Signal Processing 89 (2009) 1050–1060

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Second and fourth order statistics-based reduced polynomial rooting direction finding algorithms

Wasyl Wasylkiwskyj [a], Ivica Kopriva [b,*]

[a] Department of Electrical and Computer Engineering, The George Washington University, 801 22nd Street, NW Room 615, Washington, DC 20052, USA
[b] Rudjer Boskovich Institute, Bijenicka cesta 54, 10002 Zagreb, Croatia

ARTICLE INFO

Article history:
Received 4 April 2008
Received in revised form
10 November 2008
Accepted 8 December 2008
Available online 24 December 2008

Keywords:
Direction finding
Polynomial rooting algorithms
Second order statistics
Fourth order statistics

ABSTRACT

Polynomial rooting direction finding (DF) algorithms are a computationally efficient alternative to search-based DF algorithms and are particularly suitable for uniform linear arrays (ULA) of physically identical elements provided mutual interaction among the array elements can be either neglected or compensated for. A popular polynomial rooting algorithm is Root-MUSIC (RM) wherein, for an N-element array, the estimation of the directions of arrivals (DOA) requires the computation of the roots of a $2N-2$-order polynomial for a second order (SO) statistics- and a $4N-4$-order polynomial for a fourth order (FO) statistics-based approach, wherein the DOA are estimated from L pairs of roots closest to the unit circle, when L signals are incident on the array. We derive SO- and FO statistics reduced polynomial rooting (RPR) algorithms capable to estimate L DOA from L roots only. We demonstrate numerically that the RPR algorithms are at least as accurate as the RM algorithms. Simplified algebraic structure of RPR algorithms leads to better performance than afforded by RM algorithms in saturated array environment, especially in the case of FO methods when number of incident signals exceeds number of elements and under low SNR and/or small sample size conditions.

1. Introduction

Super-resolution direction finding (DF) algorithms for linear arrays fall into two broad categories: search-based algorithms, as exemplified by MUSIC [1,2] and root-based algorithms such as Root-MUSIC (RM) [3,4], ESPRIT [2]. Search algorithms make no assumptions about the algebraic structure of the array steering vectors but require that they be known to great accuracy, especially if a high degree of angular resolution is called for. In that case they can also be computationally quite demanding. In practice the determination of the array steering vector amounts to an accurate measurement of the magnitude and phase of the array element patterns, sometimes referred to as array manifold calibration. Normal accuracies attained in such measurements are a few tenths of a dB in amplitude and about 1° in phase, which generally is insufficient for the design of high-resolution DF systems. Admittedly an alternative technique would be to rely on numerical computer simulations (either computing the element patterns directly or inferring them from the array geometry and the computed impedance or scattering matrix). However our experience with comparisons of numerical simulations using the latest commercially available software with experimental data indicates that presently this is not yet a fruitful approach [5].

Root-based algorithms on the other hand require no array calibration and afford substantial computational efficiency over search algorithms. They require that the elements be uniformly spaced and physically identical, which a search algorithm such as MUSIC does not. The

* Corresponding author.
E-mail addresses: wasylkiw@gwu.edu (W. Wasylkiwskyj), ikopriva@irb.hr (I. Kopriva).

0165-1684/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2008.12.004

more significant restriction, however, is that the array steering vector must have the form of an array factor of a linear array of uniformly spaced elements. Unfortunately due to inter-element mutual coupling this idealized form of the steering vector is practically unattainable without compensation. Indeed, when root-based DF algorithms are applied to a real array without some form of compensation, significant angle estimation errors can result [6]. Compensation for the effects of mutual coupling can be realized by employing extra "dummy" elements to equalize the active element radiation patterns [7,8]. Under the assumption that the element radiation patterns are sufficiently equalized, the nonnegative pseudo-spectrum function becomes a polynomial and the DF problem is reduced to a polynomial rooting problem [3,4], ESPRIT [2]. In case of a covariance-based RM algorithm, for an N-element array, the degree of the polynomial equals $2N-2$, so that $2N-2$ roots have to be calculated. In case of a fourth order (FO) statistics-based RM algorithm, the degree of the polynomial equals $4N-4$ and, consequently, $4N-4$ roots have to be calculated. For L incident signals, the directions of arrivals (DOA) are calculated from the L roots closest to the unit circle. This selection process can introduce serious errors in saturated[1] array environments, especially under low SNR and/or small sample size conditions because the signal roots then do not stay close to the unit circle. Unlike RM algorithms, reduced polynomial rooting (RPR) algorithms do not generate extraneous roots,[2] i.e., all polynomial roots correspond to the actual DOA. As is demonstrated in Section 4, this feature is of particular advantage in saturated array environments, especially in the case of FO methods when number of incident signals L exceeds number of elements N and low SNR and/or small sample size conditions and results in enhanced performance of RPR algorithms over RM algorithms.

The formulation of the RPR algorithms relies on the solution of an over-determined system of linear equations that yields the coefficients of an L degree polynomial. Depending on the required accuracy, this system can be solved either by using the Moore–Penrose pseudo-inverse or by using a more accurate total-least-square (TLS) approach [13]. Our numerical studies have shown that in not too demanding scenarios, where the separation between adjacent signals in the angular domain was not very close, the two approaches gave results of comparable accuracy. As will be demonstrated in Section 4, this computationally lighter version of the RPR algorithms is not inferior to RM algorithms. The RPR algorithms themselves are derived in Sections 2 and 3. Results of comparative performance evaluations are presented in Section 4. The conclusions are given in Section 5.

[1] By a saturated array we refer to a scenario wherein the number of emitters L is close to either the number of real sensors N, in a case of the SO methods, or to the number of virtual sensors $2N-1$, in a case of the FO methods.

[2] We comment that RPR algorithms presented herein should not be confused with the algorithms we have recently derived in [16]. The latter algorithms rely on different subspace decomposition principles and require the solution of polynomials of order $2L$ instead of L and are, in that sense, computationally more demanding.

2. Linear antenna array model

Polynomial rooting-based super-resolution DF algorithms such as RM [3] offer computational efficiency in relation to the search-based DF methods [1] when the special geometry of the uniform linear arrays (ULA) is employed. In this case the problem of estimating the DOA of L signals incident on N-element array is described by

$$\mathbf{z}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{v}(t) \tag{1}$$

where $\mathbf{z}(t)$ is a complex column vector comprised of N signals at the output of the array; $\mathbf{A}$ is $N \times L$ steering matrix of the linear array comprised of the L column vectors $\mathbf{a}(\Omega_l)$ corresponding with the DOA of the l-th source signal; $\mathbf{s}(t)$ is a column vector comprised of the L source signals incident on the array and $\mathbf{v}(t)$ represents additive noise. If mutual coupling among the array elements is compensated [9] the steering vector for a ULA simplifies to

$$\mathbf{a}(\Omega_l) = \hat{f}(\Omega_l)[1 \;\; e^{jk_0 d\xi_l} \;\; e^{jk_0 2d\xi_l} \;\; \ldots \;\; e^{jk_0(N-1)d\xi_l}]^{\mathrm{T}} \tag{2}$$

where $\Omega_l = (\theta_l, \varphi_l)$, $\xi_l = \sin(\theta_l)\cos(\varphi_l)$, θ_l and φ_l represent elevation and azimuth of the l-th source DOA, $k_0 = 2\pi/\lambda$ is a free space wave number evaluated at the receiver local oscillator frequency, λ is a wavelength, d is an inter-element spacing and $\hat{f}(\Omega_l)$ represents the element radiation pattern. In the formulation of the SO MUSIC algorithm [1] one estimates $\mathbf{E}_v$, the matrix of eigenvectors that span the noise subspace and forms the nonnegative function

$$\Lambda(\Omega) = \mathbf{a}(\Omega)^H \mathbf{E}_v \mathbf{E}_v^H \mathbf{a}(\Omega) \tag{3}$$

called pseudo-spectrum and employs the locations of its zeros to estimate the DOA's. For sufficiently large sample sizes the $\mathbf{E}_v$ can be well approximated by the eigenvectors of the sample data covariance matrix

$$\hat{\mathbf{R}}_{\mathbf{zz}} = (1/T)\sum_{t=1}^{T} \mathbf{z}(t)\mathbf{z}(t)^H$$

where 'H' denotes Hermitian operation. For the ULA in (2) the $\Lambda(\Omega)$ can be written in polynomial form [2] as follows:

$$\Lambda(z) = z^{-(N-1)} P_{2N-2}(z) \tag{4}$$

where $z = e^{jk_0 d\xi}$ and $P_{2N-2}(z)$ is the $2N-2$ degree polynomial in z. From (4) DOA are found from the L pairs of complex roots of the polynomial $P_{2N-2}(z)$ that are closest to the unit circle. The corresponding direction cosines are

$$\xi_l = angle(z_l)/k_0/d, \quad l = 1, 2, \ldots, L \tag{5}$$

In view of (4), RM requires the calculation of $2N-2$ roots. For large arrays this leads to high-computational loads and becomes a source of the numerical errors alluded to previously.

By analogy with the SO MUSIC pseudo-spectrum, the quadricovariance version is formulated as follows [10,11]:

$$\Lambda(\Omega) = (\mathbf{a}(\Omega) \otimes \mathbf{a}^*(\Omega))^H \mathbf{E}_v \mathbf{E}_v^H (\mathbf{a}(\Omega) \otimes \mathbf{a}^*(\Omega)) \tag{6}$$