ORDER-CONTROLLED MULTIPLE SHIFT SBR2 ALGORITHM FOR PARA-HERMITIAN POLYNOMIAL MATRICES

Zeliang Wang, John G. McWhirter*, Jamie Corr† and Stephan Weiss†*

* School of Engineering, Cardiff University, Cardiff, Wales, UK
† Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, Scotland, UK

ABSTRACT

In this work we present a new method of controlling the order growth of polynomial matrices in the multiple shift second order sequential best rotation (MS-SBR2) algorithm which has been recently proposed by the authors for calculating the polynomial matrix eigenvalue decomposition (PEVD) for para-Hermitian matrices. In effect, the proposed method introduces a new elementary delay strategy which keeps all the row (column) shifts in the same direction throughout each iteration, which therefore gives us the flexibility to control the polynomial order growth by selecting shifts that ensure non-zero coefficients are kept closer to the zero-lag plane. Simulation results confirm that further order reductions of polynomial matrices can be achieved by using this direction-fixed delay strategy for the MS-SBR2 algorithm.

Index Terms— MS-SBR2, Polynomial Matrix EVD, Order Growth Control.

1. INTRODUCTION

Polynomial matrices [1] often arise in describing the convolutive mixing for broadband sensor array processing. Assuming the sensor output signals $\mathbf{x}[t] \in \mathbb{C}^M$ have zero mean, the space-time covariance matrix $\mathbf{R}[\tau] \in \mathbb{C}^{M\times M}$ can be used to formulate the correlation of the sensor outputs, which is expressed as $\mathbf{R}[\tau] = E\{\mathbf{x}[t]\mathbf{x}^{\mathrm{H}}[t-\tau]\}$. Here $t, \tau \in \mathbb{Z}$, $E\{\cdot\}$ represents the expectation operation and $\{\cdot\}^{\mathrm{H}}$ denotes conjugate transposition. Therefore, its z-transform yields a polynomial cross-spectral density (CSD) matrix taking the form of

$$\underline{\mathbf{R}}(z) = \sum_{\tau=-T}^{T} \mathbf{R}[\tau] z^{-\tau} = \begin{bmatrix} \underline{r}_{11}(z) & \cdots & \underline{r}_{1M}(z) \\ \vdots & \ddots & \vdots \\ \underline{r}_{M1}(z) & \cdots & \underline{r}_{MM}(z) \end{bmatrix}, \tag{1}$$

with the polynomial order given by $2T+1$, such that $\mathbf{R}[\tau] = \mathbf{0}\ \forall|\tau| > T$, and each element of this matrix is a polynomial represented by $\underline{r}_{mn}(z) = \sum_\tau r_{mn}[\tau] z^{-\tau}$. Note that the CSD matrix is para-Hermitian (PH) which satisfies $\widetilde{\underline{\mathbf{R}}}(z) = \underline{\mathbf{R}}(z)$. The notation $\widetilde{\cdot}$ upon a polynomial matrix denotes the para-conjugate operation, i.e., $\widetilde{\underline{\mathbf{R}}}(z) = \underline{\mathbf{R}}^{\mathrm{H}}(1/z)$, which means take the conjugate transposition for all the coefficient matrices $\mathbf{R}[\tau]$ and time-reversing all the elements inside. Unless otherwise stated, polynomial matrices in this paper are represented by underscored upper case bold characters.

The conventional eigenvalue decomposition (EVD) can be used to diagonalize the covariance matrix which corresponds to decorrelating instantaneously mixed signals in the narrowband situation, but it is not suitable to generate the equivalent decomposition of a polynomial matrix $\underline{\mathbf{R}}(z)$. To solve this problem, an extension of the EVD to polynomial matrices has been proposed in [2], and its idea has been generalized as

$$\underline{\mathbf{H}}(z)\underline{\mathbf{R}}(z)\widetilde{\underline{\mathbf{H}}}(z) \approx \underline{\mathbf{D}}(z) \quad , \tag{2}$$

where $\underline{\mathbf{H}}(z)$ is a paraunitary (PU) matrix, i.e., $\underline{\mathbf{H}}(z)\widetilde{\underline{\mathbf{H}}}(z) = \widetilde{\underline{\mathbf{H}}}(z)\underline{\mathbf{H}}(z) = \mathbf{I}_{M\times M}$, and it aims to diagonalize $\underline{\mathbf{R}}(z)$ by means of paraunitary similarity transformation. $\underline{\mathbf{D}}(z)$ is (ideally) a diagonal matrix.

Several algorithms exist for calculating the PEVD in (2), including the most established SBR2 algorithm [2], its faster converging version, MS-SBR2 [3] and the family of sequential matrix diagonalization (SMD) algorithms [4–6]. One common feature among these PEVD algorithms is that the order of polynomial matrices continuously increases with each iteration. This is problematic, as such order growth will lead to a significant increasing in computational complexity. In addition, paraunitary matrices with high order will cause costly implementation for applications including subband coding [7], precoding and equalization design for broadband MIMO systems [8], blind source separation from convolutive mixtures [9], and spectral factorization [10] etc.

This paper introduces a new elementary delay strategy for the MS-SBR2 algorithm which can be used to restrict the direction of all row (column) shifts throughout iterations. The benefit of doing this is that all the zero filled outer lags of polynomial matrices can be precisely tracked and removed without affecting the algorithm convergence. In other words, it is a lossless process.

In this paper, we aim to investigate how this direction-fixed delay strategy can exploited to limit the polynomial order growth in the MS-SBR2 algorithm. In particular, the objective is to see if any further order reductions can be achieved

during the truncation process [2, 11, 12] while the direction-fixed delay strategy is involved in the MS-SBR2 algorithm.

To accomplish this, we start by briefly reviewing the SBR2 and MS-SBR2 algorithms in Sec. 2. Sec. 3 presents the details of the method for controlling the polynomial order growth in the MS-SBR2 algorithm. Simulation results and conclusions are shown in Sec. 4 and Sec. 5 respectively.

2. STATE OF THE ART

2.1. SBR2 Algorithm

The SBR2 algorithm calculates the PEVD by using a sequence of elementary paraunitary operations to iteratively diagonalize the para-Hermitian matrix $\underline{\mathbf{R}}(z)$. Each elementary paraunitary operation consists of two steps, i.e., an elementary delay and a Jacobi transformation. At the i-th iteration, the SBR2 algorithm starts by finding the maximum off-diagonal element $r_{jk}^{(i)}[\tau]$ within the upper triangular area of $\mathbf{R}^{(i-1)}[\tau]$. Thus the location of $r_{jk}^{(i)}[\tau]$, $(k > j)$ satisfies

$$\{j^{(i)}, k^{(i)}, \tau^{(i)}\} = \arg \max_{j,k>j,\tau} \|\mathbf{R}^{(i-1)}[\tau]\|_\infty \quad , \tag{3}$$

where $j^{(i)}$, $k^{(i)}$ and $\tau^{(i)}$ are the corresponding row, column and time lag index.

Then the maximum element $r_{jk}^{(i)}[\tau]$ and its complex conjugate $r_{kj}^{(i)}[-\tau]$ are shifted onto the zero-lag ($\tau = 0$) by using the elementary delay matrix, such that

$$\underline{\mathbf{R}}'^{(i)}(z) = \underline{\mathbf{P}}^{(i)}(z)\underline{\mathbf{R}}^{(i-1)}(z)\widetilde{\underline{\mathbf{P}}}^{(i)}(z) \quad , \tag{4}$$

where $\underline{\mathbf{R}}'^{(i)}(z)$ denotes the intermediate matrix after the elementary delay operation, and the delay matrix $\underline{\mathbf{P}}^{(i)}(z)$ takes the form of

$$\underline{\mathbf{P}}^{(i)}(z) = \text{diag}\{\underbrace{1 \cdots 1}_{k^{(i)}-1}\ z^{-\tau^{(i)}}\ \underbrace{1 \cdots 1}_{M-k^{(i)}}\} \quad , \tag{5}$$

which means shifting the maximum element in the $k^{(i)}$-th row by $|\tau^{(i)}|$ lags onto the zero-lag. Finally the maximum element is brought onto the diagonal using the Jacobi transformation $\mathbf{Q}^{(i)}$ [2], which results in

$$\underline{\mathbf{R}}^{(i)}(z) = \mathbf{Q}^{(i)}\underline{\mathbf{R}}'^{(i)}(z)\mathbf{Q}^{\mathrm{H}(i)} \quad . \tag{6}$$

Thus the elementary paraunitary matrix $\underline{\mathbf{E}}^{(i)}(z)$ for the i-th iteration can be expressed as

$$\underline{\mathbf{E}}^{(i)}(z) = \mathbf{Q}^{(i)}\underline{\mathbf{P}}^{(i)}(z) \quad . \tag{7}$$

The algorithm continues its iterations until all the off-diagonal elements are below a given threshold, with a smaller threshold giving greater accuracy. Assuming that the algorithm has converged at the N-th iteration, the diagonalized para-Hermitian matrix in (2) takes the form of

$$\underline{\mathbf{D}}(z) = \text{diag}\{\underline{d}_1(z)\ \underline{d}_2(z)\ \cdots\ \underline{d}_M(z)\} \quad , \tag{8}$$

and the generated paraunitary polynomial matrix is given by

$$\underline{\mathbf{H}}(z) = \prod_{i=1}^{N} \underline{\mathbf{E}}^{(i)}(z) = \underline{\mathbf{E}}^{(N)}(z) \cdots \underline{\mathbf{E}}^{(2)}(z)\underline{\mathbf{E}}^{(1)}(z) \quad . \tag{9}$$

2.2. MS-SBR2 Algorithm

The MS-SBR2 algorithm [3] is an improved version of the SBR2 algorithm in terms of the convergence speed. It adopts the faster convergence property from the multiple shift maximum element SMD (MSME-SMD) algorithm [5] while still preserving the benefit of lower computational cost from the SBR2 algorithm. It uses a different search strategy of the off-diagonal elements which is akin to that of the MSME-SMD algorithm, so that it can achieve the diagonalization with less iterations than the SBR2 algorithm.

For the i-th iteration, the MS-SBR2 algorithm involves multiple shifts operations $\hat{\underline{\mathbf{P}}}^{(i)}(z)$, followed by a sequence of Jacobi transformations $\hat{\mathbf{Q}}^{(i)}$. Therefore the resulting para-Hermitian matrix is computed by

$$\underline{\mathbf{R}}^{(i)}(z) = \hat{\mathbf{Q}}^{(i)}\hat{\underline{\mathbf{P}}}^{(i)}(z)\underline{\mathbf{R}}^{(i-1)}(z)\widetilde{\hat{\underline{\mathbf{P}}}}^{(i)}(z)\hat{\mathbf{Q}}^{\mathrm{H}(i)} \quad , \tag{10}$$

where $\hat{\underline{\mathbf{P}}}^{(i)}(z) = \prod_{l=1}^{L^{(i)}} \underline{\mathbf{P}}^{(l,i)}(z)$, $\hat{\mathbf{Q}}^{(i)} = \prod_{l=1}^{L^{(i)}} \mathbf{Q}^{(l,i)}$ and $L^{(i)}$ denotes the total number of off-diagonal elements shifted onto the zero-lag at the i-th iteration ($L^{(i)} \in \mathbb{Z},\ 1 \leq L^{(i)} \leq \lfloor M/2 \rfloor$).

Accordingly the delay matrix at the l-th delay stage within i-th iteration is represented by

$$\underline{\mathbf{P}}^{(l,i)}(z) = \text{diag}\{\underbrace{1 \cdots 1}_{k^{(l,i)}-1}\ z^{-\tau^{(l,i)}}\ \underbrace{1 \cdots 1}_{M-k^{(l,i)}}\} \quad , \tag{11}$$

and the elementary paraunitary matrix can be expressed as $\hat{\underline{\mathbf{E}}}^{(i)}(z) = \hat{\mathbf{Q}}^{(i)}\hat{\underline{\mathbf{P}}}^{(i)}(z)$. Note that when $L^{(i)} = 1$, the MS-SBR2 algorithm is identical to the SBR2 algorithm. For further details of the algorithm, including numerical examples and proof of convergence, see [3].

3. POLYNOMIAL ORDER GROWTH CONTROL

The idea of controlling order growth of polynomial matrices for MS-SBR2 is implemented by using the direction-fixed delay strategy. This process also involves removing zero-filled outer matrices after each iteration. Thus the whole scheme is entitled order-controlled MS-SBR2 (OC-MS-SBR2) algorithm.

For each delay stage in MS-SBR2, the conventional delay strategy [3] operates by shifting the $k^{(l,i)}$-th row of $\underline{\mathbf{R}}^{(l,i)}(z)$ towards either positive ($\tau^{(l,i)} > 0$) or negative ($\tau^{(l,i)} < 0$) lag direction, and so the $k^{(l,i)}$-th column to the opposite direction. In some cases, the directions of row (column) shifts at different delay stages within one iteration might be different, which will result in some non-zero elements to be shifted further away from zero-lag plane and cause the unnecessary order growth of polynomial matrices.

However, the new delay strategy which constrains all the rows (columns) moving in the same direction can guarantee no interference between the subsequent delay stages. This helps to keep the non-zero elements near the zero-lag plane.

In other words, it maximises the number of zero-filled lags which can be easily eliminated without loss of any energy or affecting the accuracy of algorithm. In the context of this paper, the direction of all the row shifts is confined towards the positive time lag, while the direction of all the column shifts towards the negative time lag. The summary of the direction-fixed delay strategy is shown in Tab. 1.

Table 1. Direction-Fixed Delay Strategy for the i-th Iteration in the OC-MS-SBR2 Algorithm

1. Input parameters:
   $\underline{\mathbf{R}}^{(i-1)}(z)$, $\underline{\mathbf{H}}^{(i-1)}(z)$, $\{j^{(l,i)}, k^{(l,i)}, \tau^{(l,i)}\}$.
2. Initialization:
   $\underline{\mathbf{R}}^{(1,i)}(z) \leftarrow \underline{\mathbf{R}}^{(i-1)}(z)$, $\underline{\mathbf{H}}^{(1,i)}(z) \leftarrow \underline{\mathbf{H}}^{(i-1)}(z)$.
3. for $l = 1 : L^{(i)}$
4. if $\tau^{(l,i)} > 0$
5. Shift the $k^{(l,i)}$-th ***row*** of $\underline{\mathbf{R}}^{(l,i)}(z)$ and $\underline{\mathbf{H}}^{(l,i)}(z)$ by $|\tau^{(l,i)}|$ lags towards the ***positive*** lag direction;
6. Shift the $k^{(l,i)}$-th ***column*** of $\underline{\mathbf{R}}^{(l,i)}(z)$ by $|\tau^{(l,i)}|$ lags towards the ***negative*** lag direction.
7. elseif $\tau^{(l,i)} < 0$
8. Shift the $j^{(l,i)}$-th ***row*** of $\underline{\mathbf{R}}^{(l,i)}(z)$ and $\underline{\mathbf{H}}^{(l,i)}(z)$ by $|\tau^{(l,i)}|$ lags towards the ***positive*** lag direction;
9. Shift the $j^{(l,i)}$-th ***column*** of $\underline{\mathbf{R}}^{(l,i)}(z)$ by $|\tau^{(l,i)}|$ lags towards the ***negative*** lag direction.
10. else
11. $\underline{\mathbf{R}}^{(l,i)}(z) \leftarrow \underline{\mathbf{R}}^{(l,i)}(z)$, $\underline{\mathbf{H}}^{(l,i)}(z) \leftarrow \underline{\mathbf{H}}^{(l,i)}(z)$.
12. end
13. end
14. $\underline{\mathbf{R}}'^{(i)}(z) \leftarrow \underline{\mathbf{R}}^{(L^{(i)},i)}(z)$, $\underline{\mathbf{H}}'^{(i)}(z) \leftarrow \underline{\mathbf{H}}^{(L^{(i)},i)}(z)$.

Assuming no order truncation scheme is applied when computing the PEVD via OC-MS-SBR2, the order growth on $\underline{\mathbf{R}}^{(i)}(z)$ and $\underline{\mathbf{H}}^{(i)}(z)$ are now bounded by the maximum modulus of the delays $|\tau^{(l_{\max},i)}|$, whereby

$$l_{\max} = \arg \max_{l} |\tau^{(l,i)}|, \; \forall \, l = 1 \cdots L^{(i)} \quad , \tag{12}$$

and $|\tau^{(l,i)}|$ denotes the modulus of the delay needed for bringing the maximum element onto the zero-lag at the l-th delay stage within i-th iteration. With zero-filled outer matrices being removed, the resulting polynomial orders can be estimated as

$$\begin{aligned} \mathcal{O}_{\mathrm{R}}^{(N)} &= \mathcal{O}_{\mathrm{R}}^{(0)} + 2\sum_{i=1}^{N} |\tau^{(l_{\max},i)}| \quad , \\ \mathcal{O}_{\mathrm{H}}^{(N)} &= 1 + \sum_{i=1}^{N} |\tau^{(l_{\max},i)}| \quad , \end{aligned} \tag{13}$$

where $\mathcal{O}_{\mathrm{R}}^{(N)}$ denotes the order of para-Hermitian matrix $\underline{\mathbf{R}}^{(N)}(z)$ at the N-th iteration with the initial order value of $\mathcal{O}_{\mathrm{R}}^{(0)}$, and $\mathcal{O}_{\mathrm{H}}^{(N)}$ denotes the order of the paraunitary matrix $\underline{\mathbf{H}}^{(N)}(z)$. Bear in mind that the benefit of using the direction-fixed delay strategy can only be reflected when $L^{(i)} \geq 2, \exists \, i = 1 \cdots N$, meaning that in OC-MS-SBR2 there exists at least one iteration at which two or more shift steps arise. For example, the best scenario for $\underline{\mathbf{R}}(z)$ with dimension of 6×6 is that there are 3 off-diagonal elements to be shifted and rotated at each iteration.

Due to the manner in which PEVD algorithms operate, the resulting para-Hermitian matrix $\underline{\mathbf{R}}^{(i)}(z)$ at each iteration is usually with highly sparse outer coefficient matrices which generally accounts for a small proportion of the total energy of $\underline{\mathbf{R}}(z)$. To truncate the negligibly small amount of energy and also to reduce the computational complexity, the para-Hermitian [11] and paraunitary [12] truncation approaches are respectively applied to $\underline{\mathbf{R}}^{(i)}(z)$ and $\underline{\mathbf{H}}^{(i)}(z)$ with pre-defined truncation parameters μ_{PH} and μ_{PU} whose values indicate the proportion of the total energy of $\underline{\mathbf{R}}(z)$ and $\underline{\mathbf{H}}(z)$ to be truncated. Further details about how the OC-MS-SBR2 algorithm performs after introducing the truncation schemes will be presented in the next section.

4. RESULTS

To examine the performance of the different PEVD algorithms, the performance metrics are firstly defined, followed by the description of the simulation scenario and results of comparison.

4.1. Performance Metrics

To confirm that the OC-MS-SBR2 algorithm shifts a similar amount of energy at each iteration as the conventional MS-SBR2 algorithm, the first test is to measure the diagonalization performance, i.e., the remaining off-diagonal energy after i iterations normalized by the energy of the input para-Hermitian matrix $\underline{\mathbf{R}}(z)$,

$$\eta^{(i)} = \frac{\sum_{\tau} \sum_{m,n,m\neq n}^{M} |r_{mn}^{(i)}[\tau]|^2}{\sum_{\tau} \|\mathbf{R}[\tau]\|_{\mathrm{F}}^2} \quad , \tag{14}$$

where the notation $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm.

The paraunitary property, i.e., $\underline{\mathbf{H}}^{(i)}(z)\widetilde{\underline{\mathbf{H}}}^{(i)}(z) = \mathbf{I}_{M\times M}$, is lost after applying the truncation, therefore the difference from paraunitary is given by

$$\underline{\mathbf{\Phi}}^{(i)}(z) = \mathbf{I}_{M\times M} - \underline{\mathbf{H}}_{\mathrm{T}}^{(i)}(z)\widetilde{\underline{\mathbf{H}}}_{\mathrm{T}}^{(i)}(z) \quad , \tag{15}$$

where $\underline{\mathbf{H}}_{\mathrm{T}}^{(i)}(z)$ denotes the truncated matrix. Thus the loss of the paraunitarity can be measured as

$$\xi^{(i)} = \frac{1}{M} \sum_{\tau} \|\mathbf{\Phi}^{(i)}[\tau]\|_{\mathrm{F}}^2 \quad . \tag{16}$$

4.2. Simulation Scenario

The PEVD algorithms are run by using Monte Carlo simulations over an ensemble of 2000 different random 6×6 para-

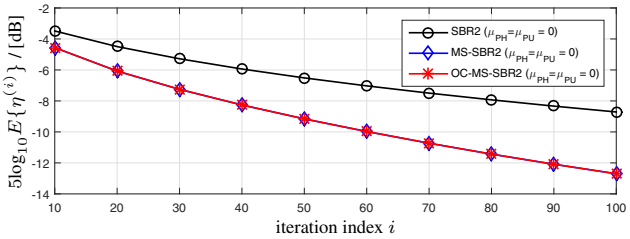

Fig. 1. Comparison of the convergence speed among different versions of SBR2, showing the ensemble averages of normalized off-diagonal energy $\eta^{(i)}$ versus iterations.

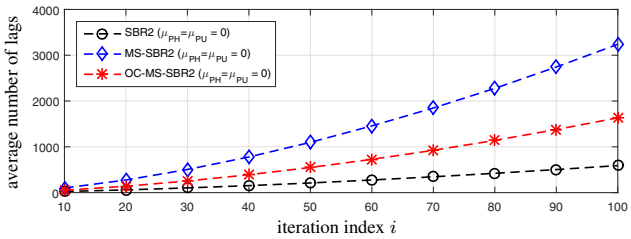

Fig. 2. Comparison of the average number of lags of $\underline{\mathbf{R}}^{(i)}(z)$ among different versions of SBR2. Note that the number of lags is not equivalent to the number of orders.

Hermitian matrices $\underline{\mathbf{R}}(z)$, which can be generated from matrices $\underline{\mathbf{A}}(z) \in \underline{\mathbb{C}}^{6\times 6}$ of order 3 with i.i.d. zero mean unit variance complex Gaussian entries, such that $\underline{\mathbf{R}}(z) = \underline{\mathbf{A}}(z)\widetilde{\underline{\mathbf{A}}}(z)$. Each of the PEVD algorithms was run for 100 iterations with the performance metrics recorded after every 10 iterations. The simulations was firstly set up with the para-Hermitian and paraunitary truncation parameters $\mu_{\text{PH}} = \mu_{\text{PU}} = 0$, i.e., no truncation scheme is applied, then repeated over the same ensemble for $\mu_{\text{PH}} = 10^{-4}$ and $\mu_{\text{PU}} = 10^{-3}$.

4.3. Algorithm Convergence & Polynomial Order

As shown in Fig. 1, both versions of MS-SBR2 algorithm require much fewer iterations than the SBR2 algorithm to achieve the same level of diagonalization. However, it should be noticed that each iteration within MS-SBR2 involves more rotation steps, which means the computational costs among them are comparable. Also, the elimination of zero-valued coefficient matrices in MS-SBR2 seems no impact on the algorithm convergence.

Without using the truncation schemes, Fig. 2 presents the results of the average number of lags versus iterations among different PEVD algorithms, and it shows almost half amount of lag reductions achieved in $\underline{\mathbf{R}}^{(i)}(z)$ for the OC-MS-SBR2 algorithm. After applying the truncations with $\mu_{\text{PH}} = 10^{-4}$ and $\mu_{\text{PU}} = 10^{-3}$, the average polynomial orders of $\underline{\mathbf{R}}^{(i)}(z)$ and $\underline{\mathbf{H}}^{(i)}(z)$ versus iterations are respectively depicted in Fig. 3 and 4. The benefit in terms of order reduction from the OC-MS-SBR2 algorithm is reduced when non-zero values are truncated. Fig. 5 shows the reconstruction error of paraunitarity for different PEVD methods. Initially the error curves start very low but they quickly increase as the truncation algorithms begin to remove the proportion of energy. In particular, both versions of MS-SBR2 algorithms have shown very similar reconstruction error throughout iterations. Again it proves that the introduction of the order growth control scheme for MS-SBR2 does not affect the convergence of the algorithm.

5. CONCLUSION

We have proposed an order-controlled version of MS-SBR2 algorithm for calculating the PEVD. The OC-MS-SBR2 algorithm uses a direction fixed delay strategy which can limit the polynomial order growth by selecting shifts that ensure non-zero coefficients are kept closer to the zero-lag plane. It preserves the similar algorithm convergence property and same level of computational cost as the conventional MS-SBR2 algorithm. Simulation results have suggested that further order reductions can be achieved after introducing the order truncation process to the OC-MS-SBR2 algorithm.

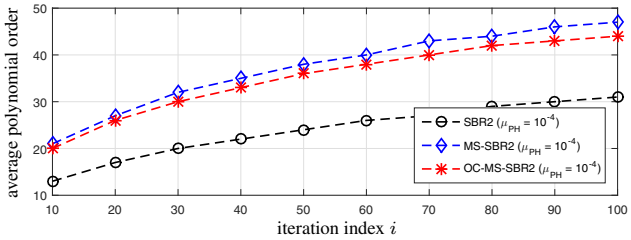

Fig. 3. Average order of $\underline{\mathbf{R}}^{(i)}(z)$ after truncation [11] with $\mu_{\text{PH}} = 10^{-4}$, showing the comparison among different versions of SBR2.

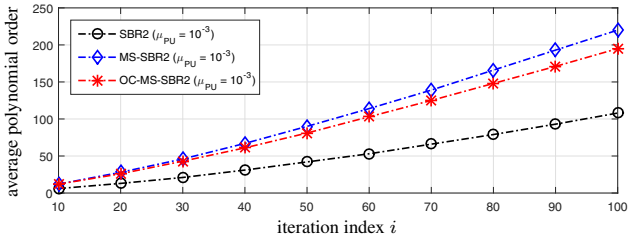

Fig. 4. Average order of $\underline{\mathbf{H}}^{(i)}(z)$ after truncation [12] with $\mu_{\text{PU}} = 10^{-3}$, showing the comparison among different versions of SBR2.

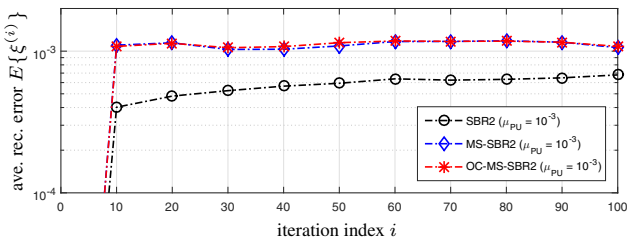

Fig. 5. Average reconstruction error $E\{\xi^{(i)}\}$ versus iterations, showing the comparison among different versions of SBR2 for $\mu_{\text{PU}} = 10^{-3}$.

6. REFERENCES

[1] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, 1993.

[2] J.G. McWhirter, P.D. Baxter, T. Cooper, S. Redif, J. Foster, "An EVD Algorithm for Para-Hermitian Polynomial Matrices," *IEEE Transactions on Signal Processing*, 55(5):2158–2169, May 2007.

[3] Z. Wang, J.G. McWhirter, J. Corr, S. Weiss, "Multiple Shift Second Order Sequential Best Rotation Algorithm for Polynomial Matrix EVD," in *European Signal Processing Conference*, pp. 844–848, Nice, France, Aug. 2015.

[4] S. Redif, S. Weiss, J.G. McWhirter, "Sequential Matrix Diagonalization Algorithms for Polynomial EVD of Parahermitian Matrices," *IEEE Transactions on Signal Processing*, 63(1):81–89, Jan. 2015.

[5] J. Corr, K. Thompson, S. Weiss, J.G. McWhirter, S. Redif, I.K. Proudler, "Multiple Shift Maximum Element Sequential Matrix Diagonalisation for Parahermitian Matrices," in *IEEE Workshop on Statistical Signal Processing*, pp. 312–315, Gold Coast, Australia, Jun. 2014.

[6] J. Corr, K. Thompson, S. Weiss, J.G. McWhirter, I.K. Proudler, "Causality-constrained Multiple Shift Sequential Matrix Diagonalisation for Parahermitian Matrices," in *European Signal Processing Conference*, pp. 1277–1281, Lisbon, Portugal, Sep. 2014.

[7] S. Redif, J.G. McWhirter, S. Weiss, "Design of FIR Paraunitary Filter Banks for Subband Coding Using a Polynomial Eigenvalue Decomposition," *IEEE Transactions on Signal Processing*, 59(11):5253–5264, Nov. 2011.

[8] C.H. Ta, S. Weiss, "A Design of Precoding and Equalisation for Broadband MIMO Systems," in *Asilomar Conference on Signals, Systems and Computers*, pp. 1616–1620, CA, USA, Nov. 2007.

[9] P. Comon, L. Rota, "Blind Separation of Independent Sources from Convolutive Mixtures," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 86(3):542–549, Mar. 2003.

[10] Z. Wang, J.G. McWhirter, S. Weiss, "Multichannel Spectral Factorization Algorithm using Polynomial Matrix Eigenvalue Decomposition," in *Asilomar Conference on Signals, Systems and Computers*, pp. 1714–1718, CA, USA, Nov. 2015.

[11] J. Foster, J.G. McWhirter, J. Chambers, "Limiting the Order of Polynomial Matrices Within the SBR2 Algorithm," in *IMA International Conference on Mathematics in Signal Processing*, Cirencester, UK, Dec. 2006.

[12] C.H. Ta, S. Weiss, "Shortening the Order of Paraunitary Matrices in SBR2 Algorithm," in *International Conference on Information, Communications & Signal Processing*, pp. 1–5, Singapore, 2007.

[13] J. Corr, K. Thompson, S. Weiss, I.K. Proudler, J.G. McWhirter, "Row-shift Corrected Truncation of Paraunitary Matrices for PEVD Algorithms," in *European Signal Processing Conference*, pp. 849–853, Nice, France, Aug. 2015.