AN EFFICIENT AND RELIABLE IMPLEMENTATION OF THE PERIODIC QZ ALGORITHM

Daniel Kressner *

* *Department of Mathematics, University of Chemnitz, Chemnitz, Germany*

Abstract: We discuss performance and accuracy aspects of the periodic QZ algorithm. Blocked formulations of the involved orthogonal transformations increase the data locality and thus address the first task. For the sake of reliability the proposed implementation includes balancing, implicit methods for computing shifts and carefully chosen deflation strategies. Algorithms for pole placement and other tasks arising from periodic discrete-time systems could benefit from these improvements.

Keywords: Accuracy, Discrete-time systems, Efficient algorithms, Eigenvalue problems, Exponentially stable, Factorization methods, Implementation

1. INTRODUCTION

For matrices $E_i, F_i \in \mathbb{R}^{n,n}$ and $G_i \in \mathbb{R}^{n,m}$ consider the linear discrete-time system

$$E_i x_{i+1} = F_i x_i + G_i u_i, \ i \in \mathbb{N}, \tag{1}$$

where x_i, u_i are vectors of states and inputs, respectively. The coefficient matrices shall satisfy $E_{i+k} = E_i$, $F_{i+k} = F_i$ and $G_{i+k} = G_i$ for some fixed $k \in \mathbb{N}$. Such periodic systems naturally arise when performing multirate sampling of continuous-time systems.

The corresponding monodromy matrix is associated with the formal product

$$E_k^{-1} F_k E_{k-1}^{-1} F_{k-1} \dots E_1^{-1} F_1. \tag{2}$$

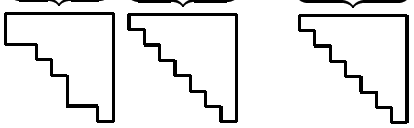
Some of the equations in (1) might have no algebraic constraints, that is, E_i is the identity matrix for some $i \in [1, k]$. In this context it is more appropriate to replace (2) by the general product

$$S = A_1 A_2^{s_2} A_3^{s_3} \dots A_k^{s_k}, \tag{3}$$

where $A_i \in \mathbb{R}^{n,n}$ and $s_i \in \{-1, 1\}$. The implicit assumption $s_1 = 1$ can always be achieved by a suitable reordering or formal inversion of the coefficient matrices. Note that the invertibility of factors A_i with $s_i = -1$ is not assumed. Even when this condition is satisfied it is not favorable to form (3) explicitly.

The periodic Schur decomposition is often the first and most expensive step in numerically reliable methods for pole placement and several other tasks related to linear periodic systems (Sreedhar and Van Dooren, 1993). In this decomposition k orthogonal matrices Q_i are constructed so that $Q_i^T A_i Q_{i+1}$, for $s_i = 1$, and $Q_{i+1}^T A_i Q_i$, for $s_i = -1$, are upper triangular, where $i = 2, \dots, k$ and $Q_{k+1} = Q_1$. The first transformed factor $Q_1^T A_1 Q_2$ is upper quasi-triangular. Illustrated:

$$\underbrace{Q_1^T A_1 Q_2}\ \underbrace{Q_2^T A_2^{s_2} Q_3}\ \dots\ \underbrace{Q_k^T A_k^{s_k} Q_1} \tag{4}$$



A numerically stable method to compute (4) is the so called Periodic QZ algorithm established by (Van Loan, 1975; Bojanczyk *et al.*, 1992; Hench and Laub, 1994). Sections 2 and 3 of this work are concerned with variations of this algorithm which significantly decrease its execution time. Reliability, a crucial aspect of any competitive

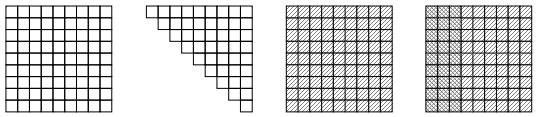
implementation, is treated in Sections 4, 5 and 6.

2. BLOCKED REDUCTION TO PERIODIC HESSENBERG FORM

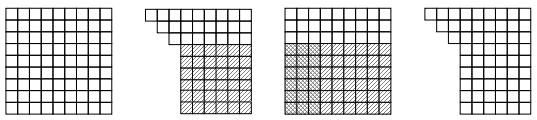
As usual for algorithm which compute a variant of the Schur decomposition, the first step consists of the reduction to a Hessenberg like form. In the context of general products, this form is almost identical to (4) besides the first factor which stays upper Hessenberg.

An efficient implementation should, in a first attempt, reduce A_1 only to block Hessenberg form, that is, $n_b \geq 1$ subdiagonals are nonzero. This concept was successfully applied to the QZ algorithm (Dackland and Kågström, 1999). Since the technique easily generalizes to the periodic case, only a brief outline of the method for the special product $A_1 A_2^{-1} A_3 A_4$ with $n = 9$ and $n_b = 3$ is presented.

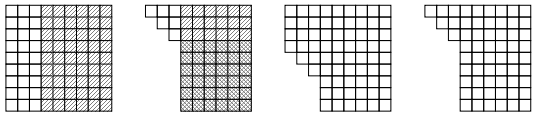
The first stage starts with an RQ decomposition of A_2 which alters the matrix A_1. Next, $A_4(:, 1:3)$ is triangularized by three Householder reflectors from the left. Their WY representation is applied to the remaining part of A_4 as well as the matrix A_3 (Golub and Loan, 1996, Section 5.1.7).



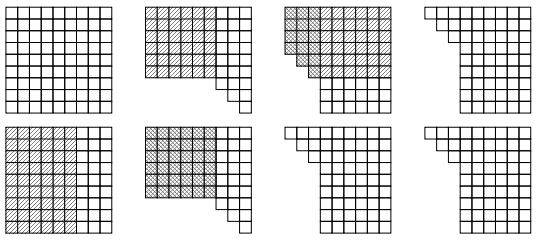
A QR decomposition triangularizes $A_3(4:9, 1:3)$ introducing nonzeros in the lower triangular part of $A_2(4:9, 4:9)$.



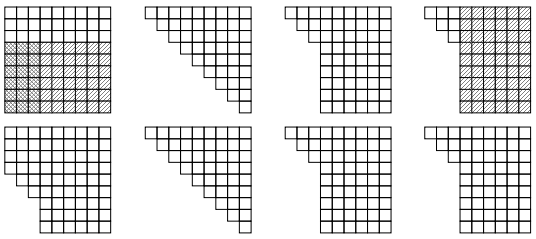
This part is immediately annihilated by an appropriate RQ decomposition.



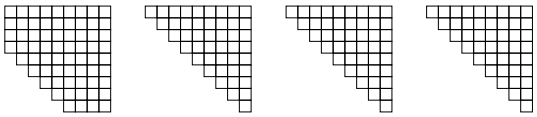
Repeating the procedure for $A_3(1:6, 1:3)$ yields the following pattern.



Now, three blocked Householder reflectors triangularize $A_1(:, 1:3)$ from the left.



Repeating the procedure for the remaining south-east 6-by-6 subproduct finally leads to the following block Hessenberg form.



It should be noted that the fill-ins in A_2 overlap for consecutive iterations. Hence, in an actual implementation RQ decompositions of smaller sized r-by-$2r$ matrices are applied in order to annihilate these fill-ins.

The second stage is to annihilate the remaining $n_b - 1$ subdiagonals of A_1, to get an upper Hessenberg matrices while keeping the other matrices upper triangular. This is accomplished via a super-sweep routine as for the general product AB^{-1} described in (Dackland and Kågström, 1999). There is virtually no difference when going to larger products, only that the description of algorithm becomes rather awkward.

Several benchmarks were run to compare the above described blocked with the original version (Bojanczyk *et al.*, 1992) of the Hessenberg reduction algorithm. The FORTRAN 77 implementations were compiled and serially executed on an Origin 2000 computer equipped with 400 MHz IP27 R12000 processors. The programs call optimized BLAS and LAPACK (Anderson *et al.*, 1994) subroutines from the SGI/Cray Scientific Library version 1.2.0.0. The block size n_b was chosen to be 64.

s	n	Original	Blocked
$(1, -1)$	1024	432	195
$(1, -1)$	2048	4740	2481
$(1, -1, 1, -1, 1, -1)$	512	110	60
$(1, -1, 1, -1, 1, -1)$	1024	1295	595
$(1, 1, 1, 1, 1, 1)$	512	62	39
$(1, 1, 1, 1, 1, 1)$	1024	1047	308

Table 1. Execution times in seconds for the reduction to periodic Hessenberg form.

In Table 1 an extract of the observed execution times is presented. Overall, significantly lower times for the blocked algorithms were noted as soon as $n \geq 128$.

3. BLOCKED PERIODIC QZ ITERATION

A similiar algorithm as for the supersweep algorithm can be used to speed up the generalized QZ iterations as well (Dackland and Kågström, 1999). Again, there is a straightforward way to adapt this technique to the periodic case. From the results in Section 2 it can be extrapolated that the execution times of such a blocked periodic QZ iteration will be significantly lower than the original formulation.

4. BALANCING

The periodic QZ algorithm is backward stable (Bojanczyk *et al.*, 1992). That is, the computed Schur decomposition corresponds to a slightly perturbed product $\prod_{i=1}^{k} \hat{A}_i^{s_i}$, where the backward error $\|\hat{A}_i - A_i\|$ is of the order unit round off times $\|A_i\|$. However, even such small perturbations can be harmful when dealing with ill-conditioned problems.

For example, consider the formal product

$$\begin{bmatrix} 5^{-26} & 3^{-14} & 6^{-16} \\ 6^{-06} & 2^{+06} & 3^{+04} \\ 4^{-16} & 2^{-04} & 5^{-06} \end{bmatrix} \begin{bmatrix} 6^{-28} & 3^{-16} & 5^{-18} \\ 7^{-09} & 3^{+03} & 7^{+01} \\ 6^{-23} & 3^{-11} & 3^{-13} \end{bmatrix}^{-1} \cdot \begin{bmatrix} 8^{-02} & 6^{-24} & 6^{-11} \\ 5^{+17} & 5^{-05} & 6^{+08} \\ 3^{+03} & 4^{-19} & 7^{-06} \end{bmatrix} \begin{bmatrix} 9^{+00} & 4^{-22} & 3^{-09} \\ 7^{+20} & 2^{-02} & 9^{+11} \\ 4^{+10} & 6^{-12} & 7^{+01} \end{bmatrix}^{-1}, \quad (5)$$

where the signed integer superscript at the end of a number represents its exponential exponent. The eigenvalues, given by the general product of the diagonal elements of the periodic Schur decomposition, and the corresponding condition numbers (Benner *et al.*, 2000) are tabulated below.

Eigenvalue	Condition number
2.88728	4.32×10^{21}
0.39941	1.77×10^{21}
0.07459	2.59×10^{21}

Not surprisingly the periodic QZ algorithm completely fails to compute eigenvalues with acceptable accuracy.

Such effects, caused from matrix entries of widely varying magnitudes, can be removed by a preceding balancing step. For positive definite diagonal matrices D_α, D_β, D_γ, D_ξ the eigenvalues of the formal product $AB^{-1}CE^{-1}$ and

$$(D_\alpha A D_\beta)(D_\gamma B D_\beta)^{-1}(D_\gamma C D_\xi)(D_\alpha E D_\xi)^{-1} \quad (6)$$

are equivalent. Different sign patterns do not pose a problem; if for example $s_B = 1$, then in the following discussion B can virtually be replaced by the matrix

$$\tilde{B} = \left[\tilde{b}_{ij}\right]_{i,j=1}^{n} := \left[\delta(b_{ji} \neq 0) \cdot \frac{1}{b_{ji}}\right]_{i,j=1}^{n}.$$

The diagonal transformations shall reduce the condition numbers and thus improve the accuracy of the computed eigenvalues. However, minimizing the conditioning of the periodic eigenvalue problem is certainly an unrealistic goal. On the other hand, reducing the magnitude ranges of the elements in the factors seems to be reasonable.

Analogously to the generalized eigenvalue problem (Ward, 1981), the balancing step can be formulated as the solution of an optimization problem. Let α_i, β_i, γ_i and ξ_i denote the binary logarithm of the i-th diagonal entry in the corresponding diagonal matrix. Then one wants to minimize the expression

$$\begin{aligned} S(\alpha,\beta,\gamma,\xi) = \sum_{i,j=1}^{n} & (\alpha_i + \beta_j + \log_2|a_{ij}|)^2 \\ & + (\gamma_i + \beta_j + \log_2|b_{ij}|)^2 \\ & + (\gamma_i + \xi_j + \log_2|c_{ij}|)^2 \\ & + (\alpha_i + \xi_j + \log_2|e_{ij}|)^2. \end{aligned} \quad (7)$$

By differentiation a minimal point $(\alpha, \beta, \gamma, \xi)$ satisfies the linear system of equations with system matrix

$$\begin{bmatrix} F(E,A) & H(A) & 0 & H(E) \\ H^T(A) & G(A,B) & H^T(B) & 0 \\ 0 & H(B) & F(B,C) & H(C) \\ H^T(E) & 0 & H^T(C) & G(C,E) \end{bmatrix}$$

and right hand side

$$-\begin{bmatrix} \mathrm{row}(A) + \mathrm{row}(E) \\ \mathrm{col}(B) + \mathrm{col}(A) \\ \mathrm{row}(C) + \mathrm{row}(B) \\ \mathrm{col}(E) + \mathrm{col}(C) \end{bmatrix},$$

where the notation is as follows:

(1) $F(X,Y)$ / $G(X,Y)$ is a diagonal matrix whose elements are given by the number of nonzero entries in the rows / columns of X and Y,
(2) $H(X)$ is the incidence matrix of X,
(3) $\mathrm{row}(X)$ / $\mathrm{col}(X)$ is the vector of row / column sums of the matrix

$$\left[\delta(x_{ij} \neq 0) \cdot \log_2|x_{ij}|\right]_{i,j=1}^{n}.$$

It can be shown that this linear system is symmetric, positive semidefinite and consistent. For its solution a generalized conjugate gradient iteration is used as described in (Ward, 1981).

To reduce the computational it is desirable to construct a suitable preconditioner. Under the assumption of completely dense factors the system matrix is for even k given by a kn-kn block circulant $M_{k,n}$ with first n rows

$$\left[\, 2nI \;\; ee^T \;\; 0 \;\; \ldots \;\; 0 \;\; ee^T \,\right],$$

and for odd k by a kn-by-kn block skew circulant $N_{k,n}$ with first n rows

$$\left[\, 2nI \;\; ee^T \;\; 0 \;\; \ldots \;\; 0 \;\; -ee^T \,\right],$$

where e is the n-vector containing a one in each element.

To be useful for preconditioning the application of the Moore-Penrose generalized inverses $M_{k,n}^{\dagger}$ and $N_{k,n}^{\dagger}$ should be cheap. Indeed, for $x \in \mathbb{R}^k$ the products $M_{k,1}^{\dagger}x$ and $N_{k,1}^{\dagger}x$ can be formed within $O(k)$ operations by using an incomplete Cholesky factorization.

One can show that for general $n \geq 1$, now $x \in \mathbb{R}^{kn}$,

$$M_{k,n}^{\dagger}x = \frac{1}{n^2}\left[\frac{n}{2}I_{kn} + \left(M_{k,1}^{\dagger} - \frac{1}{2}I_k\right) \otimes ee^T\right]x.$$

An analogous result holds for $N_{k,n}^{bot}x$.

Hence, per iteration of the conjugate gradient method $O(kn^2)$ operations are required. If the factors are reasonably dense, then the iterative scheme usually converges within 3 iterations, which was already observed in the context of the generalized eigenvalue problem (Ward, 1981).

For Example (5) the binary logarithms of the optimal scaling parameters are given by

$$\alpha = \left[\, 36.3 \;\; -30.0 \;\; 3.14 \,\right] \quad \beta = \left[\, 47.6 \;\; 8.85 \;\; 14.7 \,\right]$$
$$\gamma = \left[\, 42.7 \;\; -20.4 \;\; 26.5 \,\right] \quad \xi = \left[\, -38.8 \;\; 34.7 \;\; -8.96 \,\right]$$

The eigenvalues of the balanced product are substantially less sensitive as shown below.

Eigenvalue	Condition number
2.88728	2.49
0.39941	4.40
0.07459	3.44

As expected, the periodic QZ algorithm now reveals eigenvalues nearly to machine precision.

5. SHIFT COMPUTATION

At the start of each periodic QZ iteration an initial orthogonal matrix Q_0 is applied to both sides of the product. Given m shifts σ_i the matrix Q_0 is required to satisfy the condition that its first column is parallel to the first column of the shift polynomial

$$P_\sigma = \left(\prod_{i=1}^{k} A_i^{s_i} - \sigma_1\right) \cdots \left(\prod_{i=1}^{k} A_i^{s_i} - \sigma_m\right).$$

An usual choice of shifts is to take the two eigenvalues of the southeast two-by-two part of the product,

$$\begin{bmatrix} a_{mm}^{(1)} & a_{mn}^{(1)} \\ a_{nm}^{(1)} & a_{nn}^{(1)} \end{bmatrix} \begin{bmatrix} a_{mm}^{(2)} & a_{mn}^{(2)} \\ 0 & a_{nn}^{(2)} \end{bmatrix}^{s_2} \cdots \begin{bmatrix} a_{mm}^{(k)} & a_{mn}^{(k)} \\ 0 & a_{nn}^{(k)} \end{bmatrix}^{s_k},$$

where $m = n - 1$ and $a_{jl}^{(i)}$ denotes the (j,l)-th entry of A_i.

Especially for long products, computing the shifts and the shift polynomial desires for great care to avoid unnecessary over-/underflow and disastrous cancellations. From this point of view it is more favorable to construct Q_0 directly from the given data. For example, if the shift polynomial can be rewritten as a product of matrices, then Q_0 can be computed by a partial product QR factorization (De Moor and Van Dooren, 1992). For the double shift strategy described above a suitable product embedding is given by

$$P_\sigma = \left[\, A_1 \;\; I_n \,\right] \cdot \prod_{i=2}^{k} \begin{bmatrix} A_i & 0 \\ 0 & a_{mm}^{(i)} I_n \end{bmatrix}^{s_i} \cdot \begin{bmatrix} -I_n & 0 \\ a_{mm}^{(1)} I_n & -a_{nm}^{(1)} I_n \end{bmatrix} \begin{bmatrix} -A_1 & a_{nm}^{(1)} I_n & a_{nn}^{(1)} I_n \\ 0 & a_{mm}^{(1)} I_n & a_{mn}^{(1)} I_n \end{bmatrix} \cdot \prod_{i=2}^{k} \begin{bmatrix} A_i & 0 & 0 \\ 0 & a_{mm}^{(i)} I_n & a_{mn}^{(i)} I_n \\ 0 & 0 & a_{nn}^{(i)} I_n \end{bmatrix}^{s_i} \cdot \begin{bmatrix} I_n \\ 0 \\ I_n \end{bmatrix}.$$

By carefully exploiting the underlying structure the recursive computation of Q_0 from this embedding requires approximately $37k$ operations.

6. DEFLATION AND EXPONENTIAL SPLITTINGS

Convergence is certainly the most important aspect of an iterative algorithm.

Consider the product with factors

$$A_1 = \begin{bmatrix} 9.0 & 4.0 & 1.0 & 4.0 & 3.0 & 4.0 \\ 6.0 & 8.0 & 2.0 & 4.0 & 0.0 & 2.0 \\ 0.0 & 7.0 & 4.0 & 4.0 & 6.0 & 6.0 \\ 0.0 & 0.0 & 8.0 & 4.0 & 6.0 & 7.0 \\ 0.0 & 0.0 & 0.0 & 8.0 & 9.0 & 3.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 5.0 & 0.0 \end{bmatrix}, \qquad (8)$$

$$A_2 = \cdots = A_k = \mathrm{diag}(10^{-1}, 10^{-2}, 10^{-3}, 1, 1, 1),$$

and $s_2 = \cdots = s_k = 1$. For $k = 5$ the periodic QZ algorithm requires 29 iterations to converge, 62 for $k = 10$, 271 for $k = 40$ and for $k \geq 50$ it does not converge at all. Even worse, the breakdown can not be cured by using standard ad hoc shifts.

The reason is basically that the leading diagonal entries in the triangular factors diverge exponentially, that is, the relation

$$\prod_{i=1}^{k} \left(\frac{a_{j+1,j+1}^{(i)}}{a_{jj}^{(i)}}\right)^{s_i} = O(\alpha^k), \quad 0 \leq \alpha < 1, \qquad (9)$$

is satisfied for $j = 1, 2$. A Givens rotator acting on such a $(j, j+1)$ plane is likely to converge to the 2-by-2 identity matrix when propagated over $A_k, A_{k-1}, \ldots, A_2$ back to A_1.

It is important to note that (9) is *not* an exceptional situation. Exponentially splitted products in the sense of (Oliveira and Stewart, 2000) have the pleasant property that even for extremely large k the eigenvalues can be computed to high relative accuracy. Moreover, such products hardly ever fail to satisfy (9). One of the prominent examples is the infinite product where all factors have random entries chosen from a uniform distribution on the interval $(0, 1)$. It can be shown that the sequence of periodic Hessenberg forms related to finite truncations of this product satisfies (9) for all $j = 1, \ldots, n-1$.

In the original algorithm (Bojanczyk *et al.*, 1992), a direct deflation is only performed when a small subdiagonal element in A_1 or a small diagonal element in $A_2, \ldots, A_k$ is encountered. For the purpose that exponentially diverging diagonal entries do not represent a convergence barrier the following additional deflation strategy is proposed.

A QR decomposition is applied to the Hessenberg matrix A_0. If $s_k = 1$, the resulting $n-1$ Givens rotators (c_j, s_j) are successively applied to the columns of A_k,

$$\begin{bmatrix} a_{j,j}^{(k)} & a_{j+1,j}^{(k)} \\ 0 & a_{j+1,j+1}^{(k)} \end{bmatrix} \begin{bmatrix} c_j & s_j \\ -s_j & c_j \end{bmatrix} = \begin{bmatrix} c_j a_{j,j}^{(k)} - s_j a_{j+1,j}^{(k)} & s_j a_{j,j}^{(k)} + c_j a_{j+1,j}^{(k)} \\ -s_j a_{j+1,j+1}^{(k)} & c_j a_{j+1,j+1}^{(k)} \end{bmatrix}.$$

Whenever it happens that $\left|s_j a_{j+1,j+1}^{(k)}\right|$ is small compared to

$$\max\left(\left|c_j a_{j,j}^{(k)} - s_j a_{j+1,j}^{(k)}\right|, \left|c_j a_{j+1,j+1}^{(k)}\right|\right),$$

or, being more generous, compared to $\|A_k\|_F$, then in the following steps (c_j, s_j) can be safely set to $(1, 0)$. Otherwise, the $(j+1, j)$-th element of A_k is annihilated by a Givens rotator acting on rows $(j, j+1)$. (c_j, s_j) is overwritten with the parameters of this rotator.

The process, being similiar when $s_k = 1$, is recursively applied to $A_{k-1}, \ldots, A_2$. At the end, the rotator sequence is applied to the columns of A_1 and each pair $(c_j, s_j) = (1, 0)$ results in a zero element at position $(j+1, j)$ in A_1.

Since the above procedure is as expensive as a single shift periodic QZ iteration it should only occasionally be applied.

For Example (8) with $k = 40$ two applications of the proposed deflation strategy result in zeros at positions $(2, 1)$, $(3, 2)$ and $(7, 6)$ in A_1. Barely 7 periodic QZ iterations are required to reduce the remaining 3-by-3 product to quasi upper triangular form.

7. SOFTWARE IMPLEMENTATION

A FORTRAN 77 software package based on the described algorithms is being developed. The routines conform to the SLICOT implementation and documentation standards (Benner *et al.*, 1999) and are readily available from

`http://www.math.tu-berlin.de/~kressner/`

8. ACKNOWLEDGEMENT

The author sincerly thanks Ralph Byers and Volker Mehrmann for useful discussions and help throughout this work.

9. REFERENCES

Anderson, E., Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen (1994). *LAPACK Users' Guide*. second ed.. SIAM. Philadelphia, PA.

Benner, P., V. Mehrmann and H. Xu (2000). Perturbation analysis for the eigenvalue problem of a formal product of matrices. Berichte aus der Technomathematik 00-01. Fachbereich Mathematik und Informatik, Univ. Bremen.

Benner, P., V. Mehrmann, V. Sima, S. Van Huffel and A. Varga (1999). SLICOT—a subroutine library in systems and control theory. In: *Applied and computational control, signals, and circuits, Vol. 1*. pp. 499–539. Birkhäuser Boston. Boston, MA.

Bojanczyk, A., G. H. Golub and P. Van Dooren (1992). The periodic Schur decomposition; algorithm and applications. In: *Proc. SPIE Conference*. Vol. 1770. pp. 31–42.

Dackland, K. and B. Kågström (1999). Blocked algorithms and software for reduction of a regular matrix pair to generalized Schur form. *ACM Trans. Math. Software* **25**(4), 425–454.

De Moor, B. and P. Van Dooren (1992). Generalizations of the singular value and QR-decompositions. *SIAM J. Matrix Anal. Appl.* **13**(4), 993–1014.

Golub, G. H. and C. F. Van Loan (1996). *Matrix Computations*. third ed.. Johns Hopkins University Press. Baltimore.

Hench, J. J. and A. J. Laub (1994). Numerical solution of the discrete-time periodic Riccati equation. *IEEE Trans. Automat. Control* **39**(6), 1197–1210.

Oliveira, S. and D. E. Stewart (2000). Exponential splittings of products of matrices and accurately computing singular values of long products. In: *Proceedings of the International Workshop on Accurate Solution of Eigenvalue*

Problems (University Park, PA, 1998). Vol. 309:1-3. pp. 175–190.

Sreedhar, J. and P. Van Dooren (1993). Pole placement via the periodic Schur decomposition. In: *Proceedings Amer. Contr. Conf.*. pp. 1563–1567.

Van Loan, C. F. (1975). A general matrix eigenvalue algorithm. *SIAM J. Numer. Anal.* **12**(6), 819–834.

Ward, Robert C. (1981). Balancing the generalized eigenvalue problem. *SIAM J. Sci. Statist. Comput.* **2**(2), 141–152.