Strathprints Institutional Repository

Ta, C.H. and Weiss, S. (2007) *Shortening the order of paraunitary matrices in SBR2 algorithm.* In: 6th Conference on Information, Communications, and Signal Processing, 2007-12-10 - 2007-12-13.

Shortening the Order of Paraunitary Matrices in SBR2 Algorithm

Chi Hieu Ta and Stephan Weiss
Institute for Communications & Signal Processing
Dept. of Electronic & Electrical Engineering
University of Strathclyde, Glasgow, United Kingdom
{chi.ta, stephan.weiss}@eee.strath.ac.uk

***Abstract*— The second order sequential best rotation (SBR2) algorithm has recently been proposed as a very effective tool in decomposing a para-Hermitian polynomial matrix $\mathbf{R}(z)$ into a diagonal polynomial matrix $\mathbf{\Gamma}(z)$ and a paraunitary matrix $\mathbf{B}(z)$, extending the eigenvalue decomposition to polynomial matrices, $\mathbf{R}(z) = \mathbf{B}(z)\mathbf{\Gamma}(z)\tilde{\mathbf{B}}(z)$. However, the algorithm results in polynomials of very high order, which limits its applicability. Therefore, in this paper we evaluate approaches to reduce the order of the paraunitary matrices, either within each step of SBR2, or after convergence. The paraunitary matrix $\mathbf{B}(z)$ is replaced by a near-paraunitary quantity $\mathbf{B}_N(z)$, whose error will be assessed. Simulation results show that the proposed truncation can greatly reduce the polynomial order while retaining good near-paraunitariness of $\mathbf{B}_N(z)$.**

I. Introduction

The second order sequential best rotation (SBR2) algorithm was proposed by McWhirter et al. in [1], [2]. It is an iterative algorithm which aims to decompose a polynomial para-Hermitian matrix $\mathbf{R}(z) \in \mathbb{C}^{P\times P}(z)$ into a diagonal polynomial matrix $\mathbf{\Gamma}(z)$ and a polynomial paraunitary matrix $\mathbf{B}(z)$, such that $\mathbf{R}(z) = \mathbf{B}(z)\mathbf{\Gamma}(z)\tilde{\mathbf{B}}(z)$ can be regarded as a generalisation of the standard eigenvalue decomposition to the broadband case [2]. Different from the conventional methods which often perform the diagonalisation of polynomial matrices in the frequency domain, the SBR2 algorithm diagonalises para-Hermitian matrices in the time domain by applying a sequence of suitably chosen delays and Givens rotations.

The algorithm has been demonstrated to be very robust in diagonalising $\mathbf{R}(z)$ such that ideally

$$\mathbf{\Gamma}(z) = \mathrm{diag}\{\Gamma_0(z), \Gamma_1(z), \cdots \Gamma_{P-1}(z)\} \tag{1}$$

with on-diagonal elements $\Gamma_i(z)$, which are spectrally majorised, i.e. the power spectral densities fulfill

$$\Gamma_i(e^{j\Omega}) \geq \Gamma_{i+1}(e^{j\Omega}) \quad \forall\Omega \quad . \tag{2}$$

The SBR2 algorithm has been applied to a number of problems, such as source separation of array data [2], subband-based source coding [4], subspace-based channel coding in the presence of structured noise [6], or precoding and equalisation for MIMO systems [8]. However, the iterative SBR2 algorithm often results in matrices of rather high polynomial order, which for many applications results filter banks of considerable delay and computational complexity.

In [3], [2], the authors address the truncation of the para-Hermitian matrix during the diagonalisation process and show that with a small loss in Frobenius norm, the order of the para-Hermitian matrix can be significantly reduced. The truncation of the extracted paraunitary matrix $\mathbf{B}(z)$ is not considered there, although the matrix is utilised for a number of applications [6], [8]. The order of $\mathbf{B}(z)$ can be prohibitively high, although generally the coefficient matrices corresponding to high and low powers of z, which will be reffered to as outer coefficients matrices, in $\mathbf{B}(z)$ can be observed to tail off to very small values.

In this paper we proposed a method to shorten the order of the paraunitary matrix $\mathbf{B}(z)$. This method will discard the outer coefficient matrices which have the smallest Frobenius norm, thus replacing the paraunitary matrix $\mathbf{B}(z)$ with $\mathbf{B}(z)\tilde{\mathbf{B}}(z) = \mathbf{I}$ by a near-paraunitary matrix $\mathbf{B}_N(z)$ such that $\mathbf{B}_N(z)\tilde{\mathbf{B}}_N(z) \approx \mathbf{I}$. We aim to keep the loss in paraunitariness bounded, with simulation results showing that the order of $\mathbf{B}_N(z)$ can be significantly reduced even with a low error threshold.

The paper is organised as follow. In Sec. II, a brief description of SBR2 algorithm will be laid out. Sec. III addresses the proposed method for shortening the order of paraunitary matrices. Simulation results are given in Sec. IV and conclusions are drawn in Sec. V.

In our notation, we use lower- and uppercase boldface fonts for vector and matrix quantities, respectively. The operator $\tilde{\{\cdot\}}$ denotes the parahermitian transpose, e.g. $\tilde{\mathbf{A}}(z) = \mathbf{A}^{\mathrm{H}}(z^{-1})$. The operator $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.

II. SBR2 Algorithm [2]

The SBR2 algorithm is an iterative broadband eigenvalue decomposition technique, which in each step eliminates the largest off-diagonal element at a specific lag value of $\mathbf{R}[\tau] \circ\!\!-\!\!\bullet\, \mathbf{R}(z)$,

$$\mathbf{R}(z) = \sum_{\tau=-L}^{L} z^{-\tau}\mathbf{R}[\tau] \; . \tag{3}$$

by means of an elementary paraunitary operation. After a sufficient amount of iterations, the algorithm converges to a diagonalised version of the para-Hermitian matrix $\mathbf{R}(z)$. Note that since $\mathbf{R}(z)$ is a para-Hermitian polynomial matrix, the elements of its coefficient matrices $\mathbf{R}[\tau]$ satisfy

$$r_{ij}[\tau] = r^*_{ji}[-\tau] \quad . \tag{4}$$

1–4244–0983–7/07/$25.00

The algorithm commences its operation on the original parahermitian matrix, $\mathbf{\Gamma}^{(0)}(z) = \mathbf{R}(z)$ with $\mathbf{B}^{(0)}(z) = \mathbf{I}$, while at the ith iteration the gradually diagonalised matrix is denoted $\mathbf{\Gamma}^{(i)}(z)$ with coefficient $r_{kl}^{(i)}[\tau]$.

At the ith iteration, the algorithm finds the two largest off-diagonal polynomial coefficients — suppose $r_{kl}^{(i-1)}[T] \in \mathbf{\Gamma}^{(i-1)}[T]$ and according to (4) $r_{lk}^{(i-1)}[-T] \in \mathbf{\Gamma}^{(i-1)}[-T]$. A delay of T can be applied to all kth rows of $\mathbf{\Gamma}^{(i-1)}(z)$ by means of

$$\mathbf{\Lambda}_i(z) = \mathbf{I}_{P\times P} - \mathbf{v}_i\mathbf{v}_i^{\mathrm{H}} + z^{-T}\mathbf{v}_i\mathbf{v}_i^{\mathrm{H}} \tag{5}$$

with $\mathbf{v}_i = [0\ \cdots 0\ 1\ 0\ \cdots\ 0]^T$ containing zeros except for a unit element in the kth position. Thus $\mathbf{\Lambda}_i(z)$ is an identity matrix with the kth diagonal element replaced by a delay z^{-T}. In conjunction with advanding all kth columns of $\mathbf{\Gamma}^{(i-1)}(z)$, the maximum off-diagonal elements at lags T and $-T$ are now brought to lag zero of $\mathbf{\Lambda}_i(z)\mathbf{\Gamma}^{(i-1)}(z)\tilde{\mathbf{\Lambda}}_i(z)$.

In the same ith step, the above mentioned maximum elements now in lag zero are eliminated by a Givens rotation $\mathbf{Q}_i$, which is an identity matrix with elements at the intersections of the kth and lth rows and the kth and lth columns given by

$$\begin{bmatrix} q_{ll}^{(i)} & q_{lk}^{(i)} \\ q_{kl}^{(i)} & q_{kk}^{(i)} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta)e^{i\phi} \\ -\sin(\theta)e^{-i\phi} & \cos(\theta) \end{bmatrix} \tag{6}$$

where

$$\phi = \arg\{r_{lk}^{(i-1)}[-T]\} \tag{7}$$

$$\theta = \frac{1}{2}\arctan\left\{\frac{2|r_{lk}^{(i-1)}[-T]|}{r_{ll}^{(i-1)}[0] - r_{kk}^{(i-1)}[0]}\right\} \quad . \tag{8}$$

Therefore, the overall operation in the ith step is characterised by

$$\begin{aligned} \mathbf{\Gamma}^{(i)}(z) &= \mathbf{Q}_i\mathbf{\Lambda}_i(z)\mathbf{\Gamma}^{(i-1)}(z)\tilde{\mathbf{\Lambda}}_i(z)\mathbf{Q}_i^{\mathrm{H}} && (9)\\ &= \tilde{\mathbf{B}}^{(i)}(z)\mathbf{R}(z)\mathbf{B}^{(i)}(z) && (10)\\ \tilde{\mathbf{B}}^{(i)}(z) &= \mathbf{Q}_i\mathbf{\Lambda}_i(z)\tilde{\mathbf{B}}^{(i-1)}(z) = \prod_{j=1}^{i}\mathbf{Q}_j\mathbf{\Lambda}_j(z) \quad . && (11) \end{aligned}$$

Through the appropriate choice of θ, diagonalisation of the given para-Hermitian matrix $\mathbf{R}(z)$ can be achieved, while the parameter ϕ ensures spectral majorisation. The algorithm is stopped either after reaching a certain measure for suppressing off-diagonal terms or after exceeding a specified number of iterations [1], [2].

After N iterations, the original matrix is decomposed as

$$\mathbf{R}(z) = \mathbf{B}^{(N)}(z)\mathbf{\Gamma}^{(N)}(z)\tilde{\mathbf{B}}^{(N)}(z) \tag{12}$$

where $\mathbf{\Gamma}^{(N)}(z)$ is an approximately diagonalised polynomial matrix and $\mathbf{B}^{(N)}(z)$ is the paraunitary matrix given by

$$\mathbf{B}^{(N)}(z) = \prod_{i=1}^{N}\mathbf{Q}_i\mathbf{\Lambda}_i(z) \quad . \tag{13}$$

Example. Fig. 1 characterises a para-Hermitian matrix $\mathbf{R}(z) \in \mathbb{C}^{4\times4}(z)$ of order 20. This matrix may for example emerge from a signal vector $\mathbf{x}[n] \in \mathbb{C}^4$ obtained from a 4-element sensor array, by correlating $\mathbf{R}(z) \bullet\!\!-\!\!\circ\ \mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^{\mathrm{H}}[n-\tau]\}$, whereby $\mathcal{E}\{\cdot\}$ is the expectation operator. In this case, the responses on the main diagonal of Fig. 1 are the auto-correlation sequences of each sensor signal, while off-diagonal responses are the cross-correlation sequences between the various sensors. Fig. 1 shows the modulus of each — potentially complex valued — element in the para-Hermitian matrix.

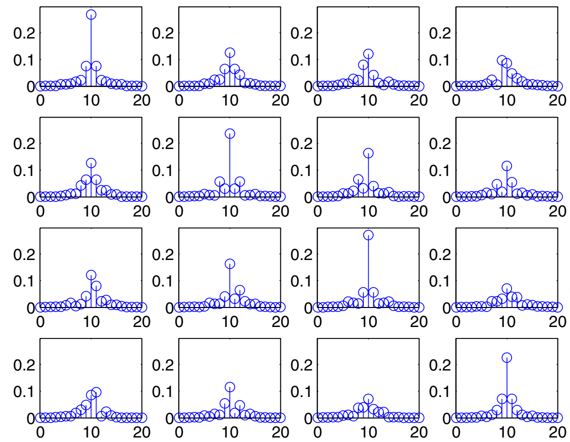

Fig. 1. Coefficients of a para-Hermitian matrix $\mathbf{R}(z) \in \mathbb{C}^{4\times4}(z)$.

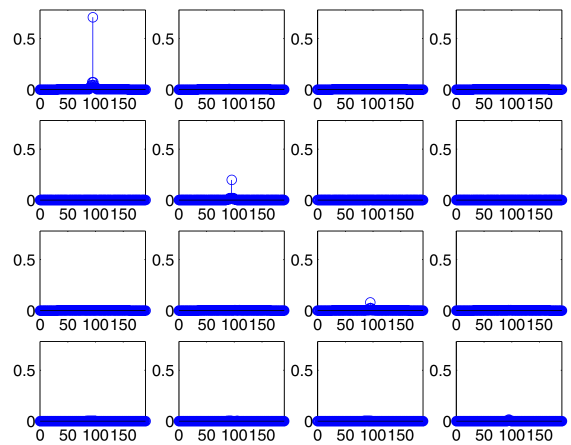

Fig. 2. Coefficients of the para-Hermitian matrix $\mathbf{\Gamma}^{(N)}(z)$ after applying SBR2.

Fig. 2 shows the modulus of the elements of the resulting $\mathbf{\Gamma}^{(N)}(z) = \tilde{\mathbf{B}}^{(N)}(z)\mathbf{R}(z)\mathbf{B}^{(N)}(z)$ obtained by applying SBR2 as described above. It is evident that off-diagonal elements have been eliminated, and only on-diagonals remain. Note however that the order of the system $\mathbf{\Gamma}^{(N)}(z)$ has significantly increased as compared to $\mathbf{R}(z)$ in Fig. 1.

The effect of spectral majorisation is highlighted in Fig. 3, where the power spectral densities along the diagonal of $\mathbf{\Gamma}^{(N)}$ are shown. As we can see, these frequency responses are ordered in descending values, which shows that the spectral majorisation according to (2) has been achieved along with

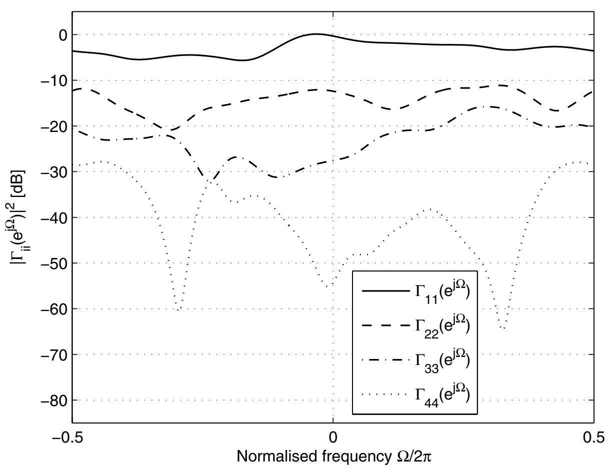

Fig. 3. Power spectral densities $\Gamma_i^{(N)}(e^{j\Omega})$ along the main diagonal of $\mathbf{\Gamma}^{(N)}(z)$ in Fig. 2.

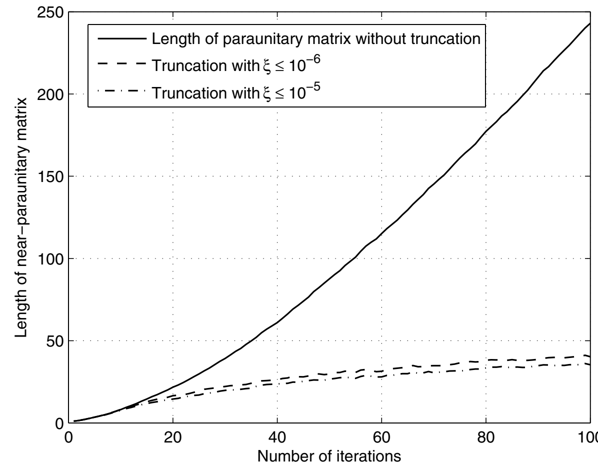

Fig. 4. Length of the paraunitary matrix with and without truncation.

the diagonalisation of $\mathbf{R}(z)$.

Due to the delays applied in the diagonalisation process, the order of the polynomial matrix $\mathbf{\Gamma}^{(N)}(z)$ as well as the order of paraunitary matrix $\mathbf{B}^{(N)}(z)$ grow as the number of iterations increases [1], [3]. This growth requires large memory to accomodate a record of both the para-Hermitian $\mathbf{\Gamma}^{(i)}(z)$ and the paraunitary $\mathbf{B}^{(i)}(z)$ within SBR2. Also, the computational complexity to perform one iteration step of SBR2 increases with the iteration number. In order to avoid this, [3] have proposed to discard the outer coefficient matrices of $\mathbf{\Gamma}^{(i)}(z)$ while allowing an acceptable small loss in its Frobenius norm. This approach can help to significantly reduce the order of para-Hermitian matrix, while the order of the paraunitary matrix remains unaltered. Note that despite of truncation, $\mathbf{\Gamma}^{(i)}(z)$ will retain its parahermitian property. However, the order of the paraunitary matrix $\mathbf{B}^{(N)}(z)$ remains very high, which affects the speed of SBR2 and is likely to inhibit the application of $\mathbf{B}^{(N)}(z)$ in practical senarios where long paraunitary filter banks with large delay and complexity are undesired. Therefore problem of lowering the order of the paraunitary matrix $\mathbf{B}^{(N)}(z)$ will be treated in the next section.

III. Truncation of Paraunitary Matrix

The truncation $\mathbf{B}^{(i)}(z)$ to a matrix $\mathbf{B}_{\mathrm{T}}^{(i)}(z)$ of lower order in the course of SBR2 leads to a loss of paraunitarity. We therefore refer to $\mathbf{B}_{\mathrm{T}}^{(i)}(z)$ as a *near-paraunitary* matrix, provided that the deviation from a paraunitary matrix can be kept small. In order to quantify how close $\mathbf{B}_{\mathrm{T}}^{(i)}(z)$ is to a paraunitary matrix, we therefore define a cost function

$$\xi = \frac{1}{P}\sum_{\tau=-\infty}^{\infty} \|\mathbf{Q}[\tau]\|_F^2 \tag{14}$$

where $\mathbf{Q}[\tau] \circ\!\!-\!\!\bullet\, \mathbf{Q}(z) = \mathbf{I}_{P\times P} - \mathbf{B}_{\mathrm{T}}^{(i)}(z)\tilde{\mathbf{B}}_{\mathrm{T}}^{(i)}(z)$. In the case of no truncation, $\mathbf{B}_{\mathrm{T}}^{(i)}(z) = \mathbf{B}_{N}^{(i)}(z)$ and ξ equals to zero. Truncation of $\mathbf{B}^{(i)}(z)$ will lead to a positive error. The truncation of $\mathbf{B}_{(i)}(z)$, either during each SBR2 step or after convergence, can be performed under the condition

$$\xi \leq \epsilon \quad . \tag{15}$$

A second measure evaluates by how much the matrix $\mathbf{R}^{'}(z) = \tilde{\mathbf{B}}_{\mathrm{T}}^{(i)}(z)\mathbf{\Gamma}^{(i)}(z)\mathbf{B}_{\mathrm{T}}^{(i)}(z)$ deviates from the original para-Hermitian matrix $\mathbf{R}(z)$ due to truncation of $\mathbf{B}^{(i)}(z)$, for which we consider the normalised error

$$\chi = \frac{\|\mathbf{R}^{'}(z) - \mathbf{R}(z)\|_F^2}{\|\mathbf{R}(z)\|_F^2} \quad . \tag{16}$$

Again, this error depends on whether $\mathbf{B}^{(i)}(z)$ is truncated only after convergence, i.e. for $i = N$, or as an ongoing operation in each step of SBR2. Unlike ξ, the error χ is not a design parameter which permits control, but a true output to assess the impact of truncation. Note that if we perform the truncation of $\mathbf{\Gamma}^{(i)}(z)$ in each step of SBR2 as proposed in [3], it causes another deviation in addition to the deviation caused by the truncation of $\mathbf{B}^{(i)}(z)$. Thus, in such case the error χ has two sources, whereby one is from the truncation of $\mathbf{\Gamma}^{(i)}(z)$ and the other is from the truncation of $\mathbf{B}^{(i)}(z)$.

IV. Simulations and Results

To highlight the advantage of the proposed truncation scheme, we consider the diagonalisation of a covariance matrix $\mathbf{R}(z)$, which is generated from a 4×4 polynomial MIMO channel matrix $\mathbf{C}(z)$ based on an indoor statistical channel model in [5]. The channel order is chosen to be 10 which leads to $\mathbf{R}(z) = \mathbf{C}(z)\tilde{\mathbf{C}}(z)$ to be of order 21. Simulation are performed over an ensemble of 50 randomly generated MIMO channels.

First consider the case when there is no truncation of $\mathbf{\Gamma}^{(i)}(z)$ in the diagonalisation process. The length of the near-paraunitary matrix as a function of the number of iterations in different cases is illustrated in Fig. 4. The error function for the truncated cases has been chosen to be $\xi \leq 10^{-6}$ and $\xi \leq 10^{-5}$. As one can see from the figure, without truncation the length of the paraunitary matrix is rather high and it can be significantly reduced whith a very small lost in

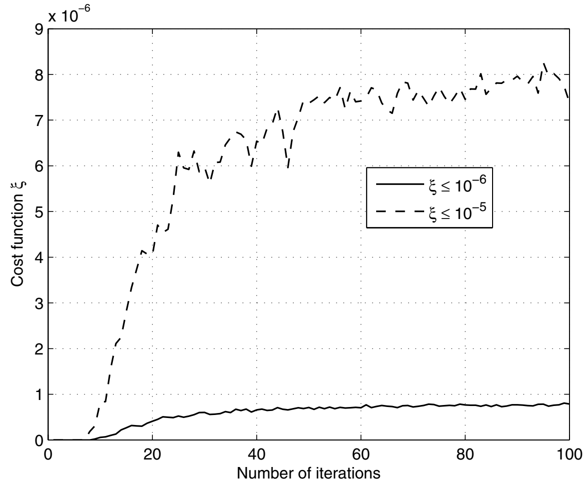

Fig. 5. Value of the cost function.

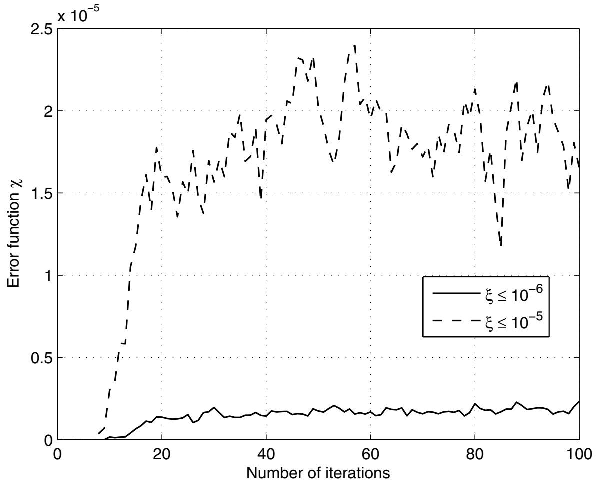

Fig. 6. Distortion caused by truncation of $\mathbf{B}_N(z)$.

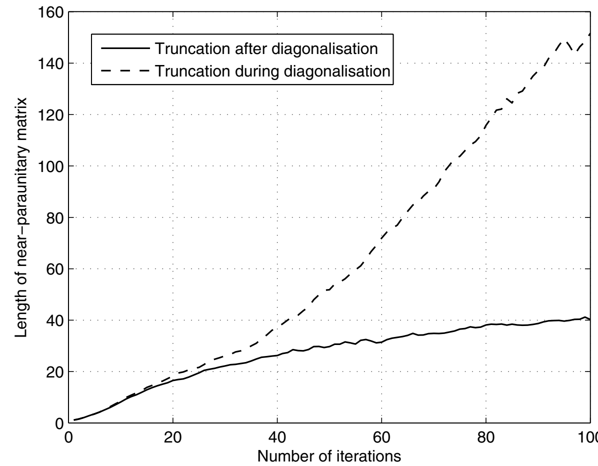

Fig. 7. Length of the near-paraunitary matrices truncated after and during diagonalisation.

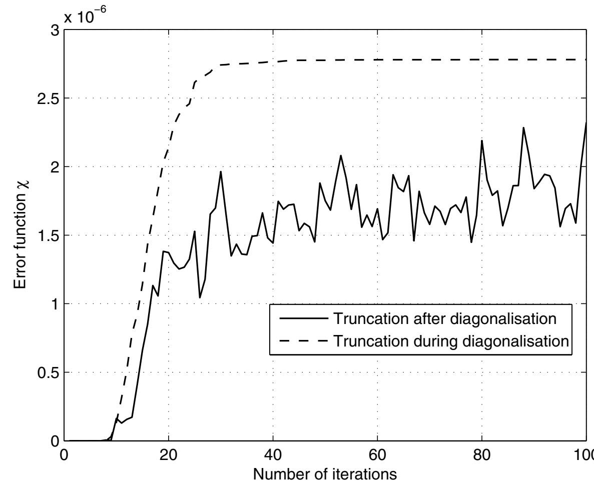

Fig. 8. Error function in two cases of truncation (before and during the diagonalisation).

the paraunitariness. Fig. 5 shows the dependence of the cost function on the number of iterations with the upper bound to be 10^{-6} and 10^{-5}. We see from the figure that the cost function is always kept under the given upper bound. The dependence of the error funtion χ on the number of iterations is illustrated in Fig. 6. From the figure, one can see that the distortion caused by the truncation of paraunitary matrix is very small. Fig. 7 and Fig. 8 show the length of near-paraunitary matrix and the error function in two cases when the truncation of paraunitary matrix is performed after the diagonalisation and during the diagonatisation. One can see from the figures that performing the truncation during the diagonalisation process results in a higher order of truncated near-paraunitary matrix. The error caused by truncation in this case is also higher but more stable than that of the case when the truncation is performed after diagonalisation process.

Next we consider the case when the truncation of $\mathbf{\Gamma}^{(i)}(z)$ is performed during diagonalisation process as proposed in [3] with a lost of 10^{-5} in Frobenius norm of $\mathbf{\Gamma}^{(i)}(z)$ after each iteration. As one can see from Fig. 9, the truncation of $\mathbf{\Gamma}^{(i)}(z)$ does not have a big effect on the length of near-paraunitary matrix, but it makes the error function increase with the number of iterations, which means the error function χ is dominated by the error component caused by the truncation of para-Hermitian matrix.

V. Conclusion

In this paper we have proposed an approach to shorten the order of the parauniraty matrix obtained from the SBR2 algorithm. The algorithm tries to reduce the order of a paraunitary matrix while still keeps the lost in its paraunitariness under a given upper bound. The simulation results show that our proposed algorithm can significantly reduce the order of the paraunitary matrix while the upper bound of the lost in paraunitariness can still be controlled and the error caused by the truncation of paraunitary matrix to the diagonalisation process is very small. Performing the truncation of paraunitary matrix after the diagonalisation process have been shown to result in a lower order of the near-paraunitary matrix than that of the case when the truncation is performed during the diagonalisation process.

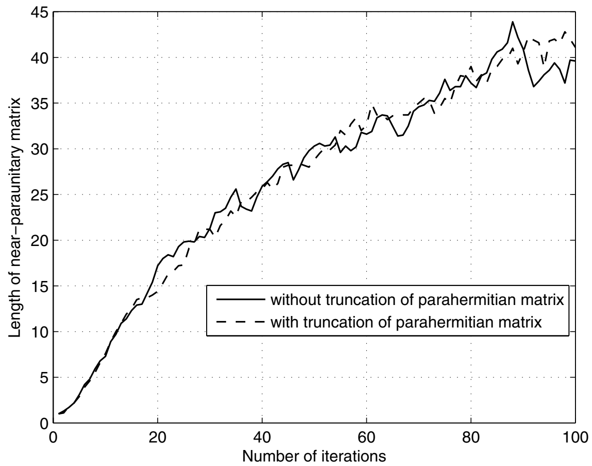

Fig. 9. Length of the near-paraunitary matrices with and without truncation of para-Hermitian matrix.

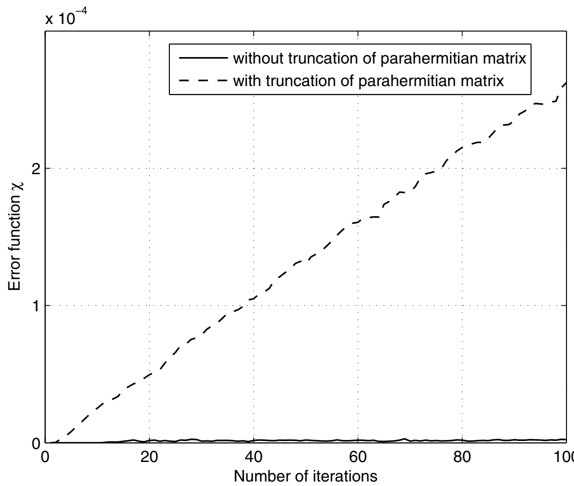

Fig. 10. Error function with and without truncation of para-Hermitian matrix.

The future works will concern the implementation issues of the algorithm in DSP hardware such as the accuracy in different number formats.

REFERENCES

[1] J. G. McWhirter and P. D. Baxter. A Novel Technqiue for Broadband SVD. In *12th Annual Workshop on Adaptive Sensor Array Processing*, MIT Lincoln Labs, Cambridge, MA, 2004.

[2] J. G. McWhirter, P. D. Baxter, T. Cooper, S. Redif, and J. Foster. An EVD Algorithm for Para-Hermitian Polynomial Matrices. *IEEE Transactions on Signal Processing*, to appear, 2007.

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

[4] S. Redif and T. Cooper. Paraunitary Filter Bank Design via a Polynomial Singular Value Decomposition. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 613–616, Philadelphia, PA, March 2005.

[5] A.A.M Saleh and R.A. Valenzuela. A Statistical Model for Indoor Multipath Propagation. *IEEE Journal on Selected Areas in Communications*, 5(2):128–137, February 1997.

[6] S. Weiss, S. Redif, T. Cooper, C. Liu, P. D. Baxter and J. G. McWhirter. Paraunitary Oversampled Filter Bank Design for Channel Coding. *EURASIP Journal on Applied Signal Processing*, 2006.

[7] S. Weiss, C. H. Ta and C. Liu. A Wiener Filter Approach to the Design of Filter Bank Based Sigle-Carrier Precoding and Equalisation. *Proc. IEEE International Symposium on Powerline Communications and Its Applications*, Pisa, Italy, March 2007.

[8] C.H. Ta and S. Weiss. A Design of Precoding and Equalisation for Broadband MIMO Systems. *Proc. 15th International Conference on Digital Signal Processing*, Cardiff, UK, July 2007.

ACKNOWLEDGMENT

The research of Chi Hieu Ta is supported by a scholarship from the Ministry of Education and Training of Vietnam.