Multiple Shift QR Decomposition for Polynomial Matrices

By Fraser K. Coutts[*], Jamie Corr[*], Keith Thompson[*], Stephan Weiss[*], Ian K. Proudler[*,†], John G. McWhirter[‡]

[*] Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, Scotland

[*,†] School of Electrical, Electronics & Systems Engineering, Loughborough Univ., Loughborough, UK

[‡] School of Engineering, Cardiff University, Cardiff, Wales, UK

Abstract

In recent years, several algorithms for the iterative calculation of a polynomial matrix QR decomposition (PQRD) have been introduced. The PQRD is a generalisation of the ordinary QRD and uses paraunitary operations to upper-triangularise a polynomial matrix. This paper addresses a multiple shift strategy that can be applied to an existing PQRD algorithm. We demonstrate that with the proposed strategy, the computation time of the algorithm can be reduced. The benefits of this are important for a number of broadband multichannel problems.

1. Introduction

Polynomial matrix representations can elegantly express broadband multichannel problems. Such formulations have been used by Ta & Weiss (2007) for broadband MIMO precoding and equalisation, and by Vaidyanathan (1993) for polyphase analysis and synthesis matrices for filter banks. For polynomial matrix problems that require an extension of the the QR decomposition (QRD), a polynomial matrix QR decomposition by columns (PQRD-BC) algorithm has been defined by Foster *et al.* (2009). PQRD-BC uses finite impulse response (FIR) paraunitary matrices to approximately upper-triangularise a polynomial matrix.

Applying a multiple shift approach to the sequential matrix diagonalisation (SMD) polynomial eigenvalue decomposition (PEVD) algorithms for parahermitian matrices has been proven to be beneficial by Corr *et al.* (2014) This paper describes a similar shifting approach applied to the PQRD-BC algorithm to create a multiple shift PQRD-BC (MS-PQRD-BC) algorithm. We demonstrate that using the developed shifting approach, increased performance relative to PQRD-BC can be achieved.

Below, Sec. 2 will provide a brief overview of the MS-PQRD-BC method. Simulation results demonstrating the performance increase are presented in Sec. 3, with conclusions drawn in Sec. 4.

2. Multiple Shift Polynomial QR Decomposition by Columns

The MS-PQRD-BC algorithm approximates the PQRD using a series of elementary paraunitary operations to iteratively triangularise a polynomial matrix $\boldsymbol{A}(z) \in \mathbb{C}^{p\times q}$. Note, $\boldsymbol{A}(z)$ is the z-transform of a set of coefficient matrices relating to different lags, $\mathbf{A}[\tau]$.

The objective of the algorithm is to calculate a paraunitary matrix $\boldsymbol{Q}(z) \in \mathbb{C}^{p\times p}$ such that

$$\boldsymbol{Q}(z)\boldsymbol{A}(z) \cong \boldsymbol{R}(z)\,, \tag{2.1}$$

where $\boldsymbol{R}(z) \in \mathbb{C}^{p\times q}$ is an upper-triangular polynomial matrix. The series of paraunitary operations used to compute the polynomial matrix $\boldsymbol{Q}(z)$ in (2.1) can be broken down into elementary delay and rotations matrices — as described by Vaidyanathan (1993). These matrices can be used to formulate an elementary polynomial Givens rotation (EPGR).

2.1. *An Elementary Polynomial Givens Rotation*

An EPGR is a polynomial matrix that can be applied to a polynomial matrix $\boldsymbol{A}(z) \in \mathbb{C}^{p\times q}$ to zero one coefficient of a polynomial element. For example, if the polynomial coefficient $a_{jk}[t]$ for $j > k$ is to be made equal to zero, the required EPGR takes the form of

$$\boldsymbol{G}^{(j,k,t,\alpha,\theta,\phi)}(z) = \begin{bmatrix} \mathbf{I}_{k-1} & & & & \\ & ce^{i\alpha} & \cdots & se^{i\phi}z^t & \\ & \vdots & \mathbf{I}_{j-k-1} & \vdots & \\ & -se^{-i\phi} & \cdots & ce^{-i\alpha}z^t & \\ & & & & \mathbf{I}_{p-j} \end{bmatrix}, \tag{2.2}$$

where c and s define the cosine and sine of the angle θ, respectively. The rotation angles θ, α, and ϕ are chosen such that

$$\theta = \begin{cases} \tan^{-1}\left(\dfrac{|a_{jk}[t]|}{|a_{kk}[0]|}\right), & \text{if } a_{kk}[0] \neq 0 \\ \pi/2, & \text{if } a_{kk}[0] = 0 \end{cases} \tag{2.3}$$

$$\alpha = -\arg(a_{kk}[0]) \;\text{ and }\; \phi = -\arg(a_{jk}[t])\,; \tag{2.4}$$

thus resulting in $a'_{jk}[0] = 0$, where $\boldsymbol{A'}(z) = \boldsymbol{G}^{(j,k,t,\alpha,\theta,\phi)}(z)\boldsymbol{A}(z)$. Furthermore, following the application of the EPGR, the coefficient $a'_{kk}[0]$ has increased in magnitude squared such that $|a'_{kk}[0]|^2 = |a_{kk}[0]|^2 + |a_{jk}[t]|^2$ and this coefficient will also be real.

2.2. *MS-PQRD-BC Algorithm*

The MS-PQRD-BC algorithm operates as a series of ordered column-steps where at each step, all coefficients relating to all polynomial elements beneath the diagonal in one column of the matrix are driven sufficiently small by applying a series of EPGR matrices interspersed with paraunitary delay matrices. The term column-step is used, as the columns of $\boldsymbol{A}(z) \in \mathbb{C}^{p\times q}$ are visited in sequence according to the ordering $k = 1, 2, \ldots, \min\{(p-1), q\}$.

At iteration i of column-step k, the search step of MS-PQRD-BC locates $n = (p-k)$ coefficients with maximum magnitude from the n rows beneath the diagonal in column k, i.e., a dominant coefficient is found for each row according to

$$\{\mathbf{t}^{(i)}\} = \underset{t_1,\ldots,t_n}{\arg\max}\, \{|a^{(i-1)}_{(k+1)k}[t_1]| + \cdots + |a^{(i-1)}_{pk}[t_n]|\} \quad , \tag{2.5}$$

where $\mathbf{t}^{(i)} = [t_1^{(i)}, \ldots, t_n^{(i)}]^T$ contains the lags of the dominant coefficients for rows $j = k+1, \ldots, p$. The scalar $a^{(i-1)}_{jk}[t]$ represents the element in the jth row and kth column of the coefficient matrix at lag t and iteration $(i-1)$.

Following the identification of $\mathbf{t}^{(i)}$ in (2.5), a delay matrix $\boldsymbol{\Lambda}_i(z) \in \mathbb{R}^{p\times p}$, which takes

the form

$$\mathbf{\Lambda}_i(z) = \begin{bmatrix} \mathbf{I}_k & & & \\ & z^{-t_1^{(i)}} & & \\ & & \ddots & \\ & & & z^{-t_n^{(i)}} \end{bmatrix}, \tag{2.6}$$

can be applied to $\boldsymbol{A}^{(i-1)}(z)$ to induce a $t_m^{(i)}$-fold delay to the mth row beneath the diagonal in column k, where $m = 1, \ldots, n$. The resulting matrix $\boldsymbol{A}^{(i-1)\prime}(z) = \mathbf{\Lambda}_i(z)\boldsymbol{A}^{(i-1)}(z)$ will contain the dominant coefficients from column k on its zero-lag.

EPGRs are then applied to $\mathbf{A}^{(i-1)\prime}[0]$ in sequence to drive elements $a_{(k+1)k}^{(i-1)\prime}[0], \ldots, a_{pk}^{(i-1)\prime}[0]$ to zero according to

$$\begin{aligned} \mathbf{A}^{(i-1)\prime\prime}[0] &= (\mathbf{G}^{(k+1,k,\gamma_1)} \cdots \mathbf{G}^{(p,k,\gamma_n)})\mathbf{A}^{(i-1)\prime}[0] \\ &= \mathbf{H}_i \mathbf{A}^{(i-1)\prime}[0] \,, \end{aligned} \tag{2.7}$$

where the elements beneath diagonal element k in $\mathbf{A}^{(i-1)\prime\prime}[0]$ are zero, and $\mathbf{G}^{(j,k,\gamma)}$ represents the EPGR required to zero element $a_{jk}^{(i-1)\prime}[0]$ of $\mathbf{A}^{(i-1)\prime}[0]$. Vector $\gamma = [\alpha, \theta, \phi]^T$ contains the calculated rotation angles from (2.3) and (2.4). Note that $\mathbf{G}^{(j,k,\gamma)}$ is an EPGR without shift parameter t, and is therefore a simple unitary matrix.

Matrix $\mathbf{H}_i$ in (2.7) is the product of EPGRs required to drive the dominant coefficients in iteration i of MS-PQRD-BC to zero. This unitary matrix is applied to all lags of $\boldsymbol{A}^{(i-1)\prime}(z)$ to generate $\boldsymbol{A}^{(i-1)\prime\prime}(z) = \mathbf{H}_i\boldsymbol{A}^{(i-1)\prime}(z)$.

The final step of an iteration of MS-PQRD-BC is to apply an inverse delay matrix $\tilde{\mathbf{\Lambda}}_i(z) = \mathbf{\Lambda}_i^{\mathrm{H}}(z^{-1})$ to produce the overall transformation

$$\begin{aligned} \boldsymbol{A}^{(i)}(z) &= \tilde{\mathbf{\Lambda}}_i(z)\mathbf{H}_i\mathbf{\Lambda}_i(z)\boldsymbol{A}^{(i-1)}(z) \\ &= \boldsymbol{Q}^{(i,k)}(z)\boldsymbol{A}^{(i-1)}(z) \,. \end{aligned} \tag{2.8}$$

This iterative process is repeated until all coefficients associated with polynomial elements beneath the diagonal in the kth column of the matrix are sufficiently small in magnitude and therefore satisy the following stopping condition:

$$|a_{jk}[t]| < \epsilon \tag{2.9}$$

for $j > k$ and $\forall t \in \mathbb{Z}$ where $\epsilon > 0$ is a prespecified small value. The overall transformation performed in the kth column-step of the algorithm is of the form

$$\boldsymbol{A}^k(z) = \boldsymbol{Q}^k(z)\boldsymbol{A}(z) \tag{2.10}$$

where $\boldsymbol{Q}^k(z) \in \mathbb{C}^{p\times p}$ is the paraunitary product of all EPGRs and delay matrices applied during iterations $i = 1, \ldots, I_k$ within column-steps $1, \ldots, k$:

$$\boldsymbol{Q}^k = \prod_{i=1}^{I_k} \boldsymbol{Q}^{(i,k)} \cdots \prod_{i=1}^{I_1} \boldsymbol{Q}^{(i,1)} \,, \tag{2.11}$$

where

$$\prod_{n=1}^{I_k} \boldsymbol{Q}^{(i,k)} = \boldsymbol{Q}^{(I_k,k)}\boldsymbol{Q}^{(I_k-1,k)} \cdots \boldsymbol{Q}^{(2,k)}\boldsymbol{Q}^{(1,k)} \tag{2.12}$$

is the product of all EPGRs and delay matrices applied during iterations $i = 1, \ldots, I_k$ and I_k is the maximum number of iterations within column-step k. $I_k = \frac{I}{p-k}$ for some I.

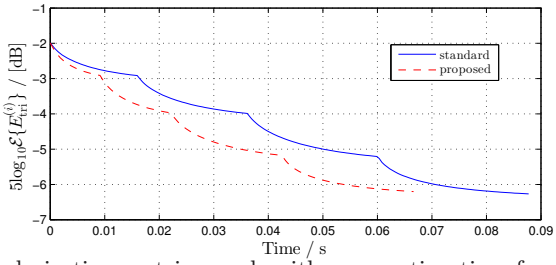

FIGURE 1. Triangularisation metric vs. algorithm execution time for the proposed and standard implementations for $\boldsymbol{A}(z) \in \mathbb{C}^{5\times 5}$ for polynomial order 14.

Following this column-step, all coefficients relating to all polynomial elements beneath the diagonal in the kth column of the matrix $\boldsymbol{A}(z)$ will be sufficiently small.

After $l = \min\{(p-1), q\}$ column-steps of the algorithm, all columns of the matrix have been visited, and the transformation is of the form

$$\boldsymbol{R}(z) = \boldsymbol{Q}(z)\boldsymbol{A}(z) \tag{2.13}$$

where $\boldsymbol{Q}(z)$ is the paraunitary product of all EPGRs and delay matrices applied during column-steps $1, \ldots, l$.

3. Results

A suitable normalised metric $E_{\text{tri}}^{(\hat{i})}$ is used for evaluating standard PQRD-BC and proposed MS-PQRD-BC implementations, which divides the lower-triangular energy in $\boldsymbol{A}(z)$ at the $\hat{i}$th algorithm iteration — ignoring column-steps — by the total energy.

Simulations were performed over an ensemble of 10^3 instantiations of $\boldsymbol{A}(z) \in \mathbb{C}^{5\times 5}$, for a polynomial order of 14, $\epsilon = 10^{-6}$, and $I = 50$. Real and imaginary components of $\boldsymbol{A}(z)$ were drawn from a Gaussian source with unit variance and zero mean.

The ensemble-averaged triangularisation versus algorithm execution time for both methods is shown in Fig. 1. The curves demonstrate that the multiple shift implementation obtains a faster triangularisation than the standard realisation.

4. Conclusion

In this paper, we have proposed a multiple shift approach to increase the performance of an existing polynomial QR decomposition algorithm. We have shown through simulation that the implementation of this approach translates to an increase in triangularisation speed of the created MS-PQRD-BC algorithm.

REFERENCES

TA, C. H. & WEISS, S. 2007 A design of precoding and equalisation for broadband MIMO systems. In *Asilomar SSC*, pp. 1616–1620, Pacific Grove, CA.

VAIDYANATHAN, P. P. 1993 Multirate Systems and Filter Banks. Prentice Hall, Englewood Cliffs.

FOSTER, J. A., MCWHIRTER, J. G. & CHAMBERS, J. A. 2009 A novel algorithm for calculating the QR decomposition of a polynomial matrix. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 3177–3180, Taipei, Taiwan.

CORR, J., THOMPSON, K., WEISS, S., MCWHIRTER, J., REDIF, S. & PROUDLER, I. 2014 Multiple shift maximum element sequential matrix diagonalisation for parahermitian matrices. In *IEEE SSP*, pp. 312–315, Gold Coast, Australia.