $\lambda_j^{(1)}$ of $\lambda_j(t_{k+1})$ is determined by applying Euler's method with constant step length h to the differential equation (9). We obtain

$$\lambda_j^{(1)} := \lambda_j(t_k) + h\frac{-f_t(\lambda_j(t_k), t_k)}{f_\lambda(\lambda_j(t_k), t_k)}. \tag{10}$$

Typically, $\lambda_j^{(1)}$ is not a sufficiently accurate approximation of $\lambda_j(t_{k+1})$. We determine improved approximations $\lambda_j^{(2)}$, $\lambda_j^{(3)}, \ldots$ of $\lambda_j(t_{k+1})$ by applying Newton's method for the equation $f(z, t_{k+1}) = 0$ and start the iterations with $z = \lambda_j^{(1)}$. We obtain

```
for $\ell := 1, 2, 3, \ldots$ until convergence do
    $h^{(\ell)} := f(\lambda_j^{(\ell)}, t_{k+1}) / f_\lambda(\lambda_j^{(\ell)}, t_{k+1})$;
    $\lambda_j^{(\ell+1)} := \lambda_j^{(\ell)} - h^{(\ell)}$;
$\lambda_j(t_{k+1}) := \lambda_j^{(\ell+1)}$.
```

Following Deuflhard and Hohmann [9], we require that the Newton corrections $h^{(\ell)}$ satisfy

$$\left|\frac{h^{(\ell+1)}}{h^{(\ell)}}\right| \leq \frac{1}{2} \tag{11}$$

for all $\ell \leq 1$. If the condition (11) is violated for some value of ℓ, then we reduce the step length h according to

$$h := 2^{-1/2}h$$

and determine a new value $\lambda_j^{(1)}$ by Euler's method (10) with the reduced step length. The value $\lambda_j^{(1)}$ now approximates $\lambda_j(t_{k+1})$ with $t_{k+1} := t_k + h$. More accurate approximations $\lambda_j^{(2)}, \lambda_j^{(3)}, \dots$ of $\lambda_j(t_{k+1})$ are, as above, determined by Newton's method applied to the equation $f(z, t_{k+1}) = 0$.

Conversely, assume that (11) is satisfied for all Newton corrections computed, and that moreover

$$\left|\frac{h^{(2)}}{h^{(1)}}\right| \leq \frac{1}{8}.$$

Then, after determining $\lambda(t_{k+1})$, we increase, if possible, the step length h by a factor $2^{1/2}$ according to

$$h := \min\{2^{1/2}h,\ 1 - t_{k+1}\}$$

before defining $t_{k+2} := t_{k+1} + h$.

Assuming that the number of arithmetic operations required to determine $\lambda_j(1)$ from $\lambda_j(0)$ for each j grows only linearly with n, the spectrum of H_n can be computed in only $O(n^2)$ arithmetic operations. Also note that, in principal, each eigenvalue $\lambda_j(1)$ can be determined from $\lambda_j(0)$, independently of the other eigenvalues. This makes the method well suited for parallel computation.

For some matrices $H_n(1)$, however, tracking the eigenvalue paths can require that certain eigenvalues $\lambda_j(t)$ be determined for a larger number of increasing values of t. This may occur when an eigenvalue path has large curvature or when there is a bifurcation of eigenvalue paths.

Generically, only two types of bifurcations can occur. The first type occurs when a conjugate pair of eigenvalue paths meet on the real axis and become two real paths moving on opposite directions. The second type occurs in an opposite situation, when two real eigenvalues meet and become a complex conjugate pair. Other bifurcations are quite unlikely, although possible. However, we have not yet observed the bifurcation of two non-conjugate complex paths. Consequently, we are developing our algorithm to handle bifurcations on the real axis only, using the approach of Li and Zeng [18].

After candidates for the terminal eigenvalues $\{\lambda_j(1)\}_{j=1}^n$ are obtained, we check these for duplication. If there is duplication, we check the derivatives to see if the duplicated eigenvalue is a multiple eigenvalue. If it is not, we repeat the path-following scheme on these two paths with more stringent conditions (e.g., with a smaller initial stepsize h_0). We also try again for those paths that did not terminate properly.

Finally, if the modified h_0 is too small, the procedure terminates with an error flag to signal to the user that all the eigenvalues were not found. If the user is not satisfied with the eigenvalues that were found, a backup procedure, such as the QR algorithm for general upper Hessenberg matrices, can be employed. Preliminary experiments indicate the procedure will efficiently and successfully compute the eigenvalues of many matrices H_n.

References

[1] E. Allgower and K. Georg. *Numerical Continuation Methods.* Springer-Verlag, Berlin, 1990.

[2] E. L. Allgower and K. Georg. Continuation and path following. *Acta Numerica*, pages 1–64, 1993.

[3] G. S. Ammar, W. B. Gragg, and L. Reichel. On the eigenproblem for orthogonal matrices. In *Proceedings of the 25th Conference on Decision and Control*, pages 1963–1966, 1986.

[4] G. S. Ammar, W. B. Gragg, and L. Reichel. Determination of Pisarenko frequency estimates as eigenvalues of an orthogonal matrix. In F. T. Luk, editor, *Advanced Algorithms and Architectures for Signal Processing II*, pages 143–145, 1987.

[5] G. S. Ammar, L. Reichel, and D. C. Sorensen. An implementation of a divide and conquer algorithm for the unitary eigenproblem. *ACM Trans. Math. Software*, 18:292–307, 1992.

[6] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide.* SIAM, Philadelphia, 1992.

[7] A. Bunse-Gerstner and L. Elsner. Schur parameter pencils for the solution of the unitary eigenproblem. *LAA*, 154–156:741–778, 1991.

[8] A. Bunse-Gerstner and C. He. A Sturm sequence of polynomials for unitary Hessenberg matrices. Preprint, 1992.

[9] P. Deuflhard and A. Hohmann. *Numerische Mathematik.* de Gruyter, Berlin, 1991.

[10] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, second edition, 1989.

[11] W. B. Gragg. The QR algorithm for unitary Hessenberg matrices. *J. Comput. Appl. Math.*, 16:1–8, 1986.

[12] W. B. Gragg. Positive definite Toeplitz matrices, the Arnoldi process for isometric operators, and Gaussian quadrature on the unit circle. *J. Comput. Appl. Math.*, 46:183–198, 1993. Originally published (in Russian) in: E.S. Nikolaev, Ed., *Numerical Methods in Linear Algebra,* Moscow Univ. Press, 1982, pp. 16–32.

[13] W. B. Gragg and L. Reichel. A divide and conquer method for the unitary eigenproblem. In M. T. Heath, editor, *Hypercube Multiprocessors 1987*, SIAM Philadelphia, 1987, pages 639–647.

[14] W. B. Gragg and L. Reichel. A divide and conquer method for unitary and orthogonal eigenproblems. *Numer. Math.*, 57:695–718, 1990.

[15] U. Grenander and G. Szegő. *Toeplitz Forms and Their Applications.* Chelsea, New York, second edition, 1984.

[16] T. Kato. *Perturbation Theory for Linear Operators.* Sprinver-Verlag, second edition, 1976.

[17] S. M. Kay. *Modern Spectral Estimation: Theory and Application.* Prentice Hall, 1988.

[18] T. Y. Li and Z. Zeng. Homotopy-determinant algorithm for solving nonsymmetric eigenvalue problems. *Math. Comp.*, 59:483–502, 1992.

[19] L. Reichel and G. S. Ammar. Fast approximation of dominant harmonics by solving an orthogonal eigenproblem. In J. C. McWhirter, editor, *Mathematics in Signal Processing II*, pages 575–591. Clarendon Press, 1990.

[20] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag, 1980.

[21] G. Szegő. *Orthogonal Polynomials.* American Math. Soc., Providence, 1939.