Periodic Lyapunov equations: some applications and new algorithms

ANDRAS VARGA[†]

The discrete-time periodic Lyapunov equation has several important applications in the analysis and design of linear periodic control systems. Specific applications considered in the paper are the solution of state- and output-feedback optimal periodic control problems, the stabilization by periodic state feedback and the square-root balancing of discrete-time periodic systems. Efficient numerically reliable algorithms based on the periodic Schur decomposition are proposed for the solution of periodic Lyapunov equations. The proposed algorithms are extensions of the direct solution methods for standard discrete-time Lyapunov equations for the case of indefinite as well as of nonnegative definite solution.

1. Introduction

Consider the linear discrete-time periodic system of the form

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k \\ y_k &= C_k x_k + D_k u_k \end{aligned} \tag{1.1}$$

where the matrices $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{n\times m}$, $C_k \in \mathbb{R}^{p\times n}$ and $D_k \in \mathbb{R}^{p\times m}$ are periodic with period $K \geq 1$. Such models arise usually by the discretization of linear continuous-time periodic models which are the primary mathematical descriptions encountered in most of practical applications (Pittelkau 1993, Varga and Pieters 1996). The main advantage of using discrete-time models instead of continuous-time ones is the possibility to develop and to use efficient computational algorithms which completely parallel those for standard discrete-time systems. The computational problems considered in this paper to illustrate this fact are: the optimal periodic LQG control with state feedback and with output feedback, the stabilization with periodic state feedback and the balancing of discrete-time periodic systems.

In the last few years there has been a constantly increasing interest for the development of numerical algorithms for the analysis and design of linear periodic discrete-time control systems (Bittanti *et al.* 1988, Hench and Laub 1994, Sreedhar and Van Dooren 1993). Of particular interest in the above mentioned applications is the efficient and numerically reliable solution of various types of periodic Lyapunov equations. Several possible computational approaches to solve periodic Lyapunov equation are discussed by Sreedhar and Van Dooren (1993). The purpose of this paper is to propose alternative techniques which improve the numerical reliability of existing algorithms.

[†]DLR - Oberpfaffenhofen, German Aerospace Research Establishment, Institute for Robotics and System Dynamics, P.O.B. 1116, D-82230 Wessling, Germany. Tel: +49-8153-282407; Fax: +49-8153-281441; e-mail address: Andreas.Varga@dlr.de .

The algorithms proposed in this paper to solve *discrete periodic Lyapunov equations* (DPLEs) represent extensions of the direct solution methods for standard systems proposed by Kitagawa (1977) and Barraud (1977) for the case of indefinite solutions and of the method of Hammarling (1982) for the case of nonnegative definite solutions. The proposed approaches resemble with the method of Bartels and Stewart (1972) and rely on an initial reduction of the Lyapunov equation to a simpler form by using the periodic Schur decomposition of a matrix product (Bojanczyk *et al.* 1992). The reduced equations are solved by using special forward and backward substitution algorithms. Important computational subproblems are the efficient and numerically stable solution of order one or order two DPLEs and *discrete periodic Sylvester equations* (DPSEs). Several computational approaches for these subproblems are described. The roundoff error properties of the proposed algorithms and the estimation of condition numbers for DPLEs are also discussed. Note that a direct solution approach similar to ours, but for a class of generalized periodic Sylvester equations, has been also recently proposed by Byers and Rhee (1995).

Notations and notational conventions. For a square time-varying matrix A_k, $k = 0, 1, \ldots$, we denote $\Phi_A(j,i) = A_{j-1}A_{j-2}\cdots A_i$ for $j > i$ and $\Phi_A(i,i) := I$. If A_k is periodic with period K, then the *monodromy matrix* of the system (1.1) at time τ is $\Phi_A(\tau+K,\tau)$. Its eigenvalues are independent of τ and are called *characteristic multipliers*. For an arbitrary periodic matrix X_k of period K we use alternatively the *script* notation $\mathcal{X}$ which associates the block-diagonal matrix $\mathcal{X} = \mathrm{diag}\,(X_0, X_1, \ldots, X_{K-1})$ to the cyclic sequence of matrices X_k, $k = 0, \ldots, K-1$. This notation is consistent with the standard matrix operations. For example the operations with block-diagonal matrices $\mathcal{X}+\mathcal{Y}$, $\mathcal{X}\mathcal{Y}$, or $\mathcal{X}^{-1}$ can be used to express the addition, the multiplication and the inversion, respectively, performed simultaneously with all individual terms in a sequence of K matrices. We denote with $\sigma\mathcal{X}$ the K-cyclic shift $\sigma\mathcal{X} = \mathrm{diag}\,(X_1, \ldots, X_{K-1}, X_0)$ applied to the cyclic sequence X_k, $k = 0, \ldots, K-1$. The notation $\mathcal{X}_{ij}$ is used to refer simultaneously to *all* (i,j) elements or *all* (i,j) blocks in the cyclic sequence X_k, $k = 0, \ldots, K-1$. This notation also applies in the case of matrix partitioning. For instance the partitioning

$$\mathcal{X} = \begin{bmatrix} \mathcal{X}_{11} & \mathcal{X}_{12} \\ \mathcal{X}_{21} & \mathcal{X}_{22} \end{bmatrix}$$

refers to the same partitioning of all matrices of the cyclic sequence X_k, $k = 0, \ldots, K-1$. We will also use the notation $\begin{bmatrix} \mathcal{X} \\ \mathcal{Y} \end{bmatrix}_{(k)}$ to refer to the compound periodic matrix $\begin{bmatrix} X_k \\ Y_k \end{bmatrix}$.

2. Optimal State Feedback Periodic Control

Consider the quadratic cost functional associated to the system (1.1)

$$J = \frac{1}{2}\sum_{k=0}^{\infty}[\, x_k^T Q_k x_k + u_k^T R_k u_k \,] \tag{2.1}$$

where $Q_k \geq 0$ and $R_k > 0$ are periodic symmetric matrices with period K. With the standard assumptions of uniform stabilizability of the pair (A_k, B_k) and uniform detectability of the pair (A_k, Q_k) we can determine the optimal periodic state-feedback control

law

$$u_k^* = F_k x_k \tag{2.2}$$

which minimizes the performance index (2.1). By using the script notation for periodic matrices, $\mathcal{F}$ can be computed as

$$\mathcal{F} = -(\mathcal{R} + \mathcal{B}^T \sigma\mathcal{X}\mathcal{B})^{-1}\mathcal{B}^T \sigma\mathcal{X}\mathcal{A}, \tag{2.3}$$

where $\mathcal{X}$ is the unique symmetric nonnegative definite periodic stabilizing solution to the *discrete periodic Riccati equation* (DPRE) (Bittanti 1991)

$$\mathcal{X} = \mathcal{Q} + \mathcal{A}^T \sigma\mathcal{X}\mathcal{A} - \mathcal{A}^T \sigma\mathcal{X}\mathcal{B}(\mathcal{R} + \mathcal{B}^T \sigma\mathcal{X}\mathcal{B})^{-1}\mathcal{B}^T \sigma\mathcal{X}\mathcal{A}. \tag{2.4}$$

A direct solution of the above equation can be computed by using the periodic Schur method (Hench and Laub 1994, Sreedhar and Van Dooren 1993). An alternative computational approach, which can be useful also for the iterative refinement of a computed approximate solution, is to solve the above equation using a Newton-type iterative technique, analogously as in the standard case (Hewer 1971). Let $\mathcal{X}^{(i)}$ be the approximation of $\mathcal{X}$ at the i-th iteration step and let $\mathcal{F}^{(i)}$ be the corresponding state feedback computed with (2.3). The Newton step to compute a new approximation $\mathcal{X}^{i+1}$ of $\mathcal{X}$ consists in solving the following *reverse-time discrete periodic Lyapunov equation* (RTDPLE)

$$\mathcal{X}^{(i+1)} = {\mathcal{A}^{(i)}}^T \sigma\mathcal{X}^{(i+1)}\mathcal{A}^{(i)} + \mathcal{Q}^{(i)}, \tag{2.5}$$

where $\mathcal{A}^{(i)} = \mathcal{A} + \mathcal{B}\mathcal{F}^{(i)}$ and $\mathcal{Q}^{(i)} = \mathcal{Q} + {\mathcal{F}^{(i)}}^T \mathcal{R}\mathcal{F}^{(i)}$. Provided an initial stabilizing periodic gain $\mathcal{F}^{(0)}$ is available, it can be shown under the usual assumptions (Bittanti *et al.* 1988) that for all $i > 0$ the computed gains $\mathcal{F}^{(i)}$ are stabilizing and the sequence $\mathcal{X}^{(i)}$ converges to the solution $\mathcal{X}$ of the DPRE (2.4). Thus to solve the DPRE (2.4) it is only necessary to solve repeatedly, until convergence, RTDPLEs of the form (2.5). The computation of an initial stabilizing state-feedback is addressed in the next section.

3. State Feedback Stabilization of Periodic Systems

To compute a stabilizing periodic feedback, we can solve a particular DPRE by setting for instance $\mathcal{Q} = 0$ and $\mathcal{R} = \mathcal{I}$. Because always it is possible to isolate the unstable part of a system by using an appropriate state space similarity transformation (Bojanczyk *et al.* 1992), we can assume that the system to be stabilized is completely unstable, that is, all characteristic multipliers lie outside of the unit circle. With the usual assumption of stabilizability, it follows that the stabilizing solution of the DPRE (2.4) satisfies $\mathcal{X} > 0$ and thus after simple matrix manipulations the DPRE can be reduced to the following *forward-time discrete periodic Lyapunov equation* (FTDPLE) satisfied by $\mathcal{Y} = \mathcal{X}^{-1}$

$$\sigma\mathcal{Y} = \mathcal{A}\mathcal{Y}\mathcal{A}^T - \mathcal{B}\mathcal{B}^T. \tag{3.1}$$

The corresponding stabilizing periodic feedback $\mathcal{F}$ can be computed with the formula

$$\mathcal{F} = -\mathcal{B}^T(\sigma\mathcal{Y} + \mathcal{B}\mathcal{B}^T)^{-1}\mathcal{A}. \tag{3.2}$$

If we intend to ensure for the characteristic values a desired stability degree β, with $0 < \beta < 1$, then we can compute the stabilizing feedback $\mathcal{F}$ from (3.1) and (3.2) with $\mathcal{A}$ and $\mathcal{B}$ replaced by $\mathcal{A}/\sqrt[K]{\beta}$ and $\mathcal{B}/\sqrt[K]{\beta}$, respectively. This technique can be also employed when the system has characteristic values on the unit circle. A similar approach to stabilize periodic systems has been also proposed by Sreedhar and Van Dooren (1994).

4. Optimal Output Feedback Periodic Control

Instead of the state feedback control law (2.2) we can try to use an optimal output feedback control law

$$u_k^* = F_k y_k \tag{4.1}$$

which minimizes the performance index

$$J = E\left\{\frac{1}{2}\sum_{k=0}^{\infty}[\, x_k^T Q_k x_k + u_k^T R_k u_k\,]\right\}, \tag{4.2}$$

where x_0 is a zero mean random vector with known covariance $G = \mathrm{cov}\,\{x_0\}$. For the solution of this problem in general, no closed form solutions can be found even for standard state space systems. Thus iterative search methods must be used to compute the optimizing periodic output feedback matrix F_k. For search methods based on gradient techniques it is necessary to evaluate for a given stabilizing periodic output feedback F_k the corresponding values of the cost functional (4.2) and of its gradient with respect to F_k. Explicit formulas for this purpose have been derived recently in (Varga and Pieters 1996). The computations involve the solution of two DPLEs: a RTDPLE

$$\mathcal{P} = \overline{\mathcal{A}}^T \sigma\mathcal{P}\overline{\mathcal{A}} + \overline{\mathcal{Q}}, \tag{4.3}$$

and a FTDPLE

$$\sigma\mathcal{S} = \overline{\mathcal{A}}\mathcal{S}\overline{\mathcal{A}}^T + \mathcal{G}, \tag{4.4}$$

where $\overline{\mathcal{A}} = \mathcal{A} + \mathcal{B}\mathcal{F}\mathcal{C}$, $\overline{\mathcal{Q}} = \mathcal{Q} + \mathcal{C}^T\mathcal{F}^T\mathcal{R}\mathcal{F}\mathcal{C}$, and $\mathcal{G} = \mathrm{diag}\,(0, 0, \ldots, G)$. The expressions for the function and its gradient are:

$$\begin{array}{rcl} J & = & \frac{1}{2}\mathrm{tr}\,(\sigma\mathcal{P}\mathcal{G}) \\ \nabla_{\mathcal{F}} J & = & (\mathcal{R}\mathcal{F}\mathcal{C} + \mathcal{B}^T\sigma\mathcal{P}\overline{\mathcal{A}})\mathcal{S}\mathcal{C}^T. \end{array}$$

For standard systems the above two Lyapunov equations can be solved efficiently with a computational cost which is marginally greater than the cost of solving a single Lyapunov equation. The preservation of this feature is even more stringent for the periodic case, because of much higher computational effort involved in solving a single periodic Lyapunov equation. This goal can be also achieved with the algorithms proposed in this paper.

5. Square-Root Balancing of Periodic Systems

The reachability Gramian of an exponentially stable time-variant system of the form (1.1) is defined as $P_k = \sum_{i=-\infty}^{k-1} \Phi_A(k, i+1)B_i B_i^T \Phi_A(k, i+1)^T$. For the periodic system (1.1) P_k is a periodic matrix and satisfies the FTDPLE

$$\sigma\mathcal{P} = \mathcal{A}\mathcal{P}\mathcal{A}^T + \mathcal{B}\mathcal{B}^T. \tag{5.1}$$

The system (1.1) is *uniformly controllable* iff $\mathcal{P} > 0$ (Halanay and Ionescu 1994). Similarly, the observability Gramian of the periodic system (1.1) defined as $Q_k = \sum_{i=k}^{\infty} \Phi_A(i, k)^T C_i^T C_i \Phi_A(i, k)$, satisfies the RTDPLE

$$\mathcal{Q} = \mathcal{A}^T\sigma\mathcal{Q}\mathcal{A} + \mathcal{C}^T\mathcal{C} \tag{5.2}$$

and the system (1.1) is *uniformly observable* iff $\mathcal{Q} > 0$ (Halanay and Ionescu 1994). For an exponentially stable periodic system the Gramians are nonnegative definite and thus can be expressed in Cholesky factorized forms $\mathcal{P} = \mathcal{S}^T\mathcal{S}$ and $\mathcal{Q} = \mathcal{R}^T\mathcal{R}$. These factors are useful for instance in determining a balancing transformation for a given uniformly controllable and uniformly observable periodic system.

Let T_k be a periodic invertible matrix of period K. Two periodic systems $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$ and $(\widetilde{\mathcal{A}}, \widetilde{\mathcal{B}}, \widetilde{\mathcal{C}}, \widetilde{\mathcal{D}})$ related by the Lyapunov transformation

$$(\widetilde{\mathcal{A}}, \widetilde{\mathcal{B}}, \widetilde{\mathcal{C}}, \widetilde{\mathcal{D}}) = (\sigma\mathcal{T}^{-1}\mathcal{A}\mathcal{T}, \sigma\mathcal{T}^{-1}\mathcal{B}, \mathcal{C}\mathcal{T}, \mathcal{D}) \tag{5.3}$$

are called *similar* and the transformation (5.3) is called a *Lyapunov similarity transformation*. The Gramians $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{Q}}$ of the transformed system $(\widetilde{\mathcal{A}}, \widetilde{\mathcal{B}}, \widetilde{\mathcal{C}}, \widetilde{\mathcal{D}})$ satisfy

$$\widetilde{\mathcal{P}} = \mathcal{T}^{-1}\mathcal{P}\mathcal{T}^{-T}, \quad \widetilde{\mathcal{Q}} = \mathcal{T}^T\mathcal{Q}\mathcal{T} \tag{5.4}$$

and thus their product satisfies

$$\widetilde{\mathcal{P}}\widetilde{\mathcal{Q}} = \mathcal{T}^{-1}\mathcal{P}\mathcal{Q}\mathcal{T} \ . \tag{5.5}$$

For a uniformly controllable and observable periodic system, in analogy with the standard case, we can determine $\mathcal{T}$ such that the transformed Gramians are equal and diagonal, that is

$$\widetilde{\mathcal{P}} = \widetilde{\mathcal{Q}} = \Sigma = \mathrm{diag}\,(\Sigma_0, \Sigma_1, \ldots, \Sigma_{K-1}),$$

where $\Sigma_k = \mathrm{diag}\,(\sigma_{k,1}, \sigma_{k,2}, \cdots, \sigma_{k,n})$. By using the singular value decomposition

$$\mathcal{R}\mathcal{S}^T = \mathcal{U}\Sigma\mathcal{V}^T,$$

$\mathcal{T}$ results as (Tombs and Postlethwaite 1987)

$$\mathcal{T} = \mathcal{S}^T\mathcal{V}\Sigma^{-1/2} \ .$$

Notice that $\mathcal{T}$ is computed exclusively on the basis of square-root information (the Cholesky factors of Gramians) and this has as direct effect the enhancement of numerical accuracy of computations. The quantities $\sigma_{k,i}, i = 1, \ldots, n$, representing the eigenvalues of the product P_kQ_k, are called the *Hankel-singular values* and the maximum of them over all k's defines the *Hankel-norm* of the given periodic system (Halanay and Ionescu 1994). Balancing transformation of this type could be useful for instance to perform model reduction of periodic systems in analogy with standard systems.

To compute the balancing transformation it is important to solve the periodic Lyapunov equations (5.1) and (5.2) directly for the Cholesky factors of the Gramians. Notice again the importance to determine the two Gramians with practically the same computational effort as that necessary to evaluate a single Gramian.

6. Solution of Indefinite DPLEs

In this section we discuss the solution of the RTDPLE

$$\mathcal{X} = \mathcal{A}^T\sigma\mathcal{X}\mathcal{A} + \mathcal{W} \tag{6.1}$$

and of the related dual FTDPLE

$$\sigma\mathcal{X} = \mathcal{A}\mathcal{X}\mathcal{A}^T + \mathcal{V}, \tag{6.2}$$

where W_k and V_k are symmetric periodic matrices. To solve the equations (6.1) and (6.2) we assume that the monodromy matrix $\Phi_A(K,0)$ has no reciprocal eigenvalues. This condition ensures the existence of a unique solution of both equations as it will be apparent below.

One class of existing numerical methods to solve periodic Lyapunov equations (Bittanti *et al.* 1988, Sreedhar and Van Dooren 1993) is based on reducing these problems to a single Lyapunov equation to compute a *periodic generator*, say X_0. It is easy to show that for the RTDPLE (6.1) X_0 satisfies the standard *discrete Lyapunov equation* (DLE)

$$X_0 = \Phi_A^T(K,0)X_0\Phi_A(K,0) + \sum_{j=0}^{K-1} \Phi_A^T(j,0)W_j\Phi_A(j,0),$$

while for the FTDPLE (6.2) X_0 fulfills the DLE

$$X_0 = \Phi_A(K,0)X_0\Phi_A^T(K,0) + \sum_{j=0}^{K-1} \Phi_A(K,j+1)V_j\Phi_A^T(K,j+1).$$

These equations can be solved by using standard methods (Barraud 1977, Kitagawa 1977), provided $\Phi_A(K,0)$ has no reciprocal eigenvalues. Once X_0 is determined, the rest of the solution is computed by backward- or forward-time recursion. The main drawback of such methods is the need to form explicitly matrix products and sums of matrix products.

An alternative approach discussed also by Sreedhar and Van Dooren (1993) is to solve the periodic Lyapunov equation (6.2) as a particular periodic Riccati equation. In this approach the construction of products is avoided but the method has a substantially higher computational complexity than usually necessary to solve DPLEs by using direct solution methods like those proposed in this paper. Among several computational variants described in the aforementioned work, even the most efficient one has a too high computational cost because the need to simultaneously reorder the diagonal blocks of K pairs of $2n$-th order upper triangular matrices by using unitary similarity transformations. This computation alone requires roughly twice as much operations than those necessary to solve a DPLE by using a direct solution method.

We describe now a new computational approach which essentially parallels the direct solution methods available for standard systems (Barraud 1977, Kitagawa 1977). The key role in the new method plays the recent discovery of the *periodic Schur decomposition* (PSD) of a cyclic matrix product and of the corresponding algorithms for its computation (Bojanczyk *et al.* 1992, Hench and Laub 1994). According to Bojanczyk *et al.* (1992), given the matrices A_k, $k = 0,1,\ldots,K-1$, there exist orthogonal matrices Z_k, $k = 0,1,\ldots,K-1$ such that $\widetilde{A}_{K-1} = Z_0^T A_{K-1} Z_{K-1}$ is in *real Schur form* (RSF) and the matrices $\widetilde{A}_k = Z_{k+1}^T A_k Z_k$ for $k = 0,\ldots,K-2$ are upper triangular. Thus by using the PSD algorithm, we can determine the orthogonal matrices Z_k, $k = 0,\ldots,K-1$ to reduce the cyclic product $A_{K-1}\cdots A_1 A_0$ to the RSF without forming explicitly this product.

The transformation to compute the PSD of the product $A_{K-1}\cdots A_1 A_0$ is useful to simplify the solution of both DPLEs (6.1) and (6.2). By using the script notation for

periodic matrices the transformation to determine the PSD can expressed as an orthogonal Lyapunov transformation $\widetilde{\mathcal{A}} = \sigma \mathcal{Z}^T \mathcal{A} \mathcal{Z}$. By multiplying the equation (6.1) with $\mathcal{Z}^T$ from left and with $\mathcal{Z}$ from right, one obtains

$$\widetilde{\mathcal{X}} = \widetilde{\mathcal{A}}^T \sigma \widetilde{\mathcal{X}} \widetilde{\mathcal{A}} + \widetilde{\mathcal{W}}, \tag{6.3}$$

where $\widetilde{\mathcal{X}} = \mathcal{Z}^T \mathcal{X} \mathcal{Z}$ and $\widetilde{\mathcal{W}} = \mathcal{Z}^T \mathcal{W} \mathcal{Z}$. Notice that by this transformation the resulted transformed equations (6.3) have exactly the same form as the original ones in (6.1). After solving these equations for $\widetilde{\mathcal{X}}$, the solution of (6.1) results as $\mathcal{X} = \mathcal{Z} \widetilde{\mathcal{X}} \mathcal{Z}^T$. The same transformation technique can be used to reduce the dual FTDPLE (6.2) to a simpler form. After multiplying it with $\sigma \mathcal{Z}^T$ from left and with $\sigma \mathcal{Z}$ from right one obtains

$$\sigma \widetilde{\mathcal{X}} = \widetilde{\mathcal{A}} \widetilde{\mathcal{X}} \widetilde{\mathcal{A}}^T + \widetilde{\mathcal{V}}, \tag{6.4}$$

where $\widetilde{\mathcal{V}} = \sigma \mathcal{Z}^T \mathcal{V} \sigma \mathcal{Z}$. Notice that only one computation of the PSD of the cyclic product $A_{K-1} \cdots A_1 A_0$ is necessary to compute the solutions of both (6.1) and (6.2). This aspect is important to enhance the efficiency of computations involving the solutions of both equations, as for instance in the solution of the periodic optimal output feedback problem or in computing balancing transformations.

We discuss now the solution of the reduced equations, where excepting $\widetilde{A}_{K-1}$ which is in a RSF, all other matrices $\widetilde{A}_k$, $k = 0, \ldots, K-2$ are upper triangular. To simplify the notations, in what follows we assume that the coefficient matrices of the original equations (6.1) are already in the reduced forms corresponding to the PSD.

Consider the partitioning of $\mathcal{A}$ according to the PSD of the product $A_{K-1} \cdots A_1 A_0$

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} & \cdots & \mathcal{A}_{1\bar{n}} \\ 0 & \mathcal{A}_{22} & \cdots & \mathcal{A}_{2\bar{n}} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{A}_{\bar{n}\bar{n}} \end{bmatrix}. \tag{6.5}$$

Let us partition analogously the symmetric matrices $\mathcal{X}$ and $\mathcal{W}$

$$\mathcal{X} = \begin{bmatrix} \mathcal{X}_{11} & \mathcal{X}_{12} & \cdots & \mathcal{X}_{1\bar{n}} \\ \mathcal{X}_{21} & \mathcal{X}_{22} & \cdots & \mathcal{X}_{2\bar{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{X}_{\bar{n}1} & \mathcal{X}_{\bar{n}2} & \cdots & \mathcal{X}_{\bar{n}\bar{n}} \end{bmatrix}, \quad \mathcal{W} = \begin{bmatrix} \mathcal{W}_{11} & \mathcal{W}_{12} & \cdots & \mathcal{W}_{1\bar{n}} \\ \mathcal{W}_{21} & \mathcal{W}_{22} & \cdots & \mathcal{W}_{2\bar{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{W}_{\bar{n}1} & \mathcal{W}_{\bar{n}2} & \cdots & \mathcal{W}_{\bar{n}\bar{n}} \end{bmatrix}.$$

From (6.1) follows that the (r,l)-th block $\mathcal{X}_{rl}$ satisfies the DPSE

$$\mathcal{X}_{rl} = \mathcal{A}_{rr}^T \sigma \mathcal{X}_{rl} \mathcal{A}_{ll} + \mathcal{M}_{rl},$$

where

$$\mathcal{M}_{rl} = \mathcal{W}_{rl} + \sum_{i=1}^{r} \mathcal{A}_{ir}^T \left(\sum_{j=1}^{l-1} \sigma \mathcal{X}_{ij} \mathcal{A}_{jl} \right) + \left(\sum_{i=1}^{r-1} \mathcal{A}_{ir}^T \sigma \mathcal{X}_{il} \right) \mathcal{A}_{ll}.$$

By exploiting the symmetry of the solution matrix $\mathcal{X}$, the above equations can be solved successively for $\mathcal{X}_{11}$, $\mathcal{X}_{21}$, ..., $\mathcal{X}_{\bar{n}1}$, $\mathcal{X}_{22}$, ..., $\mathcal{X}_{\bar{n}2}$, ..., $\mathcal{X}_{\bar{n}\bar{n}}$. The following procedure can serve for an efficient implementation of the proposed solution method:

Algorithm 1. *Periodic Schur Method to Solve the RTDPLE.*

Compute the orthogonal $\mathcal{Z}$ to reduce $\Phi_A(K,0)$ to the PSD.

$\mathcal{A} \leftarrow \sigma\mathcal{Z}^T\mathcal{A}\mathcal{Z}$, $\mathcal{W} \leftarrow \mathcal{Z}^T\mathcal{W}\mathcal{Z}$.

for $l = 1, \ldots, \bar{n}$

$\mathcal{F}_i = \sum_{j=1}^{l-1} \sigma\mathcal{X}_{ij}\mathcal{A}_{jl} \quad (i = 1, \ldots, l-1)$

for $r = l, \ldots, \bar{n}$

$\mathcal{F}_r = \sum_{j=1}^{l-1} \sigma\mathcal{X}_{rj}\mathcal{A}_{jl}$,

$\mathcal{M} = \mathcal{W}_{rl} + \sum_{i=1}^{r} \mathcal{A}_{ir}^T\mathcal{F}_i + (\sum_{i=1}^{r-1} \mathcal{A}_{ir}^T\sigma\mathcal{X}_{il})\mathcal{A}_{ll}$

Solve $\mathcal{X}_{rl} = \mathcal{A}_{rr}^T\sigma\mathcal{X}_{rl}\mathcal{A}_{ll} + \mathcal{M}$; put $\mathcal{X}_{lr} = \mathcal{X}_{rl}$

end

end

$\mathcal{X} \leftarrow \mathcal{Z}\mathcal{X}\mathcal{Z}^T$.

With the same partitions of matrices A_k, X_k and an analogous partition of V_k, we obtain from (6.2) the following DPSE satisfied by $\mathcal{X}_{rl}$

$$\sigma\mathcal{X}_{rl} = \mathcal{A}_{rr}\mathcal{X}_{rl}\mathcal{A}_{ll}^T + \mathcal{M}_{rl},$$

where

$$\mathcal{M}_{rl} = \mathcal{V}_{rl} + \sum_{i=r}^{\bar{n}} \mathcal{A}_{ri}\left(\sum_{j=l+1}^{\bar{n}} \mathcal{X}_{ij}\mathcal{A}_{lj}^T\right) + \left(\sum_{i=r+1}^{\bar{n}} \mathcal{A}_{ri}\mathcal{X}_{il}\right)\mathcal{A}_{ll}^T .$$

By starting from the bottom-right corner and exploiting the symmetry of the solution, we can compute successively $\mathcal{X}_{\bar{n}\bar{n}}$, $\mathcal{X}_{\bar{n}-1,\bar{n}}$, ..., $\mathcal{X}_{1\bar{n}}$, $\mathcal{X}_{\bar{n}-1,\bar{n}-1}$, ..., $\mathcal{X}_{11}$. The following procedure is a straightforward adaptation of Algorithm 1 to solve the FTDPLE:

Algorithm 2. *Periodic Schur Method to Solve the FTDPLE.*

Compute the orthogonal $\mathcal{Z}$ to reduce $\Phi_A(K,0)$ to the PSD.

$\mathcal{A} \leftarrow \sigma\mathcal{Z}^T\mathcal{A}\mathcal{Z}$, $\mathcal{V} \leftarrow \sigma\mathcal{Z}^T\mathcal{V}\sigma\mathcal{Z}$.

for $l = \bar{n}, \bar{n}-1, \ldots, 1$

$\mathcal{F}_i = \sum_{j=l+1}^{\bar{n}} \mathcal{X}_{ij}\mathcal{A}_{lj}^T \quad (i = l+1, \ldots, \bar{n})$

for $r = l, l-1, \ldots, 1$

$\mathcal{F}_r = \sum_{j=l+1}^{\bar{n}} \mathcal{X}_{rj}\mathcal{A}_{lj}^T$

$\mathcal{M} = \mathcal{V}_{rl} + \sum_{i=r}^{\bar{n}} \mathcal{A}_{ri}\mathcal{F}_i + (\sum_{i=r+1}^{\bar{n}} \mathcal{A}_{ri}\mathcal{X}_{il})\mathcal{A}_{ll}^T$

Solve $\sigma\mathcal{X}_{rl} = \mathcal{A}_{rr}\mathcal{X}_{rl}\mathcal{A}_{ll}^T + \mathcal{M}$; put $\mathcal{X}_{lr} = \mathcal{X}_{rl}$

end

end

$\mathcal{X} \leftarrow \mathcal{Z}\mathcal{X}\mathcal{Z}^T$.

Both Algorithm 1 and Algorithm 2 allow to overwrite $\mathcal{W}$ and $\mathcal{V}$, respectively, with the computed solution $\mathcal{X}$. Thus $(K+1)n^2 + O(Kn)$ additional storage locations are necessary

to implement these algorithms. If we neglect the effort to solve the low order DPSEs, then each algorithm performs about $N_{PSD}+4Kn^3$ *floating-point operations* (flops), where N_{PSD} is the number of *flops* necessary to determine the PSD and to accumulate the performed transformations. As a rough estimate of this value we can take $N_{PSD} = 10Kn^3$.

Remark. By interchanging the roles of $\mathcal{X}$ and $\sigma\mathcal{X}$, Algorithm 1 can be also used with minor modifications to solve the anticausal FTDPLE (Halanay and Ionescu 1994)

$$\sigma\mathcal{X} = \mathcal{A}^T\mathcal{X}\mathcal{A} + \mathcal{W}$$

and Algorithm 2 can be used to solve its dual anticausal RTDPLE

$$\mathcal{X} = \mathcal{A}\sigma\mathcal{X}\mathcal{A}^T + \mathcal{V}.$$

7. Solution of Low Order DPSE

The computation of $\mathcal{X}_{rl}$ in Algorithms 1 and 2 requires the solution of low order *discrete periodic Sylvester equations* (DPSEs) either of the form

$$Y_k = E_k^T Y_{k+1} F_k + G_k, \quad k = 0, \ldots, K-1; \quad Y_0 = Y_K \tag{7.1}$$

or of the form

$$Y_{k+1} = E_k Y_k F_k^T + G_k, \quad k = 0, \ldots, K-1; \quad Y_0 = Y_K \tag{7.2}$$

where $E_k \in \mathbb{R}^{n_1 \times n_1}$, $F_k \in \mathbb{R}^{n_2 \times n_2}$ and $G_k \in \mathbb{R}^{n_1 \times n_2}$ with $1 \le n_1, n_2 \le 2$.

We discuss two methods to solve these equations. The first method relies on rewriting the above equations with the help of Kronecker products as a system of $n_1 n_2 K$ simultaneous linear equations $Hy = g$, where the coefficient matrix H is a highly structured sparse matrix. Ignoring the sparse structure of H in solving $Hy = g$ leads, even for moderate values of K, to rather expensive computations. To exploit the structure of H, we can arrange by an appropriate grouping of unknowns in the vector y and by a suitable ordering of the equations, to obtain the coefficient matrix H in a block-Hessenberg form.

For the equation (7.1), let y and g be defined as

$$y =: \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{bmatrix} = \begin{bmatrix} \mathrm{vec}(Y_{K-1}) \\ \mathrm{vec}(Y_{K-2}) \\ \vdots \\ \mathrm{vec}(Y_0) \end{bmatrix}, \quad g =: \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_K \end{bmatrix} = \begin{bmatrix} \mathrm{vec}(G_{K-1}) \\ \mathrm{vec}(G_{K-2}) \\ \vdots \\ \mathrm{vec}(G_0) \end{bmatrix},$$

where the operator $\mathrm{vec}(\cdot)$ generates a vector from the stacked columns of a matrix. By using the $\otimes$ notation for Kronecker product, the corresponding H for $K > 1$ is given by

$$H =: \begin{bmatrix} H_{11} & & & H_{1,K} \\ H_{21} & H_{22} & & \\ & \ddots & \ddots & \vdots \\ & & H_{K,K-1} & H_{K,K} \end{bmatrix} = \begin{bmatrix} I & & & -F_{K-1}^T \otimes E_{K-1}^T \\ -F_{K-2}^T \otimes E_{K-2}^T & I & & \\ & \ddots & \ddots & \\ & & -F_0^T \otimes E_0^T & I \end{bmatrix}$$

where only the nonzero block elements of H are shown. For the equation (7.2) a coefficient matrix H with identical structure can be also constructed.

The standard Gaussian elimination method (Golub and Van Loan 1989) to solve the linear equation $Hy = g$ has two main steps. First the LU factorization of H is computed by using partial pivoting, to obtain $PH = LU$, where P is a permutation matrix, L is a unit lower triangular matrix and U is an upper triangular matrix. Then by using forward and backward substitutions the solution y is computed as $y = U^{-1}L^{-1}Pg$. For the particular structure of H above, it can be easily observed that the resulting L is block-bidiagonal and U has nonzero diagonal and supra-diagonal blocks as well a nonzero last block column. For the efficient solution of the equation $Hy = g$, it is advantageous to combine the LU factorization step with the solution steps by applying the elementary row transformations also to the right hand side g, such that in parallel with the computation of nonzero blocks of U we compute also $L^{-1}Pg$. The following algorithm can be used for this purpose:

Algorithm 3. *Kronecker-Products Method to Solve the DPSE.*

if $K = 1$, **then** solve $(I - F_0^T \otimes E_0^T)y = g$ and **exit**

Comment. Compute the block-LU factorization $PH = LU$ and $g \leftarrow L^{-1}Pg$.

for $i = 1, \ldots, K-1$

Compute the LU factorization $P_i \begin{bmatrix} H_{ii} & H_{i,i+1} \\ H_{i+1,i} & H_{i+1,i+1} \end{bmatrix} = L_i \begin{bmatrix} U_{ii} & U_{i,i+1} \\ 0 & U_{i+1,i+1} \end{bmatrix}$

Put $H_{i+1,i+1} = U_{i+1,i+1}$ and compute $\left[\begin{array}{cc|c} H_{i,K} & & g_i \\ H_{i+1,K} & & g_{i+1} \end{array}\right] := L_i^{-1} P_i \left[\begin{array}{cc|c} H_{i,K} & & g_i \\ H_{i+1,K} & & g_{i+1} \end{array}\right]$

end

Comment. Compute $y = U^{-1}g$ by backward substitution.

Solve $U_{KK}\, y_K = g_K$ and $U_{K-1,K-1}\, y_{K-1} = g_{K-1} - U_{K-1,K}\, y_K$.

for $i = K-2, \ldots, 1$

Solve $U_{ii} y_i = g_i - U_{i,i+1} y_{i+1} - U_{i,K}\, y_K$

end

The cost to solve the equation $Hy = g$ with the above algorithm can be roughly estimated as $(\frac{(2n_1n_2)^3}{3} + \frac{(n_1n_2)^3}{2})K \approx 3(n_1n_2)^3K$ *flops*. In the worst case of only 2 by 2 blocks in the PSD, this equation must be solved about $n^2/8$ times, so that the cost to solve the low order DPSEs amounts to about $24n^2K$. The implementation of Algorithm 3 requires about $12nK$ additional storage.

The second approach which we discuss is based on an iterative technique. We can split H as $H = I - N$, where

$$N = \begin{bmatrix} 0 & & & F_{K-1}^T \otimes E_{K-1}^T \\ F_{K-2}^T \otimes E_{K-2}^T & 0 & & \\ & \ddots & \ddots & \\ & & F_0^T \otimes E_0^T & 0 \end{bmatrix}. \tag{7.3}$$

It can be shown that the eigenvalues of this matrix are the K-th roots of those of the matrix $\Phi_F^T(K,0) \otimes \Phi_E^T(K,0)$. Notice that each of above monodromy matrices is 2 by 2

only if its eigenvalues are complex conjugated. Therefore we can in general determine the spectral radius of N as

$$\rho(N) = (|\lambda_1(\Phi_E(K,0))| \cdot |\lambda_1(\Phi_F(K,0))|)^{\frac{1}{K}},$$

where $\lambda_1(\cdot)$ is any of the eigenvalues of the respective matrix. Notice that for the existence of a unique solution of the DPSE a necessary and sufficient condition is $\rho(N) \neq 1$. If $\rho(N) < 1$ then the solution y of the equation $Hy = g$ can be computed by using the following iteration (Golub and Van Loan 1989)

$$y^{(i+1)} = Ny^{(i)} + g \tag{7.4}$$

initialized with any starting vector $y^{(0)}$. If $\rho(N) > 1$ we can use the iteration

$$y^{(i+1)} = N^{-1}(y^{(i)} - g) \tag{7.5}$$

which results by rewriting differently the equation $Hy = g$. Notice that N^{-1} can be computed explicitly as

$$N^{-1} = \begin{bmatrix} 0 & F_{K-2}^{-T} \otimes E_{K-2}^{-T} & & \\ & \ddots & \ddots & \\ & & 0 & F_0^{-T} \otimes E_0^{-T} \\ F_{K-1}^{-T} \otimes E_{K-1}^{-T} & & & 0 \end{bmatrix}. \tag{7.6}$$

The efficient implementation of the iterative method requires the full exploitation of the problem structure. In the following algorithm, the iterative computations are initialized by computing first a periodic generator say Y_0 of equation (7.1) by solving the corresponding ordinary discrete Sylvester equation

$$Y_0 = \Phi_E^T(K,0)Y_0\Phi_F(K,0) + \sum_{j=0}^{K-1} \Phi_E^T(j,0)G_j\Phi_F(j,0) \tag{7.7}$$

and then generating the rest of solution by forward or backward recursion. Moreover we replace the iterations based on either (7.4) or (7.5) by a more efficient accelerated iteration which also leads to a very efficient implementation of the proposed iterative solution method.

Algorithm 4. *Iterative Method to Solve the DPSE.*

Comment. Compute the initializing periodic generator Y_0.

$\Phi_E = I$, $\Phi_F = I$, $Q = 0$.

for $i = 0, \ldots, K-1$

$Q \leftarrow Q + \Phi_E^T G_i \Phi_F$, $\Phi_E \leftarrow E_i \Phi_E$, $\Phi_F \leftarrow F_i \Phi_F$.

end

$\rho = |\lambda_1(\Phi_E)| \cdot |\lambda_1(\Phi_F)|$; **if** $\rho = 1$, **then** *error* and **exit**

Solve $Y_0 = \Phi_E^T Y_0 \Phi_F + Q$; set $Y_K = Y_0$ and $i = 0$.

if $\rho < 1$, **then**

Comment. Compute the solution by backward recurrence.

while $\|Y_0 - Y_K\| > \varepsilon \|Y_K\|$ or $i < 1$

$Y_K = Y_0$

for $k = K-1, \ldots, 0$

Compute $Y_k = E_k^T Y_{k+1} F_k + G_k$.

end

$i \leftarrow i + 1$

end

else

Comment. Compute the solution by forward recurrence.

while $\|Y_0 - Y_K\| > \varepsilon \|Y_0\|$ or $i < 1$

$Y_0 = Y_K$

for $k = 0, \ldots, K-1$

Compute $Y_{k+1} = E_k^{-T}(Y_k - G_k) F_k^{-1}$.

end

$i \leftarrow i + 1$

end

end if

By examining the above algorithm it is clear that the iterates $\widetilde{Y}_k^{(i+1)}$ computed at the i-th iteration step are different from $Y_k^{(i+1)}$ resulted by using either (7.4) or (7.5). Notice that, instead of using (7.4) or (7.5), each computed new value $\widetilde{Y}_k^{(i+1)}$ in Algorithm 4 is immediately used to generate the next value $\widetilde{Y}_{k+1}^{(i+1)}$ in the same iteration step. This leads to a substantial acceleration of the convergence which can be explained easily by comparing $Y_0^{(i+1)}$ computed with the iteration formula (7.4) and $\widetilde{Y}_0^{(i+1)}$ computed with Algorithm 4 with the same $Y_k^{(i)}$, $k = 0, 1, \ldots, K-1$. In the former case we have

$$Y_0^{(i+1)} = E_0^T Y_1^{(i)} F_0 + G_0$$

while in the latter case

$$\widetilde{Y}_0^{(i+1)} = \Phi_E^T(K,0)Y_0^{(i)}\Phi_F(K,0) + \sum_{j=0}^{K-1} \Phi_E^T(j,0)G_j\Phi_F(j,0).$$

It is easy to observe that the value $\widetilde{Y}_0^{(i+1)}$ produced by Algorithm 4 in one complete iteration is equal to $Y_0^{(i+K)}$, the value computed after K iterations by using formula (7.4). Experimentally it was observed that when using the initialization (7.7), 2-3 iterations in Algorithm 4 are always sufficient to attain the limiting accuracy solution.

Because this method can also be seen as repetitive application of the periodic generator technique, it is interesting to see what happens if we employ this technique without tacking into account the stability aspect. As a simple example consider the equation (7.2) with $K = 30$, $E_k = F_k = 2.1$ and $G_k = -3.41$ for all k's. The exact solution of this equation is $Y_k = 1$ for all k's. The condition number of matrix H is of order of unity, and therefore the solution can be computed with high accuracy using Algorithm 3. Indeed, by computing the solution with Algorithm 3 in IEEE double precision arithmetic, the maximum error over all k's in the computed solution $\widehat{Y}_k$ was equal to the machine precision $\epsilon_M = 2.22 \cdot 10^{-16}$. We also computed $\widehat{Y}_0$ by solving

$$Y_0 = \Phi_E(K,0)Y_0\Phi_F^T(K,0) + \sum_{j=0}^{K-1} \Phi_E(K,j+1)G_j\Phi_F^T(K,j+1)$$

with a quite satisfactory accuracy $|\widehat{Y}_0 - Y_0| = 1.88 \cdot 10^{-15}$. The computation of the rest of solution by using forward recurrence based on (7.1) led to an unexpectedly large error in Y_{K-1}, namely $|\widehat{Y}_{K-1} - Y_{K-1}| = 8.89 \cdot 10^3$. This error can be easily explained by observing that an initial error $e_0 = \widehat{Y}_0 - Y_0$ in Y_0 can be magnified by repeated multiplications to a value about $E_{K-2}...E_1|e_0|F_1...F_{K-2}$. In our case this amounts to $9.21 \cdot 10^3$ which agrees well with the error resulted from the actual computations. Because the spectral radius $\rho = |\lambda_1(\Phi_E(K,0))| \cdot |\lambda_1(\Phi_F(K,0))| > 1$, the following backward recurrence formulas corresponding to (7.2) must be used as given below

$$Y_k = E_k^{-1}(Y_{k+1} - G_k)F_k^{-T}. \tag{7.8}$$

With the above recurrence, the rest of solution was determined with very high accuracy. More precisely, the error in the next computed component Y_{K-1} was $4.44 \cdot 10^{-16}$ and the rest of solution was computed exactly. By performing another complete iteration, we obtained all Y_k exactly. It is interesting to notice that the above aspect was not mentioned in the papers discussing the method of periodic generators (Bittanti *et al.* 1988, Sreedhar and Van Dooren 1993).

The number of operations to solve K DPSEs in m iterations with Algorithm 4 is $K[(n_1 + n_2)(n_1^2 + n_2^2) + m(n_1^2 n_2 + n_1 n_2^2)]$. In the worst case of only 2 by 2 blocks in the PSD, the total cost to solve the low order discrete Sylvester equations amounts to about $(4 + 2m)n^2K$. Because usually one or two iterations are sufficient to achieve the limiting accuracy with the proposed iterative method, Algorithm 4 is more efficient than Algorithm 3 based on LU decomposition. An additional advantage of Algorithm 4 is that it needs practically no additional memory to perform the computations. The iterative

method is especially well suited if the periodic system is uniformly stable or uniformly unstable (that is each A_k has eigenvalues only inside of or only outside of the unit circle). In these cases, the accuracy achieved with the iterative approach is systematically better than the accuracy resulted with Algorithm 3.

Combinations of the two methods can be imagined in order to exploit the advantages of both methods. For instance it is possible to compute the periodic generator Y_0 very accurately using the LU factorization with partial pivoting. Then instead of using the back substitution phase of Algorithm 3, it is possible to compute the solution by forward or backward recurrence as in Algorithm 4. A complete single iteration is usually sufficient to attain the limiting accuracy solution. Notice that in this case there is no need to store the resulting nonzero blocks of the U factor and thus practically no additional storage is necessary for implementation. It is also possible to determine simultaneously several periodic generators by replacing several Sylvester equations for successive values of k with a single equation. This "multiple-shooting" approach is advantageous especially for very large values of K because of increased performance resulted by reducing the number of equations to be solved simultaneously in Algorithm 3.

8. Rounding Errors and Estimation of Condition Number

The DPLEs (6.1) and (6.2) can be represented as standard linear systems

$$Tx = w, \qquad T^T x = v,$$

respectively, where

$$x = \begin{bmatrix} \mathrm{vec}(X_0) \\ \mathrm{vec}(X_1) \\ \vdots \\ \mathrm{vec}(X_{K-1}) \end{bmatrix}, \quad w = \begin{bmatrix} \mathrm{vec}(W_0) \\ \mathrm{vec}(W_1) \\ \vdots \\ \mathrm{vec}(W_{K-1}) \end{bmatrix}, \quad v = \begin{bmatrix} \mathrm{vec}(V_0) \\ \mathrm{vec}(V_1) \\ \vdots \\ \mathrm{vec}(V_{K-1}) \end{bmatrix}$$

and for $K > 1$

$$T = \begin{bmatrix} I & -A_0^T \otimes A_0^T & & \\ & \ddots & \ddots & \\ & & I & -A_{K-2}^T \otimes A_{K-2}^T \\ -A_{K-1}^T \otimes A_{K-1}^T & & & I \end{bmatrix}.$$

The matrix T is nonsingular if $\Phi_A(K,0)$ has no reciprocal eigenvalues, that is, any two eigenvalues λ_i, λ_j of $\Phi_A(K,0)$ obey $\lambda_i \lambda_j \neq 1$.

We can extend in a straightforward way the rounding error analysis technique for the standard Sylvester equation $AX + XB = C$ done by Golub *et al.* (1979) to our algorithms. Let $\widehat{\mathcal{X}}$ be the solution of the DPLE (6.1) computed by Algorithm 1 using Algorithm 3 as subprocedure. Algorithm 1 is essentially a forward substitution for linear systems applied to the reduced DPLE obtained after applying an orthogonal similarity transformation. By using standard error analysis for orthogonal transformations and linear systems (Wilkinson 1965) combined with standard perturbation results for linear systems (Higham 1996) we obtain finally that

$$\frac{\|\mathcal{X} - \widehat{\mathcal{X}}\|_F}{\|\mathcal{X}\|_F} \leq \epsilon \frac{2\,\mathrm{cond}(T)}{1 - \epsilon\,\mathrm{cond}(T)}\,, \tag{8.1}$$

where

$$\mathrm{cond}(T) = \|T^{-1}\|_2 \|T\|_2 \tag{8.2}$$

and ϵ is a quantity of the order of the unit roundoff u. To deduce (8.1) we tacitly assumed that $\epsilon\,\mathrm{cond}(T) < 1$. According to the terminology of Higham (1996, page 142), it follows that the proposed method is *normwise forward stable*. A similar result holds also for Algorithm 2 used in conjunction with Algorithm 3.

It is also possible to show that the residual $\mathcal{R} = \widehat{\mathcal{X}} - \mathcal{A}^T\sigma\widehat{\mathcal{X}}\mathcal{A} - \mathcal{W}$ satisfies $\|\mathcal{R}\|_F \leq \epsilon\|T\|_2\|\mathcal{X}\|_F$ and thus the relative residual is guaranteed to be bounded by a modest multiple of the unit roundoff. Because of the highly structured problem, a small unstructured residual does not automatically implies small backward errors as in the case of linear equations (Higham 1996). An attempt to derive an approximation to the backward error can be done by using a linearization technique similar to that applied to the equation characterizing the optimal perturbations for the standard DLE (Ghavimi and Laub 1995).

In analogy with the standard case (Gahinet *et al.* 1990), the sensitivity of the DPLE (6.1) depends on the quantity

$$\begin{aligned} \mathrm{sep}\,(\mathcal{A},\mathcal{I}) &= \min_{\|x\|_F\,=\,1} \|Tx\|_2 = \frac{1}{\|T^{-1}\|_2} = \sigma_{\min}(T) \\ &= \min_{\|\mathcal{X}\|_F\,=\,1} \|\mathcal{A}^T\sigma\mathcal{X}\mathcal{A} - \mathcal{X}\|_F, \end{aligned}$$

where $\sigma_{\min}(T)$ is the minimum singular value of T. The quantity $\mathrm{sep}\,(\mathcal{A},\mathcal{I})$ can be efficiently estimated by the algorithm described in (Higham 1988) to estimate the 1-norm of T^{-1}. Notice that the true reciprocal 1-norm of T^{-1} differs at most by a factor $n\sqrt{K}$ from $\sigma_{\min}(T)$. To estimate the 1-norm of T^{-1} a few (say 4 or 5) linear systems with coefficient matrices T and T^T are solved, which is equivalent to solve a few DPLEs (6.1) and (6.2). It is clear that

$$\mathrm{sep}\,(\mathcal{A},\mathcal{I}) = \mathrm{sep}\,(\widetilde{\mathcal{A}},\mathcal{I}),$$

where $\widetilde{\mathcal{A}} = \sigma\mathcal{Z}^T\mathcal{A}\mathcal{Z}$ and $\mathcal{Z}$ is the orthogonal periodic Lyapunov transformation used to compute the PSD of the cyclic product $A_{K-1}\cdots A_1A_0$. Thus it is advantageous to estimate $\mathrm{sep}\,(\widetilde{\mathcal{A}}^T,\mathcal{I})$ by solving the reduced equations (6.3) and (6.4).

The quantity $\mathrm{cond}(T)$ from (8.2) can be approximated as

$$\mathrm{cond}\,(T) \approx \frac{\|\mathcal{A}\|_F^2 + 1}{\mathrm{sep}\,(\mathcal{A},\mathcal{I})}$$

and can be considered as the *condition number* of the DPLE (6.1). An approximate error bound for the computed solution can be evaluated in software implementations with the expression

$$\frac{\|\mathcal{X} - \widehat{\mathcal{X}}\|_F}{\|\mathcal{X}\|_F} \leq \texttt{ERRBND} = u\frac{\|\mathcal{A}\|_F^2 + 1}{\mathrm{sep}\,(\mathcal{A},\mathcal{I})}.$$

For the DPLE (6.2) $\mathrm{sep}\,(\mathcal{A}^T,\mathcal{I})$ and the corresponding condition number are defined and can be computed in a completely similar way.

9. Solution of Nonnegative DPLEs

In this section we discuss two methods which can be employed to solve the nonnegative definite DPLE

$$\mathcal{U}^{\mathcal{T}}\mathcal{U} = \mathcal{A}^{\mathcal{T}}\sigma\mathcal{U}^{\mathcal{T}}\sigma\mathcal{U}\mathcal{A} + \mathcal{R}^{\mathcal{T}}\mathcal{R} \tag{9.1}$$

directly for the Cholesky factor $\mathcal{U}$. For the existence of the solution we assume that the monodromy matrix $\Phi_A(K,0)$ has only eigenvalues in the interior of the unit circle.

The first method which can be devised is to solve (9.1) by an iterative method. Let U_0 be the solution of the following standard DLE

$$U_0^T U_0 = \Phi_A^T(K,0)U_0^T U_0 \Phi_A(K,0) + \sum_{j=0}^{K-1} \Phi_A^T(j,0)R_j^T R_j \Phi_A(j,0),$$

which can be computed by using the method of Hammarling (1982). Because of matrix multiplications and additions it is likely that U_0 is quite far from the limiting accuracy solution. However, we can expect that U_0 is still a good approximation of the actual solution and by recursion we can compute the rest of Cholesky factors U_k, $k = K\text{–}1, \ldots, 1$. Because the monodromy matrix $\Phi_A(K,0)$ has only eigenvalues in the interior of the unit circle, we can perform several iterations to improve the accuracy of the initial periodic solution U_k. The following simple algorithm can be used for implementation purposes.

Algorithm 5. *Iterative Method to Solve the Nonnegative RTDPLE.*

Comment. Compute the initializing periodic Cholesky factor U_0.

$\Phi = I$, $V = 0$.

for $i = 0, \ldots, K\text{–}1$

$$V^T V \leftarrow \begin{bmatrix} V \\ R_i\Phi \end{bmatrix}^T \begin{bmatrix} V \\ R_i\Phi \end{bmatrix}, \; \Phi \leftarrow A_i\Phi.$$

end

Solve $U_0^T U_0 = \Phi^T U_0^T U_0 \Phi + V^T V$; set $U_K = U_0$ and $i = 0$.

Comment. Compute the solution by backward recurrence.

while $\|U_0^T U_0 - U_K^T U_K\| > \varepsilon \|U_K^T U_K\|$ or $i < 1$

$U_K = U_0$

for $k = K\text{–}1, \ldots, 0$

Compute the Cholesky factorization $U_k^T U_k = \begin{bmatrix} R_k \\ U_{k+1}A_k \end{bmatrix}^T \begin{bmatrix} R_k \\ U_{k+1}A_k \end{bmatrix}$.

end

$i \leftarrow i + 1$

end

It is important to notice that this algorithm works exclusively with the Cholesky factors and relies heavily on updating techniques of the Cholesky factorization (Gill *et al.* 1974). To attain the limiting accuracy solution usually at most two iterations are sufficient.

Computational enhancements of this approach based on using the PSD of the monodromy matrix $\Phi_A(K,0)$ are discussed by Sreedhar and Van Dooren (1993). This algorithm is primarily useful to solve low order nonnegative RTDPLEs in the second method discussed below.

The second approach which we propose essentially parallels the method of Hammarling (1982) for standard systems including also some enhancements of this method proposed recently by Penzl (1996). This approach is based on a transformation technique similar to that described to solve indefinite DPLEs. Let $\mathcal{Z}$ be the orthogonal Lyapunov transformation to compute the PSD of the monodromy matrix $\Phi_A(K,0)$ and define $\tilde{\mathcal{A}} = \sigma\mathcal{Z}^T\mathcal{A}\mathcal{Z}$ and the upper triangular $\widetilde{\mathcal{R}}$ such that $\widetilde{\mathcal{R}}^T\widetilde{\mathcal{R}} = \mathcal{Z}^T\mathcal{R}^T\mathcal{R}\mathcal{Z}$. The equation (9.1) becomes after premultiplication with $\mathcal{Z}^T$ and postmultiplication with $\mathcal{Z}$

$$\tilde{\mathcal{U}}^T\tilde{\mathcal{U}} = \tilde{\mathcal{A}}^T\sigma\tilde{\mathcal{U}}^T\sigma\tilde{\mathcal{U}}\tilde{\mathcal{A}} + \widetilde{\mathcal{R}}^T\widetilde{\mathcal{R}}, \tag{9.2}$$

where $\tilde{\mathcal{U}} = \mathcal{U}\mathcal{Z}$. After solving this reduced equation for $\tilde{\mathcal{U}}$, the solution of (9.1) results as $\mathcal{U} = \tilde{\mathcal{U}}\mathcal{Z}^T$. In order to simplify the notations, we assume in what follows that the equation (9.1) is already in the reduced form (9.2) and $\mathcal{R}$ is upper triangular.

Let partition the matrices $\mathcal{A}$, $\mathcal{U}$ and $\mathcal{R}$ analogously as

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ 0 & \mathcal{A}_{22} \end{bmatrix}, \quad \mathcal{U} = \begin{bmatrix} \mathcal{U}_{11} & \mathcal{U}_{12} \\ 0 & \mathcal{U}_{22} \end{bmatrix}, \quad \mathcal{R} = \begin{bmatrix} \mathcal{R}_{11} & \mathcal{R}_{12} \\ 0 & \mathcal{R}_{22} \end{bmatrix},$$

where the upper left blocks are n_1 by n_1 matrices ($n_1 = 1$ or 2). Assuming $\mathcal{U}_{11}$ nonsingular we can derive recursive formulas which are analogous with those in the standard case. By rewriting (9.1) for the partitioned matrices, we obtain

$$\begin{aligned} \mathcal{U}_{11}^T\mathcal{U}_{11} &= \mathcal{A}_{11}^T\sigma\mathcal{U}_{11}^T\sigma\mathcal{U}_{11}\mathcal{A}_{11} + \mathcal{R}_{11}^T\mathcal{R}_{11} \\ \mathcal{U}_{12} &= \mathcal{M}_1^T\sigma\mathcal{U}_{12}\mathcal{A}_{22} + \mathcal{M}_1^T\sigma\mathcal{U}_{11}\mathcal{A}_{12} + \mathcal{M}_2^T\mathcal{R}_{12} \\ \mathcal{U}_{22}^T\mathcal{U}_{22} &= \mathcal{A}_{22}^T\sigma\mathcal{U}_{22}^T\sigma\mathcal{U}_{22}\mathcal{A}_{22} + \mathcal{R}_{22}^T\mathcal{R}_{22} + \mathcal{Y}^T\mathcal{Y} \end{aligned} \tag{9.3}$$

where $\mathcal{M}_1 = \sigma\mathcal{U}_{11}\mathcal{A}_{11}\mathcal{U}_{11}^{-1}$, $\mathcal{M}_2 = \mathcal{R}_{11}\mathcal{U}_{11}^{-1}$, and $\mathcal{Y}$ is defined by the periodic matrix

$$Y_k = N_k \begin{bmatrix} \mathcal{R}_{12} \\ \sigma\mathcal{U}_{11}\mathcal{A}_{12} + \sigma\mathcal{U}_{12}\mathcal{A}_{22} \end{bmatrix}_{(k)}, \quad k = 1, \ldots, K-1$$

with each N_k satisfying

$$P_k = I_{2n_1} - \begin{bmatrix} \mathcal{M}_2 \\ \mathcal{M}_1 \end{bmatrix}_{(k)} \begin{bmatrix} \mathcal{M}_2 \\ \mathcal{M}_1 \end{bmatrix}_{(k)}^T = N_k^T N_k.$$

It can be shown that $P_k = P_k^2 \geq 0$ and $\operatorname{rank} P_k = n_1$. Moreover, because

$$P_k \begin{bmatrix} \mathcal{M}_2 \\ \mathcal{M}_1 \end{bmatrix}_{(k)} = 0,$$

N_k can be computed as $N_k = Q_{2k}^T$ from the QR-decomposition

$$\begin{bmatrix} \mathcal{M}_2 \\ \mathcal{M}_1 \end{bmatrix}_{(k)} = \begin{bmatrix} Q_{1k} & Q_{2k} \end{bmatrix} \begin{bmatrix} V_k \\ 0 \end{bmatrix}.$$

Thus by solving successively the first and second equation from (9.3) for $\mathcal{U}_{11}$ and $\mathcal{U}_{12}$, respectively, it remains to solve the third equation of lower order $n - n_1$ for $\mathcal{U}_{22}$ which, after updating a Cholesky factorization, is in the standard form. The following algorithm applies systematically this technique on the partitioned matrix $\mathcal{A}$ in (6.5) resulted by computing the PSD of the product $A_{K-1} \cdots A_1 A_0$. We also assume compatible partitioning of the matrices $\mathcal{U}$ and $\mathcal{R}$.

Algorithm 6. *Periodic Schur Method to Solve the nonnegative RTDPLE.*

Compute the orthogonal $\mathcal{Z}$ to reduce $\Phi_A(K,0)$ to the PSD; $\mathcal{A} \leftarrow \sigma\mathcal{Z}^T\mathcal{A}\mathcal{Z}$.

Compute the QR-decomposition $\mathcal{R}\mathcal{Z} = \mathcal{Q}\widetilde{\mathcal{R}}$; put $\mathcal{R} = \widetilde{\mathcal{R}}$.

for $r = 1, \ldots, \bar{n}$

Solve $\mathcal{U}_{rr}^T\mathcal{U}_{rr} = \mathcal{A}_{rr}^T\sigma\mathcal{U}_{rr}^T\sigma\mathcal{U}_{rr}\mathcal{A}_{rr} + \mathcal{R}_{rr}^T\mathcal{R}_{rr}$ (use Algorithm 5)

$\mathcal{M}_1 = \sigma\mathcal{U}_{rr}\mathcal{A}_{rr}\mathcal{U}_{rr}^{-1}$, $\mathcal{M}_2 = \mathcal{R}_{rr}\mathcal{U}_{rr}^{-1}$

Compute the QR-decompositions $\begin{bmatrix} \mathcal{M}_2 \\ \mathcal{M}_1 \end{bmatrix}_{(k)} = \begin{bmatrix} Q_k & N_k^T \end{bmatrix} \begin{bmatrix} V_k \\ 0 \end{bmatrix}$ $(k = 0, \ldots, K-1)$

for $j = r+1, \ldots, \bar{n}$

$\mathcal{G}_j = \sum_{i=r}^{j-1} \sigma\mathcal{U}_{ri}\mathcal{A}_{ij}$

Solve $\mathcal{U}_{rj} = \mathcal{M}_1^T\sigma\mathcal{U}_{rj}\mathcal{A}_{jj} + \mathcal{M}_1^T\mathcal{G}_j + \mathcal{M}_2^T\mathcal{R}_{rj}$ (use Algorithm 3 or 4)

$Y_{jk} = N_k \begin{bmatrix} \mathcal{R}_{rj} \\ \sigma\mathcal{U}_{rj}\mathcal{A}_{jj} + \mathcal{G}_j \end{bmatrix}_{(k)}$ $(k = 0, \ldots, K-1)$

end

for $i = r+1, \ldots, \bar{n}$

Compute the QR-decompositions $\begin{bmatrix} \mathcal{R}_{ii} \\ \mathcal{Y}_i \end{bmatrix}_{(k)} = Q_k \begin{bmatrix} W_k \\ 0 \end{bmatrix}$ $(k = 0, \ldots, K-1)$

$\begin{bmatrix} \mathcal{R}_{ij} \\ \mathcal{Y}_j \end{bmatrix}_{(k)} \leftarrow Q_k \begin{bmatrix} \mathcal{R}_{ij} \\ \mathcal{Y}_j \end{bmatrix}_{(k)}$ $(j = i, \ldots, \bar{n};\ \ k = 0, \ldots, K-1)$

end

end

Compute the QR-decomposition $\mathcal{U}\mathcal{Z}^T = \mathcal{Q}\widetilde{\mathcal{U}}$; put $\mathcal{U} = \widetilde{\mathcal{U}}$.

Special care is necessary to handle the cases when the diagonal blocks of $\mathcal{R}$ are zero or when the computed diagonal blocks $\mathcal{U}_{rr}$ are not invertible. Details on how to handle these cases in a numerically sound way are discussed by Hammarling (1982). An algorithm similar to Algorithm 6 can be easily devised to solve the nonnegative FTDPLE.

Algorithm 6 allows to overwrite $\mathcal{R}$ with the computed Cholesky factor $\mathcal{U}$ and thus its implementation requires $(K+1)n^2 + O(Kn)$ additional storage. Neglecting the costs to solve the low order DPLEs and DPSEs, Algorithm 6 performs about $N_{PSD} + 4Kn^3$ *flops*.

10. Conclusion

Numerically reliable computational algorithms have been proposed to solve various DPLEs. These algorithms represent extensions of similar algorithms to solve standard DLEs. Special methods have been developed to solve efficiently low order DPSEs and nonnegative DPLEs. All proposed algorithms are presented as implementable procedures suitable to exploit parallel computations. A set of LAPACK based Fortran routines have been implemented to compute the PSD and to solve four types of indefinite periodic Lyapunov equations. The implemented software was successfully used to evaluate gradients of linear-quadratic functionals for optimal periodic output feedback control (Varga and Pieters 1996).

References

Barraud, A. Y., 1977, Numerical algorithms to solve $A^TXA - X = Q$. *IEEE Trans. Autom. Control*, **22,** 883–885.

Bartels, R. H. and Stewart, G. W., 1972, Algorithm 432: Solution of the matrix equation AX+XB=C. *Comm. ACM*, **15,** 820–826.

Bittanti, S., 1991, The periodic Riccati equation. *In* S. Bittanti, A. J. Laub and J. C. Willems (eds), *The Riccati Equation*, Springer-Verlag, Berlin, pp. 127–162.

Bittanti, S., Colaneri, P. and Nicolao, G. D., 1988, The difference periodic Riccati equation for the periodic prediction problem. *IEEE Trans. Autom. Control*, **33,** 706–712.

Bojanczyk, A. W., Golub, G. and Van Dooren, P., 1992, The periodic Schur decomposition. Algorithms and applications. *In* F. T. Luk (ed.), *Proceedings SPIE Conference*, Vol. 1770, pp. 31–42.

Byers, R. and Rhee, N., 1995, Cyclic Schur and Hessenberg-Schur numerical methods for solving periodic Lyapunov and Sylvester equations. *Technical Report*, Dept. of Mathematics, Univ. of Missouri at Kansas City.

Gahinet, P. M., Laub, A. J., Kenney, C. S. and Hewer, G. A., 1990, Sensitivity of the stable discrete-time Lyapunov equation. *IEEE Trans. Autom. Control*, **35,** 1209–1217.

Ghavimi, A. R. and Laub, A. J., 1995, Residual bounds for discrete-time Lyapunov equations. *IEEE Trans. Autom. Control*, **40,** 1244–1249.

Gill, P. E., Golub, G. H., Murray, W. and Saunders, M. A., 1974, Methods for modifying matrix factorizations. *Math. Comput.*, **28,** 505–535.

Golub, G. H. and Van Loan, C. F., 1989, *Matrix Computations*, John Hopkins University Press, Baltimore.

Golub, G. H., Nash, S. and Van Loan, C., 1979, A Hessenberg–Schur method for the problem $AX + XB = C$. *IEEE Trans. Autom. Control*, **24,** 909–913.

Halanay, A. and Ionescu, V., 1994, *Time-Varying Discrete Linear Systems*, Birkhäuser.

Hammarling, S. J., 1982, Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA J. Numer. Anal.*, **2,** 303–323.

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

Hewer, G. A., 1971, An iterative technique for the computation of the steady state gains for the discrete optimal regulator. *IEEE Trans. Autom. Control.* **16,** 382-383.

Higham, N. J., 1988, FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation (Algorithm 674). *ACM Trans. Math. Software*, **14,** 381–396.

Higham, N. J., 1996, *Accuracy and Stability of Numerical Algorithms*, SIAM.

Kitagawa, G., 1977, An algorithm for solving the matrix equation $X = FXF' + S$. *Int. J. Control*, **25,** 745–753.

Penzl, T., 1996, Numerical solution of generalized Lyapunov equations. *Preprint SFB393/96-02*, Technical University Chemnitz.

Pittelkau, M. E., 1993, Optimal periodic control for spacecraft pointing and attitude determination. *J. of Guidance, Control, and Dynamics*, **16,** 1078–1084.

Sreedhar, J. and Van Dooren, P., 1993, Periodic Schur form and some matrix equations. *In* U. Helmke, R. Mennicken and J. Saurer (eds), *Proc. MTNS'93, Regensburg, Germany*, Vol. I, pp. 339–362.

Sreedhar, J. and Van Dooren, P., 1994, On finding stabilizing state feedback gains for a discrete-time periodic system. *Proc. 1994 American Control Conference, Baltimore, MD*, pp. 1167–1168.

Tombs, M. S. and Postlethwaite, I., 1987, Truncated balanced realization of a stable non-minimal state-space system. *Int. J. Control*, **46,** 1319–1330.

Varga, A. and Pieters, S., 1996, A computational approach for optimal periodic output feedback control. *Technical Report TR R199-96*, DLR-Oberpfaffenhofen, Institute for Robotics and System Dynamics (submitted for publication).

Wilkinson, J. H., 1965, *The Algebraic Eigenvalue Problem*, Oxford University Press.