SIAM J. MATRIX ANAL. APPL.
Vol. 40, No. 3, pp. 845–869

COMPUTING DELAY LYAPUNOV MATRICES AND $\mathcal{H}_2$ NORMS FOR LARGE-SCALE PROBLEMS*

WIM MICHIELS† AND BIN ZHOU‡

Abstract. A delay Lyapunov matrix corresponding to an exponentially stable system of linear time-invariant delay differential equations can be characterized as the solution of a boundary value problem involving a matrix valued delay differential equation. This boundary value problem can be seen as a natural generalization of the classical Lyapunov matrix equation. Lyapunov matrices play an important role in constructing Lyapunov functionals and in $\mathcal{H}_2$ optimal control. In this paper we present a general approach for computing delay Lyapunov matrices and $\mathcal{H}_2$ norms for systems with multiple discrete delays, whose applicability extends toward problems where the matrices are large and sparse, and the associated positive semidefinite matrix has a low rank. The problems addressed are challenging, because the boundary value problem is matrix valued with a structure that is much harder to exploit than in the delay-free case, and its solution is in the generic situation nonsmooth. In contrast to existing methods that are based on solving the boundary value problem directly, our method is grounded in solving standard Lyapunov equations of increased dimensions. It combines several ingredients: (i) a spectral discretization of the system of delay equations, (ii) a targeted similarity transformation which induces a desired structure and sparsity pattern and, at the same time, favors accurate low-rank solutions of the corresponding Lyapunov equation, and (iii) a Krylov method for large-scale matrix Lyapunov equations. The structure of the problem is exploited in such a way that the final algorithm does not involve a preliminary discretization step and provides a fully dynamic construction of approximations of increasing rank. Interpretations are also given in terms of a projection method directly applied to a standard linear infinite-dimensional system, which is equivalent to the original time-delay system. Throughout the paper two didactic examples are used to illustrate the properties of the problem, the challenges and methodological choices, while numerical experiments are reported to illustrate the effectiveness of the algorithms.

Key words. delay system, Lyapunov matrix equations, Krylov method

AMS subject classifications. 34K06, 15A24, 35P30, 93A15

DOI. 10.1137/18M1209842

1. Introduction.

We address a linear system with multiple discrete delays,

$$\dot{x}(t) = A_0 x(t) + \sum_{i=1}^{m} A_i x(t-\tau_i), \tag{1.1}$$

where $x(t) \in \mathbb{R}^n$ is the state variable at time t, $A_i \in \mathbb{R}^{n\times n}$ are the system matrices, and $\tau_i,\ i = 1, \ldots, m$, represent time delays, ordered such that

$$0 < \tau_1 < \cdots < \tau_m.$$

Throughout the paper we assume that the zero solution of (1.1) is exponentially stable. Equivalently, we assume that the characteristic roots, which are the solutions of equation

*Received by the editors August 27, 2018; accepted for publication (in revised form) by P. Benner April 19, 2019; published electronically July 3, 2019.
https://doi.org/10.1137/18M1209842
Funding: The work of the authors was supported by project C14/17/072 of the KU Leuven Research Council, by project G0A5317N of the Research Foundation–Flanders (FWO–Vlaanderen), and by the National Natural Science Foundation of China under grant 61773140.
†Department of Computer Science, KU Leuven, Heverlee, Belgium (Wim.Michiels@cs.kuleuven.be).
‡Harbin Institute of Technology, Center for Control Theory and Guidance Technology, Harbin, China (binzhou@hit.edu.cn).

$$\det\left(\lambda I - A_0 - \sum_{i=1}^{m} A_i \mathrm{e}^{-\lambda\tau_i}\right) = 0,$$

are confined to the open left half plane [20, 21]. The fundamental solution of (1.1), which we denote by $K : \mathbb{R} \to \mathbb{R}^{n\times n}$, is defined as the function satisfying

$$\text{(1.2)} \qquad \begin{cases} \dot{K}(t) = A_0 K(t) + \sum_{i=1}^{m} A_i K(t-\tau_i) & \text{for almost all } t \geq 0, \\ K(0) = I, & \\ K(t) = 0 & \text{for } t < 0. \end{cases}$$

Consider a real positive semidefinite n-by-n matrix whose rank revealing decomposition reads as BB^{T}, where $B \in \mathbb{R}^{n\times r}$ is of full rank r. The delay Lyapunov matrix for (1.1), associated with this matrix, is defined as a function $P : \mathbb{R} \to \mathbb{R}^{n\times n}$ such that

$$\text{(1.3)} \qquad P(t) = \int_0^\infty K(s) B B^{\mathrm{T}} K^{\mathrm{T}}(s+t) \mathrm{d}s.$$

Following from the exponential stability condition of (1.1), the delay Lyapunov matrix can be characterized as the *unique* solution of the matrix valued "boundary" value problem

$$\text{(1.4)} \qquad \begin{cases} \dot{P}(t) = P(t)A_0^{\mathrm{T}} + \sum_{k=1}^{m} P(t-\tau_k)A_k^{\mathrm{T}}, \quad t \geq 0, \\ P(-t) = P^{\mathrm{T}}(t), \\ -BB^{\mathrm{T}} = P(0)A_0^{\mathrm{T}} + A_0 P(0) + \sum_{k=1}^{m} \left(P(-\tau_k)A_k^{\mathrm{T}} + A_k P(\tau_k)\right); \end{cases}$$

see [15]. There is also a dual formulation: with a positive semidefinite $(n \times n)$-matrix $C^{\mathrm{T}}C$ we can associate Lyapunov matrix

$$Q(t) = \int_0^\infty K^{\mathrm{T}}(s) C^{\mathrm{T}} C K(s+t) \mathrm{d}s,$$

which corresponds to the unique solution of

$$\text{(1.5)} \qquad \begin{cases} \dot{Q}(t) = Q(t)A_0 + \sum_{k=1}^{m} Q(t-\tau_k)A_k, \quad t \geq 0, \\ Q(-t) = Q^{\mathrm{T}}(t), \\ -C^{\mathrm{T}}C = Q(0)A_0 + A_0^{\mathrm{T}} Q(0) + \sum_{k=1}^{m} \left(Q(-\tau_k)A_k + A_k^{\mathrm{T}} Q(\tau_k)\right). \end{cases}$$

Note that in the delay-free case (which can be seen as the limit case obtained by letting $\tau_m \to 0$), the third equation in (1.4) and the third one in (1.5) reduce to a standard pair of primal and dual Lyapunov matrix equations in variables $P(0)$ and $Q(0)$.

The delay Lyapunov matrix is a building block for the construction of Lyapunov functionals of complete type, which are associated with necessary and sufficient stability conditions. We refer to [15] for an excellent review. It should be noted that in the literature on complete type Lyapunov functionals, the Lyapunov matrix is usually denoted by $U(t)$, which corresponds to $Q(t)$ in our adopted notation. As another comment, in several works, the Lyapunov matrix is defined as the solution of boundary value problem (1.4) or (1.5). In this way it can also be defined for an exponentially unstable system, provided the delay system has no pair of eigenvalues (λ_1, λ_2) such that $\lambda_1 + \lambda_2 = 0$; see [16, 15]. The latter is at the price that the aforementioned connection with the fundamental solution is lost. The Lyapunov matrix also plays a major role in the characterization of the $\mathcal{H}_2$ norm for the system

$$\left\{ \begin{array}{l} \dot{x}(t) = A_0 x(t) + \sum_{i=1}^{m} A_i x(t-\tau_i) + Bu(t), \\ y(t) = Cx(t), \end{array} \right. \tag{1.6}$$

where $u \in \mathbb{C}^r$ is the input, $y \in \mathbb{C}^s$ is the output, $B \in \mathbb{R}^{n\times r}$ is the input matrix, and $C \in \mathbb{R}^{s\times n}$ is the output matrix. The transfer function of the system (1.6) is given by

$$\Upsilon(s) = C\left(sI - A_0 - \sum_{i=1}^{m} A_i \mathrm{e}^{-s\tau_i}\right)^{-1} B, \tag{1.7}$$

and its $\mathcal{H}_2$ norm is defined in the frequency domain as

$$\|\Upsilon\|_2 = \frac{1}{2\pi}\left(\int_{-\infty}^{\infty} \mathrm{Tr}\left(\Upsilon^*(\imath\omega)\Upsilon(\imath\omega)\right) \mathrm{d}\omega\right)^{\frac{1}{2}}.$$

An equivalent definition in the time domain is given by

$$\|\Upsilon\|_2 = \left(\int_0^{\infty} \mathrm{Tr}\left(h^{\mathrm{T}}(t)h(t)\right) \mathrm{d}t\right)^{\frac{1}{2}}$$

with h the impulse response. The following proposition expresses the $\mathcal{H}_2$ norm in terms of the delay Lyapunov matrix, whose proof trivially follows from the identity

$$h(t) = CK(t)B. \tag{1.8}$$

PROPOSITION 1.1 (see [14, Theorem 1]). *The $\mathcal{H}_2$ norm of* (1.7) *satisfies*

$$\|\Upsilon\|_2^2 = \mathrm{Tr}\left(CP(0)C^{\mathrm{T}}\right), \tag{1.9}$$

where $P(t)$ is the delay Lyapunov matrix associated with matrix BB^{T}.

Finally, generalized Gramians and Lyapunov matrices have been instrumental to the structure preserving model reduction approach for time-delay systems in [10], which is based on truncation of a balanced system. Lyapunov matrices have been generalized to a class of Volterra integro-differential equations in [4], also in conjunction with structure preserving model reduction.

The aim of this paper is to present a novel method for computing delay Lyapunov matrices and $\mathcal{H}_2$ norms with the following properties:

- the method is generally applicable, in the sense that there are no restrictions on the number and values of the delays, and the delay Lyapunov matrix can be easily computed or extended a posteriori beyond the interval $[-\tau_m,\ \tau_m]$;
- the number of operations scales favorably with respect to the dimension n of the system matrices, particularly if the matrices are sparse. The aim is to target (discretizations of) partial differential equations (PDEs) with delay, provided that the rank r of B is small compared to n.

In the description of the results, we restrict ourselves to the computation of Lyapunov matrix $P(t)$, since $Q(t)$ can be obtained from Lyapunov matrix $P(t)$ associated with a "dual" system, inferred from the substitutions $A_k \leftarrow A_k^{\mathrm{T}},\ k = 0, \ldots, m$, and $B \leftarrow C^{\mathrm{T}}$. The latter directly follows from a comparison between (1.4) and (1.5).

The characterization (1.4) provides a natural way to compute the Lyapunov matrix and the $\mathcal{H}_2$ norm. However, there are major challenges. First, when making the leap from ordinary to delay differential equations, the algebraic Lyapunov matrix

equation is replaced by a *matrix-valued* boundary value problem with delay. Second, bringing the equation in triangular form using a Schur decomposition, which forms the basis of the Bartels–Stewart algorithm for the matrix Lyapunov equation, is no longer possible. Third, it has been shown that function $\mathbb{R} \ni t \mapsto P(t)$ may be nonsmooth [14]. It is continuous, but it may not be differentiable at $t = 0$. On the interval $[0,\ \infty)$, to which we restrict ourselves in this paper because of the second condition in (1.4), it is continuously differentiable. However, the second derivative might be discontinuous at $t = \tau_i,\ i = 1, \ldots, m$, as we shall illustrate in the next section.

In the literature, two approaches for solving (1.4) can be identified. The so-called direct approach is based on approximating the solution on an interval by a matrix polynomial or a piecewise matrix polynomial and determining the coefficients by collocation conditions for the differential equation, next to imposing the boundary conditions and continuity requirements; see, e.g., [9, 14]. With N the number of collocation points, this results in a linear system of equations in $\mathcal{O}(n^2N)$ variables. The convergence of the obtained approximations to the solution as a function of N might be slowed down by the lack of smoothness of the solution discussed above; see [14] for a detailed analysis. The second approach can be interpreted as a shooting method. It is applicable only if the time delays are commensurate, i.e., $\tau_i = n_i h$ for some $h > 0$ and $n_i \in \mathbb{N},\ i = 1, \ldots, m$, and it exploits that the solution of (1.4) is then piecewise smooth (more precisely, smooth on intervals of form $(ih,\ (i + 1)h),\ i \in \mathbb{Z}$). Then (1.4) can be reformulated as a standard boundary value problem for an ordinary differential equation of dimensions $2n^2n_m$ on the interval $[0,\ h]$. The transition between starting and end times can be determined explicitly in the form of the action of a matrix exponential, which corresponds to the so-called semianalytic approach [15, 7], or by a numerical time-integration scheme [13].

The common factor that leads to a poor scalability of the above vectorization based approaches with respect to the dimension n is that they rely on solving a system of equations in n^2 variables, possibly multiplied with a large factor, hence when using a direct solver the number of elementary operations amounts to $\mathcal{O}(n^6)$ operations. To the best of the authors' knowledge, the only available method that allows addressing large-scale problems is the one presented in [13] for the single delay case. It falls under the umbrella of shooting methods, with the transition map determined by time integration. The key idea behind this approach, which has been shown to be effective for problems with n up to ≈ 1000, is to solve the linear system of equations arising from the shooting method using a preconditioned Krylov method, where the preconditioner is determined from the corresponding problem without delay. The latter allows an application of the preconditioner using $\mathcal{O}(n^3)$ operations. This approach is complementary to the presented approach, which is grounded in solving standard Lyapunov matrix equations.

In section 2, we present a spectral discretization of (1.6) into a system of ordinary differential equations of dimension $(N+1)n$, where N determines the resolution of the discretization. This allows us to obtain approximations of delay Lyapunov matrices and $\mathcal{H}_2$ norms by solving standard Lyapunov matrix equations. We also show how a favorable structure on the system matrices can be imposed by using a transformation. The main results are obtained in section 3, where projections on Krylov spaces are used to approximate the solutions of these Lyapunov equations. This results in a dynamic construction of Lyapunov matrix approximations. Note that Krylov methods constitute an established approach for solving large-scale matrix Lyapunov equations; see, e.g., [24, 23, 6] and the references therein. We will show that several methodological choices can be made in such a way that the overall algorithm does not depend

anymore on parameter N. In fact the only condition is that it is sufficiently large with respect to the number of iterations for building the Krylov space. This property is at the basis of an interpretation in terms of a projection method applied directly to a linear infinite-dimensional system equivalent to the original delay system. In this sense, the algorithm complements the set of "discretization free" algorithms for solving nonlinear eigenvalue problems and associated problems in [11, 12]. Numerical experiments are reported in section 4 and some concluding remarks are formulated in section 5. Preliminary results regarding the computation of $\mathcal{H}_2$ norms have been presented in [22].

2. Finite-dimensional approximation. In section 2.1 we outline how to discretize (1.6) (and as a consequence (1.1)) using a spectral method [25], resulting in a system described by ordinary differential equations. For the sake of conciseness, the derivation is slightly different from [2], in the sense that the connection of (1.6) with an abstract infinite-dimensional linear system is not explicitly made. Subsequently, we outline how an approximation of the delay Lyapunov matrix can be obtained from this discretization. In section 2.2, we discuss properties of the obtained approximations. In section 2.3, we reformulate the expressions for the delay Lyapunov approximations in a form that is more suitable for the application of a Krylov method.

2.1. A spectral discretization. Given a positive integer N, we consider a mesh Ω_N of $N+1$ distinct points in the interval $[-\tau_m,\ 0]$:

$$\Omega_N = \{\theta_{N,i},\ i = 1, \ldots, N+1\}, \tag{2.1}$$

where $-\tau_m \le \theta_{N,1} < \cdots < \theta_{N,N} < \theta_{N,N+1} = 0$. Throughout the paper we choose the nonzero mesh points as scaled and shifted zeros of the Chebyshev polynomial of the second kind and order N. More precisely, the mesh points are specified as

$$\theta_{N,i} = \frac{\tau_m}{2}(\alpha_{N,i} - 1), \quad \alpha_{N,i} = -\cos\frac{\pi i}{N+1},\ i = 1, \ldots, N+1. \tag{2.2}$$

Denoting with $l_{N,k}$ the Lagrange polynomials corresponding to Ω_N, i.e., real valued polynomials of degree N satisfying

$$l_{N,k}(\theta_{N,i}) = \begin{cases} 1, & i = k, \\ 0, & i \neq k, \end{cases}$$

and letting $x_k,\ k = 1, \ldots, N+1$, functions from $\mathbb{R}$ to $\mathbb{R}^n$, we approximate the "piece of trajectory" $x(t+\theta),\ \theta \in [-\tau_m,\ 0]$ as

$$x(t+\theta) \approx \sum_{k=1}^{N+1} l_{N,k}(\theta) x_k(t), \quad \theta \in [-\tau_m,\ 0], \tag{2.3}$$

which induces on its term the approximation

$$\begin{cases} x_1(t) & \approx & x(t+\theta_{N,1}), \\ & \vdots & \\ x_N(t) & \approx & x(t+\theta_{N,N}), \\ x_{N+1}(t) & \approx & x(t). \end{cases} \tag{2.4}$$

Along a solution of (1.1), x is differentiable almost everywhere. Hence, for almost all $t \ge 0,\ \theta \in [-\tau_m,\ 0]$ we can express

$$\frac{\partial x(t+\theta)}{\partial t} = \frac{\partial x(t+\theta)}{\partial \theta},$$

which is closely related to the modeling of linear delay equations via an advection equation [17]. Requiring that the right-hand side of (2.3) satisfies this identity for (collocation points) $\theta_{N,1}, \ldots, \theta_{N,N}$ brings us to the equations

$$\dot{x}_i(t) = \sum_{k=1}^{N+1} \dot{l}_{N,k}(\theta_{N,i}) x_k(t), \quad i = 1, \ldots, N. \tag{2.5}$$

Next, substituting the right-hand side of (2.3) into (1.6) yields

$$\left\{ \begin{array}{ll} \dot{x}_{N+1}(t) = & A_0 x_{N+1}(t) + \left(\sum_{i=1}^{m} \sum_{k=1}^{N+1} A_i l_{N,k}(-\tau_i) x_k(t) \right) + Bu(t), \\ y(t) & = Cx_{N+1}(t). \end{array} \right. \tag{2.6}$$

Letting $z(t) = [x_1^{\mathrm{T}}(t) \ \cdots \ x_{N+1}^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{(N+1)n \times 1}$, (2.5) and (2.6) can be written as

$$\left\{ \begin{array}{l} \dot{z}(t) = \mathcal{A}_N z(t) + B_N u(t), \\ y(t) = C_N z(t), \end{array} \right. \tag{2.7}$$

where

$$\begin{array}{l} \mathcal{A}_N = \left[\begin{array}{ccc} d_{1,1} & \ldots & d_{1,N+1} \\ \vdots & & \vdots \\ d_{N,1} & \ldots & d_{N,N+1} \\ a_1 & \ldots & a_{N+1} \end{array} \right], \quad B_N = \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ 1 \end{array} \right] \otimes B, \\ C_N = [0 \ \cdots \ 0 \ 1] \otimes C \end{array} \tag{2.8}$$

and

$$\left\{ \begin{array}{lll} d_{i,k} = & \dot{l}_{N,k}(\theta_{N,i}) I_n, & i \in \{1, \ldots, N\}, \ k \in \{1, \ldots, N+1\}, \\ a_k = & A_0 l_{N,k}(0) + \sum_{i=1}^{m} A_i l_{N,k}(-\tau_i), & k \in \{1, \ldots, N+1\}. \end{array} \right.$$

The advantage of approximation (2.7) is that it is in the form of a standard state space representation, for which many analysis and control design techniques exist. We refer to [28], where (2.7) is at the basis of a design method for fixed-order $\mathcal{H}_2$ optimal controllers.

According to (2.4), it is natural to relate the initial condition in the definition of the fundamental solution K (see (1.2)) with initial condition $z(0) = E_N$ of (2.7), where

$$E_N = [0 \ \cdots \ 0 \ 1]^{\mathrm{T}} \otimes I_n.$$

This gives rise to the solution $z(t) = \mathrm{e}^{\mathcal{A}_N t} E_N$. Since $z(t)$ is a discrete approximation of the piece of trajectory $K(t+\theta), \ \theta \in [-\tau_m, \ 0]$, we can obtain an approximation of $K(t)$ by selecting the last block component of $z(t)$. This brings us to the approximation K_N of the fundamental matrix K, defined by

$$K_N(t) = E_N^{\mathrm{T}} \mathrm{e}^{\mathcal{A}_N t} E_N, \tag{2.9}$$

which by invoking (1.3) leads us, in turn, to an approximation $\mathcal{P}_N$ of P,

$$\begin{array}{rl} \mathcal{P}_N(t) = & \int_0^\infty K_N(s) B B^{\mathrm{T}} K(s+t) \mathrm{d}s \\ = & \int_0^\infty E_N^{\mathrm{T}} \mathrm{e}^{\mathcal{A}_N s} B_N B_N^{\mathrm{T}} \mathrm{e}^{\mathcal{A}_N^{\mathrm{T}}(s+t)} E_N \mathrm{d}s \\ = & E_N^{\mathrm{T}} \left(\int_0^\infty E_N^{\mathrm{T}} \mathrm{e}^{\mathcal{A}_N s} B_N B_N^{\mathrm{T}} \mathrm{e}^{\mathcal{A}_N^{\mathrm{T}} s} \mathrm{d}s \right) \mathrm{e}^{\mathcal{A}_N^{\mathrm{T}} t} E_N. \end{array} \tag{2.10}$$

Similarly, we can approximate Υ in (1.7) by the transfer function of (2.7), given by

$$\Upsilon_N(s) = C_N \left(sI - \mathcal{A}_N\right)^{-1} B_N. \tag{2.11}$$

The following proposition provides a computational expression for $\mathcal{P}_N$ in terms of a Lyapunov matrix equation.

PROPOSITION 2.1. *If matrix $\mathcal{A}_N$ is Hurwitz, we can express*

$$\mathcal{P}_N(t) = E_N^{\mathrm{T}} P_N \mathrm{e}^{\mathcal{A}_N^{\mathrm{T}} t} E_N, \tag{2.12}$$

where P_N satisfies the Lyapunov equation

$$\mathcal{A}_N P_N + P_N \mathcal{A}_N^{\mathrm{T}} + B_N B_N^{\mathrm{T}} = 0. \tag{2.13}$$

The next proposition expresses that the approximation (2.10) of the delay Lyapunov matrix, and the approximation of the transfer function, are consistent with respect to property (1.9).

PROPOSITION 2.2. *Function* (2.10) *and transfer function* (2.11) *satisfy*

$$\|\Upsilon_N\|_2^2 = \operatorname{tr}\left(C \mathcal{P}_N(0) C^{\mathrm{T}}\right). \tag{2.14}$$

2.2. Properties. We discuss properties of approximations (2.7) and (2.10) which are instrumental to the developments in the next sections and which shed further light on the difficulty of the problem of computing the delay Lyapunov matrix. They are illustrated by means of the didactic example

$$\dot{x}(t) = \frac{1}{2}x(t) - x(t-1) + u(t),\ y(t) = x(t). \tag{2.15}$$

Time domain. The function $t \mapsto K(t)$ is in general not analytic on $(0,\ \infty)$, due to the propagation of the discontinuity at $t = 0$. If function K has a discontinuity in its kth derivative[1] at some time $\hat{t} \geq 0$, then the function has, in the generic case, a discontinuity in its $(k+1)$th derivative at time instants $\hat{t} + \tau_i,\ i = 1, \ldots, m$. The increase of regularity is called the smoothing property of solutions [8].

Via definition (1.3) the nonsmoothness of K propagates to the function $t \geq 0 \mapsto P(t)$. Note here that we restrict to nonnegative t because of the so-called symmetry property $P(-t) = P(t)^{\mathrm{T}}$. In [14, section 4], it has been shown that the function P is in general not infinitely many times differentiable for $t \in S$, where

$$S = \left\{\vec{\tau} \cdot \vec{z} :\ \vec{z} \in \mathbb{Z}^m,\ \vec{\tau} \cdot \vec{z} > 0\right\},$$

where $\vec{\tau} = (\tau_1, \ldots, \tau_m)$ and $\vec{z} = (z_1, \ldots, z_m)$. In the commensurate delay case, where $\vec{\tau} = h\vec{n}$ with $n \in \mathbb{N}^m$ and $\gcd(\vec{n}) = 1$, we have $S = \{kh :\ k = 0, 1, 2, \ldots\}$. In the case of noncommensurate delays, set S is dense in $[0, \infty)$. In both cases, function P is continuous for $t \geq 0$, function $\dot{P}$ is continuous on $(0, \infty)$, while $\ddot{P}$ is continuous for all $t \in (0, \infty)$ except for $t = \tau_i,\ i \in \{1, \ldots, m\}$, but still of bounded variation. For more details we refer to [14]. As an illustration we plot the functions K and P, corresponding to (2.15), in Figure 2.1.

In Figure 2.2 we plot for system (2.15) the normalized errors

$$\frac{\max_{t \in [0,\ t_{\max}]} |P(t) - \mathcal{P}_N(t)|}{\max_{t \in [0,\ t_{\max}]} |P(t)|} \tag{2.16}$$

[1] $k = 0$ for a discontinuity in the function.

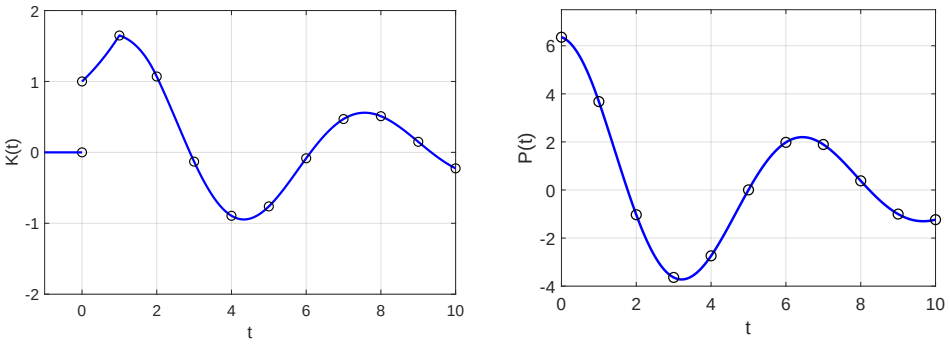

FIG. 2.1. *Plot of functions K and P for system* (2.15). *The circles correspond to time instants where the function is not infinitely many times differentiable. Function K (function P) exhibits a discontinuity in its kth derivative ($(k+1)$th derivative) at $t = k$ for all $k \in \mathbb{N}$.*

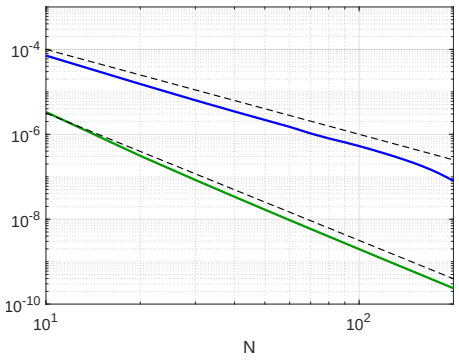

FIG. 2.2. *Normalized error* (2.16) *(blue curve) and* (2.17) *(green curve) as a function of N for system* (2.15). *The dashed lines indicate the rates $\mathcal{O}\left(N^{-2}\right)$ and $\mathcal{O}\left(N^{-3}\right)$.*

for $t_{\max} = 2$ and

$$\frac{|P(0) - \mathcal{P}_N(0)|}{|P(0)|} \tag{2.17}$$

as a function of N. Note that, as $B = C = 1$ for system (2.15), expression (2.17) corresponds to the normalized error on the squared $\mathcal{H}_2$ norm if the latter is approximated by $\|\Upsilon_N\|_2^2$; see (2.14). We observe the following rates of convergence: $\mathcal{O}\left(N^{-2}\right)$ for (2.16) and $\mathcal{O}\left(N^{-3}\right)$ for (2.17). In all other experiments, we observed the same rates of convergence.

The seemingly slow convergence, $\mathcal{O}\left(N^{-2}\right)$ for the maximum error of P on a compact interval, is expected in view of the smoothness properties of function P. As we have seen, P has discontinuities in its second derivative at $t = \tau_i$, $i = 1, \ldots, m$, with $\ddot{P}$ being of bounded variation, while function $\mathcal{P}_N$, defined by (2.10), is analytic on $\mathbb{R}$. Thus, we are approximating a nonsmooth function by a sequence of smooth functions. Note that we would obtain the same rate of convergence when approximating P on an interval by a series of polynomials interpolating in a Chebyshev mesh [26, Theorem 7.2]. As P is analytic in the interval $(0, \tau_1)$ and we only consider nonnegative

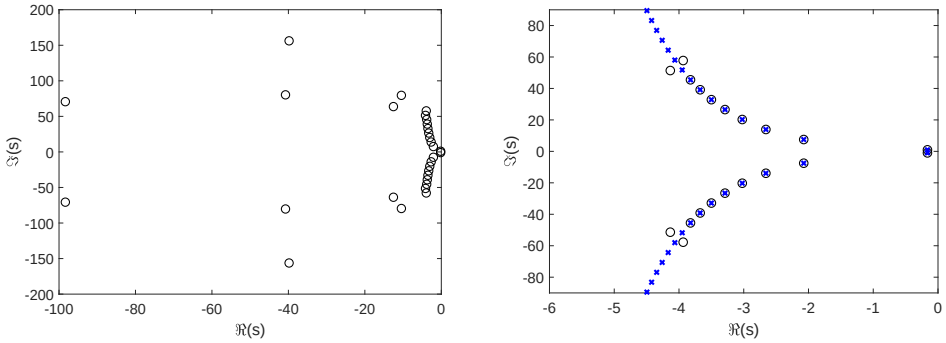

FIG. 2.3. *(left) All eigenvalues of $\mathcal{A}_N$, corresponding to system* (2.15)*, for $N = 30$ (black circles). (right) Zoom of the right part of the spectrum of $\mathcal{A}_N$, supplemented with the characteristic roots of the delay equation (blue stars). Its null solution is exponentially stable, with rightmost characteristic roots $-0.1629 \pm 0.9725j$.*

t, the convergence rate is better at $t = 0$. An extensive argumentation for the rate $\mathcal{O}(N^{-3})$ for the $\mathcal{H}_2$ norm approximation induced by $\|\Upsilon_N\|_2$ is given in [28].

Frequency domain. With the choice of the Chebyshev mesh (2.2) the asymptotic convergence of the individual eigenvalues of $\mathcal{A}_N$ to corresponding characteristic roots is fast. More specifically, in [2] it is proven that spectral accuracy (approximation error $O(N^{-N})$) is obtained. An additional property of using mesh (2.2) for discretizing (1.6), observed in extensive numerical experiments, is that the eigenvalues of $\mathcal{A}_N$, which have not yet converged to corresponding characteristic roots of (1.1), are very often located to the left of the eigenvalues that have already converged (but there is no guarantee). The latter is important with respect to preservation of stability. These properties are illustrated for system (2.15) in Figure 2.3. Finally, since the effect of the spectral discretization can be interpreted in terms of a rational approximation of functions $\lambda \to \exp(-\lambda\tau_i)$, $i = 1, \ldots, m$, around zero [30], convergence is almost invariably reached first for the smallest characteristic roots in modulus if N is gradually increased. Due to the characteristic shape on the spectrum of delay equation (exhibiting infinite root chains extending in the left half plane, along which the imaginary part grows exponentially as a function of the real part; see [29] for a detailed description), the rightmost, stability determining roots are typically among the smallest characteristic roots.

Regarding the approximation of the transfer function, the following moment matching property is proven in [19].

PROPOSITION 2.3. *The transfer functions* (1.7) *and* (2.11) *satisfy*

$$\left.\frac{\mathrm{d}^i \Upsilon_N(s)}{\mathrm{d}s^i}\right|_{s=0} = \left.\frac{\mathrm{d}^i \Upsilon(s)}{\mathrm{d}s^i}\right|_{s=0}, \quad i = 0, \ldots, N, \tag{2.18}$$

and

$$\left.\frac{\mathrm{d}^i \Upsilon_N(s^{-1})}{\mathrm{d}s^i}\right|_{s=0} = \left.\frac{\mathrm{d}^i \Upsilon(s^{-1})}{\mathrm{d}s^i}\right|_{s=0}, \quad i = 0, 1, \tag{2.19}$$

that is, the moments of $\Upsilon(s)$ and $\Upsilon_N(s)$ at zero match up to the Nth moment, and the moments at infinity match up to the first moment.

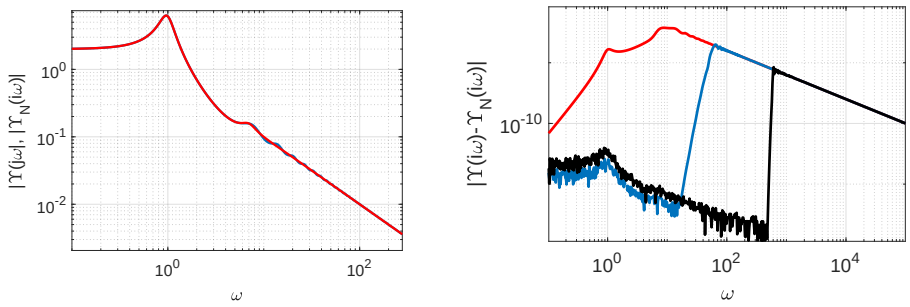

FIG. 2.4. *(left) Modulus of the transfer function of* (2.15) *(blue curve) and the corresponding approximation* (2.11) *for* $N = 5$ *(red curve), evaluated on the imaginary axis, i.e., for* $s = \imath\omega$, $\omega \geq 0$. *(right) Approximation error on the imaginary axis for* $N = 5$ *(red curve),* $N = 30$ *(blue curve), and* $N = 100$ *(black curve).*

By property (2.18), which corresponds to Hermite interpolation at $s = 0$, the region in the complex plane where the approximation is accurate extends from the origin as N is increased, consistently with the convergence behavior of characteristic root approximations sketched in the right pane of Figure 2.3. At the same time, the asymptotic decay rate of the transfer function for $\omega \to \infty$, which is described by CB/ω, is captured by property (2.19). Note that higher-order moments of (1.7) at infinity are not well defined, which is related to the property that $s = \infty$ is an *essential* singularity of Υ. As a consequence, the overall approximation error is mainly due to a mismatch in the mid-frequency range. This is illustrated in Figure 2.4, where we compare the transfer function of (2.15) and its approximation of form (2.11).

The right pane in Figure 2.4 gives a complementary explanation, besides the smoothness properties of the function $t \mapsto P(t)$, for why the convergence of $\mathcal{P}_N(0)$ to $P(0)$ has exhibited a low rate of convergence $\mathcal{O}(N^{-3})$, compared to the spectral convergence of the eigenvalues of $\mathcal{A}_N$: unlike an individual pole and the $\mathcal{H}_\infty$ norm, the $\mathcal{H}_2$ norm is a *global* characteristic of the transfer function, in the sense that an accurate computation involves approximating the transfer function well over whole the imaginary axis.

2.3. A reformulation of the discretized problem. In what follows we denote by T_i the Chebyshev polynomial of the first kind with order i, $i \in \mathbb{N}$. Let us start by defining matrix G_N as

$$G_N = \Sigma_N^{-1}\Pi_N, \tag{2.20}$$

where

$$\Pi_N = \frac{\tau_m}{4}\begin{bmatrix} \frac{4}{\tau_m} & \frac{4}{\tau_m} & \frac{4}{\tau_m} & \cdots & & \cdots & \frac{4}{\tau_m} \\ 2 & 0 & -1 & & & & \\ & \frac{1}{2} & 0 & -\frac{1}{2} & & & \\ & & \frac{1}{3} & 0 & -\frac{1}{3} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \frac{1}{N-1} & 0 & -\frac{1}{N-1} \\ & & & & & \frac{1}{N} & 0 \end{bmatrix} \otimes I \tag{2.21}$$

and

$$\Sigma_N = \begin{bmatrix} R_0 & R_1 & \cdots & R_N \\ & I_n & & \\ & & \ddots & \\ & & & I_n \end{bmatrix}, \tag{2.22}$$

with

$$R_i = A_0 T_i(1) + \sum_{k=1}^{m} A_k T_i\left(-2\frac{\tau_k}{\tau_m} + 1\right),\ i = 0, \ldots, N.$$

In addition, let us define matrices H_N and F_N as

$$H_N = \begin{bmatrix} R_0^{-1}\left(I - \frac{\tau_m}{2} R_1\right) R_0^{-1} B \\ \frac{\tau_m}{2} R_0^{-1} B \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad F_N = [R_0\ R_1\ \ \cdots\ R_N].$$

In [19] it has been proven that system (2.7) is equivalent to

$$\begin{cases} G_N \dot{\eta}(t) = \eta(t) + H_N u(t), \\ y(t) = C F_N \eta(t), \end{cases} \tag{2.23}$$

and we can express

$$\Upsilon_N(s) = C F_N (s G_N - I)^{-1} H_N. \tag{2.24}$$

In the same spirit, the following main theorem reformulates expressions (2.12)–(2.13) in terms of matrix G_N, which is similar to $\mathcal{A}_N^{-1}$. This gives the Lyapunov equation a favorable structure that will be exploited by the algorithms, presented in section 3.

THEOREM 2.4. *If matrix $\mathcal{A}_N$ is Hurwitz, then we can express $\mathcal{P}_N$ in* (2.12) *as*

$$\mathcal{P}_N(t) = F_N Q_N \mathrm{e}^{G_N^{-\mathrm{T}} t} F_N^{\mathrm{T}}, \tag{2.25}$$

where Q_N satisfies the Lyapunov equation

$$G_N Q_N + Q_N G_N^{\mathrm{T}} + H_N H_N^{\mathrm{T}} = 0. \tag{2.26}$$

Proof. In [19, section 3.1] it has been shown that

$$\mathcal{A}_N = (S_N \otimes I) G_N^{-1} (S_N^{-1} \otimes I), \tag{2.27}$$

where matrix $S_N \in \mathbb{R}^{(N+1)\times(N+1)}$ maps coefficients of a polynomial of degree N in the Chebyshev basis $\{T_i(2\frac{t}{\tau_m} + 1) :\ i = 0, \ldots, N\}$ onto the corresponding coefficients in the Lagrange basis, $\{l_{N,i}(t) :\ i = 1, \ldots, N+1\}$, defined on the mesh (2.2). Substituting (2.27) into (2.10) yields

$$\mathcal{P}_N(t) = \int_0^\infty \Big(E_N^{\mathrm{T}} (S_N \otimes I) \mathrm{e}^{G_N^{-1} s} (S_N^{-1} \otimes I) B_N \times B_N^{\mathrm{T}} (S_N^{-T} \otimes I) \mathrm{e}^{G_N^{-\mathrm{T}}(s+t)} (S_N^{\mathrm{T}} \otimes I) E_N \Big)\, \mathrm{d}s. \tag{2.28}$$

In the proof of Theorem 3.2 of [19] it has been shown that

$$(S_N^{-1} \otimes I)B_N = c_N \otimes B, \quad E_N^{\mathrm{T}}(S_N \otimes I) = \mathbf{1}_N^{\mathrm{T}} \otimes I, \tag{2.29}$$

with

$$c_N = \begin{cases} \frac{2}{N+1}\,[0\ 1\ 0\ 1\ \cdots\ 0\ 1]^{\mathrm{T}} \otimes B, & N \text{ odd}, \\ \frac{2}{N+1}\,[\frac{1}{2}\ 0\ 1\ 0\ 1\ \cdots\ 0\ 1]^{\mathrm{T}} \otimes B, & N \text{ even}, \end{cases}$$

and $\mathbf{1}_N = [1\ 1\ \cdots\ 1]^{\mathrm{T}}$. Using these expressions, as well as the identity $\mathrm{e}^{G_N^{-1}t} = G_N^{-1}\mathrm{e}^{G_N^{-1}t}G_N$, we can write (2.28) as

$$\begin{aligned} \mathcal{P}_N(t) = \int_0^\infty & \left(\mathbf{1}_N^{\mathrm{T}} \otimes I\right) G_N^{-1}\mathrm{e}^{G_N^{-1}s}G_N(c_N \otimes B) \\ & \times \left(c_N^{\mathrm{T}} \otimes B^{\mathrm{T}}\right) G_N^{\mathrm{T}}\mathrm{e}^{G_N^{-\mathrm{T}}(s+t)}G_N^{-\mathrm{T}}(\mathbf{1}_N \otimes I)\mathrm{d}s. \end{aligned} \tag{2.30}$$

A straightforward computation shows that $(\mathbf{1}_N^{\mathrm{T}}\otimes I)G_N^{-1} = F_N$ and $G_N(c_N\otimes B) = \hat{H}_N$, with $\hat{H}_N = \begin{bmatrix} (R_0^{-1}B)^{\mathrm{T}} & 0 & \cdots & 0 \end{bmatrix}^{\mathrm{T}}$. As a consequence, we can write

$$\mathcal{P}_N(t) = F_N \left(\int_0^\infty \mathrm{e}^{G_N^{-1}s}\hat{H}_N\hat{H}_N^{\mathrm{T}}\mathrm{e}^{G_N^{-\mathrm{T}}s}\ \mathrm{d}s \right) \mathrm{e}^{G_N^{-\mathrm{T}}t}F_N^{\mathrm{T}}. \tag{2.31}$$

Denoting the integral in (2.31) by $\hat{Q}_N$, we can express the latter, relying on the assumption that $\mathcal{A}_N$ and G_N^{-1} are Hurwitz, as the solution of the Lyapunov equation

$$G_N^{-1}\hat{Q}_N + \hat{Q}_N G_N^{-\mathrm{T}} + \hat{H}_N\hat{H}_N^{\mathrm{T}} = 0.$$

Premultiplying this equation with G_N and postmultiplying with G_N^{T} yields

$$\hat{Q}_N G_N^{\mathrm{T}} + G_N\hat{Q}_N + G_N\hat{H}_N\hat{H}_N^{\mathrm{T}}G_N^{\mathrm{T}} = 0.$$

Since we have $G_N\hat{H}_N = H_N$, it follows that $\hat{Q}_N = Q_N$, where Q_N uniquely solves (2.26). Hence, (2.31) corresponds to (2.25) and (2.26). ∎

Matrices Σ_N and Π_N have sparse structures that can be exploited. In what follows a key role will be played by the following property.

PROPOSITION 2.5. *Assume that $N_1, N_2 \in \mathbb{N}$ with $N_1 < N_2$. Then the matrices $\Sigma_{N_1}, \Pi_{N_1}, F_{N_1}, H_{N_1}$ in Theorem 2.4 are submatrices of $\Sigma_{N_2}, \Pi_{N_2}, F_{N_2}, H_{N_2}$.*

3. A dynamic subspace method. The discretization of the delay equation led us to the standard state space representation (2.7), which in turn led us to delay Lyapunov matrix approximations in explicit form, namely, (2.12)–(2.13) and (2.25)–(2.26). These properties are useful for the further developments. However, the dimension has increased from n to $(N+1)n$. At the same time relatively high values of N are expected for an accurate approximation, as motivated in section 2.2.

If Nn is large and matrix $B_N B_N^{\mathrm{T}}$, respectively, $H_N H_N^{\mathrm{T}}$, has low rank (in the sense of $r << Nn$), computing a low-rank approximation of P_N, respectively, Q_N, may be beneficial. In this section we construct an approximation inferred from the projection of the Lyapunov equation on a Krylov space of dimension kr. Before we present the construction in sections 3.2–3.4, we use another didactic example to motivate important methodological choices regarding (i) the relation between parameters N and k, (ii) the choice of the Krylov space, and (iii) the system matrix/Lyapunov equation to be projected on this space. We conclude with an interpretation in terms

of projecting an infinite-dimension linear system in section 3.5. Some implementation aspects are discussed in Appendix A.

Since the technical derivations involve many steps, we included Figure 3.2 at the end of the section in order to keep an overview of the main steps and corresponding notations.

3.1. Motivation of methodological choices. We consider system

(3.1)
$$\dot{x}(t) = \begin{bmatrix} -0.08 & -0.03 & 0.2 \\ 0.2 & -0.04 & -0.005 \\ -0.06 & 0.2 & -0.07 \end{bmatrix} x(t) + \begin{bmatrix} -0.0471 & -0.0504 & -0.0602 \\ -0.0942 & -0.1008 & -0.1204 \\ 0.0471 & 0.0504 & 0.0602 \end{bmatrix} x(t-5) + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u(t), \quad y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t).$$

For $N = 50, 100, 150$, and 200 we computed matrices P_N and Q_N, solving Lyapunov equations (2.13) and (2.26). We display in Figure 3.1 their ordered singular values, normalized such that the leading singular value equals to one. This experiment indicates that the solution of Lyapunov equation (2.26), inferred from the representation (2.23), is more amendable for a low-rank approximation.

Concerning the input-output behavior, Proposition 2.3 expresses that functions Υ and Υ_N match $N+1$ moments at zero and two at infinity. To have these matching moments carried over by a projection of (2.7) on a right Krylov space, one needs in general a subspace of dimension $N+2$. At the same time, if more than $N+1$ moments at zero are preserved by the projection, or more than two at infinity, the highest order moments won't match anymore with those of the original transfer function (1.7). This can be interpreted as an instance of "over-fitting" in the sense that particularities of the discretization (2.7) are captured by the projection, which are not present in the original delay equation and related to the discretization error. Similar conclusions can be made from the experiment related to Figure 3.1. On a compact interval for index i, the eigenvalue functions of Q_N uniformly converge for $N \to \infty$ to the limit function indicated in black, which is related to the original (nondiscretized) delay equation (we come back to this in section 3.5). Important to observe is that, for a given value of N, *less than* N singular values are related to the limit behavior. This indicates that the

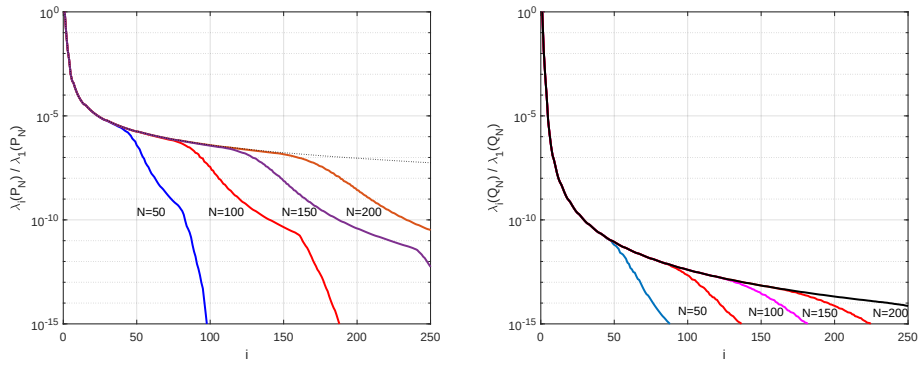

FIG. 3.1. *Normalized eigenvalues ($\lambda_i(\cdot)$ denoting the ith eigenvalue in decreasing order) of matrix P_N and Q_N, computed for system* (3.1).

choice $k > N$ could lead to a similar instance of over-fitting, at least for a best rank-k approximation of Q_N. All the above elements motivate us to ensure N is sufficiently large, such that the dimension of the subspace k satisfies

$$k \leq N \tag{3.2}$$

and, preferably, $k << N$.

The typical spectrum distribution of delay equations, with rightmost characteristic roots close to the origin, the properties of the spectral discretization illustrated in Figure 2.3, and the above reasoning with respect to matching moments, suggest building a Krylov space using matrix $\mathcal{A}_N^{-1}$. Furthermore, from extensive numerical experiments including system (3.1) we have observed an advantage of projecting matrix $\mathcal{A}_N^{-1}$ on this Krylov space, compared to projecting $\mathcal{A}_N$, which is explained by a better separation of the targeted characteristic roots after an inversion of the spectrum.

The preference for building a Krylov space for $\mathcal{A}_N^{-1}$ and for projecting this matrix, the property that G_N is similar to $\mathcal{A}_N^{-1}$, and, last but not least, the typically faster decay of singular values of Q_N than those of P_N naturally lead us to the representation (2.25)–(2.24) of the discretized system and associated approximation of the delay Lyapunov matrix. Condition (3.2), along with the convergence behavior of $\mathcal{P}_N$ to P, make a discretize-first approach less attractive, where standard Lyapunov matrix equation (2.26) is solved for a fixed value of N using state-of-the-art algorithms (such as a low-rank ADI method or a (rational) Krylov method; see [24] for an overview). They call for an algorithm that does not rely on an a priori choice of critical parameter N, similar to the infinite Arnoldi method for eigenvalue computations [11]. Such an algorithm is constructed in what follows.

3.2. Dynamic construction of a Krylov space. We fix integer k and assume N large enough such that (3.2) holds. We consider the block Krylov space

$$\mathcal{K}_k(G_N, b) = \text{span}\{b, G_N b, \ldots, G_N^{k-1} b\}, \tag{3.3}$$

where b is a block vector of size $(N+1)n \times r$, having the structure

$$b = [x_0^{\mathrm{T}}\ 0\ \cdots 0]^{\mathrm{T}}, \tag{3.4}$$

with $x_0 \in \mathbb{R}^{n\times r}$ to be specified in section 3.3. The block Arnoldi algorithm builds the Krylov sequence, block vector by block vector, where these vectors are orthogonalized. Due to the special structure of b and the fact that G_N is a block Hessenberg matrix, whose blocks have size $n\times n$, the block vectors $G_N b, \ldots, G_N^{k-1} b$ only have their first $2n$, $3n$, $\ldots$, kn block rows different from zero. Moreover, in computing the matrix vector products with (3.4), only submatrices of G_N are needed. Hence, in the computation of the Krylov space, we can restrict to storing only the nonzero part of the block vectors and using the relevant part of G_N. This leads us to the following procedure:

1. Apply Algorithm 1 for computing a basis of $\mathcal{K}_k(G_{k-1}, [x_0^{\mathrm{T}}\ 0 \cdots 0]^{\mathrm{T}})$. There we use notation common for Arnoldi iterations: we let $\underline{\mathcal{H}}_i \in \mathbb{R}^{(i+1)r\times ri}$ denote the constructed rectangular block Hessenberg matrix and $\mathcal{H}_i \in \mathbb{R}^{ri\times ri}$ the corresponding $i \times i$ upper blocks.
2. A basis for

$$\mathcal{K}_k(G_N, [x_0^{\mathrm{T}}\ 0\ \cdots\ 0]^{\mathrm{T}}) \tag{3.5}$$

is spanned by the columns of

$$V_{N,k} = \begin{bmatrix} \mathbf{V}_k^{\mathrm{T}} & 0 & \cdots & 0 \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{(N+1)n \times kr}, \tag{3.6}$$

while, due to the structure of G_N, expression $\mathcal{H}_k = V_{N,k}^{\mathrm{T}}\, G_N\, V_{N,k}$, holds, i.e., $\mathcal{H}_k$ can be considered as an orthogonal projection of G_N on a k-dimensional Krylov subspace, for *any* N satisfying (3.2).

Algorithm 1. A structure exploiting block Arnoldi algorithm.

Require: $x_0 \in \mathbb{R}^{n\times r}$ of full column rank, number of iterations k

1: Let $x_0 = Q_0\tilde{R}_0$ be the reduced QR factorization of x_0. Set $\mathbf{V}_1 = Q_0$ and let $\underline{\mathcal{H}}_0$ be the empty matrix
2: **for** $i = 1, 2, \ldots, k$ **do**
3: Let $W_i = G_i \begin{bmatrix} Q_{i-1} \\ 0 \end{bmatrix}$
4: Compute $H_i = [\mathbf{V}_i^{\mathrm{T}}\ 0]\ W_i$ and then $\hat{W}_i = W_i - \begin{bmatrix} \mathbf{V}_i \\ 0 \end{bmatrix} H_i$ (orthogonalization)
5: Compute $\hat{W}_i = Q_i\tilde{R}_i$ as the reduced QR factorization of $\hat{W}_i$ (normalization)
6: Let $\underline{\mathcal{H}}_i = \begin{bmatrix} \underline{\mathcal{H}}_{i-1} & H_i \\ 0 & \tilde{R}_i \end{bmatrix} \in \mathbb{R}^{(i+1)r\times ir}$
7: Expand $\mathbf{V}_i$ into $\mathbf{V}_{i+1} = \left[\begin{array}{c|c} \mathbf{V}_i & \\ 0 & Q_i \end{array}\right]$
8: **end for**

Output: matrix $\mathbf{V}_k$, whose columns are an orthogonal basis for $\mathcal{K}_k(G_{k-1}, [x_0^{\mathrm{T}}\ 0 \cdots\ 0]^{\mathrm{T}})$, $\mathcal{H}_k$, $\underline{\mathcal{H}}_k$, satisfying $\mathcal{H}_k = \mathbf{V}_k^{\mathrm{T}} G_{k-1} \mathbf{V}_k$.

3.3. Dynamic approximation of the transfer function. We now arrive at the derivation of an approximation of $\Upsilon_N(s)$, defined by (2.11) or (2.24), having a prescribed order kr. Here we take once again the assumption that (3.2) is satisfied. For this we construct the Krylov space (3.5) and project matrices F_N, G_N, H_N, defined in Theorem 2.4, on this Krylov space. An orthogonal projection yields the following approximation of $\Upsilon_N(s)$:

$$\boldsymbol{\Upsilon}_k(s) = \mathbf{F}_k\ (s\mathbf{G}_k - I)^{-1}\ \mathbf{H}_k, \tag{3.7}$$

where

$$\begin{array}{lll} \mathbf{F}_k = CF_N\ V_{N,k} & = CF_{k-1}\mathbf{V}_k, \\ \mathbf{G}_k = V_{N,k}^{\mathrm{T}}\ G_N\ V_{N,k} & = \mathcal{H}_k, \\ \mathbf{H}_k = V_{N,k}^{\mathrm{T}}\ H_N & = \mathbf{V}_k^{\mathrm{T}} H_{k-1}, \end{array} \tag{3.8}$$

matrix $\mathbf{V}_k$ and $\mathcal{H}_k$ refer to the output of Algorithm 1, and $V_{N,k}$ is given by (3.6). The matrices of the reduced model (3.7) do *not* depend on N. Furthermore, matrices $\mathbf{F}_k$ and $\mathbf{H}_k$ are *submatrices* of $\mathbf{F}_{k+1}$ and $\mathbf{H}_{k+1}$. Therefore, they can be constructed in a dynamic way when doing iterations of Algorithm 1, as is the case with the Hessenberg matrix $\mathbf{G}_k = \mathcal{H}_k$.

With a particular choice of the vector x_0 in (3.5), the transfer function (3.7) satisfies the following moment matching property.

PROPOSITION 3.1 (see [19, Theorem 11]). *Let $N, k \in \mathbb{N}$ with $N \geq k \geq 2$ and let the Krylov space* (3.5) *be constructed from $x_0 = R_0^{-1}B$. Then transfer function* (3.7) *satisfies*

$$\left.\frac{\mathrm{d}^i \boldsymbol{\Upsilon}_k(s)}{\mathrm{d}s^i}\right|_{s=0} = \left.\frac{\mathrm{d}^i \Upsilon(s)}{\mathrm{d}s^i}\right|_{s=0}, \quad i = 0, \ldots, k-2, \tag{3.9}$$

and

$$\left.\frac{\mathrm{d}^i \boldsymbol{\Upsilon}_k(s^{-1})}{\mathrm{d}s^i}\right|_{s=0} = \left.\frac{\mathrm{d}^i \Upsilon(s^{-1})}{\mathrm{d}s^i}\right|_{s=0}, \quad i = 0, 1. \tag{3.10}$$

Note that Proposition 3.1 concerns the matching of moments with the transfer function of the *original* delay system (1.6). This is due to to the property that the moments, preserved by projection of the discretized system, are precisely matching moments between the discretized system and the delay system, by Proposition 2.3.

3.4. Dynamic approximation of the delay Lyapunov matrix. The evaluation of $\mathcal{P}_N(t)$, defined by (2.25), relies on solving Lyapunov equation (2.26). An established way to solve large-scale Lyapunov equations consists of computing a low-rank approximation obtained from the projection of the Lyapunov equation on a Krylov space. See, e.g., [23] and the references therein.

To determine an appropriate Krylov space, it is useful to express Q_N in terms of matrix exponentials,

$$Q_N = \int_0^\infty \mathrm{e}^{G_N^{-1} s} \left(G_N^{-1} H_N\right) \left(H_N^{\mathrm{T}} G_N^{-\mathrm{T}}\right) \mathrm{e}^{G_N^{-\mathrm{T}} s} \mathrm{d}s. \tag{3.11}$$

Hence, a low-rank approximation of Q_N can be induced by approximating the action of $\mathrm{e}^{G_N^{-1} t}$ on vector(s) $(G_N^{-1} H_N)$ in a low-dimensional space. This motivates us to include

$$G_N^{-1} H_N = \left[\ (R_0^{-1} B)^T\ 0\ \cdots\ 0\ \right]^T$$

in the Krylov space. Furthermore, since the rightmost characteristic roots of a delay equation are typically very well approximated by the dominant eigenvalues of G_N (equivalently, the smallest eigenvalues of $\mathcal{A}_N$ in modulus), while the largest eigenvalues of A_N have no correspondence with characteristic roots (see the arguments in section 2.2 and the illustration in Figure 2.3), approximating the dominant eigenspace of G_N should be favored. This brings us once again to Krylov space (3.5) with starting vector $x_0 = R_0^{-1} B$.

Replacing Q_N in (2.26) by $V_{N,k} \mathbf{Q_k} V_{N,k}^{\mathrm{T}}$ and requiring the residual to be orthogonal with respect to the Krylov space, we arrive at the projected Lyapunov equation

$$\mathbf{G}_k \mathbf{Q}_k + \mathbf{Q}_k \mathbf{G}_k^T + \mathbf{H}_k \mathbf{H}_k^{\mathrm{T}} = 0. \tag{3.12}$$

Hence, under the assumption that $\mathbf{G}_k$ is invertible we can approximate

$$\begin{array}{rcl} Q_N & \approx & V_{N,k} \mathbf{Q}_k V_{N,k}^{\mathrm{T}} \\ & = & \int_0^\infty V_{N,k} \mathrm{e}^{s \mathbf{G}_k^{-1}} (\mathbf{G}_k^{-1} \mathbf{H}_k)\ (\mathbf{H}_k^{\mathrm{T}} \mathbf{G}_k^{-\mathrm{T}}) \mathrm{e}^{s \mathbf{G}_k^{-\mathrm{T}}} V_{N,k}^{\mathrm{T}}\ \mathrm{d}s. \end{array} \tag{3.13}$$

Let us now compare approximation (3.13) with expression (3.11). By construction of the Krylov space we have $G_N^{-1} H_N = V_{N,k} \beta$ for some matrix β of appropriate dimensions. As a consequence,

$$H_N = G_N V_{N,k} \beta \ \Rightarrow\ \mathbf{H}_k = \mathbf{G}_k \beta.$$

Thus, the approximation of Q_N as in (3.13) can be interpreted in terms of the approximation

$$\mathrm{e}^{tG_N^{-1}}(G_N^{-1}H_N) = \mathrm{e}^{tG_N^{-1}}(V_{N,k})\,\beta \approx V_{N,k}\mathrm{e}^{t\mathbf{G}_k^{-1}}\beta. \tag{3.14}$$

Substituting the right-hand side of (3.13) into (2.25) we get

$$\begin{aligned}\mathcal{P}_N(t) &\approx F_N V_{N,k}\mathbf{Q}_k V_{N,k}^{\mathrm{T}}\mathrm{e}^{G_N^{-\mathrm{T}}t}F_N^{\mathrm{T}}\\ &= F_N V_{N,k}\mathbf{Q}_k\left(\mathrm{e}^{tG_N^{-1}}V_{N,k}\right)^{\mathrm{T}}F_N^{\mathrm{T}}.\end{aligned} \tag{3.15}$$

To approximate $\mathrm{e}^{tG_N^{-1}}V_{N,k}$, we use the same principle underlying (3.14). More precisely, we build a Krylov space, span $\left\{V_{N,k},\ G_N V_{N,k},\ \ldots, G_N^k V_{N,k}\right\}$. Since the columns of $V_{N,k}$ already span a Krylov space, this can be done by doing k more iterations of Algorithm 1, provided condition (3.2) on N is strengthened to

$$2k \le N. \tag{3.16}$$

It results in a basis $V_{N,2k}$ such that $V_{N,k} = V_{N,2k}\left[\begin{smallmatrix} I \\ 0 \end{smallmatrix}\right]$; hence, we can approximate

$$\mathrm{e}^{tG_N^{-1}}V_{N,k} \approx V_{N,2k}\mathrm{e}^{t\mathbf{G}_{2k}^{-1}}\begin{bmatrix} I \\ 0 \end{bmatrix}. \tag{3.17}$$

Finally, combining (3.15) and (3.17) we arrive at the following approximation of $\mathcal{P}_N(t)$ and thus of the Lyapunov matrix $P(t)$:

$$\mathbf{P}_k(t) = [R_0\ R_1\ \cdots R_{k-1}]\mathbf{V}_k\mathbf{Q}_k\,[I\ 0]\,\mathrm{e}^{t\mathbf{G}_{2k}^{-T}}\mathbf{V}_{2k}^{\mathrm{T}}\begin{bmatrix} R_0^{\mathrm{T}} \\ R_1^{\mathrm{T}} \\ \vdots \\ R_{2k-1}^{\mathrm{T}} \end{bmatrix}, \tag{3.18}$$

where $\mathbf{Q}_k$ satisfies (3.12). This brings us to Algorithm 2.

Algorithm 2. Construction of a (uniformly) low-rank approximation of the delay Lyapunov matrix.

Require: $B \in \mathbb{R}^{n\times r}$ of full column rank, parameter k determining number of Arnoldi iterations

1: Set $x_0 = R_0^{-1}B$ and perform $2k$ iterations of Algorithm 1, resulting in $\mathbf{V}_{2k}$ and $\mathbf{G}_{2k} = \mathcal{H}_{2k}$; set

$$\mathbf{G}_k = \begin{bmatrix} I_{kr} & 0 \end{bmatrix}\mathbf{G}_{2k}\begin{bmatrix} I_{kr} \\ 0 \end{bmatrix}.$$

2: Construct matrices $\mathbf{H}_k = \mathbf{V}_k^{\mathrm{T}}H_{k-1}$ and $\mathbf{L}_k = [R_0\ R_1\cdots R_{2k-1}]\mathbf{V}_{2k}$.

3: Solve Lyapunov equation (3.12) for $\mathbf{Q}_k$.

Output: matrices $\mathbf{L}_k,\ \mathbf{Q}_k,\ \mathbf{G}_{2k}$ from which $\mathbf{P}_k$ can be constructed according to (3.18).

Finally we note that the low-order approximation (3.7) of transfer function Υ and the approximation (3.18) of Lyapunov matrix $P(t)$ of rank smaller than or equal to kr are still consistent, in view of Propositions 1.9 and 2.2.

PROPOSITION 3.2. *We can express* $\|\boldsymbol{\Upsilon}_k\|_2^2 = \mathrm{Tr}\left(C\mathbf{P}_k(0)C^{\mathrm{T}}\right)$.

Proof. From (3.18) we directly have

$$\begin{aligned}\mathrm{Tr}\left(C\mathbf{P}_k(0)C^{\mathrm{T}}\right) &= \mathrm{Tr}\left(CF_N\mathbf{V}_{N,k}\mathbf{Q}_k\mathbf{V}_{N,k}^{\mathrm{T}}F_N^{\mathrm{T}}C^{\mathrm{T}}\right)\\ &= \mathrm{Tr}\left(\mathbf{F}_k\mathbf{Q}_k\mathbf{F}_k^{\mathrm{T}}\right)\end{aligned}$$

The latter expression, combined with (3.12), characterizes the $\mathcal{H}_2$ norm of $\mathbf{\Upsilon}_k$. ∎

Algorithm 2 is fully dynamic, in the sense that by increasing iteration count k, matrices $\mathbf{V}_k$, $\mathbf{G}_k$, $\mathbf{L}_k$, etc., only need to be extended or updated, hence, the iteration can be resumed if the accuracy is deemed insufficient. Implementation aspects, including stopping criteria if k is not chosen a priori, and the computational complexity are briefly discussed in Appendix A.

3.5. Interpretation as a projection method applied directly to an infinite-dimensional system. The spectral discretization in section 2 resulted in a finite-dimensional approximation of dimension $(N+1)n$, which can be rewritten as

$$\left\{\begin{array}{rl}\Pi_N\dot{c}_N(t) &= \Sigma_N c_N(t) + (\mathbf{1}_N \otimes B)u(t),\\ y(t) &= (\mathbf{1}_N^{\mathrm{T}} \otimes C)x(t).\end{array}\right. \tag{3.19}$$

Here we employed the identities $G_N\Sigma_N^{-1}(\mathbf{1}_N \otimes B) = H_N$ and $(\mathbf{1}_N^{\mathrm{T}} \otimes C)G_N^{-1} = F_N$. The discretization was key in the technical derivation of Algorithm 2, yet surprisingly parameter N plays no role at the end:

- the execution of Algorithms 1–2, as well as the stopping criteria discussed in Appendix A, do not rely on a choice of this parameter;
- the algorithms are dynamic in the sense that the iterative processes can always be resumed;
- Proposition 3.1 directly connects moments of transfer functions $\mathbf{\Upsilon}_k$ and Υ.

As a matter of fact, it is only implicitly assumed that N is *sufficiently large*, such that (3.16) holds. The above properties are due to the fact that Algorithms 1–2 only rely on the use of submatrices of Σ_N, Π_N, at the top-left position. In the remainder of this section we derive an alternative representation of the original time-delay system, where the dynamics are governed by infinite extensions of these matrices. Hence, by using only information contained in the top-left blocks Algorithms 1–2 can be interpreted as directly applied to the delay system, making them truly "discretization free."

We reconsider system (1.6) and define

$$v(\theta,t) = x(t+\theta), \quad \theta \in [-\tau_m,\ 0], \quad t \geq 0.$$

Solutions of (1.6), starting at $t = 0$, are continuous for $t \geq 0$, and they satisfy the advection PDE

$$\left\{\begin{array}{ll}\frac{\partial v}{\partial t}(\theta,t) - \frac{\partial v}{\partial \theta}(\theta,t) = 0, & \theta \in [-\tau_m,\ 0), \quad t \geq \tau_m,\\ \frac{\partial v}{\partial t}(0,t) = A_0 v(0,t) + \sum_{i=1}^m A_i v(-\tau_i,t) + Bu(t), & t \geq \tau_m;\end{array}\right. \tag{3.20}$$

see [17]. Let us represent $v(\theta,t)$ in a Chebyshev series [26] in variable θ on the interval $[-\tau_m,\ 0]$,

$$v(\theta,t) = \sum_{j=0}^{\infty} c_j(t) T_j\left(\tfrac{2\theta}{\tau_m} + 1\right), \ \theta \in [-\tau_m, 0].$$

The second equation in (3.20) then becomes

(3.21)
$$\begin{aligned}\textstyle\sum_{j=0}^{\infty}\dot{c}_j(t)T_j(1) &= A_0\left(\textstyle\sum_{j=0}^{\infty}c_j(t)T_j(1)\right)+\sum_{i=1}^{m}A_i\left(\textstyle\sum_{j=0}^{\infty}c_j(t)T_j\left(-\frac{2\tau_i}{\tau_m}+1\right)\right)\\ &\quad + Bu(t)\\ &= \textstyle\sum_{j=0}^{\infty}c_j(t)\left(A_0+\sum_{i=1}^{m}A_iT_j\left(-\frac{2\tau_i}{\tau_m}+1\right)\right)+Bu(t).\end{aligned}$$

In the same way the first equation in (3.20) becomes

$$\sum_{j=0}^{\infty}\dot{c}_j(t)T_j\left(\frac{2\theta}{\tau_m}+1\right)=\sum_{j=1}^{\infty}c_j(t)\frac{2j}{\tau_m}U_{j-1}\left(\frac{2\theta}{\tau_m}+1\right), \tag{3.22}$$

where we employed the property $\dot{T}_{j+1}(\theta) = (j+1)U_j(\theta)$, with U_j the Chebyshev polynomial of the second kind and order j, for $j \geq 0$.

For $j \geq 2$ we can substitute expression

$$T_j\left(\frac{2\theta}{\tau_m}+1\right)=\frac{1}{2}\left(U_j\left(\frac{2\theta}{\tau_m}+1\right)-U_{j-2}\left(\frac{2\theta}{\tau_m}+1\right)\right)$$

in (3.22), as well as

$$T_1\left(\frac{2\theta}{\tau_m}+1\right)=\frac{1}{2}U_1\left(\frac{2\theta}{\tau_m}+1\right),\quad T_0\left(\frac{2\theta}{\tau_m}+1\right)=U_0\left(\frac{2\theta}{\tau_m}+1\right).$$

Multiplying subsequently the left- and right-hand sides of (3.22) with

$$U_{i-1}\left(\frac{2\theta}{\tau_m}+1\right)\sqrt{1-\left(\frac{2\theta}{\tau_m}+1\right)^2},$$

taking the integral in θ from $-\tau_m$ to zero, and considering the orthogonality properties of Chebyshev polynomials of the second kind, we arrive at

$$\begin{array}{l}\dot{c}_0(t)-\frac{1}{2}\dot{c}_2(t)=\frac{2}{\tau_m}c_1,\\ \frac{1}{2}\left(\dot{c}_{i-1}(t)-\dot{c}_{i+1}(t)\right)=\frac{2i}{\tau_m}c_i,\quad i\geq 2.\end{array} \tag{3.23}$$

Letting $\mathbf{c} = \begin{bmatrix}c_0^{\mathrm{T}} & c_1^{\mathrm{T}} & \cdots\end{bmatrix}^{\mathrm{T}}$, $\mathbf{e_1} = [1\ 0\ \cdots]^{\mathrm{T}}$, and $\mathbf{1} = [1\ 1\ \cdots]^{\mathrm{T}}$, differential equations (3.21) and (3.23), and the output equation in (1.6) can be written as

$$\left\{\begin{array}{lcl}\Pi_\infty\dot{\mathbf{c}}(t) &=& \Sigma_\infty\mathbf{c}(t)+(\mathbf{e_1}\otimes B)\,u(t),\\ y(t) &=& \left(\mathbf{1}^{\mathrm{T}}\otimes C\right)\mathbf{c}(t),\end{array}\right. \tag{3.24}$$

with

$$\Pi_\infty=\frac{\tau_m}{4}\begin{bmatrix}\frac{4}{\tau_m} & \frac{4}{\tau_m} & \frac{4}{\tau_m} & \cdots & \cdots & \cdots\\ 2 & 0 & -1 & & & \\ & \frac{1}{2} & 0 & -\frac{1}{2} & & \\ & & \frac{1}{3} & 0 & -\frac{1}{3} & \\ & & & \ddots & \ddots & \ddots\end{bmatrix}\otimes I,\quad \Sigma_\infty=\begin{bmatrix}R_0 & R_1 & \cdots\\ & I_n & \\ & & \ddots\end{bmatrix}.$$

Note that by truncating the matrices to the first $N+1$ block components (or, equivalently, applying a Galerkin projection on the range of $\left([I_{n(N+1)}\ 0\cdots]^{\mathrm{T}}\right)$, one arrives at (3.19).

We note that, for the case of approximating characteristic roots by the reciprocal of eigenvalues of $\mathbf{G}_k$, a related interpretation of Algorithm 1 is given in [11], in terms of the standard Arndoli's method applied to an operator eigenvalue problem.

Finally, an overview of the developments throughout sections 2–3 is given by Figure 3.2

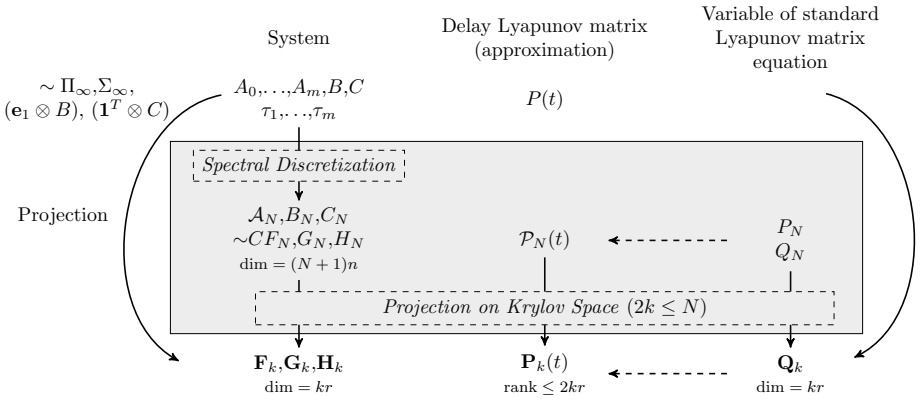

FIG. 3.2. *Overview of different steps in the derivations and corresponding notation. If only the $\mathcal{H}_2$ norm needs to be approximated, a Kyrlov space of dimension k such that $k \leq N$ is sufficient. With the relation between k and N satisfied, the delay Lyapunov matrix and $\mathcal{H}_2$ norm approximations, obtained after projection of the discretized system, do not depend on the value of N, only on k. This leads to the interpretation spelled out in section* 3.5 *and illustrated with the curved arrows.*

4. Experiments. We first consider the model for a heat exchanger described in [29], for which a controller (based on a combination of static state feedback and proportional integral control) has been determined by optimizing the spectral abscissa using the method of [18]. The closed-loop system is described by a delay equation of form (1.6) with $n = 5$ state variables and $m = 7$ delays. The nonzero elements of matrices $A_i,\ i = 0, \ldots, 7$, are specified as

(4.1)

A_0	$(2,1):\ \frac{1}{3},\ (2,2):\ -\frac{2}{3},\ (3,3):\ -\frac{1}{3}\ (5,4):\ -1$
A_1	$(4,3):\ 0.0324$
A_2	$(1,1):\ -0.07142857143$
A_3	$(4,4):\ -0.04$
A_4	$(2,4):\ \frac{1}{3}$
A_5	$(1,1):\ -0.01219364644,\ (1,2):\ -0.05460277319,\ (1,3):\ -0.1005215423$
	$(1,4):\ -0.1290047174,\ (1,5):\ 0.005063395489$
A_6	$(3,2):\ 0.3133333333$
A_7	$(1,2):\ 0.01714285714,$

while input matrix B, output matrix C, and the delay values are given by

$$B = \begin{bmatrix} 0.0278571429 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \quad C = I, \tag{4.2}$$
$$\begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 & \tau_5 & \tau_6 & \tau_7 \end{bmatrix} = \begin{bmatrix} 2.8 & 6.5 & 9.2 & 13 & 13.2 & 18 & 40 \end{bmatrix}.$$

In the left pane of Figure 4.1 we plot the normalized error on the Lyapunov matrix,

$$\frac{\max_{t\in[0,\ t_{\max}]} |P(t) - \mathbf{P}_k(t)|}{\max_{t\in[0,\ t_{\max}]} |P(t)|}, \tag{4.3}$$

for $t_{\max} = 50$ as a function of k, computed using Algorithm 2. We also show the normalized error on the $\mathcal{H}_2$ norm,

$$\frac{|\|\Upsilon\|_2 - \|\boldsymbol{\Upsilon}_k\|_2|}{\|\Upsilon\|_2}. \tag{4.4}$$

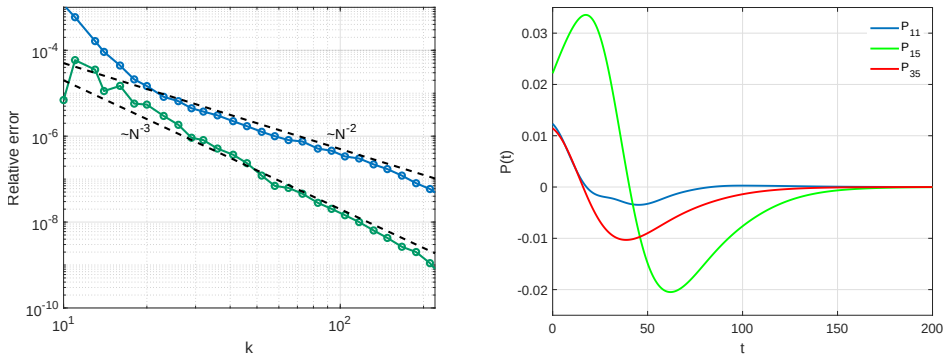

FIG. 4.1. *(left) Normalized errors* (4.3) *with* $t_{\max} = 50$ *(blue curve) and* (4.4) *(green curve) as a function of* k *for system* (1.6) *with matrices and delays* (4.1)–(4.2). *The dashed lines indicate the rates* $\mathcal{O}\left(k^{-2}\right)$ *and* $\mathcal{O}\left(k^{-3}\right)$. *(right) Some elements of* $P(t)$ *as a function of* t *for system* (1.6) *with matrices and delays* (4.1)–(4.2).

Finally, the evolution of selected elements of the Lyapunov matrix $P(t)$ is shown in the right pane of Figure 4.1.

Even though the dimension n is small, the advantage of using a projection method is significant. To illustrate this, when choosing $k = 100$ the application of Algorithm 2 involves the solution of a matrix Lyapunov equation of dimension 100×100, leading to an error on the $\mathcal{H}_2$ norm approximation smaller than $2\ 10^{-8}$; see Figure 4.1 (left). At the same time, when discretizing the delay equation into an ordinary equation as in section 2, with $N = 19$, and computing an $\mathcal{H}_2$ norm approximation via (2.14), one also has to solve a Lyapunov equation of size 100×100, but the error is then around 10^{-6}. The underlying reason is that the former approach can be interpreted in terms of a much more accurate discretization with $N > 99$ points, followed by 100 steps of an Arnoldi iteration (see Figure 3.2).

For the second and third examples we consider models described by PDEs

$$\frac{\partial v(x,t)}{\partial t} = \frac{\partial^2 v(x,t)}{\partial x^2} - \frac{1}{4} x\ v(x,t-1) \tag{4.5}$$

and

$$\frac{\partial v(x,t)}{\partial t} = \frac{\partial^2 v(x,t)}{\partial x^2} - 2\sin(x) v(x,t) + 2\sin(x) v(\pi - x, t-1), \tag{4.6}$$

with in both cases $v(0,t) = v(\pi,t) = 0$. The equations, which are variants of examples in [3], can be interpreted as heat equations describing the temperature in a rod, controlled with distributed delayed feedback. In (4.5) the feedback is proportional and localized, while in (4.6) it corresponds to Pyragas type feedback and it is nonlocalized. We discretize differential equations (4.5)–(4.6) in space using central differences. For (4.6), for instance, this results in a systems of the form (1.1) with matrices

$$A_0 = \left(\frac{n-1}{\pi}\right)^2 \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} - 2\Delta_0$$

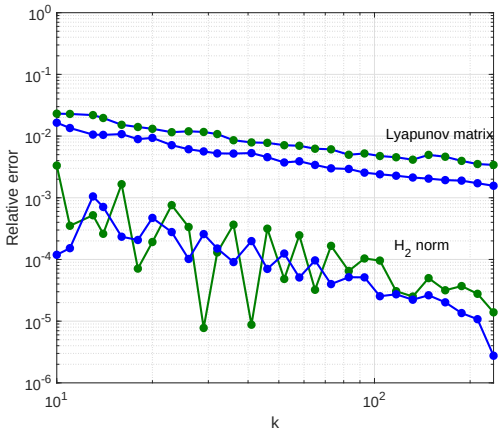

FIG. 4.2. *Normalized errors* (4.3)*, with* $t_{\max} = 3$*, and* (4.4) *as a function of* k*, with matrices obtained form the spatial discretization of* (4.5) *(blue curves) and* (4.6) *(green curves), such that* $n = 10000$*.*

and $A_1 = 2\Delta_{-1}$. Here $A_0,\ A_1 \in \mathbb{R}^{n\times n}$, and Δ_0 is a diagonal matrix containing the elements of the vector $(0, \sin(\frac{1}{n-1}\pi), \ldots, \sin(\frac{n-2}{n-1}\pi), 0)$ on its diagonal, while Δ_{-1} is the anti-diagonal matrix based on the same vector. For both (4.5) and (4.6) we take $n = 10000$ and output matrix $C = (1, 1, \ldots, 1)/\|(1, 1 \ldots, 1)\|_2$, i.e., the output is the average temperature of the rod. We further assume $B = C^{\mathrm{T}}$.

In Figure 4.2 we display the normalized error (4.3) on the Lyapunov matrix for the interval $[0,\ t_{\max}] = [0,\ 3]$, as well as the normalized error on the associated $\mathcal{H}_2$ norm approximation, as a function of k. To shed light on the computation time, for system (4.6) and $k = 100$ the computation time for the delay Lyapunov matrix, respectively, $\mathcal{H}_2$ norm,[2] was 42 seconds, respectively, 4.8 seconds, using MATLAB R2017b on a laptop with an Intel Core i7 2.80 GHz processor and 16 GB RAM.

Let us now comment on the convergence behavior shown in Figures 4.1 and 4.3. The experiments carried out for $kr >> n$, which is natural if n is small as for the first presented example, indicate an asymptotic rate of $\mathcal{O}(k^3)$, respectively, $\mathcal{O}(k^2)$, for the $\mathcal{H}_2$ norm, respectively the delay Lyapunov matrix approximation. These rates are similar to those obtained by the spectral discretization in section 2 as a function on N, hence the projection step does not result in a slowing down of the asymptotic convergence rate, although it is advantageous in terms of computational complexity. Recall the arguments in section 2.2, where the rates are, among others, related to the lack of smoothness of $P(\cdot)$). Some intuition behind this observation is given by Theorems 2.3 and 3.1: by construction precisely the matching moment between Υ and Υ_N carry over to the projected transfer function $\boldsymbol{\Upsilon}_k$. In experiments with very large n, we have $kr << n$ for a realistic range for k values as in the second and third examples, and the observed decay rate is slower, which is illustrated by a comparison between Figures 4.2 and 4.1. A possible explanation is that unlike the previous case a low-rank approximation of Lyapunov matrix $P(t) \in \mathbb{R}^{n\times n}$ is enforced by the construction.

Inherent to the projection approach, the efficiency of the computational approach depends on whether or not accurate low-rank approximations exist, whose determining factors are not well understood, and the projection into system matrix $\mathbf{G}_{2k}$ must be

[2] As can be seen from (3.18) only $\mathbf{V}_k$ needs to be available to evaluate $\mathbf{P}_k(t)$ at $t = 0$.

stability preserving. The latter is the case for most problems and it was an important consideration in the methodological choices, but not always—a counterexample is the second example in [13] for $n = 1023$, where spurious roots are observed in the right half plane. It is not necessarily a strong limitation for the $\mathcal{H}_2$ norm computation, since the $\mathcal{L}_2$ norm of the low-order, projected transfer function $\mathbf{\Gamma}_k$ can still be computed using techniques different from solving the Lyapunov equation directly. All these issues are subject for further investigation.

5. Concluding remarks. A novel algorithm for computing delay Lyapunov matrices and $\mathcal{H}_2$ norms has been presented. It is the first algorithm which is generally applicable to linear time-delay systems with multiple delays and which has at the same time favorable scaling properties with respect to dimension n. In this respect examples with $n = 10000$ in section 4 indicate the potential of the approach. The algorithm is dynamic in nature, in the sense that the computations can be resumed if the accuracy is judged insufficient. It results in approximations of the delay Lyapunov matrix in an explicit form given by (3.18).

Computing delay Lyapunov matrices induces a lot of challenges and complications compared to solving classical Lyapunov matrix equations, since this requires making the leap from an algebraic equation to matrix valued boundary problem (1.4) with a nonsmooth solution. At the same time the research is still in an initial phase. To the best of our knowledge only two, fundamentally different, methods are available for large-scale problems: the presented one and the one of [13]. Therefore we hope that the methodology, results, and observations trigger further research on the topic.

Finally we come back to the assumption of exponential stability of (1.6). It implies that computing the delay Lyapunov matrix with the presented method is not useful in the context of verifying recently proposed stability conditions, expressed in terms of the solution of boundary valued problem (1.4); see, e.g., [5] and the references therein. Yet, the overall algorithm starts with iterations of Algorithm 1, which corresponds to the infinite Arnoldi algorithm [11] for eigenvalue computations and which does not require an exponentially stable system. Consequently, from the output of the first step, more precisely from the spectrum of $\mathbf{G}_{2k}$, we directly obtain a certificate whether or not the system is exponentially stable.

Appendix A. Stopping criteria and computational complexity. If the number of iterations k in Algorithm 2 is not chosen beforehand, the most reliable stopping criterion consists of testing the residual for boundary value problem (1.4) at a set of time instants in the interval under consideration. Substituting (3.18) in (1.4) and letting the columns of $\mathcal{W}_k$ be an orthogonal basis for the column space of $[\mathbf{L}_k \; A_0\mathbf{L}_k \; \cdots \; A_m\mathbf{L}_k]$, every term in the equations has its column, respectively, row range, contained in those of $\mathcal{W}_k$, respectively, $\mathcal{W}_k^{\mathrm{T}}$. As a consequence the Euclidean norm of the residual at a given time instant can be expressed in terms of the residual for a boundary value problem where the size of the matrices is determined by the rank of $\mathcal{W}_k$.

The construction of matrix $\mathcal{W}_k$, however, introduces a significant additional computational cost. To our experience a good indicator of convergence consists of determining the residual for Lyapunov equation (2.26). Recall that $G_N Q_N$ is approximated by

$$G_N V_{N,k} \mathbf{Q}_k V_{N,k}^{\mathrm{T}} = V_{N,k+1} \underline{\mathcal{H}}_k \, [\mathbf{Q}_k \; 0] \, V_{N,k+1}^{\mathrm{T}}.$$

At the same time we have

$$H_N = V_{N,k}V_{N,k}^{\mathrm{T}}H_N = V_{N,k}\mathbf{H}_k = V_{N,k+1}\begin{bmatrix}\mathbf{H}_k\\0\end{bmatrix}.$$

Since the columns of $V_{N,k+1}$ are orthogonal, the residual of (2.26), $R_{N,k}$ satisfies

$$\|R_{N,k}\|_2 = \left\|\underline{\mathcal{H}}_k\,[\mathbf{Q}_k\ 0] + \begin{bmatrix}\mathbf{Q}_k^{\mathrm{T}}\\0\end{bmatrix}\underline{\mathcal{H}}_k^{\mathrm{T}} + \begin{bmatrix}\mathbf{H}_k\\0\end{bmatrix}\begin{bmatrix}\mathbf{H}_k^{\mathrm{T}} & 0\end{bmatrix}\right\|_2. \tag{A.1}$$

Note that the residual norm can be expressed in terms of projected matrices and is independent of N.

As far as the computational complexity is concerned, the core of Algorithm 2 consists of doing $2k$ iterations of Algorithm 1. Expressed in terms of operations on *vectors of length* n, the computational complexity is as follows:

number of backward solves:	$2rk$,
number of matrix vector products:	$O(rk^2)$,
number of scalar products (orthogonalization):	$O(r^2k^3)$.

It is important to point out that all backward solves are with the same matrix (R_0), inherent to an Arnoldi type algorithm. Hence, the first step in our implementation consists of computing a (sparse) LU factorization of matrix $R_0 = \sum_{i=0}^{m} A_i$. For the remaining steps of Algorithm 2, the dominant cost in most cases consists of solving Lyapunov equation (3.12) for $\mathbf{Q}_k$, whose complexity is described by $\mathcal{O}(r^3k^3)$ operations for the adopted Bartels–Stewart algorithm. In addition, our implementation fully exploits the property that, due to the special structure of G_k and the starting vector of the Arnoldi iteration, $\mathbf{V}_k$ can be represented in the form

$$\mathbf{V}_k = (I_k \otimes W_k)\begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,k}\\ 0 & v_{2,2} & \cdots & v_{2,k}\\ \vdots & \ddots & \ddots & \vdots\\ 0 & \cdots & 0 & v_{k,k}\end{bmatrix}, \tag{A.2}$$

where both factors are orthogonal matrices, matrix W_k has dimensions $n \times s$ with $s \le kr$, and $v_{i,j} \in \mathbb{R}^{s\times r}$, $i,j = 1,\dots,k$. Furthermore, both factors can be dynamically constructed. These properties are fundamental in the so-called tensor infinite Arnoldi method and CORK framework (compact rational Krylov algorithms) for nonlinear eigenvalue problems [27, 12], on their turn generalizing [1] for quadratic eigenvalue problems. We refer to these references for more details on representation (A.2). Obviously, for large n its use leads to a significant reduction in the memory requirements, but it is also beneficial in terms of computational complexity, as argued in [12].

Acknowledgment. The first author thanks V. L. Kharitonov for an invitation to give a talk in a session on Lyapunov matrices at the 14th IFAC Workshop on Time-Delay Systems, which was the starting point of this work.

REFERENCES

[1] Z. Bai and Y. Su, *SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 640–659.

[2] D. Breda, S. Maset, and R. Vermiglio, *Pseudospectral differencing methods for characteristic roots of delay differential equations*, SIAM J. Sci. Comput., 27 (2005), pp. 482–495.

[3] D. Breda, S. Maset, and R. Vermiglio, *Numerical approximation of characteristic values of partial retarded functional differential equations*, Numer. Math., 113 (2009), pp. 181–242.

[4] T. Breiten, *Structure-preserving model reduction for integro-differential equations*, SIAM J. Control Optim., 54 (2015), pp. 2992–3015.
[5] S. Cuvas and S. Mondié, *Necessary stability conditions for delay systems with multiple pointwise and distributed delays*, IEEE Trans. Automat. Control, 61 (2016), pp. 1987–1994.
[6] V. Druskin, L. Knizhnerman, and V. Simoncini, *Analysis of the Rational Krylov Subspace and ADI methods for solving the Lyapunov equation*, SIAM J. Numer. Anal., 49 (2011), pp. 1875–1898.
[7] M. A. Gomez, A. Egorov, S. Mondié, and W. Michiels, *Optimization of the $\mathcal{H}_2$ norm for time-delay systems, with application to control design and model approximation*, IEEE Trans. Automat. Control, 64 (2019), pp. 804–811.
[8] J. K. Hale and S. M. Verduyn Lunel, *Introduction to Functional Differential Equations*, Appl. Math. Sci. 99, Springer, New York, 1993.
[9] E. Huesca, S. Mondié, and O. Santos, *Polynomial approximations of the Lyapunov matrix of a class of time delay systems*, in Proceedings of the 8th IFAC Workshop on Time-Delay Systems, 2009, pp. 261–266.
[10] E. Jarlebring, T. Damm, and W. Michiels, *Model reduction of time-delay systems using position balancing and delay lyapunov equations*, Math. Control Signals Systems, 25 (2013), pp. 147–166.
[11] E. Jarlebring, K. Meerbergen, and W. Michiels, *A Krylov method for the delay eigenvalue problem*, SIAM J. Sci. Comput., 32 (2010), pp. 3278–3300.
[12] E. Jarlebring, G. Mele, and O. Runborg, *The waveguide eigenvalue problem and the tensor infinite Arnoldi method*, SIAM J. Sci. Comput., 39 (2017), pp. A1062–A1088.
[13] E. Jarlebring and F. Poloni, *Iterative methods for the delay Lyapunov equation with T-sylvester preconditioning*, Appl. Numer. Math., 135 (2018), pp. 173–185.
[14] E. Jarlebring, J. Vanbiervliet, and W. Michiels, *Characterizing and computing the $\mathcal{H}_2$ norm of time-delay systems by solving the delay Lyapunov equation*, IEEE Trans. Automat. Control, 56 (2011), pp. 814–825.
[15] V. L. Kharitonov, *Time-delay systems. Lyapunov Functionals and Matrices*, Birkhäuser, Basel, 2013.
[16] V. L. Kharitonov and E. Plischke, *Lyapunov matrices for time-delay systems*, Systems Control Lett., 55 (2006), pp. 697–706.
[17] M. Krstić, *Delay Compensation for Nonlinear, Adaptive and PDE Systems*, Birkhäuser, Basel, 2007.
[18] W. Michiels, *Spectrum based stability analysis and stabilization of systems described by delay differential algebraic equations*, IET Control Theory Appl., 5 (2011), pp. 1829–1842.
[19] W. Michiels, E. Jarlebring, and K. Meerbergen, *Krylov based model order reduction of time-delay systems*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1399–1421.
[20] W. Michiels and S. Niculescu, *Stability, Control, and Computation for Time-Delay Systems: An Eigenvalue Based Approach*, 2nd ed., SIAM, Philadelphia, 2014.
[21] S.-I. Niculescu, *Delay Effects on Stability. A Robust Control Approach*, Lecture Notes in Control and Inform. Sci. 269, Springer, New York, 2001.
[22] J. Peeters and W. Michiels, *Computing the $\mathcal{H}_2$ norm of large-scale time-delay systems*, in Proceedings of the IFAC Joint Conference, Grenoble, France, 2012, pp. 1–6.
[23] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288.
[24] V. Simoncini, *Computational methods for linear matrix equations*, SIAM Rev., 58 (2016), pp. 377–441.
[25] L. N. Trefethen, *Spectral Methods in MATLAB*, Software Environments Tools 10, SIAM, Philadelphia, 2000.
[26] L. N. Trefethen, *Approximation Theory and Approximation Practice*, SIAM, Philadelphia, 2013.
[27] R. Van Beeumen, K. Meerbergen, and W. Michiels, *Compact rational Krylov methods for nonlinear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 820–838.
[28] J. Vanbiervliet, W. Michiels, and E. Jarlebring, *Using spectral discretization for the optimal $\mathcal{H}_2$ design of time-delay systems*, Internat. J. Control, 84 (2011), pp. 228–241.
[29] T. Vyhlídal and P. Zítek, *Mapping based algorithm for large-scale computation of quasi-polynomial zeros*, IEEE Trans. Automat. Control, 54 (2009), pp. 171–177.
[30] Z. Wu and W. Michiels, *Reliably computing all characteristic roots of delay differential equations in a given right half plane*, J. Comput. Appl. Math., 236 (2012), pp. 2499–2514.