

SOLUTION OF POSITIVE PERIODIC DISCRETE LYAPUNOV EQUATIONS WITH APPLICATIONS TO THE BALANCING OF PERIODIC SYSTEMS

A. Varga

*DLR Oberpfaffenhofen
German Aerospace Research Establishment
Institute of Robotics and System Dynamics
D-82234 Wessling, Germany
fax: +49-8153-281441, e-mail: Andras.Varga@dlr.de*

Keywords : Numerical methods, linear time-varying systems, periodic systems, balancing transformation.

Abstract

The discrete-time positive periodic Lyapunov equations have important applications in the balancing and potentially also in the model reduction of discrete-time periodic systems. Efficient numerically reliable algorithms based on periodic Schur decomposition are proposed for the solution of these equations. The proposed algorithms are extensions of the method of Hammarling for the case of positive semidefinite solution. Special methods were developed to solve efficiently small order periodic Lyapunov and Sylvester equations.

1 Introduction

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 [2, 8, 10, 12] of the form

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

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 $K \geq 1$. Of particular interest in many applications is the efficient and numerically reliable solution of various types of *discrete periodic Lyapunov equations* (DPLEs). Several possible computational approaches to solve DPLEs are discussed in [10]. A particular family of periodic Lyapunov equations with interesting applications in balancing and model reduction are the *positive discrete periodic Lyapunov equations* (PDPLEs). Because the periodic solutions in this case are positive semidefinite, these equations

can be solved directly for the Cholesky factors of the solutions.

In this paper we propose new algorithms to solve PDPLEs which represent extensions of the method for standard systems proposed by Hammarling [7]. The proposed approach resembles with the method of Bartels and Stewart [1] and relies on an initial reduction of the Lyapunov equation to a simpler form by using the periodic Schur decomposition of a matrix product [3]. The reduced equations are solved by using a special forward substitution algorithm. Important computational subproblems are the efficient and numerically stable solution of order one or order two PDPLEs and of *discrete periodic Sylvester equations* (DPSEs). Several computational approaches for these subproblems are also described.

Notation and notational conventions. For a square time-varying matrix A_k , $k = 0, 1, \dots$, we denote $\Phi_A(j, i) = A_{j-1} A_{j-2} \cdots A_i$ for $j > i$ and $\Phi_A(i, i) := I$. The matrix $\Phi_A(\tau + K, \tau)$ is called the *monodromy matrix* of system (1) at time τ and its eigenvalues, independent of τ , are called *characteristic multipliers*. For a periodic matrix X_k of period K we use alternatively the *script* notation \mathcal{X} which associates the block-diagonal matrix $\mathcal{X} = \text{diag}(X_0, X_1, \dots, X_{K-1})$ to the cyclic matrix sequence X_k , $k = 0, \dots, K-1$. This notation is consistent with the standard matrix operations as for instance addition, multiplication, inversion as well as with several standard matrix decomposition (Cholesky, SVD). We denote with $\sigma \mathcal{X}$ the K -cyclic shift $\sigma \mathcal{X} = \text{diag}(X_1, \dots, X_{K-1}, X_0)$ applied to the cyclic sequence X_k , $k = 0, \dots, K-1$. 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 , $k = 0, \dots, K-1$. 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 $X_k, k = 0, \dots, K-1$. 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 Balancing of Periodic Systems

The reachability gramian of an exponentially stable time-variant system of the form (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) P_k is a periodic matrix and satisfies the *forward-time positive discrete periodic Lyapunov equation* (FTPDPLE)

$$\sigma \mathcal{P} = \mathcal{A} \mathcal{P} \mathcal{A}^T + \mathcal{B} \mathcal{B}^T. \quad (2)$$

The system (1) is *uniformly controllable* iff $\mathcal{P} > 0$ [6]. Similarly, the observability gramian of the periodic system (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 *reverse-time positive discrete periodic Lyapunov equation* (RTPDPLE)

$$\mathcal{Q} = \mathcal{A}^T \sigma \mathcal{Q} \mathcal{A} + \mathcal{C}^T \mathcal{C} \quad (3)$$

and the system (1) is *uniformly observable* iff $\mathcal{Q} > 0$ [6]. For an exponentially stable periodic system the gramians are non-negative 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 $(\tilde{\mathcal{A}}, \tilde{\mathcal{B}}, \tilde{\mathcal{C}}, \tilde{\mathcal{D}})$ related by the transformation

$$(\tilde{\mathcal{A}}, \tilde{\mathcal{B}}, \tilde{\mathcal{C}}, \tilde{\mathcal{D}}) = (\sigma T^{-1} \mathcal{A} T, \sigma T^{-1} \mathcal{B}, \mathcal{C} T, \mathcal{D}) \quad (4)$$

are called *similar* and (4) is called a *Lyapunov similarity transformation*. The gramians $\tilde{\mathcal{P}}$ and $\tilde{\mathcal{Q}}$ of the transformed system $(\tilde{\mathcal{A}}, \tilde{\mathcal{B}}, \tilde{\mathcal{C}}, \tilde{\mathcal{D}})$ satisfy

$$\tilde{\mathcal{P}} = T^{-1} \mathcal{P} T^{-T}, \quad \tilde{\mathcal{Q}} = T^T \mathcal{Q} T.$$

For a uniformly controllable and observable periodic system, we can determine T such that the transformed gramians are equal and diagonal, that is

$$\tilde{\mathcal{P}} = \tilde{\mathcal{Q}} = \Sigma = \text{diag}(\Sigma_0, \Sigma_1, \dots, \Sigma_{K-1}),$$

where $\Sigma_k = \text{diag}(\sigma_{k,1}, \sigma_{k,2}, \dots, \sigma_{k,n})$. In analogy with the standard case [11], by using the singular value decomposition

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

T results as

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

The balancing transformation of could be useful for instance to perform model reduction of periodic systems in

analogy with standard systems. Notice that T can be computed exclusively on the basis of square-root information (the Cholesky factors of gramians) and this leads to a guaranteed enhancement of numerical accuracy of computations. The quantities $\sigma_{k,i}, i = 1, \dots, n$, representing the eigenvalues of the product $P_k Q_k$, are called the *Hankel-singular values* and the maximum of them over all k 's defines the *Hankel-norm* of the given periodic system [6].

To compute the balancing transformation it is important to solve the periodic Lyapunov equations (2) and (3) directly for the Cholesky factors of the gramians. A numerically reliable procedure for this purpose is proposed in the next section.

3 Solution of PDPLEs

In this section we propose a new method to solve the RTPDPLE

$$\mathcal{U}^T \mathcal{U} = \mathcal{A}^T \sigma \mathcal{U}^T \sigma \mathcal{U} \mathcal{A} + \mathcal{R}^T \mathcal{R} \quad (5)$$

directly for the Cholesky factor \mathcal{U} . A completely analog method can be derived for the FTPDPLE of the form (2). To solve (5) we assume that the monodromy matrix $\Phi_A(K, 0)$ has all characteristic multipliers in the interior of the unit circle.

One class of available methods to solve positive periodic Lyapunov equations [10] is based on reducing these equations to a single standard positive discrete Lyapunov equation to compute a *periodic generator*, say U_0 , which satisfies

$$\begin{aligned} U_0^T U_0 &= \Phi_A^T(K, 0) U_0^T U_0 \Phi_A(K, 0) \\ &+ \sum_{j=0}^{K-1} \Phi_A^T(j, 0) R_j^T R_j \Phi_A(j, 0). \end{aligned} \quad (6)$$

U_0 can be computed by using Hammarling's method [7] and the rest of matrices $U_k, k = 1, \dots, K-1$ results by recursion. Even with the enhancements proposed in [10], this approach is generally not recommendable because of implied matrix products and sums of matrix products. However, as it will be shown later, an iterative variant of this approach is well suited to solve low order PDPLEs.

The approach which we propose essentially parallels the Hammarling's method for standard systems [7]. 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 [3, 8]. According to [3], given the matrices $A_k, k = 0, 1, \dots, K-1$, there exist orthogonal matrices $Z_k, k = 0, 1, \dots, K-1$ such that $\tilde{A}_{K-1} = Z_0^T A_{K-1} Z_{K-1}$ is in *real Schur form* (RSF) and the matrices $\tilde{A}_k = Z_{k+1}^T A_k Z_k$ for $k = 0, \dots, K-2$ are upper triangular. Thus by using the PSD algorithm, we can determine the orthogonal matrices $Z_k, k = 0, \dots, K-1$ 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 the RTPDPLE (5).

The approach which we propose essentially parallels the method of [7] for standard systems including also some enhancements of this method proposed recently by [9]. Let \mathcal{Z} be an 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 $\tilde{\mathcal{R}}$ such that $\tilde{\mathcal{R}}^T \tilde{\mathcal{R}} = \mathcal{Z}^T \mathcal{R}^T \mathcal{R} \mathcal{Z}$. The equation (5) becomes after premultiplication with \mathcal{Z}^T and postmultiplication with \mathcal{Z}

$$\tilde{\mathcal{U}}^T \tilde{\mathcal{U}} = \tilde{\mathcal{A}}^T \sigma \tilde{\mathcal{U}}^T \sigma \tilde{\mathcal{U}} \tilde{\mathcal{A}} + \tilde{\mathcal{R}}^T \tilde{\mathcal{R}} \quad (7)$$

where $\tilde{\mathcal{U}} = \mathcal{U} \mathcal{Z}$. After solving this reduced equation for $\tilde{\mathcal{U}}$, the solution of (5) results as $\mathcal{U} = \tilde{\mathcal{U}} \mathcal{Z}^T$. In order to simplify the notations, we assume in what follows that the equation (5) is already in the reduced form (7) and \mathcal{R} is upper triangular.

Let us partition \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} non-singular we can derive recursive formulas which are analogous with those in the standard case. By rewriting (5) for the partitioned matrices, we obtain

$$\begin{aligned} \mathcal{U}_{11}^T \mathcal{U}_{11} &= \mathcal{A}_{11}^T \sigma \mathcal{U}_{11}^T \sigma \mathcal{U}_{11} \mathcal{A}_{11} + \mathcal{R}_{11}^T \mathcal{R}_{11} \\ \mathcal{U}_{12} &= \mathcal{M}_1^T \sigma \mathcal{U}_{12} \mathcal{A}_{22} + \mathcal{M}_1^T \sigma \mathcal{U}_{11} \mathcal{A}_{12} + \mathcal{M}_2^T \mathcal{R}_{12} \\ \mathcal{U}_{22}^T \mathcal{U}_{22} &= \mathcal{A}_{22}^T \sigma \mathcal{U}_{22}^T \sigma \mathcal{U}_{22} \mathcal{A}_{22} + \mathcal{R}_{22}^T \mathcal{R}_{22} + \mathcal{Y}^T \mathcal{Y} \end{aligned} \quad (8)$$

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

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

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 $\text{rank } P_k = n_1$. 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 (8) 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}

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} & \cdots & \mathcal{A}_{1\bar{n}} \\ 0 & \mathcal{A}_{22} & \cdots & \mathcal{A}_{2\bar{n}} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{A}_{\bar{n}\bar{n}} \end{bmatrix} \quad (9)$$

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 1. *Periodic Schur Method to Solve the RTPDPLE.*

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 of $\mathcal{R} \mathcal{Z}$ as $\mathcal{R} \mathcal{Z} = \mathcal{Q} \tilde{\mathcal{R}}$; put $\mathcal{R} = \tilde{\mathcal{R}}$.

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

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 4, see Section 5))

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

Compute the QR-decompositions

$$\begin{bmatrix} \mathcal{M}_2 \\ \mathcal{M}_1 \end{bmatrix}_{(k)} = \begin{bmatrix} Q_k & N_k^T \\ 0 \end{bmatrix} \begin{bmatrix} V_k \\ 0 \end{bmatrix} \quad (k = 0, \dots, K-1)$$

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

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

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 2 or 3, see Section 4)

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

end

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

Compute the QR-decomposition

$$\begin{bmatrix} \mathcal{R}_{ii} \\ \mathcal{Y}_i \end{bmatrix}_{(k)} = Q_k \begin{bmatrix} W_k \\ 0 \end{bmatrix} \quad (k = 0, \dots, K-1)$$

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

end

end

Compute the QR decomposition of $\mathcal{U} \mathcal{Z}^T$ as $\mathcal{U} \mathcal{Z}^T = \mathcal{Q} \tilde{\mathcal{U}}$; put $\mathcal{U} = \tilde{\mathcal{U}}$.

Special care is necessary to handle the cases when 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 in [7]. An algorithm similar to Algorithm 1 can be devised to solve the FTPDPLE.

Algorithm 1 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 PDPLEs and DPSEs, Algorithm 1 performs about $14Kn^3$ flops (floating point operations).

4 Solution of Low Order DPSEs

The computation of U_{rj} for $r \neq j$ in Algorithm 1 requires the solution of a low order *discrete periodic Sylvester equation* (DPSE) of the form

$$Y_k = E_k^T Y_{k+1} F_k + G_k, \quad k = 0, \dots, K-1; \quad Y_0 = Y_K \quad (10)$$

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 $1 \leq n_1, n_2 \leq 2$.

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 (10), let y and g be defined as

$$y =: \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{bmatrix} = \begin{bmatrix} \text{vec}(Y_{K-1}) \\ \text{vec}(Y_{K-2}) \\ \vdots \\ \text{vec}(Y_0) \end{bmatrix},$$

$$g =: \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_K \end{bmatrix} = \begin{bmatrix} \text{vec}(G_{K-1}) \\ \text{vec}(G_{K-2}) \\ \vdots \\ \text{vec}(G_0) \end{bmatrix},$$

where the operator $\text{vec}(\cdot)$ generates a vector from the stacked columns of a matrix. By using the \otimes notation for the Kronecker product, the corresponding H is given by (only nonzero elements are shown)

$$H =: \begin{bmatrix} I & & & & H_{1,K} \\ H_{21} & I & & & \\ & & \ddots & & \vdots \\ & & & \ddots & \\ & & & & H_{K,K-1} & I \end{bmatrix}$$

where $H_{i+1,i} = -F_{K-i-1}^T \otimes E_{K-i-1}^T$ for $i = 1, \dots, K-1$, $H_{1,K} = -F_{K-1}^T \otimes E_{K-1}^T$.

The standard Gaussian elimination method [5] 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 2. *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 $i = 1, \dots, K-1$

 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

$$\begin{bmatrix} H_{i,K} & g_i \\ H_{i+1,K} & g_{i+1} \end{bmatrix} := L_i^{-1} P_i \begin{bmatrix} H_{i,K} & g_i \\ H_{i+1,K} & g_{i+1} \end{bmatrix}$$

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 $i = K-2, \dots, 1$

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

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. By using the splitting H as $H = I - N$, it can be shown that the eigenvalues of N are the K -th roots of the eigenvalues 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

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

where $\lambda_1(\cdot)$ is any of the eigenvalues of the respective matrix. Because $\rho(N) < 1$ then the solution y of the equation $Hy = g$ can be computed by using the following iteration [5]

$$y^{(i+1)} = Ny^{(i)} + g \quad (11)$$

initialized with any starting vector $y^{(0)}$.

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 (10) by solving the discrete Sylvester equation

$$Y_0 = \Phi_E^T(K, 0)Y_0\Phi_F(K, 0) + \sum_{j=0}^{K-1} \Phi_E^T(j, 0)G_j\Phi_F(j, 0) \quad (12)$$

and then generating the rest of solution by backward recursion. This approach corresponds to an accelerated iteration and leads to very efficient implementation of the solution method.

Algorithm 3. *Iterative Method to Solve the DPSE.*

Comment. Compute the initializing periodic generator Y_0 .

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

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

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

end

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

Comment. Iterate by using backward recurrence.

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

$$Y_K = Y_0$$

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

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

end

$$i \leftarrow i + 1$$

end

By examining the above algorithm it is clear that the iterates $\tilde{Y}_k^{(i+1)}$ computed at the i -th iteration step are different from $Y_k^{(i+1)}$ resulted by using (11). Notice that, instead of using (11), each computed new value $\tilde{Y}_k^{(i+1)}$ in Algorithm 3 is immediately used to generate the next value $\tilde{Y}_{k+1}^{(i+1)}$ 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 (11) and $\tilde{Y}_0^{(i+1)}$ computed with Algorithm 3 with the same $Y_k^{(i)}$, $k = 0, 1, \dots, K-1$. 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

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

It is easy to observe that the value $\tilde{Y}_0^{(i+1)}$ produced by Algorithm 3 in one complete iteration equals the value

$Y_0^{(i+K)}$ computed after K iterations by using formula (11). Experimentally it was observed that when using the initialization (12), 2-3 iterations in Algorithm 3 are always sufficient to attain the limiting accuracy solution.

The number of operations to solve K DPSEs in m iterations with Algorithm 3 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 $(4 + 2m)n^2 K$. Because usually one or two iterations are sufficient to achieve the limiting accuracy with the proposed iterative method, Algorithm 3 is more efficient than Algorithm 2 based on LU decomposition. An additional advantage of Algorithm 3 is that it needs practically no additional memory to perform the computations. Moreover, we observed experimentally that, the accuracy achieved with the iterative approach is systematically better than the accuracy resulted with Algorithm 2.

5 Solution of Low Order PDPLEs

The iterative Algorithm 3 to solve DPSEs can be readily adapted to solve the low order PDPLEs necessary to compute the diagonal blocks U_{rr} in Algorithm 1. The following algorithm, although mainly intended to solve small order RTPDPLEs, can be also seen as a general iterative method to solve (5).

Algorithm 4. *Iterative Solution of RTPDPLE.*

Comment. Compute the Cholesky factor U_0 .

$\Phi = I, V = 0$.

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

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

end

Solve $U_0^T U_0 = \Phi^T U_0^T U_0 \Phi + V_0^T V_0$; set $U_K = U_0$ and $i = 0$.

Comment. Iterate by using 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 $k = K-1, \dots, 0$

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

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 [4]. To attain the limiting accuracy solution usually at most two iterations are sufficient.

6 Conclusion

A numerically reliable computational algorithm has been proposed to solve RTPDPLEs. A dual version of this algorithm can be easily devised to solve FTPDPLEs as well. The new algorithms are extensions of an algorithm to solve standard DPLEs. The effectiveness of the proposed methods heavily relies on the efficiency of solving low order DPSEs and PDPLEs. Special methods have been developed for this purpose. The main application of the proposed method is the enhanced accuracy square root balancing of periodic systems.

References

- [1] R. H. Bartels and G. W. Stewart. Algorithm 432: Solution of the matrix equation $AX+XB=C$. *Comm. ACM*, 15:820–826, 1972.
- [2] S. Bittanti, P. Colaneri, and G. De Nicolao. The difference periodic Riccati equation for the periodic prediction problem. *IEEE Trans. Autom. Control*, 33:706–712, 1988.
- [3] A. W. Bojanczyk, G. Golub, and P. Van Dooren. The periodic Schur decomposition. Algorithms and applications. In F. T. Luk, editor, *Proceedings SPIE Conference*, volume 1770, pages 31–42, July 1992.
- [4] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders. Methods for modifying matrix factorizations. *Math. Comput.*, 28:505–535, 1974.
- [5] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, 1989.
- [6] A. Halanay and V. Ionescu. *Time-Varying Discrete Linear Systems*. Birkhäuser, 1994.
- [7] S. J. Hammarling. Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA J. Numer. Anal.*, 2:303–323, 1982.
- [8] J. J. Hench and A. J. Laub. Numerical solution of the discrete-time periodic Riccati equation. *IEEE Trans. Autom. Control*, 39:1197–1210, 1994.
- [9] T. Penzl. Numerical solution of generalized Lyapunov equations. Preprint SFB393/96-02, Technical University Chemnitz, May 1996.
- [10] J. Sreedhar and P. Van Dooren. Periodic Schur form and some matrix equations. In U. Helmke, R. Menicken, and J. Saurer, editors, *Proc. MTNS'93, Regensburg, Germany*, volume I, pages 339–362, 1993.
- [11] M. S. Tombs and I. Postlethwaite. Truncated balanced realization of a stable non-minimal state-space system. *Int. J. Control*, 46:1319–1330, 1987.
- [12] A. Varga and S. Pieters. A computational approach for optimal periodic output feedback control. In *Proc. CACSD'96 Symposium, Dearborn, MI*, pp. 176–181, 1996.