

# Computation of transfer function matrices of periodic systems

Andras Varga

German Aerospace Center  
DLR - Oberpfaffenhofen  
Institute of Robotics and System Dynamics  
D-82234 Wessling, Germany  
Andras.Varga@dlr.de

## Abstract

We present a numerical approach to evaluate the transfer function matrices of a periodic system corresponding to lifted state-space representations as constant systems. The proposed *pole-zero* method determines each entry of the transfer function matrix in a minimal *zeros-poles-gain* representation. A basic computational ingredient for this method is the extended periodic real Schur form of a periodic matrix, which underlies the computation of minimal realizations and system poles. To compute zeros and gains, fast algorithms are proposed, which are specially tailored to particular single-input single-output periodic systems. The new method relies exclusively on reliable numerical computations and is well suited for robust software implementations.

## 1 Introduction

Among the open computational problems listed in a recent survey [13], the computation of the transfer function matrix of a periodic system is one which has some useful applications. For example, the evaluation of frequency-response for a periodic system, can be conveniently done by using the transfer function matrix corresponding to a constant system lifted representation. Furthermore, for the manipulation of periodic systems, the state-space to transfer function matrix conversion is a useful transformation which must be provided by any software toolbox devoted to periodic systems.

In this paper we consider time-varying periodic systems 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  $A_k \in \mathbb{R}^{n_{k+1} \times n_k}$ ,  $B_k \in \mathbb{R}^{n_{k+1} \times m}$ ,  $C_k \in \mathbb{R}^{p \times n_k}$ ,  $D_k \in \mathbb{R}^{p \times m}$  are periodic with period  $K \geq 1$ . The importance of considering periodic systems with time-varying state dimensions has been revealed

by the minimal realization theory of periodic systems [2]. Note that even for constant dimension periodic systems, the corresponding minimal order (i.e., reachable and observable) state-space realizations have, in general, time-varying state dimensions.

In this paper we propose a numerically reliable approach to evaluate the transfer function matrix corresponding to lifted state space representations. The proposed *pole-zero* method determines each entry of the transfer function matrix in a minimal *zeros-poles-gain* representation. For this purpose, minimal realizations for each input-output channel are necessary to be computed. A basic numerical ingredient to compute minimal realizations and also system poles is the *extended periodic real Schur form* of a product of rectangular matrices introduced in [12]. To compute zeros and gains, fast algorithms, specially tailored to particular single-input single-output periodic systems, are developed. The proposed method is numerically reliable, relying exclusively on using orthogonal and well-conditioned transformations. This guarantees a certain form of numerical stability for the overall computation. The new method is well suited for robust software implementations. Numerical examples computed with MATLAB-based implementations show the applicability of this method to high order periodic systems.

**Notation.** To simplify the presentation we introduce first some notation. For a  $K$ -periodic matrix  $X_i$  we use alternatively the *script* notation

$$\mathcal{X}_k := \text{diag}(X_k, X_{k+1}, \dots, X_{k+K-1}),$$

which associates the block-diagonal matrix  $\mathcal{X}_k$  to the cyclic matrix sequence  $X_i$ ,  $i = k, \dots, k + K - 1$  starting at time moment  $k$ . We reserve the script notation  $\mathcal{X}$  (i.e., without subscript) for  $\mathcal{X} = \mathcal{X}_1$ . The script notation is consistent with the standard matrix operations as for instance addition, multiplication, inversion as well as with several standard matrix decompositions (Cholesky, SVD). We denote with  $\sigma \mathcal{X}_k$  the  $K$ -cyclic

shift

$$\sigma \mathcal{X}_k = \text{diag}(X_{k+1}, \dots, X_{k+K-1}, X_k)$$

of the cyclic sequence  $X_i$ ,  $i = k, \dots, k+K-1$ .

By using the script notation, the periodic system (1) will be alternatively denoted by the quadruple  $(\mathcal{A}_k, \mathcal{B}_k, \mathcal{C}_k, \mathcal{D}_k)$  or  $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$  if  $k = 1$ . The transition matrix of the system (1) is defined by the  $n_j \times n_i$  matrix  $\Phi_A(j, i) = A_{j-1}A_{j-2} \cdots A_i$ , where  $\Phi_A(i, i) := I_{n_i}$ . The state transition matrix over one period  $\Phi_A(j+K, j) \in \mathbb{R}^{n_j \times n_j}$  is called the *monodromy matrix* of system (1) at time  $j$  and its eigenvalues are called *characteristic multipliers* at time  $j$ .

## 2 Transfer function matrices of periodic systems

To define the *transfer function matrix* (TFM) of the periodic system (1), we consider the time-invariant representations corresponding to the two associated lifted systems introduced in [8] and [9, 3], respectively. The lifting technique of [8] uses the input-output behavior of the system over time intervals of length  $K$ , rather than 1. For a given sampling time  $k$ , the corresponding  $mK$ -dimensional input,  $pK$ -dimensional output, and  $n_k$ -dimensional state vectors are

$$\begin{aligned} u_k^L(h) &= [u^T(k+hK) \cdots u^T(k+hK+K-1)]^T, \\ y_k^L(h) &= [y^T(k+hK) \cdots y^T(k+hK+K-1)]^T, \\ x_k^L(h) &= x(k+hK) \end{aligned}$$

The lifted system has the form

$$\begin{aligned} x_k^L(h+1) &= F_k^L x_k^L(h) + G_k^L u_k^L(h) \\ y_k^L(h) &= H_k^L x_k^L(h) + L_k^L u_k^L(h) \end{aligned} \quad (2)$$

where

$$\begin{aligned} F_k^L &= \Phi_A(k+K, k) \\ G_k^L &= [\Phi_A(k+K, k+1)B_k \cdots B_{k+K-1}] \\ H_k^L &= \begin{bmatrix} C_k \\ \vdots \\ C_{k+K-1}\Phi_A(k+K-1, k) \end{bmatrix} \\ L_k^L &= \begin{bmatrix} D_k & 0 & \cdots & 0 \\ L_{k,2,1} & D_{k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{k,K,1} & L_{k,K,2} & \cdots & D_{k+K-1} \end{bmatrix} \end{aligned}$$

with  $L_{k,i,j} = C_{k+i-1}\Phi_A(k+i-1, k+j)B_{k+j-1}$ , for  $i = 2, \dots, K$ ,  $j = 1, 2, \dots, K-1$ , and  $i > j$ .

The system (2) is called the *standard lifted system* at time  $k$  of the given  $K$ -periodic system (1). The associated TFM  $W_k(z)$  is

$$W_k^L(z) = H_k^L(zI_{n_k} - F_k^L)^{-1}G_k^L + L_k^L \quad (3)$$

and depends on the sampling time  $k$ . Obviously  $W_{k+K}^L(z) = W_k^L(z)$  and the TFMs at two successive values of  $k$  are related by the following relation [5]

$$W_{k+1}^L(z) = \begin{bmatrix} 0 & I_{p(K-1)} \\ zI_p & 0 \end{bmatrix} W_k^L(z) \begin{bmatrix} 0 & z^{-1}I_m \\ I_{m(K-1)} & 0 \end{bmatrix}$$

Thus, computing a single TFM for an arbitrary  $k$  is sufficient to determine all TFMs at  $K$  successive values.

The second lifted representation we consider is called the *cyclic lifted system* [9, 3]. Consider the *cyclic shift matrix*

$$Z_k = \begin{bmatrix} 0 & \cdots & 0 & I_{n_{k+K-1}} \\ I_{n_k} & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & I_{n_{k+K-2}} & 0 \end{bmatrix}$$

and define (similar as done in [9] for constant dimensions) the *cyclic lifted system* at time  $k$ , with  $mK$  inputs and  $pK$  outputs, as the time-invariant system

$$\begin{aligned} x_k^C(h+1) &= F_k^C x_k^C(h) + G_k^C u_k^C(h) \\ y_k^C(h) &= H_k^C x_k^C(h) + L_k^C u_k^C(h) \end{aligned} \quad (4)$$

with  $(F_k^C, G_k^C, H_k^C, L_k^C) = (Z_k \mathcal{A}_k, Z_k \mathcal{B}_k, \mathcal{C}_k, \mathcal{D}_k)$ . The state dimension of this system is  $\mu = \sum_{i=1}^K n_i$  and its  $pK \times mK$  TFM is

$$W_k^C(z) = H_k^C(zI_\mu - F_k^C)^{-1}G_k^C + L_k^C.$$

The relationships between the TFMs of the two lifted systems is

$$W_k^C(z) = \Delta_p(z^{-1})W_k^L(z^K)\Delta_m(z), \quad (5)$$

where  $\Delta_j(z) = \text{diag}\{I_j, zI_j, \dots, z^{K-1}I_j\}$ . Thus, the TFM of the *cyclic lifted system* can be easily determined from the TFM of the *standard lifted system*. Conversely, having the TFM  $W_k^C(z)$ , we compute  $W_k^L(z^K)$  using the relation (5) and then replace  $z^K$  by  $z$ .

For the computation of the TFM, we can also use the so-called *stacked lifted representation* [6]. This is a time-invariant descriptor system representation of the form

$$\begin{aligned} E_k^S x_k^S(h+1) &= F_k^S x_k^S(h) + G_k^S u_k^L(h) \\ y_k^L(h) &= H_k^S x_k^S(h) + L_k^S u_k^L(h) \end{aligned} \quad (6)$$

where  $G_k^S = Z_k \mathcal{B}_k$ ,  $H_k^S = \mathcal{C}_k$ ,  $L_k^S = \mathcal{D}_k$ , and

$$E_k^S - zE_k^S = \begin{bmatrix} -zI_{n_k} & O & \cdots & O & A_{k+K-1} \\ A_k & -I_{n_{k+1}} & \cdots & O & O \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & \cdots & -I_{n_{k+K-2}} & O \\ O & O & \cdots & A_{k+K-2} & -I_{n_{k+K-1}} \end{bmatrix} \quad (7)$$

The TFM of the stacked lifted system is

$$W_k^S(z) = H_k^S(zE_k^S - F_k^S)^{-1}G_k^S + L_k^S$$

and it is easy to show that  $W_k^S(z) = W_k^L(z)$ , that is, the TFMs of the *stacked* and *standard lifted systems* are the same.

### 3 Computational approach

In this section we propose an efficient computational approach to determine  $W(z) := W_1^L(z)$ , the TFM at time  $k = 1$  of the *standard lifted system* (2). The TFMs for other time moments  $k = 2, \dots, K$  or for the *cycled lifted system* (4) can be easily obtained by simple algebraic manipulations via (5).

A straightforward approach to compute the  $pK \times mK$  TFM  $W(z)$  is to apply the *pole-zero* method of [15] to the *standard lifted system* (2). This amounts to compute successively the poles, zeros and gains corresponding to minimal realizations of the individual input-output channels. However, because the construction of the *standard lifted system* involves matrix multiplications, this approach is not suited for numerical computations. To avoid matrix multiplications, we can employ the same approach to the *cyclic lifted system* and compute the TFM of the *standard lifted system* via (5). Alternatively, we can compute the same TFM from the *stacked lifted system* by applying a similar *pole-zero* approach but for descriptor systems [10]. In both cases, the required computational effort and computer storage can be prohibitive for large dimensions or large periods.

The following conceptual procedure serves as basis to describe the proposed approach to compute an element  $w_{ij}(z)$  of  $W(z)$  in the *zeros-poles-gain* form

$$w_{ij}(z) = \gamma \frac{\prod_{\iota=1}^{q_z} (z - \mu_\iota)}{\prod_{\iota=1}^{q_f} (z - \rho_\iota)}$$

starting from the *stacked lifted system* (6):

#### Pole-Zero Algorithm.

1. Compute a minimal realization  $(\widehat{A} - z\widehat{E}, \widehat{b}, \widehat{c}, \widehat{d})$  of the subsystem corresponding to the  $(j, i)$ th input-output channel of the stacked lifted system  $(F_1^S - zE_1^S, G_1^S, H_1^S, L_1^S)$ .
2. Compute the  $q_f$  finite poles  $\rho_\iota$ ,  $\iota = 1, \dots, q_f$  as the finite generalized eigenvalues of the pair  $(\widehat{A}, \widehat{E})$ .
3. Compute the  $q_z$  finite zeros  $\mu_\iota$ ,  $\iota = 1, \dots, q_z$  of the descriptor system  $(\widehat{A} - z\widehat{E}, \widehat{b}, \widehat{c}, \widehat{d})$ .
4. Choose a real  $z_0$  satisfying  $|z_0| > 1$ , that is neither a pole nor zero and compute the gain

$$\gamma = \left( \widehat{c}(z_0\widehat{E} - \widehat{A})^{-1}\widehat{b} + \widehat{d} \right) \prod_{\iota=1}^{q_f} (z_0 - \rho_\iota) / \prod_{\iota=1}^{q_z} (z_0 - \mu_\iota)$$

The basis of this conceptual procedure is the method proposed in [10] for descriptor systems. This procedure is numerically reliable, since each step can be performed using numerically stable algorithms. However, because ignoring the structure of the problem, the computational complexity of this approach is too high. To

compute a single element of  $W(z)$ , the computational complexity is, in the worst case, of order  $O(\mu^3)$ , where  $\mu = \sum_{i=1}^K n_i$ . For example, in the case of a periodic system with constant state dimensions  $n_i = n$ , the computational complexity is  $O(K^3n^3)$  instead of a desirable complexity of  $O(Kn^3)$  as formulated in [13] for a satisfactory algorithm for periodic systems. In what follows, we show that such a computational complexity can be indeed achieved by cleverly exploiting the problem structure. For this purpose, we discuss each step of the above algorithm and indicate how the underlying computations can be done efficiently.

### 3.1 Computational ingredients

For periodic systems with constant dimensions, the *periodic real Schur form* (PRSF) is the basic numerical ingredient to solve periodic Lyapunov equations or to compute the poles of a periodic system. According to [1], given the matrices  $A_k \in \mathbb{R}^{n \times n}$ ,  $k = 1, \dots, K$ , there exist orthogonal matrices  $Z_k$ ,  $k = 1, \dots, K$ ,  $Z_{K+1} := Z_1$ , such that

$$\widetilde{A}_k := Z_{k+1}^T A_k Z_k \quad (8)$$

where  $\widetilde{A}_K$  is in *real Schur form* (RSF) and the matrices  $\widetilde{A}_k$  for  $k = 1, \dots, K-1$  are upper triangular. Numerically stable algorithms to compute the PRSF have been proposed in [1, 7]. For  $A_k$  in a PRSF, the computation of the eigenvalues of  $A_K \cdots A_2 A_1$  becomes very straightforward. The computation of the PRSF is numerically stable and has a computational complexity  $O(Kn^3)$ .

For systems with time-varying dimensions, the *extended periodic real Schur form* (EPRSF) represents a generalization of the PRSF which allows to address many problems with varying dimensions. According to [12], given the matrices  $A_k \in \mathbb{R}^{n_{k+1} \times n_k}$ ,  $k = 1, \dots, K$ , with  $n_{K+1} = n_1$  there exist orthogonal matrices  $Z_k \in \mathbb{R}^{n_k \times n_k}$ ,  $k = 1, \dots, K$ ,  $Z_{K+1} := Z_1$ , such that the matrices

$$\widetilde{A}_k := Z_{k+1}^T A_k Z_k = \begin{bmatrix} \widetilde{A}_{k,11} & \widetilde{A}_{k,12} \\ 0 & \widetilde{A}_{k,22} \end{bmatrix}, \quad (9)$$

are block upper triangular, where  $\widetilde{A}_{k,11} \in \mathbb{R}^{n \times n}$ ,  $\widetilde{A}_{k,22} \in \mathbb{R}^{(n_{k+1}-n) \times (n_k-n)}$  for  $k = 1, \dots, K$  and  $\underline{n} = \min_k \{n_k\}$ . Moreover,  $\widetilde{A}_{K-1,11}$  is in RSF,  $\widetilde{A}_{k,11}$  for  $k = 1, \dots, K-1$  are upper triangular and  $\widetilde{A}_{k,22}$  for  $k = 1, \dots, K$  are upper trapezoidal. The computation of the EPRSF is numerically stable and has a computational complexity  $O(Kn^3)$ , where  $n = \max\{n_i\}$ .

A main application of the PRSF and EPRSF is the computation of the poles of the TFM  $W(z)$ . For a minimal periodic system, the poles are defined as the eigenvalues of the monodromy matrix  $\Phi_A(k+K, k) = A_{k+K-1} \cdots A_{k+1} A_k$ . For constant dimension, the poles

are independent of  $k$  and can be easily computed by reducing the  $K$ -periodic matrix  $A_k$  to a PRSF. However, for time-varying dimensions, the poles set depends on the sampling time, and can be easily computed from the EPRSF. For a given  $k$ , the pole set is formed from a so-called *core* set, representing the eigenvalues of  $\Phi_{A_{11}}(k+K, k)$ , and  $n_k - \underline{n}$  null poles. Note that the core poles are independent of  $k$ .

### 3.2 Computation of minimal realizations

The minimal realization problem at Step 1 of the **Pole-Zero Algorithm** has a particular structure. To simplify notations, we denote by  $(F - zE, G, H, L)$  the stacked lifted system  $(F_1^S - zE_1^S, G_1^S, H_1^S, L_1^S)$  (at time  $k = 1$ ) and let  $G_j$  and  $H_i$  be the  $j$ th-column of  $G$  and the  $i$ th-row of  $H$ , respectively, and let  $L_{ij}$  be the element  $(i, j)$  of  $L$ . We want to compute a minimal realization of the system  $(F - zE, G_j, H_i, L_{ij})$ . The matrices  $G_j$  and  $H_i$  have a very special particular structure. Since  $i$  and  $j$  can be uniquely expressed as

$$i = (\ell_i - 1)p + t, \quad j = (\ell_j - 1)m + s,$$

it follows that  $H_i$  can be constructed only from  $c_{\ell_i, t}$ , the row  $t$  of  $C_{\ell_i}$ , and  $G_j$  can be constructed only from  $b_{\ell_j, s}$ , the column  $s$  of  $B_{\ell_j}$ , as

$$H_i = [O \cdots O c_{\ell_i, t} O \cdots O] \quad (10)$$

$$G_j = [O \cdots O b_{\ell_j, s}^T O \cdots O]^T \quad (11)$$

Further,  $L_{ij}$  is defined as

$$L_{ij} = \begin{cases} d_{\ell_i, ts}, & \text{if } \ell_i = \ell_j \\ 0, & \text{if } \ell_i \neq \ell_j \end{cases}$$

where  $d_{\ell_i, ts}$  is the element  $(t, s)$  of  $D_{\ell_i}$ . The single-input single-output system  $(F - zE, G_i, H_j, L_{ij})$  can be seen as a *stacked lifted system* corresponding to a particular single-input single-output periodic system  $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$  with zero  $\hat{B}_k$  for  $k \neq s$  and zero  $\hat{C}_k$  for  $k \neq t$  and appropriate  $\hat{D}_k$ . Thus, we can apply the minimal realization procedure of [12] to the periodic system  $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$  to obtain a minimal order periodic system  $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$ . The corresponding a *lifted stacked system*  $(\hat{A} - z\hat{E}, \hat{b}, \hat{c}, \hat{d})$ , at Step 1 of the **Pole-zero Algorithm** is a minimal realization of  $(F_1^S - zE_1^S, G_1^S, H_1^S, L_1^S)$ .

In what follows we shortly present the main steps of the minimal realization approach for a stable periodic system (for details, see [12]). The minimal realization procedure involves the computation of two truncation matrices  $\mathcal{L}$  and  $\mathcal{T}$  such that the matrices of the minimal order system  $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$  are obtained as

$$\hat{A} = \sigma \mathcal{L} \mathcal{A} \mathcal{T}, \quad \hat{B} = \sigma \mathcal{L} \tilde{B}, \quad \hat{C} = \tilde{C} \mathcal{T}, \quad \hat{D} = \mathcal{D}. \quad (12)$$

The computation of truncation matrices relies on the reachability grammian  $\mathcal{P}$  and observability grammian

$\mathcal{Q}$ , which satisfy the periodic Lyapunov equations

$$\begin{aligned} \sigma \mathcal{P} &= \mathcal{A} \mathcal{P} \mathcal{A}^T + \tilde{B} \tilde{B}^T \\ \mathcal{Q} &= \mathcal{A}^T \sigma \mathcal{Q} \mathcal{A} + \tilde{C}^T \tilde{C} \end{aligned} \quad (13)$$

Since these grammians are non-negative definite, they can be expressed in Cholesky factorized forms  $\mathcal{P} = \mathcal{S} \mathcal{S}^T$  and  $\mathcal{Q} = \mathcal{R}^T \mathcal{R}$ , where  $S_k$  and  $R_k$  for  $k = 1, \dots, K$  are  $n_k \times n_k$  upper triangular matrices.

For each  $k$ , consider the singular value decomposition of  $R_k S_k$  in the partitioned form

$$R_k S_k = [U_{k,1} \ U_{k,2}] \begin{bmatrix} \tilde{\Sigma}_k & 0 \\ 0 & 0 \end{bmatrix} [V_{k,1} \ V_{k,2}]^T, \quad (14)$$

where  $\tilde{\Sigma}_k \in \mathbb{R}^{r_k \times r_k}$ ,  $U_{k,1} \in \mathbb{R}^{n_k \times r_k}$ ,  $V_{k,1} \in \mathbb{R}^{n_k \times r_k}$  and  $\tilde{\Sigma}_k > 0$ . From the above decomposition define, with  $\tilde{\Sigma} = \text{diag}(\tilde{\Sigma}_1, \dots, \tilde{\Sigma}_K)$ , the *truncation* matrices

$$\mathcal{L} = \tilde{\Sigma}^{-\frac{1}{2}} \mathcal{U}_1^T \mathcal{R}, \quad \mathcal{T} = \mathcal{S} \mathcal{V}_1 \tilde{\Sigma}^{-\frac{1}{2}}, \quad (15)$$

which are used to determine the reduced system matrices in (12). Note that this system is *balanced*, the corresponding grammians being equal and diagonal.

The key computation in determining  $\mathcal{L}$  and  $\mathcal{T}$  is the solution of the two periodic Lyapunov equations in (13) with time-varying dimensions directly for the Cholesky factors of the Gramians. A numerically reliable procedure for this computation has been proposed in [12]. Since the computation of truncation matrices (and also of the minimal realization) can be done using only the Cholesky factors (square-roots) of Gramians, the above method is called the *square-root* method. This method leads to a guaranteed enhancement of the overall numerical accuracy of computations. Potentially more accurate is the *balancing-free square-root* method [12], which avoids balancing by using well-conditioned truncation matrices. For the purpose of the **Pole-zero algorithm**, this approach is that one to be preferred.

The computational complexity of the approach of [12] to determine a minimal realization is  $O(Kn^3)$ . The most time-consuming operation in this process is the solution of the two periodic Lyapunov equations satisfied by the grammians. When employing the procedure of [12] for this purpose, the first step of the solution method is the reduction of the periodic matrix  $A_k$  to an EPRSF. Then, the Cholesky factors of the grammians are computed directly by solving reduced periodic Lyapunov equations (i.e., with  $A_k$  in EPRSF). Since for each element of the  $pK \times mK$  TFM the minimal realization problem involves the same periodic state matrix, the reduction to EPRSF has to be done only once to put the original periodic system in a coordinate form with the state matrix in EPRSF. This is achieved by applying an orthogonal Lyapunov similarity transformation to the original system, which preserve the TFM of the system.

This minimal realization method based on balancing technique is not restricted to asymptotically stable periodic systems. For an unstable system, a simple scaling can be used to enforce the stability of the starting representation. For instance, it is possible to replace only  $A_1$  by  $\alpha A_1$ , where  $0 < \alpha < 1$  is chosen such that  $\alpha \Phi_A(K+1, 1)$  has eigenvalues in the open unit disc. For the  $\alpha$ -scaled system, we can apply either the *square-root* or *balancing-free square-root approach* to determine a minimal system. Finally, the computed  $\hat{A}_1$  needs to be rescaled to  $\hat{A}_1/\alpha$ .

### 3.3 Computation of poles and zeros

To compute the poles of a periodic system  $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ , the eigenvalues of the *monodromy* matrix  $\Phi_{\hat{A}}(K+1, 1)$  must be determined. This computation can be done without forming this matrix product explicitly, by reducing the  $K$ -periodic matrix  $\hat{A}_k$  to the PRSF in the case of constant dimensions, or to the EPRSF in the case of time-varying dimensions.

It is possible to compute the zeros by cleverly exploiting the structure of the system pencil

$$\hat{S}(z) = \begin{bmatrix} \hat{A} - z\hat{E} & \hat{b} \\ \hat{c} & \hat{d} \end{bmatrix} \quad (16)$$

by eliminating the simple eigenvalues at infinity. We present only the main idea of the procedure to compute the finite zeros. A more general algorithm which allows to compute the infinite zeros and the Kronecker structure has been recently developed in [14]. Consider  $\hat{S}(z)$ , a system pencil with permuted block rows and columns

$$\hat{S}(z) = \begin{bmatrix} \hat{S}_1 & -\hat{T}_1 & O & \cdots & O \\ O & \hat{S}_2 & -\hat{T}_2 & \cdots & O \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ O & & & \hat{S}_{K-1} & -\hat{T}_{K-1} \\ -z\hat{T}_K & O & \cdots & O & \hat{S}_K \end{bmatrix} \quad (17)$$

with

$$\hat{S}_i = \begin{bmatrix} \hat{A}_i & \hat{B}_i \\ \hat{C}_i & \hat{D}_i \end{bmatrix}, \quad \hat{T}_i = \begin{bmatrix} I_{r_{i+1}} & O \\ O & O \end{bmatrix}$$

for  $i = 1, \dots, K$ . Here  $r_i$ ,  $i = 1, \dots, K$  are the time-varying dimensions of the state vector of the minimal order periodic system  $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ .

Consider the orthogonal transformation matrix  $U_1$  to compress the matrix  $\begin{bmatrix} -\hat{T}_1 \\ \hat{S}_2 \end{bmatrix}$  to  $\begin{bmatrix} R_1 \\ O \end{bmatrix}$ , where  $R_1$  is a full row rank matrix. Applying  $U_1^T$  to the first two blocks rows of  $\hat{S}(z)$  we obtain for the nonzero blocks

$$U_1^T \begin{bmatrix} \hat{S}_1 & -\hat{T}_1 & O \\ O & \hat{S}_2 & -\hat{T}_2 \end{bmatrix} = \begin{bmatrix} * & R_1 & * \\ \bar{S}_2 & O & -\bar{T}_2 \end{bmatrix}$$

which defines the new matrices  $\bar{S}_2$  and  $\bar{T}_2$ . Then, construct the transformations  $U_i$  for  $i = 2, \dots, K-1$  such that

$$U_i^T \begin{bmatrix} \bar{S}_i & -\bar{T}_i & O \\ O & \hat{S}_{i+1} & -\hat{T}_{i+1} \end{bmatrix} = \begin{bmatrix} * & R_i & * \\ \bar{S}_{i+1} & O & -\bar{T}_{i+1} \end{bmatrix}$$

where  $R_i$  is a full row rank matrix. Applying the transformations  $U_i$  successively to the  $i$ th and  $(i+1)$ th block rows of the pencil  $\hat{S}(z)$ , we get the reduced pencil

$$\left[ \begin{array}{c|ccc} * & R_1 & * & \cdots & * \\ * & O & R_2 & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ * & O & O & \cdots & R_{K-1} \\ \hline \bar{S}_K - z\bar{T}_K & O & O & \cdots & O \end{array} \right]$$

which is orthogonally similar to the original system pencil  $\hat{S}(z)$ . Since the matrices  $R_i$  have full row rank, the subpencil  $\bar{S}_K - z\bar{T}_K$  contains all finite zeros of the original pencil. This pencil has column dimension  $r_1 + 1$  and row dimension at most  $r_1 + 1$ . To compute the finite zeros of the periodic system, we can apply now a general algorithm, like that of [11], to compute the finite eigenvalues of this low order subpencil. This method requires for a system with constant state dimension  $r = r_i$ , about  $2 \times (13.3K + 27.5)r^3$  flops in the worst case.

### 3.4 Computation of gain

To compute the gain, the main computation is to solve a potentially large order linear system  $Hw = g$ , with

$$H = \begin{bmatrix} H_{11} & & & H_{1,K} \\ H_{21} & H_{22} & & \\ & \ddots & \ddots & \vdots \\ & & H_{K,K-1} & H_{K,K} \end{bmatrix}, \quad g = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_K \end{bmatrix}$$

where  $H_{ii} = I_{r_i}$  for  $i = 1, \dots, K$ ,  $H_{i+1,i} = -\hat{A}_i$  for  $i = 1, \dots, K-1$ ,  $H_{1,K} = -A_K/z_0$ ,  $g_1 = \hat{b}_K/z_0$ , and  $g_i = \hat{b}_{i-1}$  for  $i = 2, \dots, K$ . The solution vector  $w = [w_1^T \ w_2^T \ \cdots \ w_K^T]^T$  is obtained in a partitioned form similar to that of  $g$ .

To solve  $Hw = g$  we use the standard Gaussian elimination method [4] to compute first the LU factorization of  $H$  as  $PH = LU$ , 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  $w$  is computed as  $w = U^{-1}L^{-1}Pg$ . For the particular structure of  $H$  above,  $L$  is block-bidiagonal, and  $U$  has nonzero blocks only on the diagonal, on first supra-diagonal as well as in its last block column. The following algorithm fully exploits the sparse structure of  $H$  and combines 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  $L^{-1}Pg$ .

### Algorithm.

**if**  $K = 1$ , **then** solve  $(I_{r_1} - \widehat{A}_1/z_0)w = g$  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  $P_i \begin{bmatrix} H_{ii} \\ H_{i+1,i} \end{bmatrix} = L_i \begin{bmatrix} U_{ii} \\ O \end{bmatrix}$

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

**end**

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

Solve  $U_{KK}w_K = g_K$  and  $U_{K-1,K-1}w_{K-1} = g_{K-1} - U_{K-1,K}w_K$ .

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

Solve  $U_{ii}w_i = g_i - U_{i,i+1}w_{i+1} - U_{i,K}w_K$

**end**

The main computations in this algorithm are the  $K-1$  successive LU decompositions of  $(r_i+r_{i+1}) \times r_i$  matrices and the application of  $r_i$  elementary transformations to  $(r_i+r_{i+1}) \times (r_{i+1}+r_K+1)$  matrices. For a constant system with dimension  $r = r_i$ , this algorithm performs about  $5r^3/6$  flops to compute the LU-decomposition of a  $2r \times r$  matrix and additionally,  $2r^3$  flops to apply  $r$  elementary transformations [4]. Thus, the solution of the equation  $Hw = g$  with the above algorithm can be computed with about  $\max\{3(K-1)r^3, r^3/3\}$  flops.

## 4 Conclusion

A reliable numerical algorithm has been proposed to compute the TFM corresponding to lifted representations of periodic systems. The proposed algorithm is generally applicable to periodic systems with time-varying state dimensions. No assumption on the minimality of the original representation is needed. The individual entries of the TFM are obtained in a cancelled, minimal *zeros-poles-gain* form. The computation of minimal realizations is based on accuracy enhancing balancing-free square-root techniques. All subsequent computations of poles, zeros and gains are performed using numerically stable algorithms. The computational complexity of the proposed algorithm for one element is  $O(Kn^3)$  and is much less than the computational complexity  $O(K^3n^3)$  resulting when applying existing procedures [10] directly to the *stacked lifted representation* of the periodic system.

## References

- [1] A. W. Bojanczyk, G. Golub, and P. Van Dooren. The periodic Schur decomposition. Algorithms and applications. In F. T. Luk (Ed.), *Proceedings SPIE Conference*, vol. 1770, pp. 31–42, July 1992.
- [2] P. Colaneri and S. Longhi. The realization problem for linear periodic systems. *Automatica*, 31:775–779, 1995.
- [3] D. S. Flamm. A new shift-invariant representation of periodic linear systems. *Systems & Control Lett.*, 17:9–14, 1991.
- [4] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, 1989.
- [5] O. M. Grasselli and S. Longhi. Zeros and poles of linear periodic discrete-time systems. *Circuits, Systems and Signal Processing*, 7:361–380, 1988.
- [6] O. M. Grasselli and S. Longhi. Pole-placement for nonreachable periodic discrete-time systems. *Math. Control Signals Syst.*, 4:439–455, 1991.
- [7] J. J. Hench and A. J. Laub. Numerical solution of the discrete-time periodic Riccati equation. *IEEE Trans. Autom. Control*, 39:1197–1210, 1994.
- [8] R. A. Meyer and C. S. Burrus. A unified analysis of multirate and periodically time-varying digital filters. *IEEE Trans. Circuits and Systems*, 22:162–168, 1975.
- [9] B. Park and E. I. Verriest. Canonical forms for discrete-time periodically time varying systems and a control application. *Proc. of CDC'89, Tampa*, pp. 1220–1225, 1989.
- [10] A. Varga. Computation of transfer function matrices of generalized state-space models. *Int. J. Control*, 50:2543–2561, 1989.
- [11] A. Varga. Computation of Kronecker-like forms of a system pencil: Applications, algorithms and software. *Proc. CACSD'96 Symposium, Dearborn, MI*, pp. 77–82, 1996.
- [12] A. Varga. Balancing related methods for minimal realization of periodic systems. *Systems & Control Lett.*, 36:339–349, 1999.
- [13] A. Varga and P. Van Dooren. Computational methods for periodic systems - an overview. *Proc. of IFAC Workshop on Periodic Control Systems, Como, Italy*, pp. 171–176, 2001.
- [14] A. Varga and P. Van Dooren. Computation of zeros of periodic systems. *Proc. of CDC'2002, Las Vegas, Nevada*, 2002.
- [15] A. Varga and V. Sima. A numerically stable algorithm for transfer-function matrix evaluation. *Int. J. Control*, 33:1123–1133, 1981.