An EVD Algorithm for Para-Hermitian Polynomial Matrices

John G. McWhirter, Paul D. Baxter, Tom Cooper, Soydan Redif, *Associate Member, IEEE*, and Joanne Foster

***Abstract*—An algorithm for computing the eigenvalue decomposition of a para-Hermitian polynomial matrix is described. This amounts to diagonalizing the polynomial matrix by means of a paraunitary "similarity" transformation. The algorithm makes use of "elementary paraunitary transformations" and constitutes a generalization of the classical Jacobi algorithm for conventional Hermitian matrix diagonalization. A proof of convergence is presented. The application to signal processing is highlighted in terms of strong decorrelation and multichannel data compaction. Some simulated results are presented to demonstrate the capability of the algorithm.**

***Index Terms*—Broadband sensor array, convolutive mixing, multichannel data compaction, paraunitary matrix, polynomial matrix eigenvalue decomposition, strong decorrelation.**

I. Introduction

POLYNOMIAL matrices have been used for many years in the area of control [1]. They play an important role in the realization of multivariable transfer functions associated with multiple-input multiple-output (MIMO) systems. Over the last few years they have become more widely used in the context of digital signal processing (DSP) and communications [2]. Typical areas of application include broadband adaptive sensor array processing [3], [4], broadband subspace decomposition [5], MIMO communication channels [6]–[9], and digital filter banks for subband coding [10] or data compression [11].

A polynomial matrix is simply a matrix whose elements are polynomials. It may be viewed equivalently, as a polynomial with matrix coefficients. In this paper, we will use the term polynomial to include Laurent polynomials which can include negative powers of the indeterminate variable. We denote a $p \times q$ polynomial matrix in the indeterminate variable z^{-1} by

$$\underline{\mathbf{A}}(z) = \sum_{\tau=\tau_1}^{\tau_2} \mathbf{A}(\tau) z^{-\tau} = \begin{bmatrix} \underline{a}_{11}(z) & \underline{a}_{12}(z) & \ldots, & \underline{a}_{1q}(z) \\ \underline{a}_{21}(z) & & & \\ \vdots & & & \\ \underline{a}_{p1}(z) & \ldots, & \ldots, & \underline{a}_{pq}(z) \end{bmatrix} \tag{1.1}$$

where $\tau \in Z$, $\tau_1 \leq \tau_2$ and

$$\underline{a}_{kl}(z) = \sum_{\tau=\tau_1}^{\tau_2} a_{kl}(\tau) z^{-\tau}. \tag{1.2}$$

Since the leading term of $z^{\tau_1}\underline{\mathbf{A}}(z)$ is constant, the effective order of $\underline{\mathbf{A}}(z)$ is $\tau_2 - \tau_1$. In this paper it is assumed that $a_{kl}(\tau) \in C \ \forall \ k, l, \tau$. In keeping with the standard notation for linear systems and signal processing, we have chosen to denote the indeterminate variable by z^{-1} since this is normally used to represent a unit delay. Multiplying a polynomial by z^{-1} will sometimes be referred to as applying a delay. The underscore notation is used to signify a polynomial or power series (whether matrix, vector, or scalar). This is to avoid confusion with the corresponding z-transform, which is a function of z evaluated within its region of convergence in the complex plane.

Numerical procedures have previously been developed for a range of polynomial matrix factorization and reduction operations such as the Smith–McMillan decomposition [2], [12]. To date, however, very little attention seems to have been devoted to polynomial matrix techniques equivalent to the eigenvalue decomposition (EVD) or singular value decomposition (SVD) for conventional matrices with scalar elements [13]. The development and application of such a technique is the subject of this paper.

The EVD of conventional Hermitian matrices plays a very important role in DSP. For example, it is at the heart of the Karhunen–Loeve transform for optimal data compaction [14]. It plays a particularly important role in the context of narrowband sensor array signal processing serving, for example, to separate the signal and noise subspaces in adaptive beamforming or high resolution direction finding [14].

In the case of a narrowband sensor array, the propagation of signals from q sources to p sensors may be modelled as an instantaneous mixture of the form $\mathbf{X} = \mathbf{AS}$ where the matrix $\mathbf{S} \in C^{q \times T}$ comprises T snapshots of the q source signals (assumed to have zero mean), the matrix $\mathbf{X} \in C^{p \times T}$ consists of the corresponding T snapshots received by the p sensors and $\mathbf{A} \in C^{p \times q}$ is the mixing matrix. Each element of the mixing matrix represents a scaling in amplitude, and a phase shift that accounts for the propagation delay. In the case of instantaneous mixing, the received signals may be decorrelated by performing the SVD of the data matrix $\mathbf{X}$, i.e., by computing a unitary matrix $\mathbf{U} \in C^{p \times p}$ such that $\mathbf{UX} = \mathbf{\Sigma V}$ where $\mathbf{\Sigma} = \mathrm{diag}\{\gamma_1, \gamma_2, \ldots, \gamma_p\}$ with $\gamma_1 \geq \gamma_2 \cdots \geq \gamma_p \geq 0$ and $\mathbf{V} \in C^{p \times T}$ with $\mathbf{VV}^H = \mathbf{I}_p$. This corresponds to computing the EVD of the (Hermitian) sample covariance matrix $\mathbf{XX}^H$, i.e., performing the unitary diagonalization given by $\mathbf{UXX}^H\mathbf{U}^H = \mathbf{\Sigma}^2$. Although the SVD computation is preferable in terms of arithmetic precision, the EVD of $\mathbf{XX}^H$ and SVD of $\mathbf{X}$ will be regarded as equivalent for our purposes.

Manuscript received January 5, 2006; revised August 6, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Vitor Heloiz Nascimento. This work was supported in part by the U.K. Ministry of Defence Corporate Research Programme.

J. G. McWhirter is with QinetiQ Ltd., Malvern WR14 3PS, U.K., and also with Cardiff University, CF10 3XQ Wales, U.K. (e-mail: mcwhirter@signal.qinetiq.com).

P. D. Baxter and T. Cooper are with QinetiQ Ltd., Malvern WR14 3PS, U.K.

S. Redif is with QinetiQ Ltd., Malvern WR14 3PS, U.K., and also with the University of Southampton, Southampton SO10 1BJ, U.K.

J. Foster is with QinetiQ Ltd., Malvern WR14 3PS, U.K., and also with the Centre of Digital Signal Processing, School of Engineering, Cardiff University, Wales CF10 3XQ, U.K.

Digital Object Identifier 10.1109/TSP.2007.893222

The instantaneous mixing model is not suitable for many important applications. For example, in the case of a broadband sensor array, the propagation of signals from q sources to p sensors cannot be modelled by a scalar mixing matrix. A $p \times q$ matrix of finite impulse response (FIR) filters is required instead. If each filter is represented as a polynomial in z^{-1}, corresponding to its transfer function, the propagation model takes the form of a $p \times q$ polynomial mixing matrix of the type specified in (1.1). This is often referred to as convolutive mixing [15]. Convolutive mixing can also be used to model the effects of multipath propagation, which constitutes an important factor in many areas of sensor array signal processing.

In the case of a broadband sensor array or convolutively mixed signals, the sensor outputs will generally be correlated with one another. However, they can no longer be decorrelated using the SVD or EVD, which only measure and remove instantaneous spatial correlation, i.e., correlation between pairs of signals sampled at the same instant in time. Following convolutive mixing, it is necessary to impose decorrelation, not just at the same time instant for all signals, but over a suitably chosen range of relative time delays. This is referred to as strong decorrelation or total decorrelation [10], and a matrix of suitably chosen filters is required to achieve it.

In this paper we generalize the EVD (and hence the SVD) to the case of broadband sensor arrays and convolutive mixing by requiring the strong decorrelation to be implemented using a paraunitary polynomial matrix. A paraunitary polynomial matrix represents a multichannel all-pass filter and, accordingly, it preserves the total signal power at every frequency [2]. In order to achieve strong decorrelation, the paraunitary matrix seeks to diagonalize a para-Hermitian polynomial matrix by means of a generalized similarity transformation. This constitutes a natural generalization of the EVD for scalar Hermitian matrices. We also present a novel technique, referred to as the second-order sequential best rotation (SBR2) algorithm, for computing the required paraunitary matrix using a sequence of "elementary paraunitary matrices" [16]–[18]. This constitutes a natural generalization of Jacobi's EVD algorithm [13] from scalar to polynomial matrices. It has been proven to converge and is numerically stable by virtue of the fact that the elementary paraunitary matrices, like elementary plane rotations on which they are based, are numerically stable.

Our approach is quite distinct from other methods reported to date. Lambert [4], [19] has addressed the problem of broadband blind signal separation in the context of convolutive mixing. He represents the mixing in terms of discrete Fourier transform (DFT) filter matrices as well as polynomial matrices. He has developed an EVD for polynomial matrices by generalizing some conventional linear algebra and control techniques from the complex number field to the field of rational functions. His method involves the approximate inversion of FIR filters in the frequency domain and is therefore quite distinct from the one proposed here.

Regalia and Huang [20] have addressed the problem of computing a two-channel lossless FIR filter for optimal data compaction. This leads to the determination of an optimum paraunitary matrix as required for our polynomial matrix EVD algorithm in the 2×2 case. Their approach exploits the fixed degree parameterization proposed by Vaidyanathan [2], resulting in a difficult nonlinear optimization. However, they re-formulate the problem using a state space approach and then propose an iterative solution that avoids the problems of local minima associated with gradient descent techniques.

Another fairly obvious approach to decorrelating broadband signals is to reduce the problem to narrowband form, using a DFT to split the data into narrower frequency bands. A conventional SVD is then used to decorrelate the sensor signals within each band. However, computing an independent SVD for each frequency band ignores the relatively small but important correlations that may exist between different bands. This independent frequency band (IFB) approach limits the degree to which strong decorrelation can be achieved. It can also lead to a lack of temporal (phase) coherence across the bands, since the SVD in each band will arrange the output channels in order of decreasing power irrespective of the ordering in neighboring bands. These are well-known features of the IFB technique for space-time adaptive processing in phased array radar [21].

This paper is organized as follows. Section II proposes a generalization of the EVD, appropriate for para-Hermitian polynomial matrices. The SBR2 algorithm is then presented as a novel technique for computing the required decomposition and some simple numerical examples are presented. Section III outlines how the SBR2 algorithm may be used in the context of DSP—in particular, how it may be used to implement the strong decorrelation of signals from a broadband sensor array by diagonalizing an estimate of the cross-spectral density matrix. The tendency of the algorithm to impose spectral majorization [10] on the strongly decorrelated signals and its relevance to energy compaction is also discussed very briefly in Section III. Section IV presents the results of some computer simulations designed to demonstrate the capability of the SBR2 algorithm. Section V contains some brief conclusions and suggests various avenues for further research.

1) Choice of Notation: Throughout this paper, matrices are denoted by upper case bold characters and vectors by lower case bold characters. Regular upper or lower case characters denote scalar quantities. $[\cdot]_{kl}$ denotes the (k,l) element of the matrix in square brackets. The superscripts $*$, T, and H denote the complex conjugate, matrix transpose and Hermitian conjugate, respectively. $\mathbf{I}_p$ is used to denote the $p \times p$ identity matrix. Polynomial matrices and vectors are denoted by underscored bold upper and lower case characters, respectively. The use of an underscore with scalar quantities denotes a polynomial with scalar coefficients. Any polynomial (matrix, vector, or scalar) with the qualifier (z) denotes a polynomial in the indeterminate variable z^{-1}. The $*$, used as a subscript, denotes complex conjugation of the coefficients in a polynomial matrix or vector. The use of $\sim$ above a polynomial matrix or vector denotes the paraconjugate. The underscore notation will be extended, where appropriate, to include power series or matrices and vectors whose elements comprise power series. $\|\cdot\|_F$ will be used to denote the Frobenius norm (F-norm) of a polynomial matrix as well as a matrix with scalar elements. In the case of a polynomial matrix, $\|\cdot\|_F^2$ is simply the sum of the squared F-norms for all coefficient matrices.

II. Sequential Best Rotation Algorithm (SBR2)

A. Polynomial Matrix EVD

In this section we describe a novel algorithm for extending the EVD from conventional Hermitian matrices with complex scalar elements to para-Hermitian polynomial matrices. It takes

the form of a sequential best rotation algorithm [16] but only involves second order statistics; it is therefore referred to as SBR2. We begin with some basic definitions and notation. The paraconjugate of a polynomial matrix $\underline{\mathbf{A}}(z)$ is defined as $\tilde{\underline{\mathbf{A}}}(z) = \underline{\mathbf{A}}_*^T(1/z)$. Note that in the degenerate case of an order-zero polynomial matrix, this corresponds to the usual Hermitian conjugate. A (polynomial) matrix $\underline{\mathbf{A}}(z)$ is referred to as para-Hermitian if it is identical to its paraconjugate, i.e., if $\tilde{\underline{\mathbf{A}}}(z) = \underline{\mathbf{A}}(z)$. A (polynomial) matrix $\underline{\mathbf{H}}(z)$ is said to be paraunitary if

$$\underline{\mathbf{H}}(z)\tilde{\underline{\mathbf{H}}}(z) = \tilde{\underline{\mathbf{H}}}(z)\underline{\mathbf{H}}(z) = \mathbf{I}. \tag{2.1}$$

Note that in the degenerate case of order zero, this reduces to a conventional unitary matrix.

The input to the SBR2 algorithm is a para-Hermitian polynomial matrix, which may be expressed as

$$\underline{\mathbf{R}}(z) = \sum_{\tau=-\tau_{\max}}^{\tau_{\max}} z^{-\tau}\mathbf{R}(\tau) \tag{2.2}$$

where $\mathbf{R}(\tau) \in C^{p\times p}$ with elements

$$\begin{aligned}[\mathbf{R}(\tau)]_{kl} &= r_{kl}(\tau) = r_{lk}^*(-\tau) \\ &= [\mathbf{R}(-\tau)]_{lk}^*, \quad k,l \in \{1,2,\ldots,p\}.\end{aligned} \tag{2.3}$$

The latter condition arises from the para-Hermitian property, which also requires the limits of summation in (1.1) to satisfy $\tau_2 = -\tau_1 \triangleq \tau_{\max}$. The objective of the SBR2 algorithm is to compute a paraunitary matrix $\underline{\mathbf{H}}(z)$ such that

$$\underline{\mathbf{H}}(z)\underline{\mathbf{R}}(z)\tilde{\underline{\mathbf{H}}}(z) \cong \underline{\mathbf{D}}(z) \tag{2.4}$$

where $\underline{\mathbf{D}}(z)$ denotes a diagonal polynomial matrix. This clearly constitutes a generalization of the EVD to polynomial matrices; in the degenerate case of order zero, it reduces to the definition of a conventional Hermitian matrix EVD which can, of course, be computed to very high precision. The role of the unitary matrix in the scalar matrix case is generalized to that of a paraunitary matrix. In the context of DSP and linear system theory, a paraunitary matrix represents the transfer function for a multichannel lossless filter. The spectral characteristic of a lossless filter is "all pass," which means that the combined signal power at every frequency is invariant to the transformation. The challenge then, is to compute a paraunitary matrix $\underline{\mathbf{H}}(z)$ such that the polynomial matrix $\underline{\mathbf{D}}(z)$ in (2.4) is as close to diagonal as possible. In general, it will not be possible to achieve exact diagonalization because $\underline{\mathbf{H}}(z)$ constitutes a FIR filter. However, if the order of the polynomial elements is sufficiently large, the diagonalization can be achieved to a very good approximation; in some cases, relatively low order polynomials suffice. Note that the decomposition in (2.4) could be carried out in the frequency domain by evaluating the polynomial matrix $\underline{\mathbf{R}}(z)$ at a large number of points on the unit circle, computing the resulting scalar EVD at each frequency, and transforming back to the time domain. However, irrespective of the number of discrete frequencies selected, this still constitutes a finite approximation and leads to a transformation that is not perfectly paraunitary. The SBR2 algorithm, on the other hand, is designed specifically to operate in the time domain and computes an FIR approximation, paraunitary by construction.

In order to ensure that the approximate diagonalization in (2.4) is carried out over the restricted space of paraunitary matrices, the SBR2 algorithm uses a suitably parameterized representation. Vaidyanathan [2] has shown that an arbitrary paraunitary matrix can be decomposed into a set of rotations interspersed by delays. More specifically any $p \times p$ paraunitary matrix $\underline{\mathbf{H}}_N(z)$ of degree[1] N may be written in the form

$$\underline{\mathbf{H}}_N(z) = \mathbf{Q}_N \ldots \underline{\mathbf{\Lambda}}(z)\mathbf{Q}_1\underline{\mathbf{\Lambda}}(z)\mathbf{Q}_0\mathbf{\Psi} \tag{2.5}$$

where $\mathbf{\Psi}$ is a diagonal matrix with unimodular elements and $\underline{\mathbf{\Lambda}}(z)$ denotes a unit delay applied to one row of the (polynomial) matrix on which it operates from the left; specifically

$$\underline{\mathbf{\Lambda}}(z) = \begin{bmatrix} \mathbf{I}_{p-1} & 0 \\ 0 & z^{-1} \end{bmatrix}. \tag{2.6}$$

$\mathbf{Q}_i$ represents a $p\times p$ unitary matrix which may be implemented as a product of $p(p-1)/2$ pairwise complex rotations and parameterized by the corresponding set of $p(p-1)$ rotation angles [2]. Note that in the degenerate case of degree zero, $\underline{\mathbf{H}}_0(z)$ takes the form of a single unitary matrix as required for the EVD of a conventional Hermitian matrix. For the purposes of polynomial matrix EVD, it is necessary to compute a sequence of rotation matrices $\mathbf{Q}_0, \ldots, \mathbf{Q}_N$ which minimizes the off-diagonal elements of the polynomial matrix $\underline{\mathbf{D}}(z)$. Even if the degree N could be established in advance, attempting to optimize the parameters of a paraunitary matrix in the form of (2.5) is extremely difficult since the individual rotations cannot be computed independently and a multiparameter nonlinear optimization must be carried out [20].

B. SBR2 Algorithm

In order to simplify the problem we adopt a different formula for generating the paraunitary matrices. This takes the form

$$\underline{\mathbf{H}}_L(z) = \underline{\mathbf{G}}_L(z)\ldots\underline{\mathbf{G}}_2(z)\underline{\mathbf{G}}_1(z) \tag{2.7}$$

where L represents an unspecified number of iterations. Each term in the product denotes an "elementary paraunitary matrix" of the form

$$\underline{\mathbf{G}}(z) = \mathbf{Q}^{(j,k)}(\theta,\phi)\underline{\mathbf{B}}^{(k,t)}(z) \tag{2.8}$$

with specific values being chosen for the parameters j, k, t, θ, ϕ as appropriate for each term. In (2.8), $\mathbf{Q}^{(j,k)}(\theta,\phi)$ denotes a complex elementary scalar rotation that takes the form of a $p\times p$ unit matrix except for the 2×2 submatrix $\hat{\mathbf{Q}}(\theta,\phi)$ defined by the intersection of rows j and k with columns j and k. This is given by

$$\hat{\mathbf{Q}}(\theta,\phi) = \begin{bmatrix} c & se^{i\phi} \\ -se^{-i\phi} & c \end{bmatrix} \tag{2.9}$$

where c and s denote the cosine and sine, respectively of the angle θ. $\underline{\mathbf{B}}^{(k,t)}(z)$ represents an elementary "delay" matrix designed to impose a t-fold delay (where $t \in Z$) to the kth row of the polynomial matrix on which it operates (from the left).

[1]The degree of a polynomial matrix in the indeterminate z^{-1} is defined as the number of delays needed to implement it as the transfer function of an FIR filter. This is not the same as the order which is the highest power of z^{-1} in the polynomial matrix.

It takes the form of a polynomial identity matrix except for the (k,k) element which is z^{-1}, i.e.,

$$\underline{\mathbf{B}}^{(k,t)}(z) = \begin{bmatrix} \mathbf{I}_{k-1} & 0 & 0 \\ 0 & z^{-t} & 0 \\ 0 & 0 & \mathbf{I}_{p-k} \end{bmatrix}. \tag{2.10}$$

The paraunitary matrix $\underline{\mathbf{G}}(z)$ is elementary in the sense that it involves an elementary rotation and a simple delay, but it does not necessarily have degree one. It can be seen that any polynomial matrix generated according to (2.7) is paraunitary since each term is paraunitary. However, the degree is no longer certain.

The SBR2 algorithm seeks to generate a paraunitary matrix according to (2.7) by calculating and applying an iterative sequence of elementary paraunitary matrices designed to approximately diagonalize $\underline{\mathbf{R}}(z)$ according to (2.4). Each stage of the iterative process applies a single elementary paraunitary matrix, chosen to eliminate the dominant off-diagonal polynomial coefficient. It is clearly analogous to a single step of the classical Jacobi algorithm for diagonalizing conventional Hermitian matrices, but generalizes the concept to para-Hermitian polynomial matrices.

The algorithm begins by locating the dominant off-diagonal coefficient of the input polynomial matrix $\underline{\mathbf{R}}(z)$, i.e., the off-diagonal coefficient whose magnitude is greatest. If the dominant coefficient is not unique, any dominant coefficient may be chosen. Assume that this is the coefficient $r_{jk}(t)$ where $j < k$, and denote its magnitude by $g = |r_{jk}(t)|$. Note that the search may be restricted to the upper off-diagonal elements due to the para-Hermitian property defined in (2.3). The specific values j, k and t that define the dominant coefficient are now used to specify the corresponding parameters of the first elementary paraunitary matrix in (2.7). In accordance with (2.4), the elementary delay matrix is applied first to generate the transformed polynomial matrix

$$\underline{\mathbf{R}}'(z) = \underline{\mathbf{B}}^{(k,t)}(z)\underline{\mathbf{R}}(z)\tilde{\underline{\mathbf{B}}}^{(k,t)}(z). \tag{2.11}$$

The effect of this transformation is to shift the dominant coefficient $r_{jk}(t)$ to the coefficient plane of order zero so that $r'_{jk}(0) = r_{jk}(t) = r_{kj}(-t)^* = r'_{kj}(0)^*$. The diagonal elements are not affected.

The parameters θ and ϕ of the elementary rotation matrix $\mathbf{Q}^{(j,k)}(\theta,\phi)$ are now chosen to drive the dominant coefficient to zero. More specifically, they are chosen such that

$$\begin{bmatrix} c & se^{i\phi} \\ -se^{-i\phi} & c \end{bmatrix} \begin{bmatrix} r'_{jj}(0) & r'_{jk}(0) \\ r'_{kj}(0) & r'_{kk}(0) \end{bmatrix} \begin{bmatrix} c & -se^{i\phi} \\ se^{-i\phi} & c \end{bmatrix} = \begin{bmatrix} r''_{jj}(0) & 0 \\ 0 & r''_{kk}(0) \end{bmatrix}. \tag{2.12}$$

This condition is satisfied when

$$\phi = \arg\left(r'_{jk}(0)\right) \tag{2.13}$$

and

$$\tan 2\theta = \frac{2\left|r'_{jk}(0)\right|}{r'_{jj}(0) - r'_{kk}(0)}. \tag{2.14}$$

Equation (2.14) has multiple solutions any of which may be selected. The order of the two output channels depends on the choice. Using the basic arctangent function to compute 2θ leads to the conventional choice of inner rotations [13], i.e., solutions in the range $-\pi/4 < \theta \leq \pi/4$. However, the four quadrant arctangent is preferred since it generally leads to output channels which are ordered in terms of decreasing power.

Having computed the values of θ and ϕ, the elementary rotation matrix is used to perform the transformation

$$\underline{\mathbf{R}}''(z) = \mathbf{Q}^{(j,k)}(\theta,\phi)\underline{\mathbf{R}}'(z)\mathbf{Q}^{(j,k)^H}(\theta,\phi). \tag{2.15}$$

Note that this involves all terms in the relevant polynomial elements. It should be clear that the polynomial matrices $\underline{\mathbf{R}}(z)$ and $\underline{\mathbf{R}}''(z)$ are related by a generalized similarity transformation of the form

$$\underline{\mathbf{R}}''(z) = \underline{\mathbf{G}}(z)\underline{\mathbf{R}}(z)\tilde{\underline{\mathbf{G}}}(z) \tag{2.16}$$

where $\underline{\mathbf{G}}(z)$ constitutes an elementary paraunitary matrix of the type defined in (2.8). The generalized similarity transformation in (2.16) constitutes one stage of the SBR2 algorithm, designed to zero the dominant off-diagonal coefficient of $\underline{\mathbf{R}}(z)$.

The algorithm continues by making the substitution $\underline{\mathbf{R}}(z) \leftarrow \underline{\mathbf{R}}''(z)$ and repeating the process outlined above, i.e., applying a new elementary paraunitary transformation of the form given in (2.16) designed, this time, to zero the most significant off-diagonal coefficient of the updated polynomial matrix $\underline{\mathbf{R}}(z)$. This iterative process is repeated until the magnitude of the dominant off-diagonal coefficient of the polynomial matrix is sufficiently small. The SBR2 algorithm is guaranteed to converge in this respect, and a proof of this important property is given below. Assuming L iterations, the result will be a generalized similarity transformation of the form

$$\underline{\mathbf{H}}_L(z)\underline{\mathbf{R}}(z)\tilde{\underline{\mathbf{H}}}_L(z) = \underline{\mathbf{D}}_L(z) \tag{2.17}$$

where $\underline{\mathbf{H}}_L(z)$ is the product of L elementary paraunitary matrices as expressed in (2.7) and $\underline{\mathbf{D}}_L(z)$ is approximately diagonal. The compound transformation $\underline{\mathbf{H}}_L(z)$ is clearly paraunitary by construction.

In order to explain the basic rationale behind the SBR2 algorithm, we define the following measures for the polynomial matrix $\underline{\mathbf{R}}(z)$:

$$N_1 \triangleq \sum_{j=1}^{p} |r_{jj}(0)|^2, \quad N_2 \triangleq \sum_{j=1}^{p}\sum_{k=1}^{p} |r_{jk}(0)|^2,$$
$$N_3 \triangleq \sum_{j=1}^{p}\sum_{\substack{k=1 \\ k\neq j}}^{p} |r_{jk}(0)|^2, \quad N_4 \triangleq \sum_{\tau}\sum_{j=1}^{p}\sum_{k=1}^{p} |r_{jk}(\tau)|^2. \tag{2.18}$$

These are, respectively, the squares of: the trace norm at $\tau = 0$; the F-norm at $\tau = 0$; the off-diagonal F-norm at $\tau = 0$; and $\|\underline{\mathbf{R}}(z)\|_F$. Similarly, we define (N'_1, N'_2, N'_3, N'_4) and $(N''_1, N''_2, N''_3, N''_4)$ for $\underline{\mathbf{R}}'(z)$ and $\underline{\mathbf{R}}''(z)$. Note the following:

- N_1 is invariant to the application of an elementary delay matrix;
- N_2 is invariant to the application of an elementary rotation;
- N_4 is invariant to the application of an elementary rotation or delay matrix;
- $N_2 = N_1 + N_3$ and $N_1 \leq N_4$.

After applying the delay matrix $\underline{\mathbf{B}}^{(k,t)}(z)$ to generate $\underline{\mathbf{R}}'(z)$ in the procedure outlined above, we have $N_1' = N_1$ and $N_4' = N_4$ with new values for N_2' and N_3'. The effect of applying the rotation matrix $\mathbf{Q}^{(j,k)}(\theta,\phi)$ in the procedure outlined above is to reduce N_3' by an amount $2g^2$ where $g = \left|r'_{jk}(0)\right|$. By construction, this is the maximum value of $|r_{lm}(\tau)|$ taken over all off-diagonal coefficients in $\underline{\mathbf{R}}(z)$. Since N_2' is unchanged by the application of $\mathbf{Q}^{(j,k)}(\theta,\phi)$, it follows that

$$\begin{aligned} N_1'' &= N_1' + 2g^2 = N_1 + 2g^2, \quad N_2'' = N_2', \\ N_3'' &= N_3' - 2g^2, \quad N_4'' = N_4' = N_4. \end{aligned} \tag{2.19}$$

The result of each iteration is to increase N_1 by $2g^2$ which constitutes twice the magnitude squared of the greatest off-diagonal polynomial coefficient. The proof of convergence now follows quite simply. Since N_1 increases monotonically and is bounded from above by N_4, which is constant, it must have a supremum S. It follows that for any $\varepsilon > 0$ there must be an iteration number L, say, for which $|S - N_1| < 2\varepsilon$ and so the increase in N_1 at any subsequent stage must satisfy $2g^2 \leq |S - N_1| < 2\varepsilon$. In other words, for any $\varepsilon > 0$ there must be an iteration by which the maximum off-diagonal polynomial coefficient is bounded (in magnitude squared) by ε.

Note that the value of g does not necessarily decline monotonically. Each rotation is computed with reference to elements in the coefficient-plane of order zero in $\underline{\mathbf{R}}'(z)$, and is guaranteed to increase N_1' by driving the (maximized) off-diagonal element to zero, thus reducing N_3'. However, in any other plane of the polynomial matrix $\underline{\mathbf{R}}'(z)$, where the same rotation is being applied, it could have the effect of increasing the magnitude of the off-diagonal element while reducing the sum of the squares of the diagonal elements. As a result, the dominant off-diagonal element, taken over all values of τ, could be larger at the start of the next iteration. Note also, that the difference between N_4 and the supremum of N_1 will generally be nonzero. The algorithm does not seek to reduce the on-diagonal coefficients for nonzero values of τ, let alone drive them to zero. In the context of strong decorrelation, as discussed in Section III, this would correspond to temporal whitening of the decorrelated signals, which is often highly undesirable and cannot occur as the result of a paraunitary transformation (which preserves the total power spectral density).

It is important to realize that the order of the polynomial matrix $\underline{\mathbf{R}}(z)$, and the corresponding paraunitary matrix $\underline{\mathbf{H}}(z)$, will grow as the number of iterations in the SBR2 algorithm increases. This is a natural consequence of using an FIR filter to compute the decomposition in (2.4). In practice, the order of the polynomials can grow much larger than is necessary to achieve a good approximation, leading to an increase in the computational load and corresponding slowing of the algorithm. However, the growth of $\underline{\mathbf{R}}(z)$ may be reduced very effectively by truncating the highest order coefficient matrices at each iteration such that the sum of the squares of all discarded coefficients is a very small fraction of the total initial value represented by N_4. This has been achieved in practice by applying the following "trim" function:

$$\text{trim}(\underline{\mathbf{R}}(z),\mu) = \sum_{\tau=-\tau_{\text{lim}}}^{\tau_{\text{lim}}} z^{-\tau}\mathbf{R}(\tau) \tag{2.20}$$

where τ_{lim} is the largest positive integer $\leq \tau_{\max}$ such that $\sum_{\tau_{\text{lim}}}^{\tau_{\max}} \|\mathbf{R}(\tau)\|_F^2 \geq \mu N_4/2$. Note that, since N_1 still increases monotonically and is bounded from above by the initial value of N_4, this procedure does not invalidate the proof of convergence given above. Furthermore, the modulus squared of the biggest off-diagonal coefficient which could be lost due to the truncation, is bounded by $\mu N_4/2$ times the number of iterations. Growth in the order of the paraunitary matrix $\underline{\mathbf{H}}(z)$ may be controlled in a similar manner, bearing in mind that it is not para-Hermitian. Since the computation of $\underline{\mathbf{H}}(z)$ (which could be performed after the main iterative process) has little effect on the speed of the algorithm, the procedure will not be specified here. A brief summary of the basic SBR2 algorithm is given in Table I.

The proof of convergence given for the SBR2 algorithm shows that the off-diagonal coefficients tend uniformly to zero. However, since the order of the polynomials can increase, this does not imply that the off-diagonal F-norm tends to zero. From a mathematical point of view, it would be much better to have a proof of convergence based on the F-norm but we have not succeeded in finding one. Nonetheless, the form of convergence proved here is quite powerful in its own right. We know that the sum of the squares of the off-diagonal elements is bounded by $N_4 - N_1$. Furthermore, since the maximum off-diagonal element can be made arbitrarily small in magnitude, the criterion for convergence could be specified in terms of the smallness of g relative to $\sqrt{N_1/p}$. In the context of strong decorrelation, $\sqrt{N_1/p}$ constitutes a lower bound for the maximum zero-lag autocorrelation value, so the form of convergence proved above is consistent with the essential objective for that particular application.

C. Worked Examples

We will now illustrate the operation of the SBR2 algorithm by means of some simple but insightful examples. In the first example, the algorithm was applied to the para-Hermitian matrix given by

$$\underline{\mathbf{R}}_1(z) = \begin{bmatrix} 1 & -0.4iz & 0 \\ 0.4iz^{-1} & 1 & 0.5z^{-2} \\ 0 & 0.5z^2 & 1 \end{bmatrix}. \tag{2.21}$$

In seven iterations, it converged to produce the factorization given by

$$\begin{aligned} \underline{\mathbf{H}}_1(z) &= \begin{bmatrix} 0.4417 & -0.7071iz & -0.5522iz^{-1} \\ -0.7809i & 0 & 0.6247z^{-1} \\ 0.4417i & -0.7071z & 0.5522z^{-1} \end{bmatrix} \\ \underline{\mathbf{D}}_1(z) &= \begin{bmatrix} 1.6403 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0 & 0 & 0.3597 \end{bmatrix}. \end{aligned} \tag{2.22}$$

The final value of g was zero (to computational precision). The value of N_1 increased from 3.0000 to 3.8200 which, in this case, is equal to the value of N_4. This reflects the fact that the sum of the squares of the diagonal elements of $\underline{\mathbf{D}}_1(z)$ was also zero except in the coefficient-plane of order zero. Since $\underline{\mathbf{H}}_1(z)$ is paraunitary, the inverse decomposition is given very simply by $\underline{\mathbf{R}}_1(z) = \tilde{\underline{\mathbf{H}}}_1(z)\underline{\mathbf{D}}_1(z)\underline{\mathbf{H}}_1(z)$. When this computation was carried out, the F-norm of the difference between the result obtained and the original matrix $\underline{\mathbf{R}}_1(z)$ was also zero (to computational precision). This example is particularly simple in the

TABLE I
SUMMARY OF THE SBR2 ALGORITHM

Input $p \times p$ para-Hermitian polynomial matrix $\underline{\mathbf{R}}(z)$.
Compute $N_4 = \|\underline{\mathbf{R}}(z)\|_{\mathrm{F}}^2$.
Specify maximum number of iterations, Max, convergence parameter, δ and trim factor μ.
Initialise parameters: iter $\leftarrow$ 0, g $\leftarrow$ 1+δ and $\underline{\mathbf{H}}(z) \leftarrow \mathbf{I}_p$.
while iter < Max and $g > \delta$
    locate dominant off-diagonal coefficient, $r_{jk}(t)$, in upper part of $\underline{\mathbf{R}}(z)$.
    set $g = |r_{jk}(t)|$.
    if $g > \delta$
        iter $\leftarrow$ iter + 1
        set $\underline{\mathbf{R}}'(z) = \underline{\mathbf{B}}^{(k,t)}(z)\underline{\mathbf{R}}(z)\widetilde{\underline{\mathbf{B}}}^{(k,t)}(z)$, according to (2.11).
        set $\underline{\mathbf{H}}'(z) = \underline{\mathbf{B}}^{(k,t)}(z)\underline{\mathbf{H}}(z)$.
        Compute rotation parameters (θ, ϕ), according to (2.13) and (2.14).
        update $\underline{\mathbf{R}}(z) = \mathbf{Q}^{(j,k)}(\theta,\phi)\underline{\mathbf{R}}'(z)\mathbf{Q}^{(j,k)\mathrm{H}}(\theta,\phi)$, according to (2.15).
        update $\underline{\mathbf{H}}(z) = \mathbf{Q}^{(j,k)}(\theta,\phi)\underline{\mathbf{H}}'(z)$.
        set $\underline{\mathbf{R}}(z) \leftarrow \mathrm{trim}(\underline{\mathbf{R}}(z), \mu)$ according to (2.20)
    end
end

sense that the matrix $\underline{\mathbf{R}}_1(z)$ can be reduced to a scalar matrix by initially applying two successive delay transformations of the type specified in (2.11). The problem then reduces to one of standard Jacobi diagonalization. Note, however, that this "trick" is not exploited by the SBR2 algorithm which performs a strict sequence of alternating delay and rotate operations as specified previously. The sequence of rotations applied during this process was, not surprisingly, identical to the sequence of rotations computed when applying the Jacobi algorithm to the reduced scalar matrix. Despite the simplicity of this example, it is worth bearing in mind that the same decomposition would be much more complicated if carried out in the frequency domain.

In the second example, the SBR2 algorithm was applied to the para-Hermitian matrix

$$\underline{\mathbf{R}}_2(z) = \begin{bmatrix} 1 & 0.8z^2 - 0.4z & 0.7z \\ 0.8z^{-2} - 0.4z^{-1} & 1 & 0.5z^{-2} \\ 0.7z^{-1} & 0.5z^2 & -1 \end{bmatrix}. \tag{2.23}$$

Note that the scalar reduction trick does not apply in this case. For ease of graphical presentation, all coefficients in this example were chosen to be real. A negative value was assigned to the (3, 3) element so that the constant coefficient matrix is not positive definite. The trim function defined in (2.20) was applied to the updated polynomial matrix $\underline{\mathbf{R}}_2(z)$ after each iteration with $\mu = 10^{-4}$. The equivalent, nonsymmetric trim function was applied to the paraunitary matrix $\underline{\mathbf{H}}_2(z)$ generated in the process. In this case, the SBR2 algorithm converged as shown in Fig. 1, reaching a level where $g < 10^{-3}$ in 23 iterations. The F-norm of the off-diagonal elements of the approximately diagonalized polynomial matrix $\underline{\mathbf{D}}_2(z)$ was 1.8×10^{-3}, which is clearly very small compared to $\|\underline{\mathbf{D}}_2(z)\|_F = \|\underline{\mathbf{R}}_2(z)\|_F = 2.47$ and also the value of $\sqrt{N_1}$ which increased from 1.73 to 2.45.

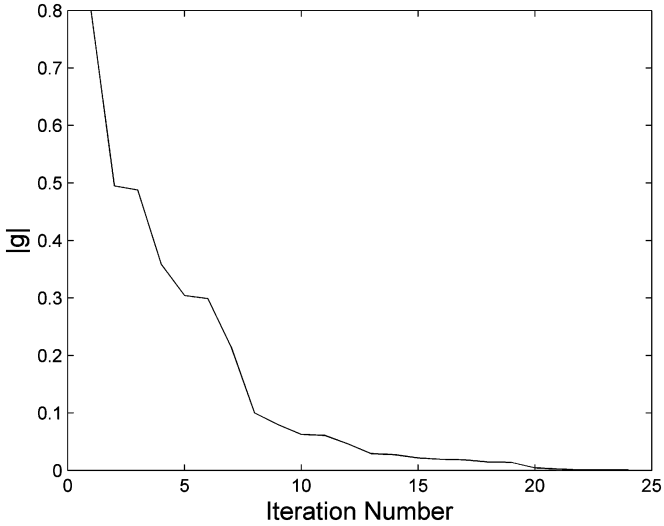


Fig. 1. Convergence of SBR2 algorithm for the example in (2.23).

The algorithm generated the polynomial matrices $\underline{\mathbf{D}}_2(z)$ (of order 4) and $\underline{\mathbf{H}}_2(z)$ (of order 6) depicted in Figs. 2 and 3, respectively. When the polynomial matrix $\underline{\mathbf{R}}_2(z)$ was reconstructed by computing the inverse decomposition $\underline{\mathbf{H}}_2(z)\underline{\mathbf{D}}_2(z)\underline{\mathbf{H}}_2(z)$, the F-norm of the error matrix was $\sim 5 \times 10^{-2}$. The same result was obtained using only the diagonal component of $\underline{\mathbf{D}}_2(z)$. Clearly, for this example, an excellent approximate decomposition can be achieved using polynomial matrices of very modest order. It is worth noting that when the computation was performed without applying the trim function, the algorithm took 37 iterations to converge to the same level. The order of the polynomial matrix $\underline{\mathbf{R}}_2(z)$ grew to 224 but the vast majority of higher order terms were either zero, or negligibly small. The corresponding

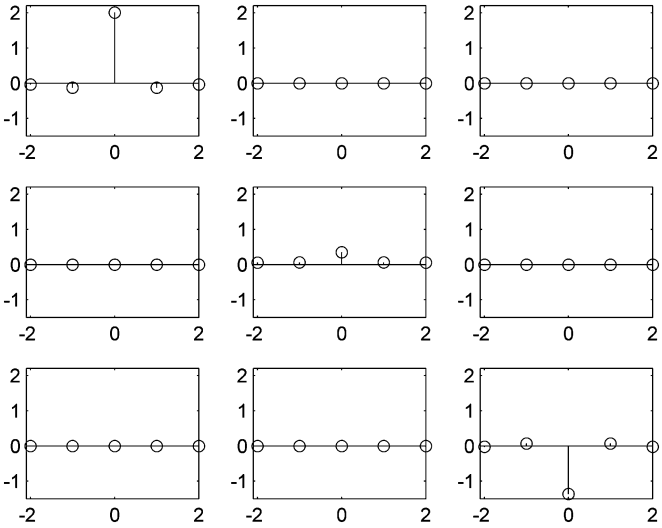


Fig. 2. Diagonalized polynomial matrix obtained using SBR2 algorithm for example in (2.23).

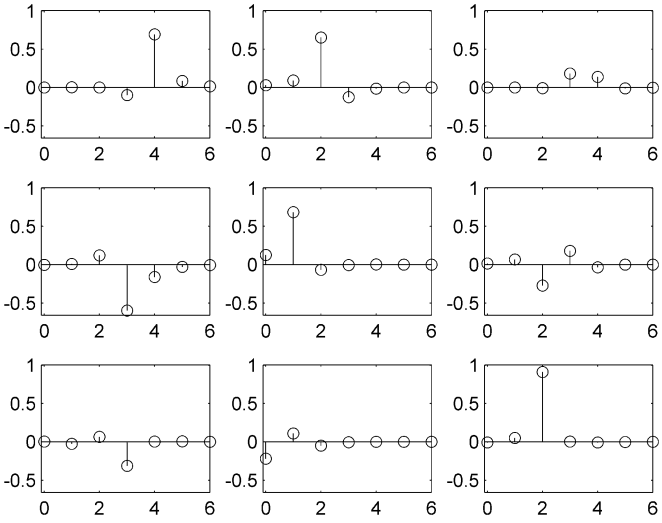


Fig. 3. Paraunitary matrix obtained using SBR2 algorithm for example in (2.23).

paraunitary matrix $\underline{\mathbf{H}}_2(z)$ was of order 111 and the F-norm of the reconstruction error matrix was $\sim 2 \times 10^{-3}$.

III. Application to Signal Processing

A. Strong Decorrelation

The SBR2 algorithm was originally developed for the purpose of generating a lossless (stable, all-pass) filter bank to decorrelate the signals received by a broadband sensor array. As mentioned in the introduction, strong decorrelation is required in the case of broadband signals. In other words, the decorrelation must be achieved over a suitable range of relative time delays. Assuming the signals have zero mean, the space-time covariance matrix is given by

$$\mathbf{R}_{xx}(\tau) = \mathrm{E}\{\mathbf{x}(t)\mathbf{x}^H(t-\tau)\} \quad \tau \in Z \tag{3.1}$$

and we denote the individual elements as

$$[\mathbf{R}_{xx}(\tau)]_{kl} = \mathrm{E}\{x_k(t)x_l^*(t-\tau)\} = r_{kl}^{(x)}(\tau). \tag{3.2}$$

The cross-spectral density matrix for a given frequency corresponds to evaluating the power series

$$\underline{\mathbf{R}}_{xx}(z) = \sum_{\tau=-\infty}^{\infty} \mathbf{R}_{xx}(\tau)z^{-\tau} \tag{3.3}$$

at a point on the unit circle (where it is assumed to converge). In general, the received signals are correlated, so $\mathbf{R}_{xx}(\tau)$ will not be diagonal $\forall\, \tau$ and hence $\underline{\mathbf{R}}_{xx}(z)$ will not be diagonal.

In practice, an estimate of the cross-spectral density matrix must be generated from the received data. This is typically given by a polynomial covariance matrix of the form

$$\hat{\underline{\mathbf{R}}}_{xx}(z) \triangleq \sum_{\tau=-W}^{W} \hat{\mathbf{R}}_{xx}(\tau)z^{-\tau} \tag{3.4}$$

where

$$\hat{\mathbf{R}}_{xx}(\tau) \triangleq \frac{\sum_{t=0}^{T-1} \mathbf{x}(t)\mathbf{x}^H(t-\tau)}{T}. \tag{3.5}$$

It is assumed that $\hat{\mathbf{R}}_{xx}(\tau) \cong 0$ for $|\tau| > W$. This reflects the fact that for broadband signals, the space-time correlation function is negligibly small if $|\tau|$ is large compared to the coherence time. In practice, the value of W is often measured experimentally. It is assumed that $T \gg W$ and also that $\mathbf{x}(t) = 0$ for values of t outside the sample interval $[0, T-1]$. It follows that $\left[\hat{\mathbf{R}}_{xx}(\tau)\right]_{kl} = \left[\hat{\mathbf{R}}_{xx}(-\tau)\right]_{lk}^*$, and so the polynomial matrix $\hat{\underline{\mathbf{R}}}_{xx}(z)$ is para-Hermitian by construction.

The SBR2 algorithm, as presented in Section II, may be used to generate a paraunitary matrix $\underline{\mathbf{H}}(z)$ s.t.

$$\underline{\mathbf{H}}(z)\hat{\underline{\mathbf{R}}}_{xx}(z)\tilde{\underline{\mathbf{H}}}(z) = \hat{\underline{\mathbf{D}}}(z) \tag{3.6}$$

where $\hat{\underline{\mathbf{D}}}(z)$ is approximately diagonal; more specifically,

$$\hat{\underline{\mathbf{D}}}(z) \cong \mathrm{diag}\{\hat{\underline{d}}_1(z), \hat{\underline{d}}_2(z), \ldots, \hat{\underline{d}}_p(z)\}. \tag{3.7}$$

The lossless filter represented by $\underline{\mathbf{H}}(z)$ is then applied to the received signals $\mathbf{x}(t)$ to produce the transformed sequence according to

$$\underline{\mathbf{v}}(z) = \underline{\mathbf{H}}(z)\underline{\mathbf{x}}(z) \tag{3.8}$$

where $\underline{\mathbf{x}}(z)$ and $\underline{\mathbf{v}}(z)$ denote the algebraic power series

$$\underline{\mathbf{x}}(z) = \sum_{t=-\infty}^{\infty} \mathbf{x}(t)z^{-t}$$
$$\underline{\mathbf{v}}(z) = \sum_{t=-\infty}^{\infty} \mathbf{v}(t)z^{-t}. \tag{3.9}$$

By analogy with (3.3), the transformed cross-spectral density matrix is given by

$$\underline{\mathbf{R}}_{vv}(z) = \sum_{\tau=-\infty}^{\infty} \mathbf{R}_{vv}(\tau)z^{-\tau} \tag{3.10}$$

where

$$\mathbf{R}_{vv}(\tau) = \mathrm{E}\{\mathbf{v}(t)\mathbf{v}^H(t-\tau)\}, \quad \tau \in Z \tag{3.11}$$

and we denote the individual elements by

$$[\mathbf{R}_{vv}(\tau)]_{kl} = \mathrm{E}\{v_k(t)v_l^*(t-\tau)\} = r_{kl}^{(v)}(\tau). \tag{3.12}$$

To a good approximation, the signals $\mathbf{v}(t)$ are strongly decorrelated since

$$\underline{\mathbf{R}}_{vv}(z) = \underline{\mathbf{H}}(z)\underline{\mathbf{R}}_{xx}(z)\tilde{\underline{\mathbf{H}}}(z) \tag{3.13}$$

and, hence, $\hat{\underline{\mathbf{R}}}_{vv}(z)$ is estimated by

$$\hat{\underline{\mathbf{R}}}_{vv}(z) = \underline{\mathbf{H}}(z)\hat{\underline{\mathbf{R}}}_{xx}(z)\tilde{\underline{\mathbf{H}}}(z) = \hat{\underline{\mathbf{D}}}(z). \tag{3.14}$$

It follows that $\mathbf{R}_{vv}(\tau)$ is given approximately by

$$\hat{\mathbf{R}}_{vv}(\tau) \cong \mathrm{diag}\{\hat{d}_1(\tau), \hat{d}_2(\tau), \ldots, \hat{d}_p(\tau)\} \tag{3.15}$$

where

$$\hat{d}_j(\tau) \cong r_{jj}^{(v)}(\tau) = \mathrm{E}\left\{v_j(t)v_j^*(t-\tau)\right\}, \quad (j = 1, 2, \ldots, p). \tag{3.16}$$

Note that the paraunitary matrix $\underline{\mathbf{H}}(z)$ can be applied to the received signals $\mathbf{x}(t)$ stage by stage during the SBR2 computation. At the nth stage, the elementary paraunitary matrix $\underline{\mathbf{G}}_n(z)$ contributing to (2.7) may be applied to the transformed signals produced as a result of the first $n-1$ stages, in order to generate the output data from stage n. This constitutes a sequence of pairwise delay and rotate operations since each elementary paraunitary matrix only affects two of the signal channels. Note, also, that the polynomial covariance matrix in (3.14) is updated as part of the SBR2 algorithm and does not need to be recomputed from the transformed data. The effectiveness of SBR2 as a technique for strong decorrelation is illustrated by means of the simulation results in Section IV.

B. Spectral Majorization and Power Compaction

By direct analogy with the narrowband case, the decorrelated output signals produced by the SBR2 algorithm may be numbered in decreasing order of their estimated powers. Noting, from (3.16), that

$$\mathrm{E}\left\{|v_j(t)|^2\right\} \cong \hat{d}_j(0) \tag{3.17}$$

the output signals are ordered such that

$$\hat{d}_1(0) \geq \hat{d}_2(0) \cdots \geq \hat{d}_p(0). \tag{3.18}$$

This amounts to ordering them in terms of their total estimated spectral power. Because the cross-spectral density matrix transforms according to (3.13), and the matrix $\underline{\mathbf{H}}(z)$ is paraunitary, it can be shown that

$$\begin{aligned}\mathrm{trace}\{\underline{\mathbf{R}}_{vv}(z)\} &= \mathrm{trace}\{\underline{\mathbf{H}}(z)\underline{\mathbf{R}}_{xx}(z)\tilde{\underline{\mathbf{H}}}(z)\} \\ &= \mathrm{trace}\{\underline{\mathbf{R}}_{xx}(z)\}.\end{aligned} \tag{3.19}$$

Setting $z = \mathbf{e}^{i\omega}$ then reveals the well-known property that the combined power of the received signals at every frequency is invariant to a paraunitary transformation [2], and hence that the total signal power is invariant. A paraunitary transformation can redistribute the spectral power between channels, but it cannot increase or decrease it. Without this property, the power of the output signals would have no physical significance.

We have observed that the SBR2 algorithm has a very strong tendency to produce spectrally majorized output signals, thereby compacting the maximum power into the least number of signal channels [10]. This is exemplified by the results in Section IV. Two signals $v_1(t)$ and $v_2(t)$ are said to be spectrally majorized if

$$s_1^{(v)}(\omega) \geq s_2^{(v)}(\omega) \quad \forall\, \omega \in [-\pi, \pi). \tag{3.20}$$

In other words, the expected power in $v_1(t)$ is greater than that in $v_2(t)$ at every frequency. This additional feature of the algorithm is attributed to the fact that it imposes a monotonic increase in, and often maximizes, the value of N_1 as defined in (2.18). N_1 is a convex function of the (estimated) output signal powers $r_{ii}(0)$ and, in the context of orthonormal filterbanks, Akkarakaran and Vaidyanathan [22] have shown that optimizing such a cost function corresponds to generating a principal component filterbank (PCFB) which ensures both spectral majorization and optimal power compaction. Note, however, that there is no guarantee that the SBR2 algorithm will always maximize N_1 and thereby achieve spectral majorization. In fact it is easy to see that two signals in nonoverlapping subbands (or two different sinusoids), while not spectrally majorized, will be strongly decorrelated from the outset, so the SBR2 algorithm can make no further improvement. However, situations such as this seem to represent unstable fixed points for the algorithm; they rarely occur in practice and can be circumvented by means of very small perturbations.

IV. Computer Simulated Results

We consider a situation in which the signals $\mathbf{s}(t) \in C^q$ emitted from q statistically independent broadband sources are received by an array of p sensors (where $p \geq q$) over a convolutive channel. In this case, the received signals $\mathbf{x}(t) \in C^p$ may be represented by a convolutive mixing model of the form

$$\mathbf{x}(t) = \sum_{k=0}^{L} \mathbf{A}(k)\mathbf{s}(t-k) + \boldsymbol{\eta}(t) \tag{4.1}$$

where $\mathbf{A}(k) \in C^{p\times q}$, $k \in \{0, 1, 2, \ldots, L\}$ and $\boldsymbol{\eta}(t)$ represents a Gaussian noise process with variance $\sigma^2\mathbf{I}_p$. The convolutive mixing process may also be written in the form

$$\underline{\mathbf{x}}(z) = \underline{\mathbf{A}}(z)\underline{\mathbf{s}}(z) + \underline{\boldsymbol{\eta}}(z) \tag{4.2}$$

where $\underline{\mathbf{x}}(z)$, $\underline{\mathbf{s}}(z)$, and $\underline{\boldsymbol{\eta}}(z)$ again denote algebraic power series of the form given in (3.9) and

$$\underline{\mathbf{A}}(z) = \sum_{k=0}^{L} \mathbf{A}(k)z^{-k}. \tag{4.3}$$

As a result of the mixing process in (4.2), the received signals $\underline{\mathbf{x}}(t)$ will generally be correlated over multiple time lags. Accordingly, their cross-spectral density matrix, given by

$$\begin{aligned}\underline{\mathbf{R}}_{xx}(z) &= \underline{\mathbf{A}}(z)\underline{\mathbf{R}}_{ss}(z)\tilde{\underline{\mathbf{A}}}(z) + \sigma^2\mathbf{I}_p \\ &= \underline{\mathbf{A}}(z)\tilde{\underline{\mathbf{A}}}(z) + \sigma^2\mathbf{I}_p\end{aligned} \tag{4.4}$$

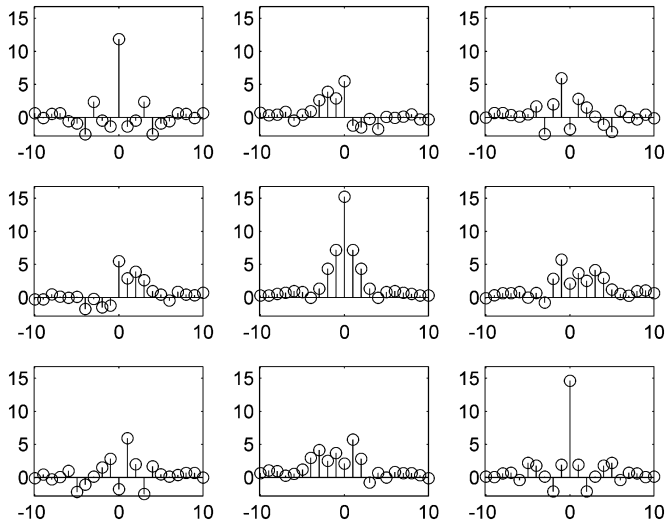


Fig. 4. Polynomial covariance matrix for three mixed signals generated from 2 i.i.d source signals at 5.3 dB SNR.

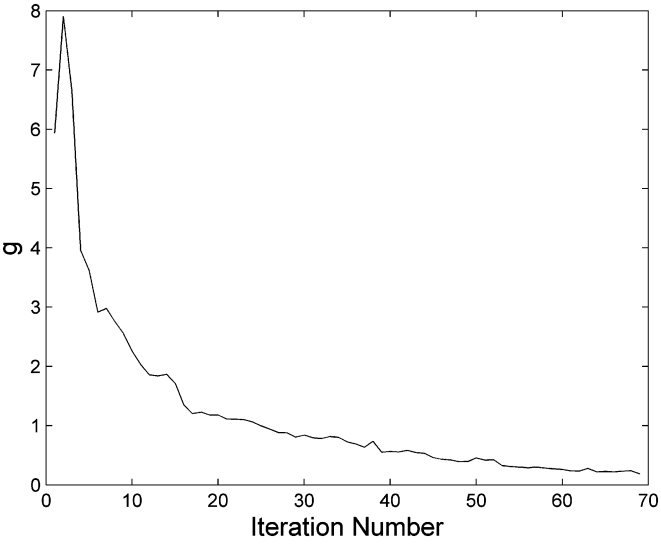


Fig. 5. Convergence of SBR2 algorithm for diagonalization of polynomial covariance matrix in Fig. 4.

will not be diagonal. However, once an estimate of $\underline{\mathbf{R}}_{xx}(z)$ has been generated according to (3.5), a paraunitary transformation may be computed using the SBR2 algorithm and applied to the received signals $\mathbf{x}(t)$ in order to generate strongly decorrelated outputs $\mathbf{v}(t)$ as in (3.6) and (3.8).

In order to demonstrate the effectiveness of the SBR2 algorithm, we first present the results of a simple computer simulation in which it was assumed that $q = 2$ and $p = 3$. The mixing process was represented by a 3×2 polynomial matrix $\underline{\mathbf{A}}(z)$ whose entries comprised order-5 FIR filters with coefficients drawn randomly from a uniform distribution in the range $[-1, 1]$. The source signals took the form of independent identically distributed (i.i.d.) sequences for which each sample was assigned the value ± 1 with probability 1/2. Gaussian random noise was added to each simulated sensor output with $\sigma = 1.8$, which corresponds to a signal-to-noise ratio (SNR) of 5.3 dB. The SNR for this experiment, is simply given by $\beta/3\sigma^2$ where $\beta = \left.\mathrm{Tr}\{\underline{\mathbf{A}}(z)\tilde{\underline{\mathbf{A}}}(z)\}\right|_{\tau=0}$. The number of samples, T, used to estimate the space-time covariance matrix in (3.5) was chosen to be 1000. The correlation window parameter, W, as defined in (3.4), was set to 10, although 5 would have been sufficient to reflect the statistics of the data and the order of the mixing matrix.

The polynomial covariance matrix $\hat{\underline{\mathbf{R}}}_{xx}(z)$ computed from the received signals $\mathbf{x}(t)$ for one specific realization is plotted in Fig. 4. The SBR2 algorithm was applied to diagonalize this matrix using the trim function in (2.20) with $\mu = 10^{-4}$. As shown in Fig. 5, the algorithm converged in 68 iterations to a point where $g \leq \sqrt{N_1/3} \times 10^{-2}$. The value of g can be driven much lower by setting a tighter convergence bound but the additional iterations are of no benefit here, due to the levels of noise and statistical error estimating $\hat{\underline{\mathbf{R}}}_{xx}(z)$ from 1000 data samples. The order of the resulting diagonal matrix $\hat{\underline{\mathbf{D}}}(z)$ in this case was 26. The corresponding paraunitary matrix $\underline{\mathbf{H}}(z)$ was of order 31 and is illustrated in Fig. 6. When the decomposition was inverted using the computed polynomial matrices $\hat{\underline{\mathbf{D}}}(z)$ and $\underline{\mathbf{H}}(z)$, the F-norm of the error matrix was 1.2 compared to a total value of 35.84. The computed matrix $\underline{\mathbf{H}}(z)$ was used to

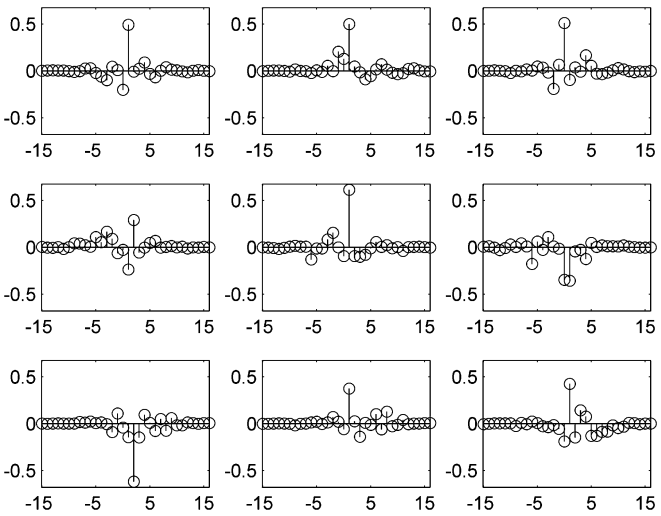


Fig. 6. Paraunitary matrix produced by SBR2 algorithm to diagonalize polynomial matrix in Fig. 4.

generate transformed signals $\mathbf{v}(t)$ as indicated in (3.8) and the polynomial covariance matrix $\hat{\underline{\mathbf{R}}}_{vv}(z)$ estimated from those signals was effectively diagonal as shown in Fig. 7. The F-norm of the off-diagonal elements of this matrix was 1.66 compared to the total F-norm of 35.31. A more appropriate statistical measure of the strong decorrelation properties of $\underline{\mathbf{H}}(z)$ was derived by computing the expected polynomial covariance matrix $\underline{\mathbf{R}}_{vv}(z) = \underline{\mathbf{H}}(z)\underline{\mathbf{A}}(z)\tilde{\underline{\mathbf{A}}}(z)\tilde{\underline{\mathbf{H}}}(z)$ which relates to the case of unit power, i.i.d. signals transmitted over the same convolutive channel in the absence of receiver noise. The F-norm of the off-diagonal elements of this matrix was found to be 3.82 compared to $\|\underline{\mathbf{R}}_{vv}(z)\|_F = 32.88$. This indicates that good strong decorrelation has been achieved in the true statistical sense.

It is evident from Fig. 7 that the SBR2 algorithm has concentrated most of the output power into $v_1(t)$ and most of the remaining power into $v_2(t)$. Furthermore, the power of $v_3(t)$ is comparable to the receiver noise power given by σ^2. The power

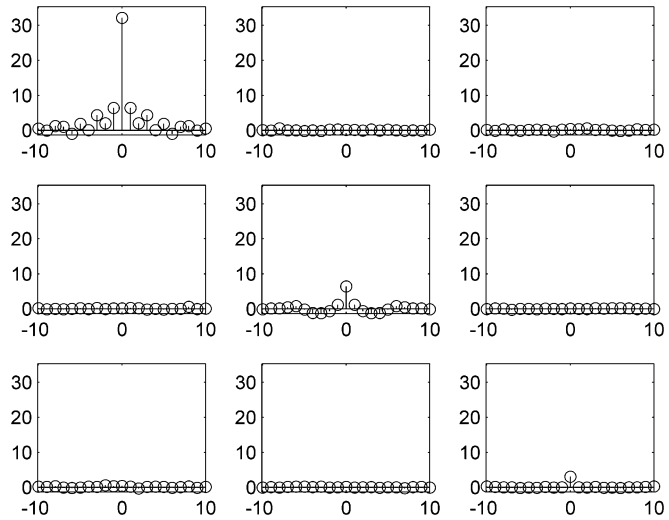


Fig. 7. Polynomial covariance matrix for the three transformed signals produced using paraunitary matrix in Fig. 6.

compaction properties of $\underline{\mathbf{H}}(z)$ can also be judged more reliably from the expected polynomial covariance matrix as defined above. This reveals that $r_{11}^{v}(0) = 28.86$, $r_{22}^{v}(0) = 3.73$, and $r_{33}^{v}(0) = 0.28$ demonstrating that the algorithm has achieved a very effective broadband subspace decomposition whereby almost all of the i.i.d. signal power is confined to the first two output channels. The power spectral density of the output signals $\mathbf{v}(t)$ is shown in Fig. 8(b) which demonstrates that approximate spectral majorization has also been achieved. The corresponding plots for the mixed signals $\mathbf{x}(t)$ prior to applying SBR2 are shown for comparison in Fig. 8(a).

In order to demonstrate how the SBR2 algorithm handles larger scale problems, the computer simulation described above was repeated for the case of five i.i.d. signals and ten sensors ($q = 5, p = 10$) using the same criterion for convergence and the same SNR (5.3 dB). When the resulting polynomial covariance matrix $\underline{\hat{\mathbf{R}}}_{xx}(z)$, of dimension 10×10, was diagonalized using SBR2 with the same trim factor ($\mu = 10^{-4}$), the algorithm converged in 616 iterations to achieve very good strong decorrelation as shown in Fig. 9. This increase in the number of iterations, by approximately one order of magnitude, is consistent with the number of polynomial matrix elements p^2, increasing from 9 to 100. In this case, the computed polynomial matrices $\underline{\hat{\mathbf{D}}}(z)$ and $\underline{\mathbf{H}}(z)$ were of order 34 and 65, respectively.

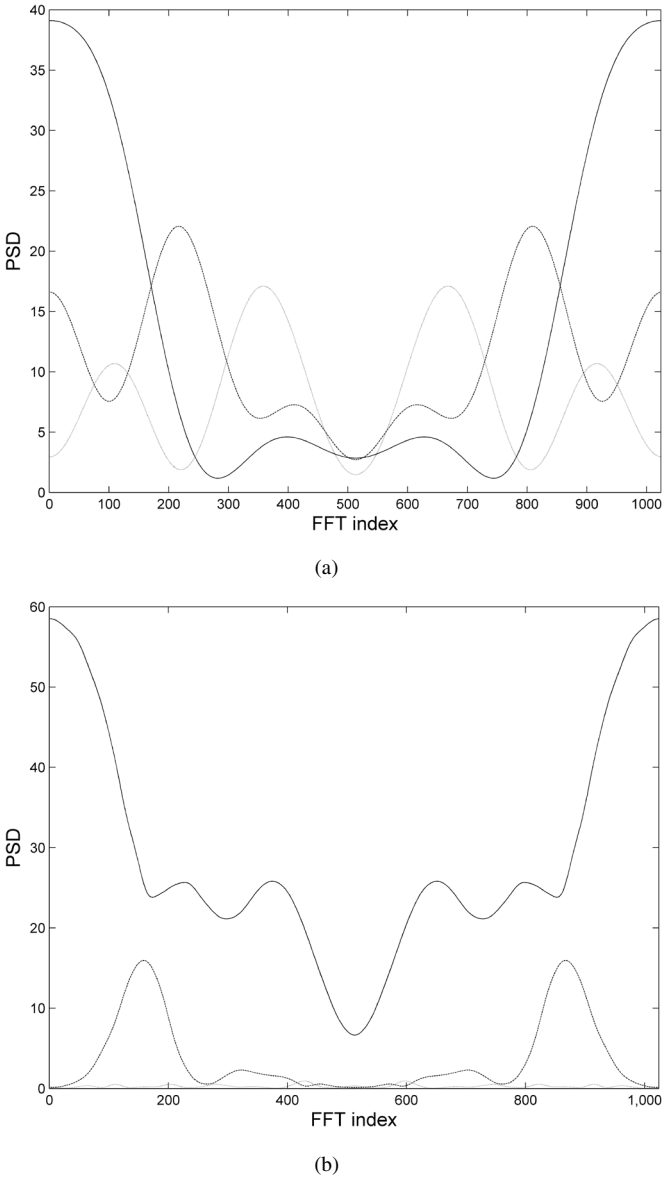


Fig. 8. (a) Power spectral densities of the three mixed signals associated with Fig. 4. (b) Power spectral densities of the three transformed signals associated with Fig. 7. Solid line: output 1; Broken line: output 2; Dashed line: output 3.

V. Comments and Conclusion

In this paper, we have presented the SBR2 algorithm for computing the EVD of a para-Hermitian polynomial matrix. It constitutes a novel generalization of the classical Jacobi algorithm for computing the EVD of conventional Hermitian matrices and has been shown to converge after sufficient iterations. An alternative version, based on the cyclic-by-rows Jacobi algorithm [13], has also been developed but was found to offer little advantage in most circumstances. The SBR2 algorithm can also be used to compute the SVD of a more general polynomial matrix representing, for example, the convolutive mixing process inherent in a MIMO communication channel. By analogy with the case of conventional matrices, the SVD of a general polynomial matrix $\underline{\mathbf{A}}(z)$ may be derived by carrying out the EVD of the para-Hermitian matrices $\underline{\mathbf{A}}(z)\underline{\tilde{\mathbf{A}}}(z)$ (to obtain the left hand polynomial singular vectors) or $\underline{\tilde{\mathbf{A}}}(z)\underline{\mathbf{A}}(z)$ (to obtain the right hand polynomial singular vectors). This technique has been used successfully to design orthogonal space-time channels for optimal data transmission over a simulated MIMO channel and the results will be reported in a future publication.

In principle, the concept of an elementary paraunitary matrix as introduced in this paper, could be used to generalize other, more sophisticated, EVD or SVD algorithms for application to polynomial matrices. An obvious objective is to develop an algorithm for direct computation of the polynomial matrix SVD without forming the products $\underline{\mathbf{A}}(z)\underline{\tilde{\mathbf{A}}}(z)$ or $\underline{\tilde{\mathbf{A}}}(z)\underline{\mathbf{A}}(z)$ as suggested above. Elementary paraunitary transformations could also be used to generate polynomial matrix versions of other important numerical procedures such as the QR decomposition, which is often used in narrowband array processing. Significant

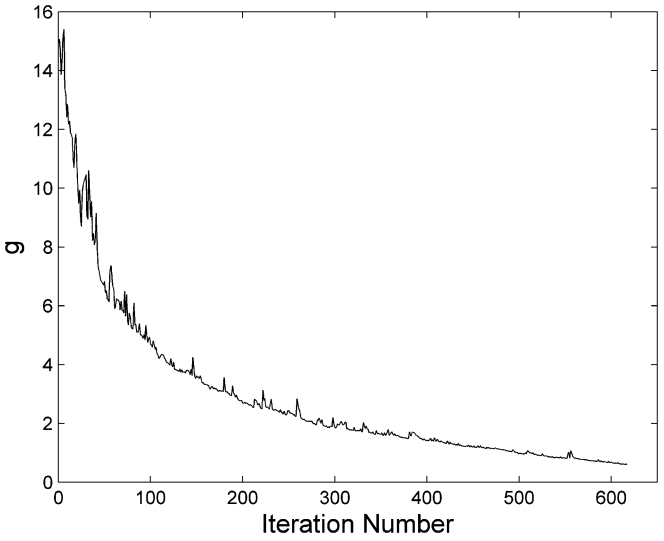


Fig. 9. Convergence of SBR2 algorithm in the case of ten mixed signals generated from five i.i.d. source signals at 5.3 dB SNR.

progress has been made on this topic but further discussion is beyond the scope of this paper.

In this paper, we have only attempted to indicate the relevance of the SBR2 algorithm to signal processing in the context of strong decorrelation and energy compaction. However, it could have as wide a range of applications for convolutive (broadband) sensor array signal processing as the conventional EVD or SVD algorithm does for instantaneous (narrowband) sensor array signal processing. It has already been applied successfully to data obtained from real sensor arrays in a number of application areas including sonar and seismology. However, discussion of the specific applications and results is beyond the scope of this paper. It has also been adopted successfully by other researchers for the purpose of designing oversampled filterbanks for channel coding [23] and for second order blind signal separation, applied to polarized signals from a three-axis seismic sensor array using quaternion (hypercomplex) arithmetic [24].

In conclusion, the purpose of this paper was to outline a novel technique based on some highly original ideas which seem to have great potential for sensor array signal processing. By virtue of its novelty, the paper raises numerous interesting and important questions which require more detailed theoretical research. For example: can a stronger proof of convergence be obtained for SBR2? Could a suitably modified cost function lead to an alternative algorithm with stronger convergence? Under what circumstances can a good decomposition be guaranteed using polynomial matrices of reasonable order? Can other algorithms from numerical linear algebra be extended to polynomial matrices in this way? The authors hope that this paper will stimulate other researchers to address some of these questions or, at least, try the SBR2 technique for themselves.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and reviewers for their helpful comments and observations which have enabled them to improve the presentation of this paper.

REFERENCES

[1] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice Hall, 1980.

[2] P. P. Vaidyanathan, *Multirate Systems and Filterbanks*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[3] N. Delfosse and P. Loubaton, "Adaptive blind separation of independent sources: A second-order stable algorithm for the general case," *IEEE Trans. Circuits Systems I, Fundam. Theory Appl.*, vol. 47, no. 7, pp. 1056–1070, Jul. 2000.

[4] R. H. Lambert, M. Joho, and H. Mathis, "Polynomial singular values for number of wideband sources estimation and principal components analysis," in *Proc Int. Conf. Independent Component Analysis*, 2001, pp. 379–383.

[5] P. A. Regalia and P. Loubaton, "Rational subspace estimation using adaptive lossless filters," *IEEE Trans. Signal Process.*, vol. 40, no. 10, pp. 2392–2405, Oct. 1992.

[6] S.-Y. Kung, Y. Wu, and X. Zhang, "Bezout space-time precoders and equalizers for MIMO channels," *IEEE Trans. Signal Process.*, vol. 50, no. 10, pp. 2499–2514, Oct. 2002.

[7] X. Xia, G. Wang, and P. Fan, "Space-time modulated codes for memory channels: Capacity and information rates, zero-forcing decision feedback equalizer," in *Proc. IEEE Sensor Array Multichannel Signal Process. Workshop*, Mar. 2000, pp. 183–187.

[8] A. Medles and D. T. M. Slock, "Linear precoding for spatial multiplexing MIMO systems: Blind channel estimation aspects," in *Proc. IEEE Int. Conf. Commun.*, Apr. 2002, pp. 401–404.

[9] L. Rota, P. Comon, and S. Icart, "Blind MIMO paraunitary equalizer," in *Proc IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2003, pp. 285–288.

[10] P. P. Vaidyanathan, "Theory of optimal orthonormal subband coders," *IEEE Trans. Signal Process.*, vol. 46, no. 6, pp. 1528–43, Jun. 1998.

[11] P. Moulin and M. K. Mihcak, "Theory and design of signal-adapted FIR paraunitary filter banks," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 920–929, Apr. 1998.

[12] X. Gao, T. Q. Nguyen, and G. Strang, "On factorization of M-channel paraunitary filterbanks," *IEEE Trans. Signal Process.*, vol. 49, no. 7, pp. 1433–1446, Jul. 2001.

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: John Hopkins Univ. Press, 1996.

[14] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2002.

[15] K. J. Pope and R. E. Bogner, "Blind signal separation II, linear convolutive systems," *Digit. Signal Process.*, vol. 6, pp. 17–28, 1996.

[16] P. D. Baxter and J. G. McWhirter, "Blind signal separation of convolutive mixtures," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2003, pp. 124–128.

[17] J. G. McWhirter and P. D. Baxter, "A novel technique for broadband SVD," presented at the 12th Annu. Workshop Adaptive Sensor Array Process., Mar. 2004.

[18] S. Redif and T. Cooper, "Paraunitary filterbank design via a polynomial singular value decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2005, pp. 613–616.

[19] R. Lambert, "Multichannel blind deconvolution: FIR matrix algebra and separation of multipath mixtures," Ph.D. dissertation, Univ. Southern California, Los Angeles, 1996.

[20] P. A. Regalia and D.-Y. Huang, "Attainable error bounds in multirate adaptive lossless FIR filters," in *Proc. IEEE Conf. Acoust., Speech Signal. Process.*, 1995, pp. 1460–63.

[21] R. Klemm, *Space-time Adaptive Processing Principles and Applications*, ser. 9. London, U.K.: IEE Radar, Sonar, Navigation and Avionics, 1998.

[22] S. Akkarakaran and P. P. Vaidyanathan, "Filterbank optimization with convex objectives and the optimality of principal component forms," *IEEE Trans. Signal Process.*, vol. 49, no. 1, pp. 100–114, Jan. 2001.

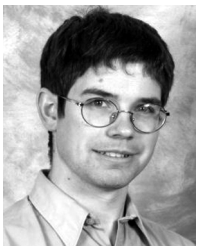
[23] S. Weiss, S. Redif, T. Cooper, C. Liu, P. D. Baxter, and J. G. McWhirter, "Paraunitary oversampled filterbank design for channel coding," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 1–10, 2006, 31346.

[24] N. Le Bihan, "Diagonalisation de matrices polynomiales quaternioniques: Application a la seperation de melanges convolutifs d'ondes polarisees," presented at the GRETSI Conf., Louvaine-la-Neuve, Belgium, Sep. 2005.

John G. McWhirter received the degree (first class hons.) in mathematics and the Ph.D. degree in theoretical physics from the Queen's University Belfast, Belfast, U.K., in 1970 and 1973.

He joined the Royal Radar Establishment, Malvern, U.K., (later to become the Royal Signals and Radar Establishment, and now part of QinetiQ Ltd.) in 1973. He is currently a Senior Fellow in the Centre for Signal and Information Processing Group, QinetiQ Ltd., Malvern, U.K. He is also a Distinguished Research Professor in the School of Engineering at Cardiff University, Wales, U.K., and a Visiting Professor in Electrical Engineering at the Queen's University Belfast. He has been carrying out research on adaptive signal processing since 1980. He has published more than 140 research papers and holds numerous patents. His current research is devoted to broadband sensor arrays, convolutive blind signal separation, and polynomial matrix techniques.

Dr. McWhirter was awarded the J. J. Thomson Medal by the IEE, U.K., in 1994 for his research on systolic arrays. The signal processing group, which he built up in Malvern over many years, received the EURASIP Group Technical Achievement Award for 2003. He was elected as a Fellow of the Royal Academy of Engineering in 1996 and as a Fellow of the Royal Society in 1999. He is a Fellow of the Institute of Mathematics and its Applications (IMA) and served as President of the IMA in 2002 and 2003. He is also a Fellow of the IEE, U.K., a Fellow of the Institute of Physics, and a member of the London Mathematical Society.



Paul D. Baxter received the B.A. degree in mathematics and a distinction in the certificate of advanced study in mathematics from the University of Cambridge, Cambridge, U.K., and the Ph.D. degree in electrical engineering from Imperial College, University of London, London, U.K., in 1998, 1999, and 2005, respectively.

He is currently working for QinetiQ, Malvern, U.K., after joining its predecessor, the Defence Evaluation and Research Agency in 1999. His research interests are in blind signal separation, signal processing for broadband signals and applications of polynomial matrix decompositions to signal processing.

Tom Cooper received the B.Sc. degree (first class hons.) in pure mathematics from Reading University, Reading, U.K., in 1989 and the M.Sc. and Ph.D. degrees from Warwick University, Coventry, U.K., in 1991 and 1994, respectively, both in mathematics. The subject of his Ph.D. dissertation was singularity theory.

In 1995, he joined the Defence Research Agency as a research scientist, working initially on ocean waves. In 2001 he joined QinetiQ Ltd., Malvern, U.K., in the Advanced Signal and Information Processing group, and has worked on Cramér–Rao bounds and algorithms for direction of arrival estimation, polynomial matrix algorithms for signal processing, and algorithms for processing radar data.

Soydan Redif (A'99) received the B.Eng. degree (first class hons.) in electronic engineering from Middlesex University, London, U.K., in 1998, and the Ph.D. degree from the University of Southampton, Southampton, U.K., in 2006

From 1999 to 2000, he was with the Communications Department at the Defence, Evaluation and Research Agency, Defford, where he worked on airborne mobile SHF and EHF satellite communications systems. In October 2000, he joined the Advanced Signal and Information Processing group at QinetiQ Ltd., Malvern, U.K., as a Research Scientist. His research contributions have been in broadband adaptive filtering and beamforming, blind signal separation, multirate systems and algorithms for polynomial matrix computations.

Dr. Redif is a Chartered Engineer, U.K., a Chartered Mathematician, U.K., a member of the IET, and a member of the IMA. He was the recipient of the IEE Award for the Best Engineering Student in 1998.

Joanne Foster received the M.S. degree in mathematics from the University of Bath, Bath, U.K., in 2004. She is currently working toward the Ph.D. degree in the Centre of Digital Signal Processing, School of Engineering, Cardiff University, Wales, U.K., under the supervision of Prof. J. Chambers.

She is currently based at QinetiQ Ltd., Malvern, U.K., since her research is funded through an EPSRC industrial CASE award, partly supported by QinetiQ.