Rational interpolation, minimal realization and model reduction

Falk Ebert* Tatjana Stykel†

Abstract

In this paper we consider the rational interpolation problem consisting in finding a rational matrix-valued function that interpolates a given set of parameters. We briefly describe two different numerical methods for solving this problem. These are the vector fitting and the frequency domain subspace identification method. Several numerical examples are given that compare the properties of these methods. Furthermore, we discuss the computation of a (minimal) state space realization of a rational function. Model order reduction methods such as modal approximation and balanced truncation are also presented. These methods can be used to compute a reduced-order approximation of the realized dynamical system.

1 Introduction

Consider the rational interpolation problem: *Given a set of parameters*

$$\begin{array}{c|c|c|c} i\omega_1 & i\omega_2 & \cdots & i\omega_q \\ \hline G_1 & G_2 & \cdots & G_q \end{array} \tag{1.1}$$

where $\omega_j \in \mathbb{R}$, $\omega_j \neq \omega_k$ *for* $j \neq k$ *and* $G_j \in \mathbb{C}^{p,m}$, *find matrices* $E, A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$, $C \in \mathbb{R}^{p,n}$ *and* $D \in \mathbb{R}^{p,m}$ *such that the rational matrix-valued function*

$$\boldsymbol{G}(s) = C(sE - A)^{-1}B + D \tag{1.2}$$

interpolates these parameters, i.e., $\boldsymbol{G}(i\omega_j) = G_j$ *for* $j = 1, \ldots, q$. Such a rational function $\boldsymbol{G}(s)$ gives an external description of a system to be modelled. In the time domain, this system can be written in the generalized state space (or descriptor) form

$$\begin{array}{rcl} E\dot{x}(t) & = & Ax(t) + Bu(t), \\ y(t) & = & Cx(t) + Du(t), \end{array} \tag{1.3}$$

where $x(t) \in \mathbb{R}^n$ is a state vector, $u(t) \in \mathbb{R}^m$ is an input and $y(t) \in \mathbb{R}^p$ is an output. Then $\boldsymbol{G}(s)$ as in (1.2) is known as the *transfer function* of system (1.3). It describes the input-output relation of (1.3) in the frequency domain.

For any rational matrix-valued function $\boldsymbol{G}(s)$, there exist matrices E, A, B, C and D such that $\boldsymbol{G}(s) = C(sE - A)^{-1}B + D$, see [31]. Such a matrix set $[\,E,\ A,\ B,\ C,\ D\,]$ is called a *realization* of $\boldsymbol{G}(s)$. Note that the realization of $\boldsymbol{G}$ is, in general, not unique. A realization of the smallest possible state space dimension n is called *minimal realization.* One can show, see [27, 31], that the realization $\boldsymbol{G} = [\,E,\ A,\ B,\ C,\ D\,]$ is minimal if and only if

*Institut für Mathematik, MA 4-5, Technische Universität Berlin, Straße des 17. Juni 136, D-10623 Berlin, Germany, e-mail: ebert@math.tu-berlin.de. Supported by the DFG Reseach Center Matheon.

†Institut für Mathematik, MA 4-5, Technische Universität Berlin, Straße des 17. Juni 136, D-10623 Berlin, Germany, e-mail: stykel@math.tu-berlin.de. Supported by the DFG Reseach Center Matheon.

- system (1.3) is completely controllable, i.e., $\operatorname{rank}[\,\lambda E - A,\, B\,] = n$ for all $\lambda \in \mathbb{C}$ and $\operatorname{rank}[\,E,\, B\,] = n$;
- system (1.3) is completely observable, i.e., $\operatorname{rank}[\,\lambda E^T - A^T,\, C^T\,] = n$ for all $\lambda \in \mathbb{C}$ and $\operatorname{rank}[\,E^T,\, C^T\,] = n$;
- the nilpotent block in E in the Weierstrass canonical form [8] of the pencil $\lambda E - A$ does not have any 1×1 Jordan blocks, i.e., $A \ker(E) \subseteq \operatorname{im}(E)$.

We are interested in parameterizing all solutions of the rational interpolation problem such that the state space dimension n of system (1.3) is as small as possible and the finite eigenvalues of the pencil $\lambda E - A$ belong to the open left half-plane. The latter is equivalent to the asymptotic stability of the descriptor system (1.3), e.g., [5].

Notation. We will denote by $\mathbb{R}^{n,m}$ and $\mathbb{C}^{n,m}$ the space of $n \times m$ real and complex matrices, respectively. The real and imaginary parts of $s \in \mathbb{C}$ are denoted by $\Re e(s)$ and $\Im m(s)$, respectively, and $i = \sqrt{-1}$. The open left half-plane is denoted by $\mathbb{C}^- = \{\, s \in \mathbb{C} \;:\; \Re e(s) < 0\,\}$. The matrices A^T and A^H stand, respectively, for the transpose and the complex conjugate transpose of $A \in \mathbb{C}^{n,m}$. The matrix A^+ denotes the Moore-Penrose inverse of $A \in \mathbb{C}^{n,m}$. An identity matrix of order n is denoted by I_n or simply by I. We denote by e_n a vector of all ones of length n. The vector formed by stacking the columns of the matrix A is denoted by $\operatorname{vec}(A)$, and $A \otimes B$ denotes the Kronecker product of the matrices A and B. The rank and the image of $A \in \mathbb{C}^{n,m}$ are denoted by $\operatorname{rank}(A)$ and $\operatorname{im}(A)$, respectively.

2 Rational approximation

A survey on rational interpolation can be found in [1, 4]. Here we briefly consider only two methods that can be used for solving the rational interpolation problem. These are the *vector fitting technique* [12, 13, 14] and the *frequency domain subspace identification method* [6, 30].

2.1 Vector fitting

The fitting technique [12, 13, 14] aims at interpolating the parameters (1.1) by a rational function $\boldsymbol{G}(s)$ of the form

$$\boldsymbol{G}(s) = \sum_{k=1}^{d} \frac{R_k}{s - a_k} + R_0 + sR_{-1}, \tag{2.1}$$

were $a_k \in \mathbb{C}$ and $R_k \in \mathbb{C}^{p,m}$ are *finite poles* and *residues* of $\boldsymbol{G}(s)$, respectively. Starting with an initial guess of the poles a_k, the function $\boldsymbol{G}(s)$ is computed by the iterative relocation of the poles followed by the identification of the residues R_k. The fitting can be done matrix-wise, column-wise and element-wise. In the matrix-wise fitting, all components of $\boldsymbol{G}(s)$ have the same poles. The fitting of columns results in $\boldsymbol{G}(s)$ whose components in one column have a common set of poles. In the element-wise fitting, the different sets of poles can be obtained for each single entry of $\boldsymbol{G}(s)$.

Next, we present the vector fitting algorithm for computing a rational vector-valued function $\boldsymbol{G}(s) \in \mathbb{C}^p$ as in (2.1) that approximately interpolates the given samples $G_j \in \mathbb{C}^p$ at the frequencies $i\omega_j$, i.e., $\boldsymbol{G}(i\omega_j) \approx G_j$ for $j = 1, \ldots, q$, see [14] for details.

Algorithm 2.1. Vector fitting.
INPUT: $\omega_1, \ldots, \omega_q \in \mathbb{R}$, $G_1, \ldots, G_q \in \mathbb{C}^p$ *and an initial guess of poles* $a_1, \ldots, a_d \in \mathbb{C}^-$.
OUTPUT: *the identified poles* $a_1, \ldots, a_d \in \mathbb{C}$ *and residues* $R_{-1}, R_0, R_1, \ldots, R_d \in \mathbb{C}^p$.

1. *FOR* $l = 1, 2, \ldots$

   (a) *Compute the least squares solution of the linear system*

   $$M\rho_l = g, \tag{2.2}$$

   where $\rho_l = [r_1^T, \ldots, r_d^T, \tilde{r}_1, \ldots, \tilde{r}_d, r_0^T, r_{-1}^T]^T \in \mathbb{C}^{d(p+1)+2p}$, $g = [G_1^T, \ldots, G_q^T]^T \in \mathbb{C}^{qp}$ *and the j-th block row of* $M = [M_j]_{j=1}^q$ *is given by*

   $$M_j = \left[\frac{I_p}{s_j - a_1}, \ldots, \frac{I_p}{s_j - a_d}, \frac{-G_j}{s_j - a_1}, \ldots, \frac{-G_j}{s_j - a_d}, I_p, s_j I_p \right] \in \mathbb{C}^{p, d(p+1)+2p}.$$

   (b) *Compute the eigenvalues* $\lambda_1, \ldots, \lambda_d$ *of the matrix* $\mathrm{diag}(a_1, \ldots a_d) - e_d[\tilde{r}_1, \ldots, \tilde{r}_d]$ *and update* $a_1 \leftarrow \lambda_1, \ldots, a_d \leftarrow \lambda_d$.

   END FOR

2. *Compute the least squares solution of the linear system*

   $$\hat{M}\rho = g, \tag{2.3}$$

   where $\rho = [R_1^T, \ldots, R_d^T R_0^T, R_{-1}^T]^T$ *and the j-th block row of* $\hat{M} = [\hat{M}_j]_{j=1}^q$ *has the form*

   $$\hat{M}_j = \left[\frac{I_p}{s_j - a_1}, \ldots, \frac{I_p}{s_j - a_d}, I_p, s_j I_p \right].$$

Algorithm 2.1 usually yields complex residues R_k. To ensure that the complex poles and the corresponding residues appear in complex conjugate pairs $a_k = \alpha_k + i\beta_k$, $a_{k+1} = \bar{a}_k = \alpha_k - i\beta_k$ and $r_k = \delta_k + i\gamma_k$, $r_{k+1} = \bar{r}_k = \delta_k - i\gamma_k$, we replace the elements $I_p/(s_j - a_k)$ and $I_p/(s_j - a_{k+1})$ of the matrices M and $\hat{M}$ corresponding to the complex conjugate poles by

$$\frac{I_p}{s_j - a_k} + \frac{I_p}{s_j - \bar{a}_k} \qquad \text{and} \qquad \frac{iI_p}{s_j - a_k} + \frac{iI_p}{s_j - \bar{a}_k},$$

respectively, and also substitute the subvectors r_k,r_{k+1}, $\tilde{r}_k$, $\tilde{r}_{k+1}$ in ρ_l and R_k, R_{k+1} in and ρ with $\Re e(r_k)$, $\Im m(r_k)$, $\Re e(\tilde{r}_k)$, $\Im m(\tilde{r}_k)$ and $\Re e(R_k)$, $\Im m(R_k)$, respectively. In order to guarantee for the solutions of systems (2.2) and (2.3) to be real, we solve the real linear systems

$$\begin{bmatrix} \Re e(M) \\ \Im m(M) \end{bmatrix} \rho_l = \begin{bmatrix} \Re e(g) \\ \Im m(g) \end{bmatrix}, \qquad \begin{bmatrix} \Re e(\hat{M}) \\ \Im m(\hat{M}) \end{bmatrix} \rho = \begin{bmatrix} \Re e(g) \\ \Im m(g) \end{bmatrix}.$$

As numerical experiments show, the approximation properties of the fitted rational function $\boldsymbol{G}(s)$ depend strongly on the choice of starting poles. The optimal choice of these poles remains an open problem, see [14] for some heuristics. Furthermore, so far, there is no convergence analysis for the pole relocation. Efficient algorithms for computing the eigenvalues of the matrix $\Lambda = \mathrm{diag}(a_1, \ldots a_d) - e_d[\tilde{r}_1, \ldots, \tilde{r}_d]$ should be used, exploiting the diagonal plus rank-one structure. Note that even all a_k lie in the open left half-plane, the eigenvalues of Λ do not necessarily have negative real part. This may result in a rational function $\boldsymbol{G}(s)$ with poles in the right half-plane. To avoid the instability of $\boldsymbol{G}(s)$, it was proposed in [14] to remove the unstable poles or to flip them into the left half-plane. Here further investigations are required.

Matrix-wise fitting (MF)

Matrix-wise fitting is performed by stacking the columns of the matrix-valued function $\boldsymbol{G}(s) \in \mathbb{C}^{p,m}$ in a column vector $\mathrm{vec}(\boldsymbol{G}(s))$ of length pm and fitting this vector with respect to the samples $\mathrm{vec}(G_j) \in \mathbb{C}^{pm}$. The obtained residues $[\, r_{k1}^T, \ldots, r_{km}^T \,]^T$ are then partitioned into m vectors of length p and the required rational function $\boldsymbol{G}(s)$ is computed as in (2.1) with $R_k = [\, r_{k1}, \ldots, r_{km} \,]$ for $k = -1, 0, 1, \ldots, d$. This function can be realized, for example, as

$$E = \left[\begin{array}{ccc|cc} I_m & & & & \\ & \ddots & & & \\ & & I_m & & \\ \hline & & & 0 & -I_m \\ & & & 0 & 0 \end{array}\right], \quad A = \left[\begin{array}{ccc|cc} a_1 I_m & & & & \\ & \ddots & & & \\ & & a_d I_m & & \\ \hline & & & I_m & 0 \\ & & & 0 & I_m \end{array}\right], \quad B = \left[\begin{array}{c} I_m \\ \vdots \\ I_m \\ \hline 0 \\ I_m \end{array}\right],$$

$$C = [\, R_1, \ \ldots, \ R_d \mid R_{-1}, \ 0 \,], \qquad D = R_0.$$

Such a realization is minimal if and only if the matrices $R_1, \ldots, R_d$ and R_{-1} have full column rank. If this condition is not satisfied, then the minimal realization of $\boldsymbol{G}(s)$ takes the form

$$E = \left[\begin{array}{ccc|cc} I_{n_1} & & & & \\ & \ddots & & & \\ & & I_{n_d} & & \\ \hline & & & 0 & -I_{n_{-1}} \\ & & & 0 & 0 \end{array}\right], \quad A = \left[\begin{array}{ccc|cc} a_1 I_{n_1} & & & & \\ & \ddots & & & \\ & & a_d I_{n_d} & & \\ \hline & & & I_{n_{-1}} & 0 \\ & & & 0 & I_{n_{-1}} \end{array}\right], \quad B = \left[\begin{array}{c} B_1 \\ \vdots \\ B_d \\ \hline 0 \\ B_{-1} \end{array}\right],$$

$$C = [\, C_1, \ \ldots, \ C_d \mid C_{-1}, \ 0 \,], \qquad D = R_0,$$

where $C_j \in \mathbb{C}^{p,n_j}$ and $B_j \in \mathbb{C}^{n_j,m}$ are full rank factors of R_j satisfying $R_j = C_j B_j$ for $j = -1, 1, \ldots, d$.

Column-wise fitting (CF)

In column-wise fitting, the columns $\boldsymbol{g}_l(s)$ of $\boldsymbol{G}(s)$ are fitted separately with respect to the sample vectors $G_k(:,l)$ for $l = 1, \ldots, m$, where $G_k(:,l)$ denotes the l-th column of the matrix G_k. Let $a_{k,l}$ and $R_{k,l}$ be the determined poles and residues of $\boldsymbol{g}_l(s)$. Then the realization of the resulting rational function $\boldsymbol{G}(s) = [\, \boldsymbol{g}_1(s), \ldots, \boldsymbol{g}_m(s) \,]$ is given by

$$E = \left[\begin{array}{ccc|cc} I_{d_1} & & & & \\ & \ddots & & & \\ & & I_{d_m} & & \\ \hline & & & 0 & -I_{d_0} \\ & & & 0 & 0 \end{array}\right], \quad A = \left[\begin{array}{ccc|cc} A_1 & & & & \\ & \ddots & & & \\ & & A_m & & \\ \hline & & & I_{d_0} & 0 \\ & & & 0 & I_{d_0} \end{array}\right], \quad B = \left[\begin{array}{ccc} e_{d_1} & & \\ & \ddots & \\ & & e_{d_m} \\ \hline & 0 & \\ & B_{-1} & \end{array}\right], \tag{2.4}$$

$$C = [\, C_1, \ \ldots, C_m \mid C_{-1}, \ 0 \,], \qquad D = [\, R_{0,1}, \ \ldots, \ R_{0,m} \,], \tag{2.5}$$

where

$$A_l = \operatorname{diag}(a_{1,l}, \ldots, a_{d_l,l}), \qquad C_l = [\, R_{1,l}, \ldots, R_{d_l,l} \,], \quad l = 1, \ldots, m,$$

and $C_{-1} \in \mathbb{C}^{p,d_0}$, $B_{-1} \in \mathbb{C}^{d_0,m}$ are full rank factors of $[\, R_{-1,1}, \ \ldots, \ R_{-1,m} \,] = C_{-1}B_{-1}$. This realization is completely controllable but not necessary completely observable. The minimal realization can be computed as above by reordering the columns and rows of C and B such that the new blocks C_j and B_j correspond to the same finite eigenvalues of $\lambda E - A$ and then computing the full rank factors of $C_j B_j$.

Element-wise fitting (EF)

In element-wise fitting, every element $\boldsymbol{g}_{jl}(s)$ of $\boldsymbol{G}(s)$ is fitted separately. We obtain

$$\boldsymbol{g}_{jl}(s) = \sum_{k=1}^{d_{jl}} \frac{r_{k,jl}}{s - a_{k,jl}} + r_{0,jl} + s r_{-1,jl},$$

where $r_{k,jl} \neq 0$ for $k = 1, \ldots, d_{jl}$. Then $\boldsymbol{G}(s)$ can be realized as in (2.4) and (2.5), where $d_l = d_{1l} + \ldots + d_{pl}$, $R_{0,l} = [\, r_{0,1l}, \ \ldots, \ r_{0,pl} \,]^T$ and

$$\begin{aligned} A_l &= \operatorname{diag}(A_{1l}, \ldots, A_{pl}), \qquad & A_{jl} &= \operatorname{diag}(a_{1,jl}, \ldots, a_{d_{jl},jl}), \\ C_l &= \operatorname{diag}(C_{1l}, \ldots, C_{pl}), \qquad & C_{jl} &= [\, r_{1,jl}, \ \ldots, r_{d_{jl},jl} \,], \end{aligned}$$

for $l = 1, \ldots, m$, $j = 1, \ldots, p$, and $C_{-1} \in \mathbb{C}^{p,d_0}$ and $B_{-1} \in \mathbb{C}^{d_0,m}$ are full rank factors of $[r_{-1,jl}]_{j,l=1}^{p,m} = C_{-1}B_{-1}$. This realization is minimal and has the state space dimension $n = \sum_{j=1}^{p} \sum_{l=1}^{m} d_{jl} + 2d_0$.

Note that the considered above realizations of $\boldsymbol{G}(s)$ have, in general, complex system matrices. Using the fact that the residues R_k and the poles a_k appear in complex conjugate pairs, one can find a block diagonal transformation matrix S such that the transformed realization $\boldsymbol{G} = [\, T^{-1}ET, \, T^{-1}AT, \, T^{-1}B, \, CT, \, D \,]$ is real and has the form

$$T^{-1}ET = \begin{bmatrix} I & \\ & E_\infty \end{bmatrix}, \; T^{-1}AT = \begin{bmatrix} A_f & \\ & I \end{bmatrix}, \; T^{-1}B = \begin{bmatrix} B_f \\ B_\infty \end{bmatrix}, \; CT = [\, C_f, \ C_\infty \,], \quad (2.6)$$

where A_f is block diagonal with block of size 1×1 and 2×2 corresponding, respectively, to the real and complex eigenvalues of $\lambda E - A$, and B_f has elements 0, 1 and 2, see [14] for details.

Table 1 shows the number and the size of linear systems of the form (2.2) and (2.3) that should be solved in the matrix-wise, column-wise and element-wise fitting. If $m = p$ and the same number d of poles is used in all three types of vector fitting, i.e., $d_l = d$ in the CF and $d_{jl} = d$ in the EF, then the matrix-wise, column-wise and element-wise fitting have, respectively, $\mathcal{O}(qd^2m^6)$, $\mathcal{O}(qd^2m^4)$ and $\mathcal{O}(qd^2m^2)$ complexity and provide, in general, systems of the state space dimension $(d+2)m$, $d(m+2)$ and $d(m^2+2)$, respectively. One can see that for multi-input multi-output systems, the element-wise fitting is less expensive but it results in systems of larger dimension.

	# systems	size
MF	1	$2qpm \times (dpm + 2pm + d)$
CF	m	$2qp \times (d_l p + 2p + d_l)$
EF	pm	$2q \times 2d_{jl} + 2$

Table 1: Comparison of the matrix-wise, column-wise and element-wise fitting.

2.2 Frequency domain subspace identification

The frequency domain subspace identification method [6, 30] is based on the extraction of the state space system (1.3) from the frequency response data-matrices

$$\hat{G} = \begin{bmatrix} G_1 & \cdots & G_q \\ i\omega_1 G_1 & \cdots & i\omega_q G_q \\ \vdots & & \vdots \\ (i\omega_1)^{k-1} G_1 & \cdots & (i\omega_q)^{k-1} G_q \end{bmatrix}, \qquad \hat{H} = \begin{bmatrix} I_m & \cdots & I_m \\ i\omega_1 I_m & \cdots & i\omega_q I_m \\ \vdots & & \vdots \\ (i\omega_1)^{k-1} I_m & \cdots & (i\omega_q)^{k-1} I_m \end{bmatrix},$$

$$G = [\,\Re e(\hat{G}),\ \Im m(\hat{G})\,] \in \mathbb{R}^{pk,2mq}, \qquad H = [\,\Re e(\hat{H}),\ \Im m(\hat{H})\,] \in \mathbb{R}^{mk,2mq}.$$

Assume that $E = I$ in (1.3). Consider the extended observability matrix O_k and the block Toeplitz matrix T_k associated with (1.3) that are given by

$$O_k = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{k-1} \end{bmatrix}, \qquad T_k = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{k-2}B & CA^{k-3}B & \cdots & D \end{bmatrix}$$

for $n < k \le 2q$, respectively. Let $G \backslash H^\perp$ denote the projection of the row space of G onto the kernel of H, i.e., $G \backslash H^\perp = G(I - H^T(HH^T)^{-1}H)$. One can show [30] that

$$\mathrm{im}(G \backslash H^\perp) = \mathrm{im}(O_k), \qquad \mathrm{rank}(G \backslash H^\perp) = n.$$

These relations allow to determine the state space dimension n of (1.3) and the image of O_k. Then the matrices A, B, C and D can be computed from the singular value decomposition of $G \backslash H^\perp$. Note that for large k the matrices G and H become ill-conditioned which may lead to the poor performance of the subspace identification algorithm. To improve the condition numbers of the data-matrices involved, it was proposed in [30] to replace G and H by the matrices

$$G_F = [\,\Re e(\hat{G}_F),\ \Im m(\hat{G}_F)\,], \qquad H_F = [\,\Re e(\hat{H}_F),\ \Im m(\hat{H}_F)\,], \tag{2.7}$$

where

$$\hat{G}_F = \begin{bmatrix} Y_0^{-1/2} R_0 \\ \vdots \\ Y_{k-1}^{-1/2} R_{k-1} \end{bmatrix}, \qquad \hat{H}_F = \begin{bmatrix} Z_0^{-1/2} S_0 \\ \vdots \\ Z_{k-1}^{-1/2} S_{k-1} \end{bmatrix}.$$

Here the matrices R_j, Y_j and S_j, Z_j are determined from the Forsythe recursions

$$\begin{array}{ll} R_0 = [\,G_1,\, \ldots,\, G_q\,], & Y_0 = \mathrm{diag}(\mathrm{diag}(R_0 R_0^H)), \\ R_1 = R_0 D_\omega, & Y_1 = \mathrm{diag}(\mathrm{diag}(R_1 R_1^H)), \\ R_j = R_{j-1} D_\omega + Y_{j-1} Y_{j-2}^{-1} R_{j-2}, & Y_j = \mathrm{diag}(\mathrm{diag}(R_j R_j^H)),\ \ j = 2, \ldots, k-1, \end{array} \tag{2.8}$$

with $D_\omega = \mathrm{diag}(i\omega_1 I_m, \ldots, i\omega_q I_m)$ and

$$\begin{array}{ll} S_0 = [\, I_m, \, \ldots, \, I_m \,], & Z_0 = \mathrm{diag}(\mathrm{diag}(S_0 S_0^H)), \\ S_1 = S_0 D_\omega, & Z_1 = \mathrm{diag}(\mathrm{diag}(S_1 S_1^H)), \\ S_j = S_{j-1} D_\omega + Z_{j-1} Z_{j-2}^{-1} S_{j-2}, & Z_j = \mathrm{diag}(\mathrm{diag}(S_j S_j^H)), \;\; j = 2, \ldots, k-1. \end{array}$$

Using G_F and H_F, the system matrices A, B, C and D can be computed by the following algorithm, see [6, 30] for details.

Algorithm 2.2. *Frequency domain subspace identification method*
INPUT: $\omega_1, \ldots, \omega_q \in \mathbb{R}$, $G_1, \ldots, G_q \in \mathbb{C}^{p,m}$, *the weighting matrices* W_1 *and* W_2.
OUTPUT: $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$, $C \in \mathbb{R}^{p,n}$, $D \in \mathbb{R}^{p,m}$.

1. *Compute the matrices* G_F, H_F *and* $Y_0, \ldots, Y_{k-1}$ *as in* (2.7) *and* (2.8), *respectively.*

2. *Compute the thin singular value decomposition*

$$W_1 G_F (I - H_F^T H_F) = U \Sigma V^T,$$

where U *and* V *have orthonormal columns and* $\Sigma \in \mathbb{R}^{n,n}$ *is nonsingular.*

3. *Compute the matrix* $M = W_1^{-1} U \Sigma^{-1/2}$.

4. *Compute the matrices* C *and* A *as* $C = Y_0^{1/2} M(1:p,:)$ *and*

$$\begin{array}{l} A = \big(D_1 M(p+1:p(k-1),:)\big)^+ \big(M(2p+1:pk,:) - D_2 M(1:p(k-1),:)\big), \\ D_1 = \mathrm{diag}(Y_1^{1/2} Y_2^{-1/2}, \ldots, Y_{k-2}^{1/2} Y_{k-1}^{-1/2}), \\ D_2 = \mathrm{diag}(Y_1 (Y_0 Y_2)^{-1/2}, \ldots, Y_{k-2} (Y_{k-3} Y_{k-1})^{-1/2}). \end{array}$$

5. *Compute the matrices* B *and* D *from the least squares solution of the linear system*

$$W_2 \left[\begin{array}{c} I_m \otimes \Re e(M_1) \\ I_m \otimes \Im m(M_1) \end{array} \right] \left[\begin{array}{c} \mathrm{vec}(B) \\ \mathrm{vec}(D) \end{array} \right] = W_2 \left[\begin{array}{c} \mathrm{vec}(\,\Re e(M_2)\,) \\ \mathrm{vec}(\Im m(M_2)) \end{array} \right],$$

where

$$M_1 = \left[\begin{array}{cc} C(i\omega_1 I - A)^{-1} & I_p \\ \vdots & \vdots \\ C(i\omega_q I - A)^{-1} & I_p \end{array} \right], \qquad M_2 = \left[\begin{array}{c} G_1 \\ \vdots \\ G_q \end{array} \right].$$

3 Model reduction

The model reduction problem for the descriptor system (1.3) consists in the approximation of (1.3) by a reduced-order model

$$\begin{array}{rcl} \tilde{E}\, \dot{\tilde{x}}(t) & = & \tilde{A}\, \tilde{x}(t) + \tilde{B}\, u(t), \\ \tilde{y}(t) & = & \tilde{C}\, \tilde{x}(t) + \tilde{D}\, u(t), \end{array} \qquad (3.1)$$

where $\tilde{E}$, $\tilde{A} \in \mathbb{R}^{\ell,\ell}$, $\tilde{B} \in \mathbb{R}^{\ell,m}$, $\tilde{C} \in \mathbb{R}^{m,\ell}$, $\tilde{D} \in \mathbb{R}^{m,m}$ and $\ell \ll n$. The transfer function of (3.1) is given by $\tilde{G}(s) = \tilde{C}(s\tilde{E} - \tilde{A})^{-1}\tilde{B} + \tilde{D}$. Note that systems (1.3) and (3.1) have the same input $u(t)$. We require for the approximate model (3.1) to preserve important properties of the original system (1.3) like regularity, stability and passivity. It is also desirable that the approximation error measured by $\|\tilde{y}-y\|$ or $\|\tilde{\boldsymbol{G}}-\boldsymbol{G}\|$ is small. Moreover, the computation of the reduced-order system should be numerically reliable and efficient.

There exist many different model reduction approaches for linear time-invariant systems, see [1, 3] for recent books on this topic. Here we consider only two methods which are more appropriate for our structured systems computed by the vector fitting method. These are *modal truncation* and *balanced truncation*.

3.1 Modal truncation

A general idea of modal truncation is the projection of the system (1.3) onto the invariant subspace of the pencil $\lambda E - A$ corresponding to some subset of eigenvalues. Usually, one chooses the non-dominant eigenvalues, i.e., the eigenvalues with smallest real part. Using the special structure of the transfer function $\boldsymbol{G}(s)$ in (2.1) computed by the vector fitting method, we compute the reduced-order system (3.1) by the projection of (1.3) onto the subspace corresponding to the infinite eigenvalues together with the finite eigenvalues a_k of $\lambda E - A$ that satisfy the condition $\|R_k\|/|\Re e(a_k)| > tol$ for small enough tolerance tol. Without loss of generality we may assume that the first ℓ finite eigenvalues of $\lambda E - A$ satisfy this condition. Then the transfer function of the reduced-order system has the form

$$\tilde{\boldsymbol{G}}(s) = \sum_{k=1}^{\ell} \frac{R_k}{s-a_k} + R_0 + sR_{-1},$$

and we obtain the following $\mathbb{H}_\infty$-norm error bound

$$\|\tilde{\boldsymbol{G}}-\boldsymbol{G}\|_{\mathbb{H}_\infty} = \sup_{\omega\in\mathbb{R}} \|\tilde{\boldsymbol{G}}(i\omega)-\boldsymbol{G}(i\omega)\| \le \sum_{k=\ell+1}^{n} \frac{\|R_k\|}{|\Re e(a_k)|} \le (n-\ell)tol,$$

where $\|\cdot\|$ denotes the spectral matrix norm and a_k, $k=\ell+1,\ldots,n$, are the truncated finite eigenvalues of $\lambda E - A$.

3.2 Balanced truncation

Balanced truncation is one of the well studied model reduction approaches proposed first for standard state space systems [7, 9, 21] and then extended to descriptor systems in [20, 25, 28]. An important property of this approach is that the asymptotic stability is preserved in the reduced-order system. Moreover, the existence of an a priori error bound [7, 9] allows an adaptive choice of the state space dimension of the approximate model. A disadvantage of balanced truncation is that (generalized) matrix Lyapunov equations have to be solved. However, recent results on low rank approximations to the solutions of Lyapunov equations [18, 22, 29] make the balanced truncation model reduction approach attractive for large-scale systems.

In balanced truncation model reduction of descriptor systems, a special attention must be paid to the approximation of the systems with an improper transfer function $\boldsymbol{G}(s)$. Such a function can be additively decomposed as

$$\boldsymbol{G}(s) = \boldsymbol{G}_{sp}(s) + \boldsymbol{P}(s), \tag{3.2}$$

where $\boldsymbol{G}_{sp}(s)$ is the strictly proper part with $\lim_{s\to\infty} \boldsymbol{G}_{sp}(s) = 0$ and $\boldsymbol{P}(s)$ is the polynomial part of $\boldsymbol{G}(s)$. Then the reduced-order system can be obtained by the approximation of the strictly proper part $\boldsymbol{G}_{sp}(s)$, while the polynomial part $\boldsymbol{P}(s)$ is kept unmodified. Note that the polynomial part of $\boldsymbol{G}(s)$ corresponds to the constraints in the descriptor system (1.3) that define a manifold in which the solution dynamics take place. Any approximation of this part may lead to physically meaningless results, see [20] for details.

The additive decomposition (3.2) can be performed via the transformation of the system matrices into the following form

$$WET = \begin{bmatrix} E_f & \\ & E_\infty \end{bmatrix}, \; WAT = \begin{bmatrix} A_f & \\ & A_\infty \end{bmatrix}, \; WB = \begin{bmatrix} B_f \\ B_\infty \end{bmatrix}, \; CT = [\, C_f,\, C_\infty \,], \tag{3.3}$$

where W and T are the nonsingular transformation matrices, the pencil $\lambda E_f - A_f$ has the finite eigenvalues and the pencil $\lambda E_\infty - A_\infty$ has the eigenvalue at infinity only. This can be done, for example, via the reduction of the pencil $\lambda E - A$ to the Weierstrass-like form, see [16]. Then the strictly proper and polynomial parts of $\boldsymbol{G}(s)$ have the form $\boldsymbol{G}_{sp}(s) = C_f(sE_f - A_f)^{-1}B_f$ and $\boldsymbol{P}(s) = C_\infty(sE_\infty - A_\infty)^{-1}B_\infty + D$. In this case, the reduced-order system (3.1) can be computed as

$$\tilde{E} = \begin{bmatrix} \tilde{E}_f & \\ & \tilde{E}_\infty \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} \tilde{A}_f & \\ & \tilde{A}_\infty \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} \tilde{B}_f \\ \tilde{B}_\infty \end{bmatrix}, \quad \tilde{C} = [\, \tilde{C}_f,\, \tilde{C}_\infty \,], \quad \tilde{D},$$

where $\tilde{\boldsymbol{G}}_{sp} = [\, \tilde{E}_f,\, \tilde{A}_f,\, \tilde{B}_f,\, \tilde{C}_f,\, 0 \,]$ is an approximation to $\boldsymbol{G}_{sp} = [\, E_f,\, A_f,\, B_f,\, C_f, 0 \,]$ and $\tilde{\boldsymbol{P}} = [\, \tilde{E}_\infty,\, \tilde{A}_\infty,\, \tilde{B}_\infty,\, \tilde{C}_\infty,\, \tilde{D} \,]$ is a minimal realization of $\boldsymbol{P}(s)$.

Note that the system matrices (2.4), (2.5) computed by the vector fitting are already in the form (3.3) with $E_f = I$, (bi)diagonal A_f and minimal $\boldsymbol{P} = [\, E_\infty,\, A_\infty,\, B_\infty,\, C_\infty,\, D \,]$. Therefore, in the following we will consider model reduction of the standard state space system $\boldsymbol{G}_{sp} = [\, I,\, A_f,\, B_f,\, C_f,\, 0 \,]$ only. For sake of simplicity, we will omit the index f.

Assume that the dynamical system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) \end{aligned} \tag{3.4}$$

is asymptotically stable, i.e., all the eigenvalues of A have negative real part. The balanced truncation method for such a system is closely related to the *controllability* and *observability Gramians* $\mathcal{P}$ and $\mathcal{Q}$ that are unique symmetric, positive semidefinite solutions of the Lyapunov equations

$$A\,\mathcal{P} + \mathcal{P}A^T = -BB^T, \tag{3.5}$$

$$A^T\mathcal{Q} + A\,\mathcal{Q} = -C^TC. \tag{3.6}$$

Using these Gramians, we can define the *Hankel singular values* of system (3.4) which characterize the ‘importance’ of state variables in (3.4). The Hankel singular values σ_j of system (3.4) are defined as the square roots of the eigenvalues of the matrix $\mathcal{P}\mathcal{Q}$, i.e., $\sigma_j = \sqrt{\lambda_j(\mathcal{P}\mathcal{Q})}$. We will assume that the Hankel singular values are ordered decreasingly. A reduced-order system can be computed by the truncation of the states corresponding to the small Hankel singular values using the following algorithm, see [17, 21] for details.

Algorithm 3.1. *Square root balanced truncation method.*
INPUT: *System* $\boldsymbol{G}(s) = C(sI - A)^{-1}B$ *with* $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$, $C \in \mathbb{R}^{p,n}$.
OUTPUT: *Reduced-order system* $\tilde{\boldsymbol{G}}(s) = \tilde{C}(sI - \tilde{A})^{-1}\tilde{B}$ *with* $\tilde{A} \in \mathbb{R}^{\ell,\ell}$, $\tilde{B} \in \mathbb{R}^{\ell,m}$, $\tilde{C} \in \mathbb{R}^{p,\ell}$.

1. *Compute the Cholesky factors* R *and* L *of the Gramians* $\mathcal{P} = RR^T$ *and* $\mathcal{Q} = L^TL$, *that satisfy the Lyapunov equations* (3.5) *and* (3.6), *respectively.*

2. *Compute the singular value decomposition*
$$LR = [\,U_1,\, U_2\,] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} [\,V_1,\, V_2\,]^T,$$
where the matrices $[\,U_1,\, U_2\,]$ *and* $[\,V_1,\, V_2\,]$ *are orthogonal,* $\Sigma_1 = \operatorname{diag}(\sigma_1, \ldots, \sigma_\ell)$ *and* $\Sigma_2 = \operatorname{diag}(\sigma_{\ell+1}, \ldots, \sigma_n)$.

3. *Compute the reduced-order system* $\tilde{A} = W^TAT$, $\tilde{B} = W^TB$ *and* $\tilde{C} = CT$ *with the projection matrices* $W = L^TU_1\Sigma_1^{-1/2}$ *and* $T = RV_1\Sigma_1^{-1/2}$.

One can show that the reduced-order system computed by this method is asymptotically stable and the $\mathbb{H}_\infty$-norm error bound
$$\|\widetilde{\boldsymbol{G}} - \boldsymbol{G}\|_{\mathbb{H}_\infty} \leq 2(\sigma_{\ell+1} + \ldots + \sigma_n)$$
holds, where $\sigma_{\ell+1}, \ldots, \sigma_n$ are the truncated Hankel singular values of system (3.4), see [7, 9].

To solve the Lyapunov equations (3.5) and (3.6) for the Cholesky factors without forming the Gramians explicitly, we can use the Hammarling method [15] for problems of moderate size and the ADI or Smith method [19, 26, 32] for large-scale systems. To reduce the computation time we can use the special structure of the matrices A, B and C obtained from the column-wise or element-wise fitting. Recall that the matrices $A = \operatorname{diag}(A_1, \ldots, A_m)$ and $B = \operatorname{diag}(B_1, \ldots, B_m)$ are block diagonal, where $A_j \in \mathbb{R}^{d_j,d_j}$ and $B_j \in \mathbb{R}^{d_j}$. Then the matrix BB^T is also block diagonal, and, hence, the solution of the Lyapunov equation (3.5) and its Cholesky factor have also block diagonal form $\mathcal{P} = \operatorname{diag}(\mathcal{P}_1, \ldots, \mathcal{P}_m)$ and $R = \operatorname{diag}(R_1, \ldots, R_m)$, where R_j is the Cholesky factor of the solution $\mathcal{P}_j = R_jR_j^T$ of the Lyapunov equation
$$A_j\mathcal{P}_j + \mathcal{P}_jA_j^T = -B_jB_j^T. \tag{3.7}$$
In the element-wise fitting, the matrix C has also some special block structure. We can find the block permutation matrix Π such that
$$\Pi A \Pi = \operatorname{diag}(\hat{A}_1, \ldots, \hat{A}_m), \qquad C\Pi = \operatorname{diag}(\hat{C}_1, \ldots, \hat{C}_m)$$
with $\hat{A}_m \in \mathbb{R}^{\hat{d}_j,\hat{d}_j}$, $C_j \in \mathbb{R}^{1,\hat{d}_j}$ and $\hat{d}_j = d_{j1} + \ldots + d_{jm}$. Then the Cholesky factor L of the solution of the Lyapunov equation (3.6) has the form $L = \operatorname{diag}(L_1, \ldots, L_m)\Pi$, where L_j is the Cholesky factor of the solution $\mathcal{Q}_j = L_j^TL_j$ of the Lyapunov equation
$$\hat{A}_j^T\mathcal{Q}_j + \mathcal{Q}_j\hat{A}_j = -\hat{C}_j^T\hat{C}_j. \tag{3.8}$$
Unfortunately, the matrix $LR = \operatorname{diag}(L_1, \ldots, L_m)\,\Pi\operatorname{diag}(R_1, \ldots, R_m)$ is not block diagonal any more, and we have to compute the singular values decomposition of the full matrix LR. Note that the Lyapunov equations (3.7) and (3.8) have the right-hand side of rank 1. It was

observed in [2, 10, 23] that the solution of the Lyapunov equation with low-rank right-hand side can often be approximated by the matrices of low rank. The low-rank Cholesky factors $\tilde{R}_j$ and $\tilde{L}_j$ of the solutions $\mathcal{P}_j \approx \tilde{R}_j \tilde{R}_j^T$ and $\mathcal{Q}_j \approx \tilde{L}_j \tilde{L}_j^T$ of the Lyapunov equations (3.7) and (3.8) can be computed from the Cholesky factors using the rank-revealing QR factorizations

$$R_j^T = Q_{1j} \begin{bmatrix} R_{1j} & R_{2j} \\ 0 & R_{3j} \end{bmatrix} \Pi_1, \qquad L_j^T = Q_{2j} \begin{bmatrix} L_{1j} & L_{2j} \\ 0 & L_{3j} \end{bmatrix} \Pi_2,$$

where Q_{1j} and Q_{2j} are orthogonal matrices, Π_1 and Π_2 are permutation matrices, R_{1j} and L_{1j} are nonsingular, and $\|R_{3j}\|$ and $\|L_{3j}\|$ are small. Then $\tilde{R}_j = \Pi_1[\, R_{1j}^T,\ R_{2j}^T\,]^T$ and $\tilde{L}_j = \Pi_2[\, L_{1j}^T,\ L_{2j}^T\,]^T$. In this case, in the second step of Algorithm 3.1 we have to compute the singular value decomposition of the matrix $\operatorname{diag}(\tilde{L}_1^T, \ldots, \tilde{L}_m^T)\, \mathit{\Pi} \operatorname{diag}(\tilde{R}_1, \ldots, \tilde{R}_m)$ that is smaller than $L^T R$. Note that the low-rank Cholesky factors $\tilde{R}_j$ and $\tilde{L}_j$ can also be computed directly using the low-rank ADI or Smith method [11, 18, 22, 24].

4 Numerical examples

4.1 Comparison of vector fitting and subspace identification

Numerous numerical tests have been performed with the data provided by CST GmbH. The examples were given in the form of complex matrices, representing the transfer function at various frequencies. The usually 1001 samples were taken at frequencies ranging from 0 to 1 GHz minimum and 22 GHz maximum.

We have tested the column-wise and the element-wise vector fitting as well as the frequency domain subspace identification method described above. For the latter, it turned out that computation time was extremely large. Thus, as computation time for subspace identification heavily depends on the number of samples, tests have also been carried out with only one out of 10 samples. This led to greatly reduced time consumption, often comparable to vector fitting computation times.

In order to produce comparable and reproducible results, all tests with the same method were carried out with the same set of input parameters, i.e., the number of starting poles for the vector fitting methods and truncation tolerances for the subspace identification methods. For the vector fitting methods 100 pairs of starting poles a_k have been used. Following the advice given in [14] these were initialized as $a_k = \alpha_k \pm i\beta_k$, with $\alpha_k = -10^{-2}\beta_k$ and β_k logarithmically spaced between f_{min} and $10^3 f_{max}$, where f_{min} and f_{max} were the lowest and the highest frequencies of the sample, respectively. These 100 pairs of poles have been used for both the fitting of each column of the transfer function and the fitting of each single component. Thus, for the element-wise fitting, the resulting system is, in general, larger than for the column-wise fitting. The vector fitting algorithm was used with pole-flipping to enforce stability of the state space model. For comparability reasons, only the matrices A, B, C, D were generated and E was assumed to be the identity.

The subspace identification algorithm has been carried out as described above. Two parameters have been provided. One of them, *order* limits the number of iterations performed to obtain R_j and S_j. The parameter *tol* serves two purposes. First, the assembly of R_j and S_j is stopped as soon as the condition number of H_F becomes greater than $1 + \sqrt{tol}$. Note that in exact arithmetic, the rows of H_F are orthogonal. However, it has been observed that numerically, at some stage, this orthogonality is lost and further computations comprise large

Example	$m = p$	EF	CF	SysId 1 out of 10	SysId full
`3dtransline`	2	32	31	-	-
`branch_line_coupler_improved`	4	161	283	83	5537
`circular_patch_antenna`	1	5	4	34	3250
`coaxdiscontinuity`	2	34	33	-	-
`defected_ground`	2	17	17	38	-
`dr_antenna`	1	9	9	41	4082
`folded_patch`	1	8	8	41	3897
`ic_package_14`	14	1120	2684	-	-
`inductor4`	4	128	253	59	4181
`lowpassfilter`	2	18	17	-	-
`microstrip_coupler`	4	117	181	55	4064
`radial_stub`	2	18	18	-	-
`rj45`	8	1006	3196	337	-
`shaped_end_radiator`	1	9	10	44	3961
`single_line`	2	34	33	48	4428
`two_lines`	4	206	294	82	-

Table 2: Computation time for different methods.

errors. Thus, the iteration for R_j and S_j is stopped as soon as this loss of orthogonality surpasses the given tolerance. Secondly, tol is used for the numerical determination of the rank of the matrix Σ. The singular values with $\sigma_i/\sigma_1 < tol$ are treated as zeros. The weighting matrices W_1 and W_2 were taken as the identity matrices.

In Table 2, the number m of inputs and the computation time for the different system identification approaches are given. A bar '-' indicates that the algorithm did not produce meaningful results. Those breakdowns only occurred for the experimental subspace identification code. The vector fitting methods did not produce any breakdowns. In Table 3, the errors for the different system identification methods are shown. For the vector fitting methods, the RMS error, returned by the vector fitting codes is used. In the element-wise fitting, this error is evaluated more often than for the column-wise fitting. Hence, in the latter case, the error is scaled by the square root of the system size to make it comparable to the element-wise error. For the subspace identification methods, the difference between the measured S-parameters and evaluations of the identified system at the given frequencies is determined and the norms of all these deviations are summed and averaged to give a quantity comparable to the RMS error in the vector fitting approach. All errors depicted in the table are given relative to the maximum absolute entry of all S-parameters for each example.

From Tables 2 and 3, several conclusions can be drawn. Although, the subspace identification method usually yields more accurate results than the vector fitting approach, it cannot be guaranteed to work at all. Except for some few examples, notably `inductor4` and `rj45`, even the test with a reduced number of samples required more computational effort than vector fitting. Subspace identification with the full number of samples was prohibitively slow for all considered examples. Two more drawbacks of subspace identification are that the generated systems lack any structure and they are usually unstable. The systems coming out of vector fitting are sparse and nicely structured, see (2.4), (2.5). While it took some time

Example	EF	CF	SysId 1 out of 10	SysId full
`3dtransline`	9.26E-3	2.02E-2	-	-
`branch_line_coupler_improved`	1.09E-1	1.56E+2	3.13E-3	2.91E-1
`circular_patch_antenna`	1.26E-2	1.26E-2	7.67E-3	3.78E-5
`coaxdiscontinuity`	6.11E-1	2.29E-1	-	-
`defected_ground`	7.75E-2	7.43E-2	8.95E-3	-
`dr_antenna`	7.17E-4	7.17E-4	1.58E-3	1.36E-5
`folded patch`	3.78E-2	3.78E-2	1.64E-4	4.05E-3
`ic_package_14`	6.44E-4	2.14E-2	-	-
`inductor4`	2.56E-1	2.31E+2	1.18E-1	6.68E-2
`lowpassfilter`	5.09E-2	3.34E-2	-	-
`microstrip_coupler`	2.35E-1	9.40E-1	1.73E-3	2.42E-3
`radial_stub`	1.39E-1	3.88E-2	-	-
`rj45`	6.64E-2	5.23E+0	7.15E-2	-
`shaped_end_radiator`	4.94E+1	4.94E+1	3.70E-3	3.47E-3
`single_line`	1.44E-2	1.08E-3	2.81E-3	2.52E-2
`two_lines`	9.13E-5	1.51E-2	1.33E-5	-

Table 3: Relative errors for different methods.

finding suitable starting poles for the vector fitting algorithm, the choice of poles given at the beginning of this section turned out satisfactory. Comparing element-wise and column-wise vector fitting, it turns out that the element-wise approach is usually faster and more accurate than the column-wise approach. The gain in computation time is especially notable for larger dimensions of the S-parameters. For one dimensional fittings, both algorithms are principally identical and produce identical results. It should be noted, that column-wise fitting failed for three examples, i.e., it produced relative errors larger than one. With component-wise fitting, this occurred only once.

4.2 Modal truncation and balanced truncation

In the second set of numerical experiments we have tested the modal and balanced truncation model reduction methods. Here, we present the numerical results for only four examples: `coaxdiscontinuity`, `two_lines`, `largefreqDS` and `smallfreqDS`. The state space systems $\boldsymbol{G} = [\,A,\,B,\,C,\,D\,]$ computed by the vector fitting method have first been approximated by the reduced models $\boldsymbol{G}_{mt} = [\,A_{mt},\,B_{mt},\,C_{mt},\,D\,]$ of order ℓ_m using the modal truncation method. Then, we applied the balanced truncation method to these models and obtained the approximate systems $\boldsymbol{G}_{bt} = [\,A_{bt},\,B_{bt},\,C_{bt},\,D\,]$. In modal truncation, we truncated the terms satisfying the condition $\|R_k\|/\Re e(a_k) \leq tol$ with $tol = \min(10^{-4}, err)$, where

$$err = \left(\sum_{j=1}^{q} \|\boldsymbol{G}(i\omega_j) - G_j\|_F^2 \right)^{1/2} \tag{4.1}$$

is the error after vector fitting and $\|\cdot\|_F$ denotes the Frobenius matrix norm. The order ℓ_b of the approximate systems $\boldsymbol{G}_{bt}$ has been chosen as a largest index of the Hankel singular values σ_j satisfying $\sigma_{\ell_b}/\sigma_1 \leq tol$ with the same tolerance tol as above.

For each example, we present

(a) the absolute error $\|\boldsymbol{G}(i\omega_j)-G_j\|$, $j=1,\ldots,q$, where $\boldsymbol{G}(s)=C(sI-A)^{-1}B+D$ is computed by the element-wise vector fitting method and (ω_j, G_j) are given S-parameters;

(b) the Hankel singular values σ_k of the system $\boldsymbol{G}_{mt}=[\,A_{mt},\,B_{mt},\,C_{mt},\,D\,]$;

(c) the spectral norms $\|\boldsymbol{G}(i\omega)\|$, $\|\boldsymbol{G}_{mt}(i\omega)\|$ and $\|\boldsymbol{G}_{bt}(i\omega)\|$ of the frequency responses

$$\begin{array}{rcl}\boldsymbol{G}(i\omega) & = & C(i\omega I-A)^{-1}B+D,\\ \boldsymbol{G}_{mt}(i\omega) & = & C_{mt}(i\omega I-A_{mt})^{-1}B_{mt}+D,\\ \boldsymbol{G}_{bt}(i\omega) & = & C_{bt}(i\omega I-A_{bt})^{-1}B_{bt}+D\end{array}$$

for the frequency range $\omega\in[\,\omega_{\min},\,\omega_{\max}\,]$;

(d) the absolute errors $\|\boldsymbol{G}_{mt}(i\omega)-\boldsymbol{G}(i\omega)\|$ and $\|\boldsymbol{G}_{bt}(i\omega)-\boldsymbol{G}_{mt}(i\omega)\|$ for the same frequency range and the error bounds

$$\|\boldsymbol{G}_{mt}-\boldsymbol{G}\|_{\mathbb{H}_\infty}\le\zeta_{mt},\qquad\qquad \|\boldsymbol{G}_{bt}-\boldsymbol{G}_{mt}\|_{\mathbb{H}_\infty}\le\zeta_{bt},\tag{4.2}$$

where $\zeta_{mt}=\displaystyle\sum_{k=\ell_m+1}^{n}\|R_k\|/\Re e(a_k)$ and $\zeta_{bt}=2\displaystyle\sum_{k=\ell_b+1}^{\ell_m}\sigma_k$.

We give also the computation time required for the computation of the systems $\boldsymbol{G}$, $\boldsymbol{G}_{mt}$, $\boldsymbol{G}_{bt}$ as well as the values of the vector fitting error err as in (4.1) and the bounds ζ_{mt}, ζ_{bt}.

It should be noted that the reduced-order systems computed by modal truncation have the same sparse structure as the original system, whereas the system matrices provided by balanced truncation are full.

Example 1: `coaxdiscontinuity`
Using the element-wise vector fitting method we obtained a system of order $n = 320$ with $m = 2$ inputs and $p = 2$ outputs. This system has been approximated by the reduced-order models $\boldsymbol{G}_{mt}$ and $\boldsymbol{G}_{bt}$ of order $\ell_m = 72$ and $\ell_b = 34$, respectively.

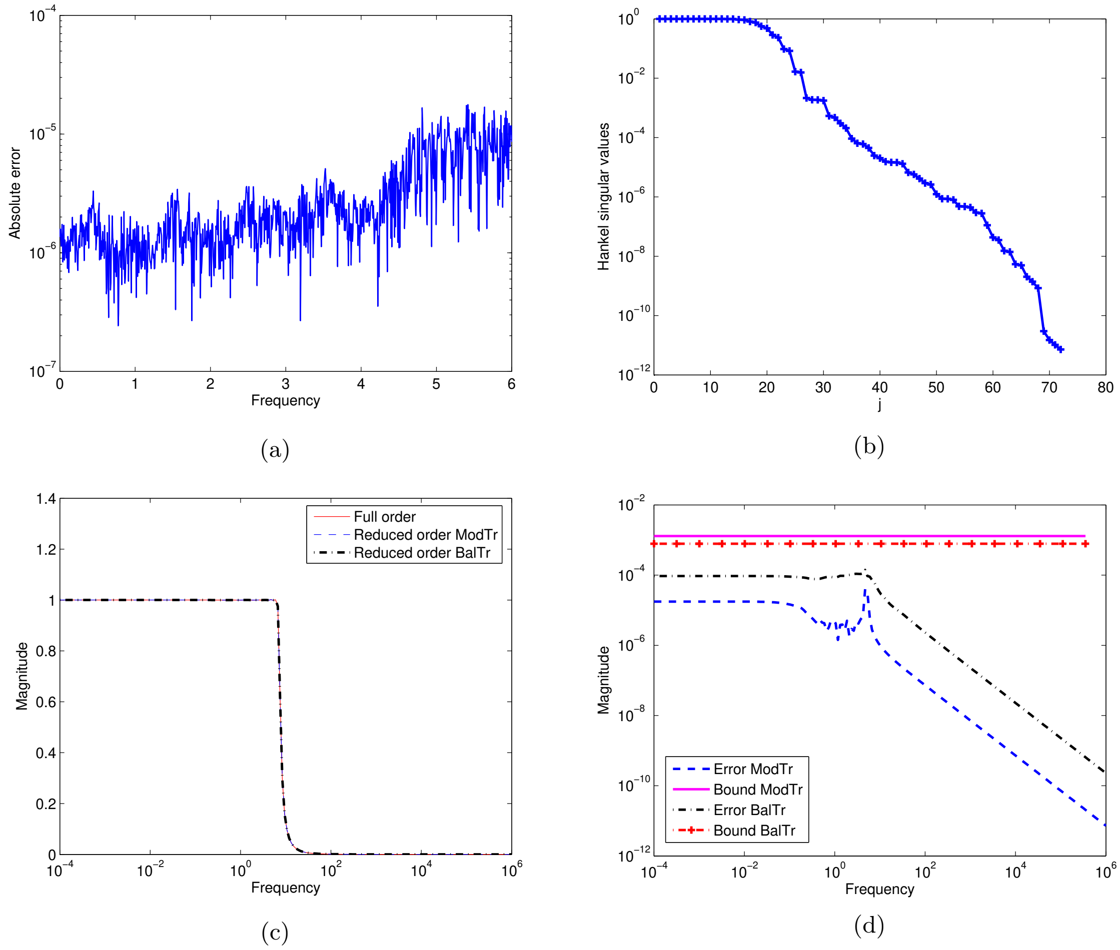

Figure 1: Example 1: (a) the absolute errors of vector fitting; (b) the Hankel singular values; (c) the frequency responses; (d) the absolute errors and the error bounds.

$m = p = 2$	VecFit ($n = 320$)	ModTr ($\ell_m = 72$)	BalTr ($\ell_b = 34$)
CPU time (s)	8.6	0.25	0.27
Error (error bound)	1.886E-04	(1.297E-03)	(7.831E-04)

Table 4: Example 1: computation time, vector fitting error and model reduction error bounds.

Example 2: `two_lines`

Using the column-wise vector fitting method we obtained a system of order $n = 600$ with $m = 4$ inputs and $p = 4$ outputs. This system has been approximated by the reduced-order models $\boldsymbol{G}_{mt}$ and $\boldsymbol{G}_{bt}$ of order $\ell_m = 36$ and $\ell_b = 30$, respectively.

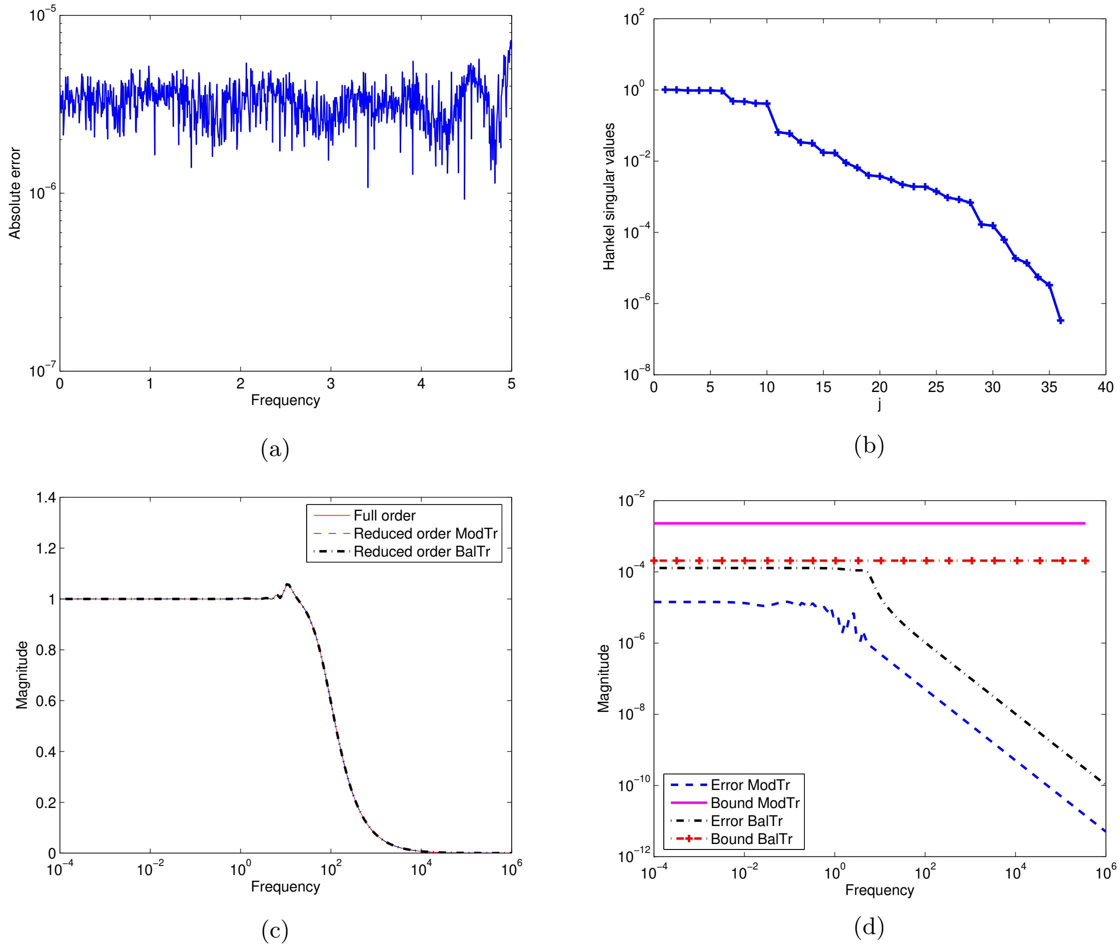

Figure 2: Example 2: (a) the absolute errors of vector fitting; (b) the Hankel singular values; (c) the frequency responses; (d) the absolute errors and the error bounds.

$m = p = 4$	VecFit ($n = 600$)	ModTr ($\ell_m = 36$)	BalTr ($\ell_b = 30$)
CPU time (s)	403	0.28	0.28
Error (error bound)	1.481E-04	(2.326E-03)	(2.076E-04)

Table 5: Example 2: computation time, vector fitting error and model reduction error bounds.

Example 3: `smallfreqDS`

Using the vector fitting method we obtained a single-input single-output system of order $n = 100$. This system has been approximated by the reduced-order models $\boldsymbol{G}_{mt}$ and $\boldsymbol{G}_{bt}$ of order $\ell_m = 28$ and $\ell_b = 12$, respectively.

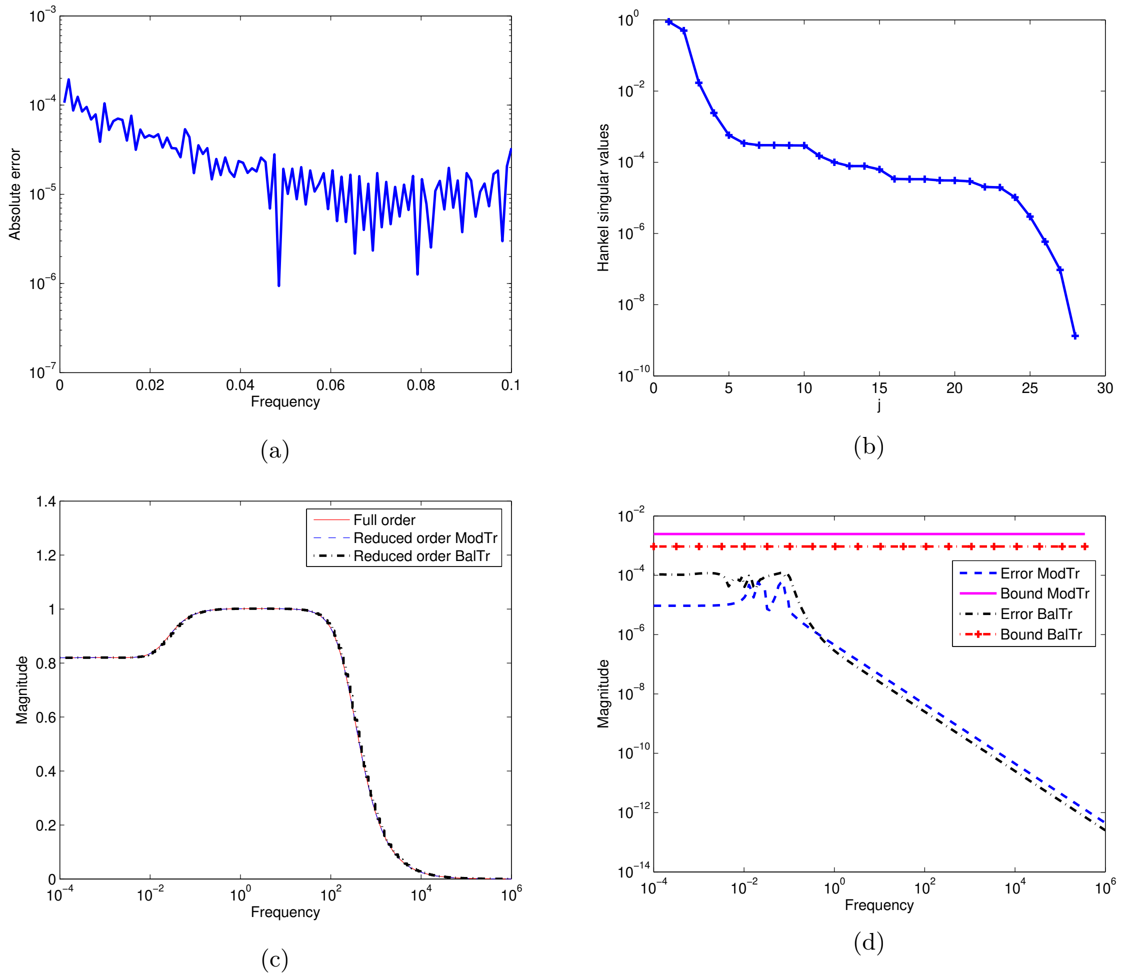

Figure 3: Example 3: (a) the absolute errors of vector fitting; (b) the Hankel singular values; (c) the frequency responses; (d) the absolute errors and the error bounds.

$m = p = 1$	VecFit ($n = 100$)	ModTr ($\ell_m = 28$)	BalTr ($\ell_b = 12$)
CPU time (s)	2.5	0.12	0.24
Error (error bound)	4.214E-04	(2.467E-03)	(9.324E-04)

Table 6: Example 3: computation time, vector fitting error and model reduction error bounds.

Example 4: `largefreqDS`

Using the vector fitting method we obtained a single-input single-output system of order $n = 100$. This system has been approximated by the reduced-order models $\boldsymbol{G}_{mt}$ and $\boldsymbol{G}_{bt}$ of order $\ell_m = 26$ and $\ell_b = 11$, respectively.

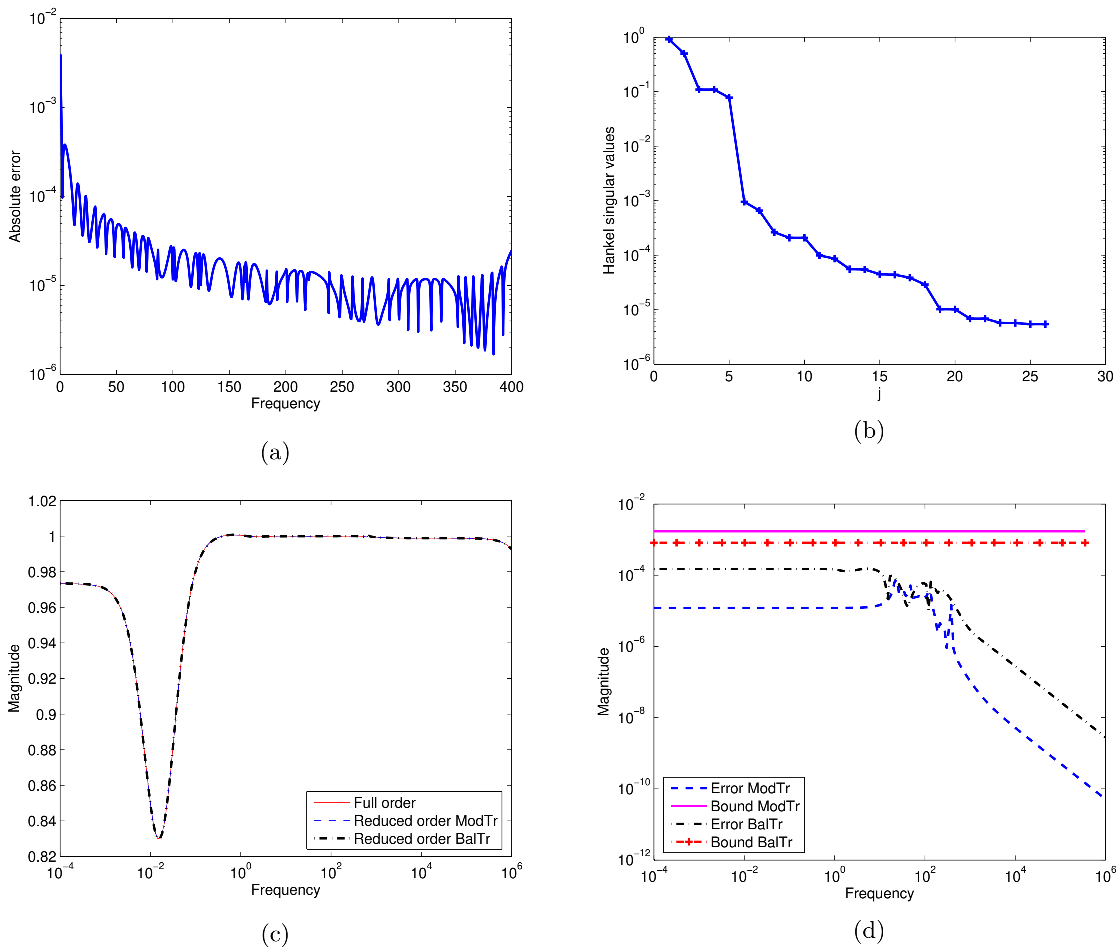

Figure 4: Example 4: (a) the absolute errors of vector fitting; (b) the Hankel singular values; (c) the frequency responses; (d) the absolute errors and the error bounds.

$m = p = 1$	VecFit ($n = 100$)	ModTr ($\ell_m = 26$)	BalTr ($\ell_b = 11$)
CPU time (s)	4.6	0.17	0.26
Error (error bound)	5.596E-03	(1.725E-03)	(8.179E-04)

Table 7: Example 4: computation time, vector fitting error and model reduction error bounds.

Examples 1-4 show that using the vector fitting method we can first compute a large state space system to guarantee small interpolation error. Then we can apply model reduction to compute an approximate system of lower dimension that has the approximation error of the same order as the interpolation error. The existence of the error bounds (4.2) for the modal truncation and the balanced truncation methods allows an adaptive choice of the state space dimension of the reduced model depending on how accurate the approximation is needed.

References

[1] A.C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, Philadelphia, PA, 2005.

[2] A.C. Antoulas, D.C. Sorensen, and Y. Zhou. On the decay rate of the Hankel singular values and related issues. *Systems Control Lett.*, 46(5):323–342, 2002.

[3] P. Benner, V. Mehrmann, and D. Sorensen, editors. *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, Heidelberg, 2005.

[4] A. Bultheel and B. De Moor. Rational approximation in linear systems and control. *J. Comput. App. Math.*, 121:355–378, 2000.

[5] L. Dai. *Singular Control Systems*. Lecture Notes in Control and Information Sciences, 118. Springer-Verlag, Berlin, Heidelberg, 1989.

[6] B. De Moor, P. Van Overschee, and W. Favoreel. Algorithms for subspace state-space system identification: an overview. In B. Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 1 of *Appl. Comput. Control Signals Circuits*, pages 247–311. Birkhäuser Boston, Boston, MA, 1999.

[7] D. Enns. Model reduction with balanced realization: an error bound and a frequency weighted generalization. In *Proceedings of the 23rd IEEE Conference on Decision and Control (Las Vegas, 1984)*, pages 127–132. IEEE, New York, 1984.

[8] F.R. Gantmacher. *Theory of Matrices*. Chelsea Publishing Company, New York, 1959.

[9] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. *Internat. J. Control*, 39(6):1115–1193, 1984.

[10] L. Grasedyck. Existence of a low rank of H-matrix approximation to the solution of the Sylvester equation. *Numer. Linear Algebra Appl.*, 11:371–389, 2004.

[11] S. Gugercin, D.C. Sorensen, and A.C. Antoulas. A modified low-rank Smith method for large-scale Lyapunov equations. *Numerical Algorithms*, 32(1):27–55, 2003.

[12] B. Gustavsen. User's manual for VECTFIT, version 2.1 for MATLAB, 2005. Available from `http://www.energy.sintef.no/Produkt/VECTFIT/index.asp`.

[13] B. Gustavsen. Improving the pole relocating properties of vector fitting. *IEEE Trans. Power Delivery*, 21(3):1587–1592, 2006.

[14] B. Gustavsen and A. Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Trans. Power Delivery*, 14:1052–1061, 1999.

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

[16] B. Kågström and P. Van Dooren. A generalized state-space approach for the additive decomposition of a transfer function. *J. Numer. Linear Algebra Appl.*, 1(2):165–181, 1992.

[17] A.J. Laub, M.T. Heath, C.C. Paige, and R.C. Ward. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *IEEE Trans. Automat. Control*, AC-32(2):115–122, 1987.

[18] J.-R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 24(1):260–280, 2002.

[19] A. Lu and E. Wachspress. Solution of Lyapunov equations by alternating direction implicit iteration. *Comput. Math. Appl.*, 21(9):43–58, 1991.

[20] V. Mehrmann and T. Stykel. Balanced truncation model reduction for large-scale systems in descriptor form. In P. Benner, V. Mehrmann, and D. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*, pages 83–115. Springer-Verlag, Berlin/Heidelberg, 2005.

[21] B.C. Moore. Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, AC-26(1):17–32, 1981.

[22] T. Penzl. A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 1999/2000.

[23] T. Penzl. Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. *Systems Control Lett.*, 40(2):139–144, 2000.

[24] T. Penzl. LYAPACK Users Guide. Preprint SFB393/00-33, Fakultät für Mathematik, Technische Universität Chemnitz, Chemnitz, Germany, 2000. Available from `http://www.tu-chemnitz.de/sfb393/sfb00pr.html`.

[25] K. Perev and B. Shafai. Balanced realization and model reduction of singular systems. *Internat. J. Systems Sci.*, 25(6):1039–1052, 1994.

[26] R.A. Smith. Matrix equation $XA + BX = C$. *SIAM J. Appl. Math.*, 16:198–201, 1968.

[27] V.I. Sokolov. *On minimal realizations of rational functions*. Ph.D. thesis, Fakultät für Mathematik, Technische Universität Chemnitz, Germany, 2006.

[28] T. Stykel. Gramian-based model reduction for descriptor systems. *Math. Control Signals Systems*, 16:297–319, 2004.

[29] T. Stykel. Low rank iterative methods for projected generalized Lyapunov equations. Preprint 198, DFG Research Center Matheon, Technische Universität Berlin, 2004.

[30] P. Van Overschee and B. De Moor. Continuous-time frequency domain subspace system identification. *Signal Process.*, 52(2):179–194, 1996.

[31] G.C. Verghese, B.C. Lévy, and T. Kailath. A generalized state-space for singular systems. *IEEE Trans. Automat. Control*, AC-26(4):811–831, 1981.

[32] E. Wachspress. Iterative solution of the Lyapunov matrix equation. *Appl. Math. Lett.*, 1:87–90, 1988.