

Computational issues for linear periodic systems: paradigms, algorithms, open problems

Andreas Varga*

German Aerospace Center, DLR-Oberpfaffenhofen, Institute of System Dynamics and Control, D-82234 Weßling, Germany

(Received 28 September 2012; final version received 31 January 2013)

Linear periodic systems originate in various control fields involving periodic phenomena. In the beginnings of algorithmic developments for periodic systems, computational detours have been often employed to reduce the computational problems for periodic systems to those for standard systems. Well-known techniques are the employment of lifted representations in discrete-time or the use of periodic generators in continuous-time. New computational paradigms evolved later, whose main ingredients are numerically reliable and efficient algorithms for manipulating matrix products without forming explicitly them, reduction of large-scale structured matrix pencils without building the underlying pencils, computing with discrete-time periodic system models with time-varying state dimensions, or solving periodic matrix differential equations using multiple shooting techniques. The paper will give a succinct overview of all these new developments and will point out some of still existing open computational problems.

Keywords: periodic systems; linear systems; computational methods

1. Introduction

Linear periodic system descriptions appear in various control applications involving periodic phenomena, for example, satellite attitude control, wind turbine disturbance attenuation, vibration reduction in helicopter forward flight, multi-rate sampled-data system analysis and design. The analysis and synthesis of periodic control systems, both in continuous- and discrete-time, represents nowadays a mature domain of the systems and control theory as documented in the recent monograph (Bittanti & Colaneri, 2009). For discrete-time periodic systems, this book provides a solid theoretical basis of various modelling and analysis problems such as lifting techniques, stability, reachability/observability/minimality, poles/zeros, system norms, and also addresses the basic aspects of periodic filter and controller synthesis problems. Unfortunately, the book contains very few references to the associated computational problems and their algorithmic solutions.

In the beginnings of algorithmic developments for periodic systems, computational detours have been often employed to reduce the computational problems for periodic systems to those for standard systems, for which a powerful arsenal of computational methods exist (Patel, Laub, & Van Dooren, 1994; Varga, 2004b). We mention in this context, two well-known detour techniques, namely the use of lifted representations in discrete-time and the method of periodic generators in continuous-time. Both techniques have severe limitations in terms of numerical reliability and computational efficiency.

To face the challenging computational problems raised by practical applications, new computational paradigms evolved later, whose main ingredients are numerically reliable and efficient algorithms for computing eigenvalues of matrix products without forming explicitly these products, reduction of large-scale structured matrix pencils without building these pencils, computing with discrete-time periodic system models with time-varying state dimensions, or solving periodic matrix differential equations (e.g. Lyapunov, Sylvester, Riccati) using multiple shooting techniques in conjunction with structure exploiting (symplectic) integration methods. In this paper, we present an overview of these new paradigms and point out several immediate computational applications. In the conclusion, we will mention some of still existing open computational problems.

2. Computational detours

Computational detours are often used when proposing computational algorithms for control systems. The typical approach is to reformulate the computational problems in a form, for which computational techniques are known, and then recover the problem solution from that of the reformulated problem. The two main reasons for using computational detours are the lack of a suitable algorithm for solving the original problem or, more frequently, the lack of awareness of authors of better techniques and tools. In this section, we present two of the commonly used computation detours to solve computational problems for periodic

*Email: andreas.varga@dlr.de

systems, namely computing with lifted representations in discrete-time and employing (single point) periodic generators in continuous-time. Both techniques have severe limitations in terms of numerical reliability and computational efficiency, and cannot generally serve as a basis for developing satisfactory computational algorithms.

2.1 Lifting-based techniques

In the case of *linear time-periodic* (LTP) systems, a common detour is to use a certain lifting technique to arrive at an equivalent *linear time-invariant* (LTI) system based problem formulation, then apply a suitable computational method available for LTI systems, and finally, recover the solution in an LTP representation. As we will see, difficulties can arise in both of the last two computational steps and are related to worsened problem conditioning, large dimensions, and numerical instability. Unfortunately, the use of lifting techniques is even nowadays advocated by some authors (Zhang, Ding, & Liu, 2012).

The *standard discrete-time lifting* proposed by Meyer and Burrus (1975) for standard LTP systems with constant state vector dimension can be extended to the more general case of systems with time-varying state dimensions of the form

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

where $x_k \in \mathbb{R}^{n_k}$ is the state vector at time k , $u_k \in \mathbb{R}^m$ is the input vector at time k , $y_k \in \mathbb{R}^p$ is the output vector at time k and for all k , $A_k = A_{k+N}$, $B_k = B_{k+N}$, $C_k = C_{k+N}$ and $D_k = D_{k+N}$ are N -periodic matrices of appropriate dimensions. Let $\Phi_A(j, i) := A_{j-1} \cdots A_{i+1} A_i$, with $\Phi_A(i, i) := I_{n_i}$, be the transition matrix over the time interval $[i, j]$. The monodromy matrix is the $n_k \times n_k$ transition matrix over one period; thus, $\Psi_A(k) := \Phi_A(k + N, k)$. The n_k eigenvalues of $\Psi_A(k)$ are called the *characteristic multipliers* at time moment k of the N -periodic matrix A_k . It is well known that the characteristic multipliers can be used to assess the *stability* of the system (1) as follows: the system is *stable* if all characteristic multipliers have moduli less than 1 and *unstable* otherwise.

The lifted system is defined according to Meyer and Burrus (1975) with the following input, output and state vectors:

$$u_{k;h}^L = \begin{bmatrix} u_{k+hN} \\ u_{k+hN+1} \\ \vdots \\ u_{k+hN+N-1} \end{bmatrix}, \quad y_{k;h}^L = \begin{bmatrix} y_{k+hN} \\ y_{k+hN+1} \\ \vdots \\ y_{k+hN+N-1} \end{bmatrix},$$

$$x_{k;h}^L = x_{k+hN}$$

which lead to a standard (lifted) LTI system at the k th time moment of the form

$$\begin{aligned} x_{k;h+1}^L &= F_k^L x_{k;h}^L + G_k^L u_{k;h}^L, \\ y_{k;h}^L &= H_k^L x_{k;h}^L + L_k^L u_{k;h}^L. \end{aligned} \tag{2}$$

The matrices of the state-space representation (2) are

$$\begin{aligned} F_k^L &= \Phi_A(k + N, k) \\ G_k^L &= [\Phi_A(k + N, k + 1) B_k \cdots B_{k+N-1}] \\ H_k^L &= \begin{bmatrix} C_k \\ \vdots \\ C_{k+N-1} \Phi_A(k + N - 1, k) \end{bmatrix} \\ L_k^L &= \begin{bmatrix} D_k & 0 & \cdots & 0 \\ L_{k;2,1} & D_{k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{k;N,1} & L_{k;N,2} & \cdots & D_{k+N-1} \end{bmatrix} \end{aligned} \tag{3}$$

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

The lifted representation (2) is a LTI standard system with state vector dimension n_k , mN inputs and pN outputs. For this system, it is possible to use the whole arsenal of available computational tools for LTI systems. The main disadvantages of using the standard lifting are a general worsening of the problem conditioning and precluding reliable numerical computations, both due to the need to form explicitly matrix products to build the lifted system matrices in (3). Moreover, recovering the problem solutions in the form of periodic system realisations as in (1) is not always straightforward or even possible, because of the need to fulfil the causality constraints (as expressed by the lower block triangular structure of L_k^L). In spite of these aspects, the lifted representation (2) has an important theoretical and conceptual value, by allowing the study of basic LTP system properties using the corresponding LTI system concepts.

The so-called *stacked discrete-time lifting* proposed by Grasselli and Longhi (1991b) for standard periodic systems can be easily extended to the most general periodic discrete-time linear descriptor systems of the form

$$\begin{aligned} E_k x_{k+1} &= A_k x_k + B_k u_k, \\ y_k &= C_k x_k + D_k u_k, \end{aligned} \tag{4}$$

where $E_k \in \mathbb{R}^{v_k \times n_{k+1}}$, $A_k \in \mathbb{R}^{v_k \times n_k}$, $B_k \in \mathbb{R}^{v_k \times m}$, $C_k \in \mathbb{R}^{p \times n_k}$, $D_k \in \mathbb{R}^{p \times m}$ are N -periodic matrices and $\sum_{k=1}^N v_k = \sum_{k=1}^N n_k$. The lifting introduced by Grasselli and Longhi

(1991b) uses

$$u_{k;h}^S = \begin{bmatrix} u_{k+hN} \\ u_{k+hN+1} \\ \vdots \\ u_{k+hN+N-1} \end{bmatrix}, \quad y_{k;h}^S = \begin{bmatrix} y_{k+hN} \\ y_{k+hN+1} \\ \vdots \\ y_{k+hN+N-1} \end{bmatrix},$$

$$x_{k;h}^S = \begin{bmatrix} x_{k+hN} \\ x_{k+hN+1} \\ \vdots \\ x_{k+hN+N-1} \end{bmatrix}$$

as lifted variables, leading to a LTI descriptor system representation of the form

$$\begin{aligned} E_k^S x_{k;h+1}^S &= F_k^S x_{k;h}^S + G_k^S u_{k;h}^S, \\ y_{k;h}^L &= H_k^S x_{k;h}^S + L_k^S u_{k;h}^S, \end{aligned} \quad (5)$$

where

$$F_k^S - zE_k^S = \begin{bmatrix} A_k & -E_k & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & A_{k+N-2} & -E_{k+N-2} \\ -zE_{k+N-1} & 0 & \cdots & 0 & A_{k+N-1} \end{bmatrix}, \quad (6)$$

$$\begin{aligned} G_k^S &= \text{diag} \{B_k, \dots, B_{k+N-1}\}, \\ H_k^S &= \text{diag} \{C_k, \dots, C_{k+N-1}\}, \\ L_k^S &= \text{diag} \{D_k, \dots, D_{k+N-1}\}. \end{aligned}$$

The lifted representation (5) is a LTI descriptor system with state vector dimension $\sum_{i=1}^N n_i$, mN inputs and pN outputs. Interestingly, the transfer function matrices of the *stacked lifted system* (5) and *standard lifted systems* (2) for the standard periodic system (1) are the same. Note that a third lifting scheme introduced by Park and Verriest (1989) does not share this useful property.

There exist many numerically reliable algorithms and accompanying software tools for descriptor systems which can be applied to solve computational problems for the lifted LTI descriptor system (5) (see Varga, 2000, for an overview of algorithms and software). However, using these algorithms leads to excessive computational efforts, because of a computational complexity of $O(N^3 \bar{n}^3)$ instead of the expected $O(N \bar{n}^3)$, where $\bar{n} = \max_{i=1}^N n_i$. Since the structures of the matrices in the lifted representation (5) are not generally preserved during the computations, the determination of a standard or descriptor periodic system realisation is a delicate final step. For example, if the result is a non-causal system, then apparently there exists even no

algorithm to compute the corresponding periodic descriptor system realisation.

2.2 Periodic generator methods

Consider the continuous-time LTP system of the form

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t), \\ y(t) &= C(t)x(t) + D(t)u(t), \end{aligned} \quad (7)$$

where $x(t) \in R^n$, $u(t) \in R^m$ and $y(t) \in R^p$ are the state, control and output vectors, and $A(t)$, $B(t)$, $C(t)$ and $D(t)$ are T -periodic matrices [e.g. $A(t) = A(t + T)$ for all t]. Let $\Phi_A(t, \tau)$ be the transition matrix over the time interval $[\tau, t]$. The monodromy matrix is the $n \times n$ transition matrix over one period; thus, $\Psi_A(\tau) := \Phi_A(\tau + T, \tau)$. The n eigenvalues of $\Psi_A(\tau)$ are called the *characteristic multipliers* of the periodic matrix $A(t)$ and are independent of the time moment τ . The characteristic multipliers can be used to assess the *stability* of the system (7) in a similar way to that for the discrete-time LTP system (1). The *characteristic exponents* $\mu(A)$ of the T -periodic matrix $A(t)$ are defined in terms of the eigenvalues $\lambda(\Psi_A(0))$ as

$$\mu(A) = \frac{1}{T} \log \lambda(\Psi_A(0)).$$

These quantities play the role of eigenvalues of time-varying matrices and it is a well-known fact that they are invariant to a Lyapunov-type similarity transformation (see the example at the end of this subsection).

In addressing various computational problems for continuous-time periodic systems, we encounter periodic matrix differential equations, whose periodic solutions can be used to solve analysis and synthesis problems. For example, periodic state feedback stabilisation problems can be addressed by solving for the T -periodic stabilising solution $X(t)$ the *periodic Riccati differential equation* (PRDE)

$$\dot{X}(t) = A(t)X(t) + X(t)A^T(t) + R(t) - X(t)Q(t)X(t), \quad (8)$$

where $R(t)$ and $Q(t)$ are suitable T -periodic symmetric matrices. Two related matrix differential equations, which are also useful for solving stabilisation problems, are the *periodic Lyapunov differential equation* (PLDE)

$$\dot{X}(t) = A(t)X(t) + X(t)A^T(t) + R(t) \quad (9)$$

and the *periodic Sylvester differential equation* (PSDE)

$$\dot{X}(t) = A(t)X(t) + X(t)F(t) + G(t), \quad (10)$$

where $F(t)$, $G(t)$ and $R(t)$ are T -periodic matrices.

The periodic generator method can be used to solve the above equations, by exploiting the T -periodicity of the

solution $X(t)$ to obtain standard discrete-time algebraic matrix Riccati, Lyapunov or Sylvester equations fulfilled by $X(0) = X(T)$. After solving these equations to obtain the *periodic generator* $X(0)$, we can subsequently integrate the differential equation in question to compute the solution $X(t)$ in an arbitrary point $t \in [0, T]$.

All derivations of the respective algebraic matrix equations involve an explicit computation of a monodromy matrix of a specially defined periodic system. Therefore, difficulties are to be expected on applying this method in the case when an unstable $A(t)$ leads to unstable integration of the underlying *ordinary differential equations* (ODEs). Also, severe accuracy losses may result for large periods T . To illustrate the difficulties related to the computation of the monodromy matrix, we consider a very simple example involving the computation of the characteristic exponent of a 2×2 periodic matrix $A(t)$.

Example (P. Colaneri, personal communication, 2004): Consider

$$A(t) = \begin{bmatrix} 0 & 1 \\ -2\dot{\alpha}(t) & 6 - 2\alpha(t) \end{bmatrix},$$

with $\alpha(t) = 15 + 5 \sin t$ and $T = 2\pi$. With

$$P(t) = \begin{bmatrix} 1 & 0 \\ 6 - 2\alpha(t) & 1 \end{bmatrix}$$

we perform the Lyapunov transformation

$$\tilde{A}(t) := P^{-1}(t)A(t)P(t) - P^{-1}(t)\dot{P}(t) = \begin{bmatrix} 6 - 2\alpha(t) & 1 \\ 0 & 0 \end{bmatrix}.$$

The characteristic exponents of $\tilde{A}(t)$ [and thus also of $A(t)$] are $\mu_1 = 0$ and $\mu_2 = -24$.

Using numerical integration with a relative accuracy tolerance $reltol = 10^{-10}$, we determined $\Psi_A(0) = \Phi_A(T, 0)$ by integrating the matrix differential equation satisfied by the transition matrix $\Phi_A(t, \tau)$:

$$\frac{\partial \Phi_A(t, \tau)}{\partial t} = A(t)\Phi_A(t, \tau), \quad \Phi_A(\tau, \tau) = I_n. \quad (11)$$

We obtained the computed values $\bar{\mu}$ of the characteristic exponents as

$$\bar{\mu}_1 = -3.9 \times 10^{-7}, \quad \bar{\mu}_2 = -2.05 \dots$$

which shows no accurate digits in the computed value of μ_2 .

This example clearly illustrates that even in this very simple 2×2 case, severe numerical difficulties may be encountered due to the presence of a moderate stiffness in the underlying ODE (11) used for computing the monodromy matrix.

3. Case studies and computational challenges

In this section, we present four applications involving periodic system descriptions, which lead to challenging computational problems. The solution of these problems by applying the described computational detours in the previous section is highly questionable. In each case, we emphasise the expected numerical difficulties and indicate possible computational approaches to overcome them, partly relying on the new computational paradigms described in the next section.

3.1 Satellite attitude control

The linearised dynamics of a *low earth orbiting* (LEO) satellite with magnetic actuation employed by Pittelkau (1993) have the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B(t)u(t), \\ y(t) &= Cx(t), \end{aligned}$$

where $x(t) \in R^4$, $u(t) \in R$ and $y(t) \in R^2$. The 4×4 state matrix A has only imaginary eigenvalues, while $B(t)$ is a T -periodic matrix, that is, $B(t + T) = B(t)$ for all t . The periodic nature of $B(t)$ is a consequence of the interaction of the revolving actuation system (based on magnetic coils) with the magnetic field of the Earth.

To stabilise the satellite dynamics, a periodic discrete-time output feedback controller has been determined by Varga and Pieters (1998). The employed discretised periodic model for a sampling period Δ has been obtained in the form (1), where $x_k := x((k - 1)\Delta)$, $u_k = u((k - 1)\Delta)$, $y_k := y((k - 1)\Delta)$ and

$$\begin{aligned} A_k &= \exp(A\Delta), \quad B_k = \int_{(k-1)\Delta}^{k\Delta} e^{A(k\Delta-\tau)} B(\tau) d\tau, \\ C_k &= C, \quad D_k = 0. \end{aligned}$$

For the resulting periodic discrete-time system of period $N = T/\Delta$, a periodic output feedback control law of the form

$$u_k = F_k y_k$$

has to be determined such that the periodic state matrix $A_k + B_k F_k C_k$ of the closed-loop system is stable (i.e. has all characteristic multipliers with moduli less than 1).

The computational approach proposed by Varga and Pieters (1998) relies on the minimisation of a quadratic performance criterion and has been solved by using a gradient descent optimisation approach, based on explicit expressions derived for the gradients. A main computational ingredient for efficient gradient evaluation is the solution of periodic Lyapunov equations for which reliable numerical algorithms are available (Varga, 1997).

The main computational challenge in solving this problem is caused by the large revolving period of the LEO satellite of about $T = 5400$ s (i.e. about 1.5 hours), which for a typical value of the sampling period $\Delta = 10$ s leads to a large period $N = 540$ for the discrete-time periodic system. In this case, the unconstrained optimisation problem to solve the periodic stabilisation has a large dimension, involving $2 \times 540 = 1080$ free variables.

3.2 Wind turbine stability analysis

The prediction of the damping characteristics of a wind turbine in closed-loop operation requires building a high-fidelity model of the control system (Riziotis, Politis, Voutsinas, & Chaviaropoulos, 2007). This model includes the full dynamics of the generator combined with the aeroelastic equations of motion for the complete wind turbine, as well as controller components (e.g. for pitch and torque regulation) and various filters. After the linearisation of the nonlinear model in a periodic equilibrium state, linear stability analysis techniques can be applied to study the damping characteristic of the closed-loop control system.

In the absence of external disturbance inputs, the linearised closed-loop control system has the continuous-time periodic autonomous form

$$\dot{x}(t) = A(t)x(t), \quad (12)$$

where $x(t) \in R^n$ is the state vector and the state matrix $A(t)$ is T -periodic. For a proper operation of the wind turbine, the closed-loop state system must be stable, which is equivalent to the requirement that $A(t)$ has only characteristic exponents with negative real parts, or equivalently the characteristic multipliers lie in the interior of the unit circle in the origin.

The Floquet–Lyapunov method for stability analysis checks that all characteristic multipliers [i.e. the eigenvalues of the monodromy matrix $\Psi_A(0)$] have moduli less than 1. $\Psi_A(0)$ can be computed by integrating the matrix differential equation satisfied by the transition matrix $\Phi_A(t, \tau)$ in (11) for $\tau = 0$ and $t \in [0, T]$ to obtain $\Psi_A(0) = \Phi_A(T, 0)$.

Due to the lightly damped structure of typical wind turbines, most of the eigenvalues of $\Psi_A(0)$ will have nearly unity magnitudes. Therefore, to guarantee the robustness of this check, the eigenvalue computation must provide accurate results. The practical difficulties of employing the Floquet analysis lie not only in the very large state dimension (typical values about $n = 1200$), but also in the fact that due to the specifics of the model building (see e.g. Jonkman & Buhl, 2005), only N values of $A(i\Delta)$, for $i = 0, 1, \dots, N-1$, are available over one time period $[0, T]$, where $N\Delta = T$. Typical values are $T = 6$ s and $\Delta = 0.2$ s which lead to $N = 30$.

To determine $\Phi_A(T, 0)$ we need to integrate (11) for $t \in [0, T]$, where at each integration step it is necessary

to evaluate $A(t)$ at arbitrary values of t . For this, the use of suitable interpolation techniques is necessary (e.g. cubic splines based interpolation).

Alternatively, we can approximate $\Phi_A(T, 0)$ in a product form. Consider the time values $t_i = i\Delta$ for $i = 0, 1, \dots, N$. Then, $\Phi(T, 0)$ can be expressed as the product of N transition matrices

$$\Phi_A(T, 0) = \Phi_A(t_N, t_{N-1}) \cdots \Phi_A(t_2, t_1) \Phi_A(t_1, t_0), \quad (13)$$

where each $\Phi_A(t_i, t_{i-1})$ is determined by integrating (11) for $t \in [t_{i-1}, t_i]$. Since only the values of $A(t_i)$ are available, we can approximate $\Phi_A(t_i, t_{i-1})$ using the trapezoidal rule as

$$\Phi_A(t_i, t_{i-1}) \approx I_n + \frac{A(t_i) + A(t_{i-1})}{2} \Delta.$$

To compute the eigenvalues of the resulting $\Phi_A(T, 0)$, it is even possible to avoid forming of the matrix product by employing the periodic Schur form method of Bojanczyk, Golub, and Van Dooren (1992), and Hench and Laub (1994). This approach is described in Section 4.1.

3.3 Vibration reduction in helicopter forward flight

The attenuation and even elimination of rotor-induced persistent vibrations in helicopters is one of the basic goals of helicopter control. These vibrations manifest primarily in a forward flight, when various asymmetries lead to considerable oscillating aerodynamic loads with higher harmonic components. The linearised helicopter model in forward flight can be expressed in the form

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) + E(t)d(t), \\ y(t) &= C(t)x(t) + D(t)u(t) + F(t)d(t), \end{aligned} \quad (14)$$

where $x(t) \in R^n$, $u(t) \in R^m$, $d(t) \in R^q$ and $y(t) \in R^p$ are the state, control, disturbance and output vectors, respectively, and $A(t)$, $B(t)$, $E(t)$, $C(t)$, $D(t)$ and $F(t)$ are T -periodic matrices. In this model, the disturbance input $d(t)$ consists of higher harmonics aerodynamic loads.

The main control task is the rotor stabilization using, for example, multi-blade control approaches. Of increased importance is, however, the rejection (exact or approximative) of a periodic disturbance with known harmonic components which are present in the input vector d . For this purpose, advanced control architectures relying on the internal model principle and periodic dynamic control laws are candidates to provide the best possible solution.

A possible computational approach for solving the rotor stabilisation control problem simultaneously with the vibration rejection (attenuation) aspect is to use a periodic output feedback controller jointly with an internal model (formed

as a collection of notch filters). Since both components of the controller can be parameterised by constant gains (see e.g. Viganò, Bergamasco, Lovera, & Varga, 2010), it is possible to use an optimisation-based tuning procedure for the closed-loop stabilisation, which automatically also solves the disturbance rejection problem. The stabilisation can be performed both in continuous-time and in discrete-time, by minimising suitable measures of the closed-loop system stability degree (e.g. the largest magnitude of characteristic multipliers). The open-loop model extended with the internal model can be stabilised using, for example, linear-quadratic regulator stabilisation techniques. The resulting periodic state feedback can be implemented with the help of a periodic state estimator. A similar approach is described by Arcara, Bittanti, and Lovera (2000).

The computational challenge to solve this problem lies first in the complexity of the open-loop model, and secondly in the need of solving a difficult periodic stabilisation problem. Regarding the first aspect, a realistic helicopter model (BO105) is described by Konstanzer (2001). The model has been built using the dedicated helicopter modelling tool CAMRAD II, described by Johnson (1992). The order of this model is relatively large, $n = 56$, and the system matrices in (14) are only available for $N = 48$ azimuth angles. A typical revolving period is $T = 0.15$ s. Analytic expressions of the system matrices can be determined in the form of Fourier series, whose coefficients can be computed by employing the fast Fourier transform. For stabilisation using a periodic output feedback, the control law can be chosen as

$$u(t) = F(t)y(t) + F_M(t)x_M(t),$$

where $F(t)$ and $F_M(t)$ are periodic matrices to be determined and $x_M(t)$ is the output of an internal model

$$\dot{x}_M(t) = A_M x_M(t) + B_M y(t), \quad (15)$$

where A_M and B_M are known constant matrices, with A_M having purely imaginary eigenvalues reflecting the frequencies to be rejected. The closed-loop system state matrix is [assuming for simplicity $D(t) = 0$]

$$A_c(t) = \begin{bmatrix} A(t) + B(t)F(t)C(t) & B(t)F_M(t) \\ B_M C(t) & A_M \end{bmatrix}.$$

For stabilisation purposes, we can use the techniques described in Viganò et al. (2010), where the periodic matrices $F(t)$ and $F_M(t)$ are parameterised with constant matrices, which represent the coefficient matrices of truncated Fourier series. A brute force optimisation could aim at the minimisation of the maximum real part of the characteristic exponent $\mu(A_c)$. This can be followed by the application of the optimal output feedback stabilisation techniques of Viganò et al. (2010) [after suitably rewriting $A_c(t)$]. Decentralised feedback structures, as proposed by Konstanzer

(2001), can be easily accommodated with this approach. Both steps are computationally very intensive, involving either repeatedly performing the Floquet–Lyapunov stability analysis step or repeatedly solving PLDEs for gradient evaluations (see Varga, 2005, for suitable methods based on multiple-shooting techniques and also Section 4.4).

3.4 Multi-rate sampling of continuous-time systems

The multi-rate sampling of continuous-time LTI systems of the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \quad (16)$$

with $x(t) \in R^n$, $u(t) \in R^m$ and $y(t) \in R^p$, involves using a basic sampling period Δ to sample the j th input u_j with sampling period $k_j \Delta$ and the i th output y_i with sampling period $l_i \Delta$, where both k_j and l_i are integers. It is well known (see e.g. Bittanti & Colaneri, 2009) that the resulting multi-rate sampled-data system can be expressed as a LTP system of the form (1), where $u_k := u((k-1)\Delta) \in R^m$, $y_k := y((k-1)\Delta) \in R^p$, $x_k \in R^{n_k}$, and A_k , B_k , C_k and D_k are N -periodic matrices, with N being the least common multiple of k_j , $j = 1, \dots, m$ and l_i , $i = 1, \dots, p$. The state dimensions $n_k \geq n$ are usually chosen constant over a period of N time steps, as $n_k = n + m$. The expressions of the system matrices in (1) are (Bittanti & Colaneri 2009)

$$\begin{aligned} A_k &= \begin{bmatrix} \bar{A} & \bar{B} S_k \\ 0 & I_m - S_k \end{bmatrix}, \quad B_k = \begin{bmatrix} \bar{B} S_k \\ S_k \end{bmatrix}, \\ C_k &= T_k C, \quad D_k = T_k D, \end{aligned} \quad (17)$$

where $\bar{A} = e^{A\Delta}$, $\bar{B} = \int_0^\Delta e^{A(\Delta-\tau)} B d\tau$, S_k and T_k are diagonal matrices with diagonal elements defined as

$$S_{k;j,j} = \begin{cases} 1, & \text{if } k \text{ is a multiple of } k_j \\ 0, & \text{otherwise} \end{cases},$$

$$T_{k;i,i} = \begin{cases} 1, & \text{if } k \text{ is a multiple of } l_i \\ 0, & \text{otherwise} \end{cases}$$

To a less extent is known that for a minimal system (i.e. reachable and observable) in (16), the resulting periodic system (1) with the matrices in (17) is usually not minimal, and more exactly, not observable (Longhi, 1994). The lack of observability is an intrinsic effect of the employed sampling and hold mechanisms. Therefore, in general, reachable and observable periodic representations of the multi-rate sampled-data systems involve time-varying state dimensions. An even more refined modelling would lead to periodic system models with time-varying dimensions of input and output vectors, which arise by removing at

each k the input components corresponding to zero columns in S_k and output components corresponding to zero rows in T_k .

In developing generally applicable numerical algorithms for discrete-time linear periodic systems, a main aspect raised by Varga and Van Dooren (2001) is to allow for time-varying state dimensions. This requirement raises several computational and implementation challenges for the analysis and synthesis algorithms for periodic systems and therefore may be one of the reasons why time-varying state dimensions received little consideration even from a theoretical point of view. For example, in the textbook of Bittanti and Colaneri (2009), time-varying dimensions are discussed only in the context of minimal realisations of periodic systems. In Sections 4.2 and 4.3, we discuss some general algorithms for solving computational problems involving time-varying state dimensions.

4. New computational paradigms

Before discussing the new computational paradigms for computational algorithms for periodic systems, we first briefly recall three key requirements for a *satisfactory* numerical algorithm for periodic systems as formulated by Varga and Van Dooren (2001): generality, numerical stability and efficiency. A *general* algorithm is one which has no limitations for its applicability of any technical nature. For any of the periodic system representations (1), (4) or (7), it should be able to handle the most general class of periodic systems. For example, a pole assignment algorithm for a standard periodic system (1) which is able to assign *only* distinct poles should not be considered satisfactory. Since the minimal realisation of a periodic system has in general time-varying state dimensions (Colaneri & Longhi, 1995), it is also highly desirable to develop algorithms for the analysis and design of periodic systems which are able to handle systems with time-varying dimensions.

Numerical stability (more precisely, *backward stability*) of an algorithm means that the results computed by that algorithm are exact for slightly perturbed original data. As a consequence, a numerically stable algorithm applied to a well-conditioned problem will produce guaranteed accurate results. This is why numerical stability is a key feature for a satisfactory algorithm. A basic ingredient to achieve numerical stability is the use of orthogonal transformations wherever possible. The use of these transformations often leads to bounds for perturbations of the initial data which are equivalent to the cumulative effect of round-off errors occurring during the computations. This is a way to prove the numerical stability of such an algorithm. The main idea when developing numerically stable algorithms for periodic systems is to exploit the problem structure by applying only orthogonal transformations on the original problem data, and thereby trying to reduce the original problem to an equivalent one which is easier to solve. For this reason,

algorithms based on computational detours, such as those described in Section 2, cannot be satisfactory.

Because of the intrinsic complexity of several computational problems in systems theory, it is not always possible to develop numerically stable algorithms for them. Therefore, one often imposes this requirement only on the individual steps of the algorithm. Although this is not sufficient to guarantee numerical stability of the global algorithm, one can still expect that it will perform accurately on well-conditioned problems.

The *efficiency* of an algorithm involves two main aspects: avoiding extensive storage use and keeping the computational complexity level as low as possible. For discrete-time periodic systems, the first requirement implies that the storage should be proportional to the amount of data defining the system, i.e. it should be $O(N\bar{n}(\bar{n} + m + p))$, where $\bar{n} = \max_{k=1}^N \{n_k\}$. Concerning the second requirement applied to a periodic system of period N , one requires a computational complexity of at most $O(N\bar{n}^3)$, that is, roughly N times the complexity of algorithms for standard linear systems. These requirements rule out all computational approaches based on lifted representations.

In what follows, we present the new computational paradigms, which emerged in the last decade in the sustained efforts to develop satisfactory numerical algorithms for periodic systems.

4.1 Paradigm 1: avoiding forming of matrix products

Algorithms to compute the eigenvalues of matrix products which are able to avoid forming explicitly these products by manipulating instead only their individual factors are instrumental in solving many computational problems for both continuous-time and discrete-time periodic systems. The main advantage of these techniques is a significantly increased reliability of the computations, which leads to increased guaranteed accuracy of the computed results. Besides the computation of characteristic exponents or characteristic multipliers, the underlying reduction algorithms (e.g. the computation of the periodic Schur form) often are used as a first computational step in several more involved algorithms, for example, the solution of various periodic discrete-time matrix equations (Riccati, Lyapunov, Sylvester).

The use of condensed forms of the system matrices, obtained under orthogonal transformations, is a basic ingredient for solving many computational problems in control (Van Dooren & Verhaegen, 1985). Accordingly, the system matrices are transformed to a particular coordinate system in which they have particular condensed forms, such that the solution of the original problem becomes straightforward. A frequently encountered computational problem when working with periodic systems is the computation of eigenvalues of the monodromy matrix. For an N -periodic

discrete-time system of the form (1), this involves the computation of eigenvalues of the matrix product

$$\Psi_A(0) = A_{N-1} \cdots A_1 A_0 .$$

For periodic systems with constant dimensions, the *periodic real Schur form* (PRSF) plays an important role in solving this problem. According to Bojanczyk et al. (1992), given the matrices $A_k \in \mathbb{R}^{n \times n}$, $k = 0, 1, \dots, N - 1$, there exist orthogonal matrices Z_k , $k = 0, 1, \dots, N - 1$, $Z_N := Z_0$, such that

$$\tilde{A}_k := Z_{k+1}^T A_k Z_k, \tag{18}$$

where \tilde{A}_{N-1} is in a quasi-upper triangular form with only 1×1 or 2×2 blocks on the diagonal [the so-called *real Schur form* (RSF)] and the matrices \tilde{A}_k for $k = 0, \dots, N - 2$ are upper triangular. Numerically stable algorithms to compute the PRSF have been proposed by Bojanczyk et al. (1992) and Hench and Laub (1994). By using these algorithms, we can determine the orthogonal matrices Z_k , $k = 0, \dots, N - 1$ to reduce $\Psi_A(0)$ to $\Psi_{\tilde{A}}(0) = Z_0^T \Psi_A(0) Z_0$ in the RSF, without forming explicitly the underlying matrix product. The characteristic multipliers are the eigenvalues of the reduced $\Psi_{\tilde{A}}(0)$ and can be simply computed, by forming explicitly the products of corresponding 1×1 or 2×2 diagonal blocks of the reduced matrices \tilde{A}_k .

In the case of a standard periodic system (1), the reduction of A_k to the PRSF can be interpreted as a periodic Lyapunov coordinate transformation $x_k = Z_k \tilde{x}_k$, to obtain the transformed periodic system

$$\begin{aligned} \tilde{x}_{k+1} &= \tilde{A}_k \tilde{x}_k + \tilde{B}_k u_k, \\ y_k &= \tilde{C}_k \tilde{x}_k + D_k u_k, \end{aligned} \tag{19}$$

with the transformed matrices

$$\tilde{A}_k := Z_{k+1}^{-1} A_k Z_k, \quad \tilde{B}_k := Z_{k+1}^{-1} B_k, \quad \tilde{C}_k := C_k Z_k . \tag{20}$$

For the orthogonal reduction of the N -periodic matrix A_k to the PRSF in (18), we have $Z_N^{-1} = Z_0^T$.

An extension of the PRSF addresses the case of square products of possibly non-square matrices, which can be used to compute the characteristic exponents for standard periodic systems (1) with time-varying state dimensions. In this case, the *extended periodic real Schur form* has been proposed by Varga (1999), with (quasi) upper trapezoidal matrices. Another extension of the PRSF considers quotient-products of the form $E_{N-1}^{-1} A_{N-1} \cdots E_1^{-1} A_1 E_0^{-1} A_0$, which can be used to compute the characteristic multipliers for descriptor periodic systems of the form (4). The corresponding condensed form is the *generalized periodic real Schur form* of a periodic matrix pair (A_k, E_k) , which extends the PRSF to so-called regular periodic systems (with square and non-singular E_k ,

see e.g. Bojanczyk et al., 1992). Given the matrices $A_k, E_k \in \mathbb{R}^{n \times n}$, $k = 0, 1, \dots, N - 1$, there exist orthogonal matrices $Z_k, Q_k \in \mathbb{R}^{n \times n}$, $k = 0, 1, \dots, N - 1$, $Z_N := Z_0$, such that the matrices

$$\tilde{A}_k := Q_k^T A_k Z_k, \quad \tilde{E}_k := Q_k^T E_k Z_{k+1}, \tag{21}$$

are all upper triangular, excepting \tilde{A}_{N-1} , which is in a quasi-upper triangular form.

In the case of a descriptor periodic system (4), the above reduction can be interpreted as a periodic Lyapunov coordinate transformation $x_k = Z_k \tilde{x}_k$, to obtain the transformed periodic system

$$\begin{aligned} \tilde{E}_k \tilde{x}_{k+1} &= \tilde{A}_k \tilde{x}_k + \tilde{B}_k u_k, \\ y_k &= \tilde{C}_k \tilde{x}_k + D_k u_k, \end{aligned} \tag{22}$$

with the transformed matrices

$$\begin{aligned} \tilde{A}_k &:= Q_k^T A_k Z_k, \quad \tilde{E}_k := Q_k^T E_k Z_{k+1}, \\ \tilde{B}_k &:= Q_k^T B_k, \quad \tilde{C}_k := C_k Z_k . \end{aligned}$$

Example (P. Colaneri, personal communication, 2004) (continued): To illustrate the effectiveness of avoiding the forming of matrix products, we describe an alternative approach to compute the characteristic exponents for the example considered in Section 2.2. For $N \geq 1$, define $t_i = iT/N$ for $i = 0, 1, \dots, N$. We computed the eigenvalues of $\Phi_A(T, 0)$ via the PRSF of

$$\Phi_A(T, 0) = \Phi_A(t_N, t_{N-1}) \cdots \Phi_A(t_2, t_1) \Phi_A(t_1, t_0)$$

The obtained accuracy results for different choices of N are shown in Table 1. The result for $N = 1$ corresponds to forming explicitly the above matrix product and can be also interpreted as a (single-point) periodic generator method (see Section 2.2). The results for $N \gg 1$ can be also interpreted in terms of multi-point periodic generators to be discussed in subsection 4.4.

As it can be observed, for $N = 2$ a complete failure occurs with the $\tilde{\mu}_2$ complex! However, for $N = 500$ the

Table 1. Single vs. multi-point methods accuracy results.

N	$ \tilde{\mu}_1 $	$\tilde{\mu}_2$
1	3.9×10^{-7}	-2.05...
2	2.6×10^{-7}	Complex!!
5	4.5×10^{-7}	-11.2653
10	2.1×10^{-9}	-19.7407
25	7.7×10^{-12}	-23.9921
50	2.4×10^{-15}	-23.9982
100	2.7×10^{-15}	-23.99995
200	3.2×10^{-15}	-23.9999993
500	1.9×10^{-14}	-23.999999998

achievement of the limiting accuracy of 10 digits is possible (recall that this is also the relative tolerance used for numerical integrations). This high accuracy is the combined effect of using the product form representation of the monodromy matrix with a sufficiently high number of factors and of avoiding the explicit building of the product of these factors!

4.2 Paradigm 2: structure exploiting

For discrete-time periodic systems, several so-called ‘fast’ algorithms implicitly work on the matrices of the stacked lifted representations and fully exploit their zero–nonzero block structure. By avoiding forming explicitly the matrices of the lifted representation in (5), these methods are highly efficient regarding both storage needs and required computational efforts. By relying exclusively on orthogonal transformations, frequently a kind of (week) numerical stability can be proven for them.

A typical algorithm in this category is the algorithm to compute the zeros of periodic systems. The significance of zero-based analysis of periodic descriptor systems can be easily described, by considering the most general N -periodic descriptor system representation in (4), where for simplicity we only assume time-varying state dimensions for $x_k \in R^{n_k}$ and assume E_k are square $n_{k+1} \times n_{k+1}$ matrices. Consider the stacked lifted system (5), for which several analysis problems can be addressed by computing the zeros of appropriate pencils. For example, the solvability of the system is ensured if the pencil

$$S_k(z) := F_k^S - zE_k^S$$

is regular (i.e. has nonzero determinant). The system *poles* are the finite and infinite zeros of the above pencil (counting also multiplicities). In general, for the system (4), the system *zeros* are the zeros of the system matrix

$$S_k(z) := \begin{bmatrix} F_k^S - zE_k^S & G_k^S \\ H_k^S & L_k^S \end{bmatrix}. \quad (23)$$

To study the reachability/stabilisability properties, the *input decoupling zeros* of the particular system pencil

$$S_k(z) := [F_k^S - zE_k^S \ G_k^S]$$

are relevant (Grasselli & Longhi, 1991a). More exactly, the system is *reachable* if there are no input decoupling zeros, or equivalently $S_k(z)$ has full row rank. The system is *stabilisable*, if all input decoupling zeros lie in the interior of the unit circle in the origin. For observability/detectability, the *output decoupling zeros* of

$$S_k(z) := \begin{bmatrix} F_k^S - zE_k^S \\ H_k^S \end{bmatrix}$$

are relevant (Grasselli & Longhi, 1991a). Similarly, the system is *observable* if there are no output decoupling zeros, or equivalently $S_k(z)$ has full column rank. The system is *detectable*, if all output decoupling zeros lie in the interior of the unit circle in the origin. It follows that any algorithm to compute reliably all the above categories of zeros can be seen as a universal analysis tool.

We sketch now a typical structure exploiting reduction algorithm for the computation of system zeros which employs exclusively orthogonal transformations. Consider the following pencil obtained by permuting the blocks of $S_k(z)$ in (23):

$$\tilde{S}_k(z) := \begin{bmatrix} S_k & -T_k & O & \cdots & O \\ O & S_{k+1} & -T_{k+1} & \cdots & O \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ O & & & S_{k+N-2} & T_{k+N-2} \\ -zT_{k+N-1} & O & \cdots & O & S_{k+N-1} \end{bmatrix}, \quad (24)$$

where for $i = k, k + 1, \dots, k + N - 1$

$$S_i := \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}, \quad T_i := \begin{bmatrix} E_i & O \\ O & O \end{bmatrix} \quad (25)$$

are the matrices which result by permuting the blocks of the system pencil (23). Since $S_k(z)$ and $\tilde{S}_k(z)$ are related by row and column block permutations, they have the same Kronecker form, thus the same zeros.

The so-called *fast* structure exploiting algorithm for zeros computation (Varga & Van Dooren, 2003) basically determines a ‘large’ orthogonal matrix Q , such that the reduced system pencil has the compressed form

$$Q\tilde{S}_k(z) = \begin{bmatrix} \tilde{F} & * & R \\ \tilde{F} - z\tilde{E} & 0 & \end{bmatrix},$$

where the ‘large’ size constant matrix R (of order $N\bar{n}$) has full row rank and the ‘small’ size pencil $\tilde{F} - z\tilde{E}$ (of order \bar{n}) has the same finite/infinite zeros as $\tilde{S}_k(z)$. To compute the zeros, we can apply the consecrated methods for standard linear systems as the algorithm proposed by Misra, Van Dooren, and Varga (1994).

The pencil reduction algorithm of Varga and Van Dooren (2003) performs $N - 1$ reduction steps (to determine only the finite zeros) or $N - 2$ reduction steps (if both finite and infinite zeros are determined), performing at each step i block row compressions with ‘small’ size orthogonal matrices U_i to obtain

$$U_i \begin{bmatrix} -\hat{T}_i \\ S_{i+1} \end{bmatrix} = \begin{bmatrix} R_i \\ O \end{bmatrix},$$

where R_i are matrices of full row rank. The overall structure exploiting algorithm updates the block rows i and $i + 1$ and

block columns 1, $i + 1$ and $i + 2$ of the pencil, grouped in the form

$$U_i \begin{bmatrix} \widehat{S}_i & -\widehat{T}_i & O \\ O & S_{i+1} & -T_{i+1} \end{bmatrix} = \begin{bmatrix} \widetilde{S}_i & R_i & -\widetilde{T}_i \\ \widehat{S}_{i+1} & O & -\widehat{T}_{i+1} \end{bmatrix}.$$

The recurring structure in the block rows starting with $i + 1$ is thus preserved. Since the operations are performed only locally, the computational complexity of each step is the usual $O(\bar{n}^3)$, while the overall reduction has complexity $O(N\bar{n}^3)$, and thus satisfactory. Since the zero–nonzero structure of $\widetilde{S}_k(z)$ is destroyed, this algorithm is not structure preserving. However, because for the whole reduction only orthogonal transformations are performed, the weak numerical stability can be proven in that the resulting pencil $Q\widetilde{S}_k(z)$ is the exact pencil corresponding to a slightly perturbed original pencil $\widehat{S}_k(z)$. A similar ‘fast’ algorithm has been developed to solve periodic Riccati equations (Varga, 2008).

4.3 Paradigm 3: structure preserving

The most advanced category of numerical methods for discrete-time periodic systems is formed by the so-called structure preserving methods, which implicitly work on the stacked lifted representations, while fully preserving the zero–nonzero structure of the matrices of the lifted representation (5). Most algorithms in this category are recent developments and for several methods the strong (structural) backward stability can be proven. This means that it can be proven that the computed results are exact for slightly perturbed original system data. Representative for these methods is the computation of the periodic Kronecker-like form of a periodic pair of matrices, which underlies many of recently proposed reliable numerical algorithms.

An algorithm to compute periodic Kronecker-like forms has been developed by Varga (2004a). For given N -periodic matrix pairs (S_i, T_i) with $S_i \in R^{\mu_i \times \nu_i}$ and $T_i \in R^{\mu_i \times \nu_{i+1}}$, this algorithm determines orthogonal N -periodic transformation matrices Q_i and Z_i such that

$$Q_i S_i Z_i = \begin{bmatrix} B_i^r & A_i^r & * & * & * \\ O & O & A_i^\infty & * & * \\ O & O & O & A_i^f & * \\ O & O & O & O & A_i^l \\ O & O & O & O & C_i^l \end{bmatrix},$$

$$Q_i T_i Z_{i+1} = \begin{bmatrix} O & E_i^r & * & * & * \\ O & O & E_i^\infty & * & * \\ O & O & O & E_i^f & * \\ O & O & O & O & E_i^l \\ O & O & O & O & O \end{bmatrix},$$

where:

- (a) E_i^r is invertible and the periodic pair $((E_i^r)^{-1} A_i^r, (E_i^r)^{-1} B_i^r)$ is completely reachable;
- (b) E_i^l is invertible and the periodic pair $(C_i^l, (E_i^l)^{-1} A_i^l)$ is completely observable;
- (c) A_i^∞ is invertible and the product $(A_i^\infty)^{-1} E_i^\infty \dots (A_{i+N-1}^\infty)^{-1} E_{i+N-1}^\infty$ is nilpotent;
- (d) E_i^f is non-singular.

Note that $Q_i S_i Z_i$ and $Q_i T_i Z_{i+1}$ have the same row partition which, however, generally depends on i . For a fixed column partitioning of $Q_i S_i Z_i$, the corresponding column partitioning of $Q_i T_i Z_{i+1}$ is uniquely determined by the conditions (a)–(d) above. The periodic pair (A_i^∞, E_i^∞) specifies the structure at infinity of the periodic pair (S_i, T_i) , while the pair (A_i^f, E_i^f) specifies its finite structure. Similarly, the periodic triples (A_i^r, E_i^r, B_i^r) and (A_i^l, E_i^l, C_i^l) specify the right-hand and left-hand Kronecker structures of the pair (S_i, T_i) , respectively.

The algorithm of Varga (2004a) performs implicitly the reduction of $\widetilde{S}_k(z)$ in (24) to a structurally identical form, where the constituent blocks formed of the pairs $(Q_i S_i Z_i, Q_i T_i Z_{i+1})$ have the Kronecker-like structure above. The reduction algorithm employs exclusively orthogonal transformations applied to individual pairs, which corresponds to apply block-diagonal orthogonal transformations to $\widetilde{S}_k(z)$ which thus preserves its structure. It is possible to prove that the computed transformed pairs $(Q_i S_i Z_i, Q_i T_i Z_{i+1})$ are exact for slightly perturbed initial matrices $\widetilde{S}_i, \widetilde{T}_i$, which satisfy

$$\|\bar{X} - X\| \leq \varepsilon_X \|X\|, \quad X = S_i, T_i,$$

where, in each case, ε_X is a modest multiple of the relative machine precision ε_M . It follows that the algorithm is *strongly* backward stable.

There are many applications of this reduction technique. For example, to compute different types of zeros, the algorithm of Varga (2004a) can be applied to particular periodic pairs. The system zeros correspond to the definition (25), while the poles correspond to the definition

$$S_i := A_i, \quad T_i := E_i.$$

In a similar way, with

$$S_i := [A_i \ B_i], \quad T_i := [E_i \ O]$$

or

$$S_i := \begin{bmatrix} A_i \\ C_i \end{bmatrix}, \quad T_i := \begin{bmatrix} E_i \\ O \end{bmatrix}$$

the algorithm can be used to compute the *input decoupling zeros* and *output decoupling zeros*, respectively (Grasselli & Longhi, 1991a).

Other applications are described by Varga (2007a), the most notable being the minimal realisation algorithm for periodic descriptor systems (Varga, 2007b) and the solution of periodic Riccati equations in the most general setting (Varga, 2008).

4.4 Paradigm 4: multi-point periodic generators

The multi-point periodic generator method evolved as an enhancement of the periodic generator method described in Section 2.2, driven by the need of accurately solving several computational problems for periodic continuous-time systems, for example, the solution of various periodic matrix differential equations. Using appropriate exact discretisations, the continuous-time problems are transformed into equivalent multi-point discrete-time periodic problems for which reliable computational algorithms are available. By solving the discrete-time problems, multi-point periodic generators are computed, which serve to determine the continuous periodic solutions (usually by integrating the underlying ordinary matrix differential equations with known multi-point conditions). Besides increased accuracy, these methods are amenable to highly efficient solutions by exploiting the potential of parallel computations.

We will discuss these aspects succinctly on the concrete examples of solving the PLDE (9), PSDE (10) and PRDE (8). To discretise the PLDE for a given period T , consider a uniform time grid of N intervals $[(k - 1)\Delta, k\Delta]$, for $k = 1, \dots, N$, where $\Delta := T/N$ is the discretisation interval. $X(t)$ and $X(t + \Delta)$ are related as

$$X(t + \Delta) = \Phi_A(t + \Delta, t)X(t)\Phi_A^T(t + \Delta, t) + \int_t^{t+\Delta} \Phi_A(t + \Delta, \tau)C(\tau)\Phi_A^T(t + \Delta, \tau)d\tau.$$

With this, we can formulate the discretised problem as the computation of N values of the solution $X_k := X((k - 1)\Delta)$, $k = 1, \dots, N$ which satisfy the periodic discrete-time Lyapunov equation

$$X_{k+1} = F_k X_k F_k^T + W_k, \quad k = 1, \dots, N; \quad X_{N+1} = X_1 \tag{26}$$

where

$$F_k := \Phi_A(k\Delta, (k - 1)\Delta),$$

$$W_k := \int_{(k-1)\Delta}^{k\Delta} \Phi_A(k\Delta, \tau)C(\tau)\Phi_A^T(k\Delta, \tau)d\tau.$$

The computation of F_k involves the integration on $[t_0, t_0 + \Delta]$ of (11) for $t_0 = (k - 1)\Delta$. For the determination of W_k , see the approach proposed by Varga (2005). An entirely similar approach can be used to discretise PSDEs as in (10).

The solution approach of the periodic Riccati differential equation (8) employs an equivalent multi-point formulation, which involves the discretisation of a differential

equation as in (11), with $A(t)$ replaced by the $2n \times 2n$ Hamiltonian matrix

$$H(t) = \begin{bmatrix} -A^T(t) & Q(t) \\ R(t) & A(t) \end{bmatrix}.$$

The corresponding monodromy matrix $\Phi_H(T, 0)$ is represented as a product of N symplectic matrices $\Phi_H(T, 0) = H_{N-1} \dots H_1 H_0$. The matrices H_k can be freely interpreted as the state matrices of an N -periodic discrete-time system.

In the second step, the discretised problems are solved. To solve the periodic discrete-time Lyapunov equation (26), a numerically reliable algorithm has been proposed by Varga (1997). This algorithm relies on the reduction of the N -periodic matrix F_k to the PRSF, which allows to solve reduced equations with matrices in condensed forms. Similar techniques apply for the solution of periodic discrete-time Sylvester equations. For the solution of PRDEs, the stable invariant subspace S_0 of the monodromy matrix $\Phi_H(T, 0)$ is computed, by employing eigenvalue reordering techniques of matrix products (Bojanczyk et al., 1992; Granat, Kågström, & Kressner, 2007). This algorithm produces in fact N $2n \times n$ matrices S_k , $k = 0, 1, \dots, N - 1$ with orthonormal columns, satisfying $H_k S_k = S_{k+1} \tilde{H}_k$, where \tilde{H}_k is an $n \times n$ N -periodic matrix with stable characteristic multipliers. Thus, S_k forms an orthogonal basis of the periodic stable invariant subspace of the periodic matrix H_k . The corresponding multi-point solution results as

$$X_k = S_{k,12} S_{k,11}^{-1},$$

where $S_{k,11}$ is the leading $n \times n$ block of

$$S_k = \begin{bmatrix} S_{k,11} \\ S_{k,12} \end{bmatrix}.$$

At the third step, the continuous solution $X(t)$ can be computed by integrating the appropriate differential equation on the intervals $[(k - 1)\Delta, k\Delta]$, for $k = 1, \dots, N$, using $X((k - 1)\Delta) = X_k$ as the initial condition. Alternatively, for a sufficiently dense discretisation grid, interpolation techniques can be employed to evaluate $X(t)$ for an arbitrary value of time t .

The multi-point periodic generator method can be also used to address other computational problems for continuous-time periodic systems as the computation of various system norms (H_2 -norm, H_∞ -norm, Hankel-norm), the solution of the linear-quadratic periodic output feedback stabilisation problem (Viganò et al., 2010) or the high-accuracy solution of H_∞ -type PRDEs (Feng, Varga, Anderson, & Lovera, 2011). Recently performed evaluations of the multi-point approach to solve periodic Riccati equations in conjunction with the use of enhanced structure preserving (symplectic) integrators of ODEs to determine the symplectic matrices H_k , $k = 0, 1, \dots, N_1$

have demonstrated the effectiveness of these techniques in achieving high-accuracy solutions (Gusev, Johansson, Kågström, Shiriaev, & Varga, 2010).

5. Concluding remarks and open problems

In a historical perspective, the development of numerically satisfactory algorithms for linear periodic systems started with the development of the algorithms to compute and re-order the PRSF (Bojanczyk et al., 1992; Hench & Laub, 1994). The first applications of these new techniques were the solution of periodic discrete-time Riccati equations. Sustained efforts of several research groups followed in developing new algorithms for periodic systems in the next few years. An account of the main algorithmic achievements by 2001 is presented in an overview paper (Varga & Van Dooren, 2001), where many open problems have also been mentioned. The situation in 2007 is presented in a second overview paper (Varga, 2007a), where many new developments in solving the formulated open problems in 2001 are described. An important aspect worth mentioning was the development of the first reliable computational algorithms for the continuous-time periodic systems. The last developments in the field are the so-called integrated algorithms to solve specific synthesis problems (e.g. fault detection problems, Varga 2012), whose main strength lies in their ability to exploit at each computational step all available structural information at the previous step, which overall leads to very efficient structure exploiting computations.

There are many open computational problems, for both discrete- and continuous-time periodic systems. For discrete-time systems, we mention the lack of algorithms for computing inner–outer factorisation (in the non-standard case), Hankel-norm approximation, periodic descriptor system realisation of arbitrary lifted systems, or for the solution of controller/observer/fault detection filter synthesis problems using exact or approximate model matching approaches. Moreover, there exist no efficient algorithms to solve periodic linear matrix inequalities (LMIs). For continuous-time periodic systems, there are no algorithms for efficient computation of frequency responses, stabilisation or pole assignment. New methods are also necessary for the solution of periodic differential LMIs. Some of existing computational methods employ the frequency-lifted representation of continuous-time periodic systems (see e.g. Bittanti & Colaneri, 2009), which leads to large-scale structured system matrices with block Toeplitz or block Toeplitz plus diagonal structures. Efficient structure exploiting/preserving algorithms using this representation are still missing and their development is an open field for research.

Acknowledgements

The algorithm development research of the author between 2002–2008 has been partly supported in the framework of a Swedish Strategic Research Foundation Grant: ‘Matrix Pencil

Computations in Computer-Aided Control System Design: Theory, Algorithms and Software Tools’.

References

- Arcara, P., Bittanti, S., & Lovera, M. (2000). Periodic control of helicopter rotors for attenuation of vibrations in forward flight. *IEEE Transactions on Control Systems Technology*, *8*, 883–894.
- Bittanti, S., & Colaneri, P. (2009). *Periodic systems: Filtering and control*. Berlin: Springer-Verlag.
- Bojanczyk, A.W., Golub, G., & Van Dooren, P. (1992). The periodic Schur decomposition. Algorithms and applications. *Proceedings SPIE Conference*, *1770*, 31–42. doi: 10.1117/12.130915.
- Colaneri, P., & Longhi, S. (1995). The realization problem for linear periodic systems. *Automatica*, *31*, 775–779.
- Feng, Y., Varga, A., Anderson, B., & Lovera, M. (2011). A new iterative algorithm to solve periodic Riccati differential equations with sign indefinite quadratic terms. *IEEE Transactions on Automatic Control*, *56*, 929–934.
- Granat, R., Kågström, B., & Kressner, D. (2007). Computing periodic deflating subspaces associated with a specified set of eigenvalues. *BIT Numerical Mathematics*, *47*, 763–791.
- Grasselli, O.M., & Longhi, S. (1991a). Finite zero structure of linear periodic discrete-time systems. *International Journal of Systems Science*, *22*, 1785–1806.
- Grasselli, O.M., & Longhi, S. (1991b). Pole-placement for non-reachable periodic discrete-time systems. *Mathematics of Control, Signals and Systems*, *4*, 439–455.
- Gusev, S., Johansson, S., Kågström, B., Shiriaev, A., & Varga, A. (2010). A numerical evaluation of solvers for the periodic Riccati differential equation. *BIT Numerical Mathematics*, *50*, 301–329.
- Hench, J.J., & Laub, A.J. (1994). Numerical solution of the discrete-time periodic Riccati equation. *IEEE Transactions on Automatic Control*, *39*, 1197–1210.
- Johnson, W. (1992). *CAMRAD II: comprehensive analytical model of rotorcraft aerodynamics and dynamics*. Palo Alto, CA: Johnson Aeronautics.
- Jonkman, J.M., & Buhl, M.L. (2005). *FAST users guide* (Technical Report No. NREL/EL–500–38230). USA: NREL.
- Konstanzer, P. (2001). *Decentralized vibration control for active helicopter rotor blades*. Ph.D. Thesis, Germany: University of Stuttgart.
- Longhi, S. (1994). Structural properties of multirate sampled-data systems. *IEEE Transactions on Automatic Control*, *39*, 692–695.
- Meyer, R.A., & Burrus, C.S. (1975). A unified analysis of multirate and periodically time-varying digital filters. *IEEE Transactions on Circuits and Systems*, *22*, 162–168.
- Misra, P., Van Dooren, P., & Varga, A. (1994). Computation of structural invariants of generalized state-space systems. *Automatica*, *30*, 1921–1936.
- Park, B., & Verriest, E.I. (1989). Canonical forms for discrete-time periodically time varying systems and a control application. *Proceedings of CDC’89* (pp. 1220–1225). Tampa, Florida, USA.
- Patel, R.V., Laub, A.J., & Van Dooren, P. (Eds.). (1994). *Numerical linear algebra techniques for systems and control*. Piscataway, NJ, USA: IEEE Press.
- Pittelkau, M.E. (1993). Optimal periodic control for spacecraft pointing and attitude determination. *Journal of Guidance, Control, and Dynamics*, *16*, 1078–1084.

- Riziotis, V.A., Politis, E.S., Voutsinas, S.G., & Chaviaropoulos, P.K. (2007). Stability analysis of pitch-regulated, variable speed wind turbines in closed loop operation using a linear eigenvalue approach. *Journal of Physics: Conference Series*, 75, 1–16.
- Van Dooren, P., & Verhaegen, M. (1985). On the use of unitary state-space transformations, *Vol. 47 of Special Issue of Contemporary Mathematics in Linear Algebra and Its Role in Systems Theory*. Providence, RI: American Mathematical Society.
- Varga, A. (1997). Periodic Lyapunov equations: Some applications and new algorithms. *International Journal of Control*, 67, 69–87.
- Varga, A. (1999). Balancing related methods for minimal realization of periodic systems. *Systems & Control Letter*, 36, 339–349.
- Varga, A. (2000). A Descriptor Systems toolbox for MATLAB. *Proceedings of CACSD'2000 Symposium* (pp. 150–155). Anchorage, Alaska, USA.
- Varga, A. (2004a). Computation of Kronecker-like forms of periodic matrix pairs. *Proceedings of MTNS'04*. Leuven, Belgium.
- Varga, A. (2004b). Numerical awareness in control. *Special Issue of IEEE Control Systems Magazine*, 24, 14–17.
- Varga, A. (2005). On solving periodic differential matrix equations with applications to periodic system norms computation. *Proceedings of CDC'05* (pp. 6545–6550). Seville, Spain.
- Varga, A. (2007a). An overview of recent developments in computational methods for periodic systems. *Proceedings of IFAC Workshop on Periodic Control Systems* (pp. 157–162). St. Petersburg, Russia.
- Varga, A. (2007b). On computing minimal realizations of periodic descriptor systems. *Proceedings of IFAC Workshop on Periodic Control Systems* (pp. 175–180). St. Petersburg, Russia.
- Varga, A. (2008). On solving periodic Riccati equations. *Numerical Linear Algebra with Applications*, 15, 809–835.
- Varga, A. (2012). A numerically reliable approach for the synthesis of periodic FDI filters. *Proceedings of SAFEPROCESS'12* (pp. 498–503). Mexico City, Mexico.
- Varga, A., & Pieters, S. (1998). Gradient-based approach to solve optimal periodic output feedback control problems. *Automatica*, 34, 477–481.
- Varga, A., & Van Dooren, P. (2001). Computational methods for periodic systems – An overview. *Proceedings of IFAC Workshop on Periodic Control Systems* (pp. 171–176). Como, Italy.
- Varga, A., & Van Dooren, P. (2003). Computing the zeros of periodic descriptor systems. *Systems & Control Letter*, 50, 371–381.
- Viganò, L., Bergamasco, M., Lovera, M., & Varga, A. (2010). Optimal periodic output feedback control: A continuous-time approach and a case study. *International Journal of Control*, 83, 897–914.
- Zhang, P., Ding, S.X., & Liu, P. (2012). A lifting based approach to observer based fault detection of linear periodic systems. *IEEE Transactions on Automatic Control*, 57, 457–462.