A program for solving the L_2 reduced-order model problem with fixed denominator degree

W. Krajewski*

Systems Research Institute, Polish Academy of Sciences, ul. Newelska 6, 01-447 Warsaw, Poland

A. Lepschy, M. Redivo-Zaglia** and U. Viaro

Department of Electronics and Informatics, University of Padova, via Gradenigo 6/A, 35131 Padova, Italy

Received 17 May 1993; revised 22 February 1995
Communicated by C. Brezinski

A set of necessary conditions that must be satisfied by the L_2 optimal rational transfer matrix approximating a given higher-order transfer matrix, is briefly described. On its basis, an efficient iterative numerical algorithm has been obtained and implemented using standard MATLAB functions. The purpose of this contribution is to make the related computer program available and to illustrate some significant applications.

Keywords: Linear dynamic systems, rational approximation, L_2 norm.

AMS subject classification: 41A20.

1. Introduction

The model reduction problem consists in the approximation of a high-order linear system by a lower-order model according to a suitable criterion. Its interest is motivated by the need for time-efficient numerical simulation and control system design that is easier to implement with reference to simplified controlled plant models.

Quite different approaches to model reduction have been proposed in the literature; some of them are based on optimality criteria, some others on system-theoretic arguments, like the popular balancing method of Moore [14]; according to this method, the subsystems which contribute little to the impulse response of the overall system are identified and eliminated. Although for many problems the "weak subsystem" hypothesis leads to a nearly optimal (in the L_2 sense)

* Work partially supported by *Consorzio Padova Ricerche* through the IRI programme "Consortia Città Ricerche – Central European Initiative".

** e-mail: elen@elett1.dei.unipd.it

© J.C. Baltzer AG, Science Publishers

reduced-order model, there are cases in which the balancing method is much worse with respect to the least-squares criterion than the quadratically optimal reduced-order model [8]. These reasons motivate the enduring interest for optimality-based approaches and the search for simple algorithms to achieve the optimum, which is testified by many recent papers [3–5,10,20,24].

Most existing algorithms, however, are computationally demanding and their convergence is seldom guaranteed. The interesting and informative paper [20], where a convergence analysis is developed, is devoted to single-input single-output (SISO) continuous-time systems. The multi-input multi-output (MIMO) case was first considered by Wilson [22,23] and Mishra and Wilson [13], who proposed some algorithms involving the iterative solution of two algebraic matrix Lyapunov equations, whereas in [8] and [24] a projection approach using a rather large number of variables is adopted.

In this paper, we present an efficient alternative algorithm which is based on a re-formation of the first-order necessary conditions of optimality in terms of interpolation constraints and does not require gradient computations. In fact, it has long been known [1,4,12,21] that in the SISO case the best approximating function must satisfy suitable interpolation conditions; similar conditions, however, hold in the MIMO case too [10]. These can be expressed in a compact form which is suggestive of an iterative numerical procedure consisting in the solution of a sequence of linear sets of equations. A similar approach has also been considered by Rosencher [17], Ruckebush [18] and Olivi and Steer [16]. In fact, though this is not immediately apparent, the iteration step of the present method in the scalar case corresponds to the one of Rosencher's algorithm for discrete-time systems (under the assumption that the iterates are stable, a condition which is not required by the algorithm illustrated in the following sections). Rosencher's heuristics has been generalized to the matrix case with bounded McMillan degree in [16] in the generic case of cyclic approximants. Here, instead, we refer to the case in which the degree of the least common denominator (l.c.d.) ρ of the reduced model has been fixed and to the necessary conditions of optimality first derived in [10]; let us recall that the related McMillan degree is included between ρ and $\rho \cdot \min(m_o, m_i)$, where m_o and m_i are the number of outputs and inputs, respectively, and is generally equal to $\rho \cdot \min(m_o, m_i)$.

In the following we illustrate the main features of the algorithm and describe the corresponding program that has been implemented using MATLAB functions. Despite the rather cumbersome notation used in section 3 to derive the basic equations of the algorithm, this is computationally much simpler than most optimal, or even suboptimal, reduction techniques as it only requires the solution of sets of linear equations. The MATLAB program implementing the algorithm, illustrated in section 4, is user-friendly and does not require the detailed knowledge of the method: the user must only supply the numerator and denominator coefficients of the original system and a starting guess for those of the reduced model. Finally, we discuss the results obtained by applying the above method to some meaningful examples taken from the literature.

2. Optimality conditions and iterative scheme

By denoting the original stable $m_{\mathrm{o}} \times m_{\mathrm{i}}$ transfer matrix by

$$F(s) = \frac{N(s)}{d(s)}, \tag{1}$$

where $d(s)$ is the l.c.d. of degree ν of all input-output transfer functions and $N(s)$ is the $m_{\mathrm{o}} \times m_{\mathrm{i}}$ matrix formed from the corresponding numerators of degree at most $\nu - 1$, and by

$$G(s) = \frac{M(s)}{c(s)} \tag{2}$$

the stable approximating transfer matrix whose l.c.d. $c(s)$ has degree $\rho < \nu$, the approximation error is defined as

$$E(s) := F(s) - G(s) = \frac{N_E(s)}{d_E(s)}, \tag{3}$$

where $N_E(s) = N(s)c(s) - M(s)d(s)$ and $d_E(s) = d(s)c(s)$.

The index to be minimized with respect to the parameters of $G(s)$ is

$$J = ||E(s)||^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathrm{tr}\ E(j\omega)E^*(j\omega)\ \mathrm{d}\omega, \tag{4}$$

where the star indicates complex conjugate transpose.

By referring for notational simplicity to the case in which $G(s)$ has ρ simple poles $-p_k$, $G(s)$ can be written in the form:

$$G(s) = \sum_{k=1}^{\rho} \frac{R_k}{s + p_k}$$

where R_k is an $m_{\mathrm{o}} \times m_{\mathrm{i}}$ complex matrix.

It has been shown [10] that the following interpolation conditions are necessary for optimality:

$$G(p_k^*) = F(p_k^*), \qquad k = 1, \ldots, \rho, \tag{5}$$

$$\mathrm{tr}\{[F'^*(p_k^*) = G'^*(p_k^*)] \cdot R_k\} = 0, \quad k = 1, \ldots, \rho, \tag{6}$$

in which the prime denotes derivative with respect to s. If the coefficients of the polynomials involved are real, equations (5) and (6) can be given the compact form of the following polynomial identities:

$$N(s)c(s) - M(s)d(s) = Q_1(s)c(-s), \tag{7}$$

$$\mathrm{tr}\{Q_1^{\mathrm{T}}(s)M(-s)\} = q_2(s)c(-s), \tag{8}$$

where $Q_1(s)$ is an $m_{\mathrm{o}} \times m_{\mathrm{i}}$ matrix of polynomials whose degree is at most $\nu - 1$ and $q_2(s)$ is a polynomial of degree at most $\nu - 2$ (the superscript T denotes transpose).

Equating the coefficients of the equal powers of s on both sides of (7) and (8), we obtain, respectively, $(\nu+\rho)\cdot(m_o\cdot m_i)$ and $\nu+\rho-1$ equations in the $\rho\cdot(m_o\cdot m_i+1)$ unknown numerators and denominator coefficients of $G(s)$ and in the $\nu\cdot(m_o\cdot m_i+1)-1$ *auxiliary* unknown coefficients of $Q_1(s)$ and $q_2(s)$. These equations have degree 2 in the considered unknowns but can be solved by means of an iterative procedure that, at each iteration, makes use of linear equations and essentially requires the inversion of matrices of order ρ and ν only, as shown in the next section.

Precisely, by denoting with superscript (h) the quantities computed in the hth iteration and with superscript $(h+1)$ those to be evaluated in the current $(h+1)$th iteration, the basic equations of the procedure are:

$$N(s)c^{(h+1)}(s)-M^{(h+1)}(s)d(s)=Q_1^{(h+1)}(s)c^{(h)}(-s), \tag{9}$$

$$\operatorname{tr}\{Q_1^{(h+1)^{\mathrm{T}}}(s)M^{(h)}(-s)\}=q_2^{(h+1)}(s)c^{(h)}(-s), \tag{10}$$

which in the case of SISO systems reduce to:

$$n(s)c^{(h+1)}(s)-m^{(h+1)}(s)d(s)=q^{(h+1)}(s)[c^{(h)}(-s)]^2, \tag{11}$$

where $n(s)$ and $m^{(h+1)}(s)$ are the numerator polynomials of the original and reduced (scalar) transfer functions, respectively, and $q^{(h+1)}(s)$ is an auxiliary polynomial.

Clearly, in the first iteration the values of the coefficients of $c^{(0)}(s)$ and $M^{(0)}(s)$ should be suitably guessed, and a stopping criterion should be provided (on the basis of the difference between two consecutive vectors of the coefficients of $c(s)$). Since, in general, the problem exhibits more than one solution, the procedure must be started from different initial points in order to identify and compare the local minima of J. Usually, the globally optimal model is "close" to, and can rapidly be reached from, the reduced model obtained through balancing, even if this is not always the case [8]. According to experience [20], the poles of the locally optimal models are often near those of the original system and, in particular, the poles of the best model are near the dominant poles (in the L_2 sense) of the original system which can be determined as suggested in [9].

Some considerations on the algorithm convergence are developed in the appendix. It is shown there that the critical points that are not minima of the index (4) are always repelling; usually, the minima are attracting, even if some very "flat" minima can be repelling too.

3. Outline of the algorithm

In this section we describe the structure of the equations resulting from the polynomial identities (9) and (10), and the steps of the algorithm that has been implemented using standard MATLAB functions.

To this purpose, it is necessary to introduce first some notation.

Matrices will be denoted by capital letters, scalars by small letters, and vectors by bold small letters. Zero matrices and column vectors will be indicated by $O_{p,q}$ and $\mathbf{o}_p$, where p and q correspond to their row and/or column dimensions. To simplify notation, the matrix entry of position (i,j) will be assigned a single subscript k according to the lexicographic order:

$$k = m_{\mathrm{i}}(i-1)+j, \quad i=1,\ldots,m_{\mathrm{o}}, \quad j=1,\ldots,m_{\mathrm{i}},$$

and the product $m_{\mathrm{o}} \cdot m_{\mathrm{i}}$ will be indicated by μ, so that $k=1,2,\ldots,\mu$.

According to this notation, the entry $n_{ij}(s)$ of $N(s)$ will be denoted by $n_k(s)$ and its coefficients by $n_{k,l}$, i.e.,

$$n_k(s) = \sum_{l=0}^{\nu-1} n_{k,l}s^l.$$

From these coefficients we form the row vector:

$$\mathbf{n}_k = [n_{k,0}, n_{k,1}, \ldots, n_{k,\nu-1}] \in \mathbb{R}^{\nu}.$$

Similarly, from the polynomial entries $m_k^{(l)}(s)$ of $M^{(l)}(s), l=h,h+1$, and $q_{1k}^{(h+1)}(s)$ of $Q_1^{(h+1)}(s)$, from the polynomial $q_2^{(h+1)}(s)$ and from the *monic* polynomials $d(s)$ and $c^{(l)}(s)$, $l=h,h+1$, we form the row vectors:

$$\begin{aligned}
\mathbf{m}_k^{(l)} &= [m_{k,0}^{(l)}, m_{k,1}^{(l)}, \ldots, m_{k,\rho-1}^{(l)}] \in \mathbb{R}^{\rho},\\
\mathbf{q}_{1k}^{(h+1)} &= [q_{1k,0}^{(h+1)}, q_{1k,1}^{(h+1)}, \ldots, q_{1k,\nu-1}^{(h+1)}] \in \mathbb{R}^{\nu},\\
\mathbf{q}_2^{(h+1)} &= [q_{2,0}^{(h+1)}, q_{2,1}^{(h+1)}, \ldots, 1_{2,\nu-2}^{(h+1)}] \in \mathbb{R}^{\nu-1},\\
\mathbf{d} &= [d_0, d_1, \ldots, d_{\nu-1}] \in \mathbb{R}^{\nu},\\
\mathbf{c}^{(l)} &= [c_0^{(l)}, c_1^{(l)} \ldots, c_{\rho-1}^{(l)}] \in \mathbb{R}^{\rho.}
\end{aligned}$$

By properly ordering the unknowns to be determined at the current $(h+1)$th iteration, the column vector $\mathbf{x}$ collecting these unknowns can be written as

$$\mathbf{x} = [\mathbf{x}_1^{\mathrm{T}}, \mathbf{x}_2^{\mathrm{T}}, \ldots, \mathbf{x}_{\mu}^{\mathrm{T}}, \mathbf{x}_{\mu+1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{(\nu+\rho)(\mu+1)-1},$$

where

$$\begin{aligned}
\mathbf{x}_k^{\mathrm{T}} &= [\mathbf{m}_k^{(h+1)}, \mathbf{q}_{1k}^{(h+1)}], \quad k=1,2,\ldots,\mu,\\
\mathbf{x}_{\mu+1}^{\mathrm{T}} &= [\mathbf{q}_2^{(h+1)}, \mathbf{c}^{(h+1)}].
\end{aligned}$$

In this way, the *linear* set of equations for the current $(h+1)$th iteration, which is denoted by

$$A\mathbf{x} = \mathbf{b}, \tag{12}$$

takes on the particularly convenient structure illustrated in the following.

To this purpose, let us introduce the matrices:

$$D_1 = \begin{bmatrix} d_0 & 0 & \dots & 0 \\ d_1 & d_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ d_{\rho-1} & d_{\rho-2} & \dots & d_0 \end{bmatrix} \in \mathbb{R}^{\rho\times\rho},$$

$$D_2 = \begin{bmatrix} d_\rho & d_{\rho-1} & \dots & d_1 \\ \vdots & \vdots & & \vdots \\ 1 & d_{\nu-1} & \dots & d_{\nu-\rho+1} \\ 0 & 1 & \dots & d_{\nu-\rho+2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{\nu\times\rho},$$

$$C_1^{(h)} = \begin{bmatrix} -c_0^{(h)} & 0 & \dots & 0 & 0 & \dots & 0 \\ c_1^{(h)} & -c_0^{(h)} & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ -(-1)^{\rho-1}c_{\rho-1}^{(h)} & -(-1)^{\rho-1}c_{\rho-2}^{(h)} & \dots & -c_0^{(h)} & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{\rho\times\nu},$$

$$C_2^{(h)} = \begin{bmatrix} -(-1)^\rho & -(-1)^{\rho-1}c_{\rho-1}^{(h)} & \dots & c_1^{(h)} & -c_0^{(h)} & 0 & \dots & 0 \\ 0 & -(-1)^\rho & \dots & -c_2^{(h)} & c_1^{(h)} & -c_0^{(h)} & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 & \dots & -(-1)^\rho \end{bmatrix} \in \mathbb{R}^{\nu\times\nu},$$

$$\Delta = \left[\begin{array}{c|c} D_1 & C_1^{(h)} \\ \hline D_2 & C_2^{(h)} \end{array}\right] \in \mathbb{R}^{(\nu+\rho)\times(\nu+\rho)}.$$

Let us now indicate with $\overline{C}_1^{(h)} \in \mathbb{R}^{\rho\times(\nu-1)}$ the matrix obtained from $C_1^{(h)}$ by deleting its last column and with $\overline{C}_2^{(h)} \in \mathbb{R}^{(\nu-1)\times(\nu-1)}$ the matrix obtained from $C_2^{(h)}$ by deleting its last row and column, and form the square matrix:

$$\overline{\Delta} = \left[\begin{array}{c|c} \overline{C}_1^{(h)} & O_{\rho,\rho} \\ \hline \overline{C}_2^{(h)} & O_{\nu-1,\rho} \end{array}\right] \in \mathbb{R}^{(\nu+\rho-1)\times(\nu+\rho-1)}.$$

Let us also define the matrices:

$$N_{k,1} = \begin{bmatrix} n_{k,0} & 0 & \dots & 0 \\ n_{k,1} & n_{k,0} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ n_{k,\rho-1} & n_{k,\rho-2} & \dots & n_{k,0} \end{bmatrix} \in \mathbb{R}^{\rho\times\rho},$$

$$N_{k,2} = \begin{bmatrix} n_{k,\rho} & n_{k,\rho-1} & \dots & n_{k,1} \\ \vdots & \vdots & & \vdots \\ n_{k,\nu-1} & n_{k,\nu-2} & \dots & n_{k,\nu-\rho} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & n_{k,\nu-1} \\ 0 & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{\nu\times\rho},$$

$$N_k = \left[\begin{array}{c|c} O_{\rho,\nu-1} & N_{k,1} \\ \hline O_{\nu,\nu-1} & N_{k,2} \end{array}\right] \in \mathbb{R}^{(\nu+\rho)\times(\nu+\rho-1)},$$

$$M_{k,1}^{(h)} = \begin{bmatrix} m_{k,0}^{(h)} & 0 & \dots & 0 & 0 & \dots & 0 \\ -m_{k,1}^{(h)} & m_{k,0}^{(h)} & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ (-1)^{\rho-1} m_{k,\rho-1}^{(h)} & (-1)^{\rho-2} m_{k,\rho-2}^{(h)} & \dots & m_{k,0}^{(h)} & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{\rho\times\nu},$$

$$M_{k,2}^{(h)} =$$

$$\begin{bmatrix} 0 & (-1)^{\rho-1}m_{k,\rho-1}^{(h)} & \dots & -m_{k,1}^{(h)} & m_{k,0}^{(h)} & 0 & \dots & 0 \\ 0 & 0 & \dots & m_{k,2}^{(h)} & -m_{k,1}^{(h)} & m_{k,0}^{(h)} & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & (-1)^{\rho-1}m_{k,\rho-1}^{(h)} \end{bmatrix} \in \mathbb{R}^{(\nu-1)\times\nu},$$

and

$$M_k^{(h)} = \left[\begin{array}{c|c} O_{\rho,\rho} & M_{k,1}^{(h)} \\ \hline O_{\nu-1,\rho} & M_{k,2}^{(h)} \end{array}\right] \in \mathbb{R}^{(\nu+\rho-1)\times(\nu+\rho)},$$

With the above notation, matrix A in (12) takes the form:

$$A = \begin{bmatrix} \Delta & 0 & \dots & 0 & N_1 \\ 0 & \Delta & \dots & 0 & N_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & \Delta & N_\mu \\ M_1^{(h)} & M_2^{(h)} & \dots & M_\mu^{(h)} & \overline{\Delta} \end{bmatrix}, \tag{13}$$

and column vector $\mathbf{b}$ is given by:

$$\mathbf{b} = [\mathbf{b}_1^{\mathrm{T}}, \mathbf{b}_2^{\mathrm{T}}, \dots, \mathbf{b}_\mu^{\mathrm{T}}, \mathbf{o}_{\nu+\rho-1}^{\mathrm{T}}]^{\mathrm{T}},$$

with

$$\mathbf{b}_k^{\mathrm{T}} = [\mathbf{o}_\rho^{\mathrm{T}}, \overline{\mathbf{b}}_k^{\mathrm{T}}],$$

where

$$\overline{\mathbf{b}}_k^{\mathrm{T}} = [-n_{k,0}, -n_{k,1}, \dots, -n_{k,\nu-1}] \in \mathbb{R}^\nu.$$

The direct solution of (12) would thus require the inversion of matrix (13) which is of order $[(\nu+\rho)(\mu+1)-1]$. However, by eliminating the auxiliary variables $\mathbf{q}_{1k}^{(h+1)}$ and $\mathbf{q}_2^{(h+1)}$, which can be done by inverting the (triangular) nonsingular matrices $C_2^{(h)}$ and $\overline{C}_2^{(h)}$, we obtain a set of $[\rho(\mu+1)-1]$ linear equations in the same number of unknowns. The related coefficient matrix is certainly invertible at least in the neighbourhood of the considered stationary points, as shown in the appendix.

The computational complexity of the algorithm can, in general, be further reduced by exploiting the sparseness of A. This result is achieved with the simplified procedure illustrated below. It essentially requires the inversion of matrices of order ν and ρ only, whose invertibility is generically guaranteed (cf. appendix).

From the first μ block rows of A, we obtain:

$$D_1\mathbf{m}_k^{(h+1)^{\mathrm{T}}} + C_1^{(h)}\mathbf{q}_{1k}^{(h+1)^{\mathrm{T}}} + N_{k,1}\mathbf{c}^{(h+1)^{\mathrm{T}}} = \mathbf{o}_\rho, \quad k = 1,\dots,\mu, \tag{14}$$

$$D_2\mathbf{m}_k^{(h+1)^{\mathrm{T}}} + C_2^{(h)}\mathbf{q}_{1k}^{(h+1)^{\mathrm{T}}} + N_{k,2}\mathbf{c}^{(h+1)^{\mathrm{T}}} = \bar{\mathbf{b}}_k, \quad k = 1,\dots,\mu. \tag{15}$$

Solving (15) for $\mathbf{q}_{1k}^{(h+1)}$ (note that $C_2^{(h)}$ is always invertible), we get:

$$\mathbf{q}_{1k}^{(h+1)^{\mathrm{T}}} = (C_2^{(h)})^{-1}[\bar{\mathbf{b}}_k - D_2\mathbf{m}_k^{(h+1)^{\mathrm{T}}} - N_{k,2}\mathbf{c}^{(h+1)^{\mathrm{T}}}] \tag{16}$$

and substituting this expression into (14), we have:

$$\Gamma\mathbf{m}_k^{(h+1)^{\mathrm{T}}} + \Theta_k\mathbf{c}^{(h+1)^{\mathrm{T}}} = \mathbf{t}_k, \quad k = 1,\dots,\mu, \tag{17}$$

where

$$\Gamma = D_1 - C_1^{(h)}(C_2^{(h)})^{-1}D_2 \in \mathbb{R}^{\rho\times\rho},$$

$$\Theta_k = N_{k,1} - C_1^{(h)}(C_2^{(h)})^{-1}N_{k,2} \in \mathbb{R}^{\rho\times\rho},$$

$$\mathbf{t}_k = -C_1^{(h)}(C_2^{(h)})^{-1}\bar{\mathbf{b}}_k \in \mathbb{R}^{\rho}.$$

From the last $(\mu + 1)$th block row of (13), we get:

$$\sum_{k=1}^{\mu} M_{k,1}^{(h)}\mathbf{q}_{1k}^{(h+1)^{\mathrm{T}}} + \overline{C}_1^{(h)}\mathbf{q}_2^{(h+1)^{\mathrm{T}}} = \mathbf{o}_\rho, \tag{18}$$

$$\sum_{k=1}^{\mu} M_{k,2}^{(h)}\mathbf{q}_{1k}^{(h+1)^{\mathrm{T}}} + \overline{C}_2^{(h)}\mathbf{q}_2^{(h+1)^{\mathrm{T}}} = \mathbf{o}_{\nu-1}. \tag{19}$$

Solving (19) for $\mathbf{q}_2^{(h+1)}$ (note that $\overline{C}_2^{(h)}$ is always invertible), we obtain:

$$\mathbf{q}_2^{(h+1)^{\mathrm{T}}} = -(\overline{C}_2^{(h)})^{-1}\left[\sum_{k=1}^{\mu} M_{k,2}^{(h)}\mathbf{q}_{1k}^{(h+1)^{\mathrm{T}}}\right] \tag{20}$$

Using (16) and (20), from (18) we have:

$$\sum_{k=1}^{\mu} \Pi_k\mathbf{m}_k^{(h+1)^{\mathrm{T}}} + \Phi\mathbf{c}^{(h+1)^{\mathrm{T}}} = \mathbf{t}_{\mu+1}, \tag{21}$$

where

$$\Pi_k = [M_{k,1}^{(h)} - \overline{C}_1^{(h)}(\overline{C}_2^{(h)})^{-1}M_{k,2}^{(h)}](C_2^{(h)})^{-1}D_2 \in \mathbb{R}^{\rho\times\rho},$$

$$\Phi = \sum_{k=1}^{\mu}[M_{k,1}^{(h)} - \overline{C}_1^{(h)}(\overline{C}_2^{(h)})^{-1}M_{k,2}^{(h)}](C_2^{(h)})^{-1}N_{k,2} \in \mathbb{R}^{\rho\times\rho},$$

$$\mathbf{t}_{\mu+1} = \sum_{k=1}^{\mu}[M_{k,1}^{(h)} - \overline{C}_1^{(h)}(\overline{C}_2^{(h)})^{-1}M_{k,2}^{(h)}](C_2^{(h)})^{-1}\overline{\mathbf{b}}_k.$$

From (17) we get:

$$\mathbf{m}_k^{(h+1)^{\mathrm{T}}} = -\Gamma^{-1}\Theta_k\mathbf{c}^{(h+1)^{\mathrm{T}}} + \mathbf{t}, \quad k = 1, \ldots, \mu, \tag{22}$$

and substituting these expressions into (21), we finally obtain:

$$\mathbf{c}^{(h+1)^{\mathrm{T}}} = H^{-1}\mathbf{v}, \tag{23}$$

where

$$H = \Phi - \sum_{k=1}^{\mu}\Pi_k\Gamma^{-1}\Theta_k \in \mathbb{R}^{\rho\times\rho},$$

$$\mathbf{v} = \mathbf{t}_{\mu+1} - \sum_{k=1}^{\mu}\Pi_k\mathbf{t}_k \in \mathbb{R}^{\rho}.$$

On the basis of the previous relations, the solution algorithm entails the successive evaluation of vectors $\mathbf{c}^{(h+1)}$ and $\mathbf{m}_k^{(h+1)}$ using (23) and (22), respectively. Equation (23) requires the inversion of the $\rho \times \rho$ matrix H, which in turn requires the inversion of the $\rho \times \rho$ matrix Γ. Equations (22) do not require the inversion of other matrices. Considerations on the invertibility of the mentioned matrices are made in the appendix.

The above procedure is not computationally demanding. In fact, it is only necessary to invert at each step the triangular matrix $C_2^{(h)}$ of order ν, its triangular submatrix $\overline{C}_2^{(h)}$ of order $\nu - 1$, and the $\rho \times \rho$ matrices Γ and H.

4. Program description

The program implementing the algorithm of section 3 consists of 10 MATLAB routines (called: `sino`, `mino`, `l2siso`, `l2mimo`, `energy`, `sinr`, `sour`, `minr`, `mour`, `rever`). The package is self-contained and does not require extra MATLAB functions. Even if the reduction of SISO systems could well be performed by resorting to the more complex procedure for the general case of MIMO systems, two separate routines have been developed for the two cases.

For convenience, polynomial coefficients are to be supplied according to ascending powers of s and so are stored. In the MIMO case, the polynomials in the numerator matrix are arranged row by row. If the number of numerator coefficients specified by the user is less than the model order, the related routine autonomously sets to zero the coefficients of the relevant higher powers of s. Also, if the supplied denominator is not monic, the input function divides all the numerator and denominator coefficients by the coefficient of the highest power of s in the denominator. Various controls are made in the execution of the program functions to check the data consistency.

Only two functions need be used to run the program from a MATLAB session. Specifically, in the SISO case they are:

```
[num,den] = sino;
[nr,dr,jr,kr,np,dp,jp,kp] = l2siso(num,den,epsa,kmax,kdisp);
```

and in the MIMO case they are:

```
[num,den,mi,mo] = mino;
[nr,dr,jr,kr,np,dp,jp,kp] = l2mimo(num,den,mi,mo,epsa,kmax,kdisp);
```

Functions `sino` and `mimo` perform the input operators. They store the vectors of the original numerator(s) and denominator coefficients (`num` and `den`, respectively) as required by the routines `l2siso` and `l2mimo`. Clearly, it is not necessary to repeat the input operation to find reduced models of different order for the same original system.

The parameters `epsa`, `kmax` and `kdisp` are optional. Parameter `epsa` determines the stopping criterion; it sets the tolerance on the difference between the parameter values computed at two consecutive iterations according to:

$$\max_i |c_i^{(h+1)} - c_i^{(h)}| \leq \texttt{epsa} \cdot \min_i |c_i^{(h)}|.$$

The default value for `epsa` is 0.001. If this stopping criterion is not satisfied within `kmax` iterations, the procedure is arrested. The default value for `kmax` is 50. Partial results are displayed every `kdisp` iterations. If this parameter is not specified, no partial results are displayed.

When a solution has been found, the corresponding polynomial coefficients are displayed according to ascending powers of s and stored in the output parameter vectors `nr` and `dr`. The output parameter `jr` gives the related index value and `kr` the number of iterations needed to reach the solution. The index value (squared error norm of the error between the original and the reduced transfer function) is computed by a suitable function, called `energy`, according to the Routh-like algorithm suggested by Åström [2]. Note that the final solution corresponding to the satisfaction of the adopted stopping criterion might lead to an index value (`jr`) slightly greater than that characterizing a previous iteration. In this case, the program also gives the latter minimum value `jp`, together with the related reduced numerator `np`, denominator `dp` and number of iterations `kp`.

Four specific functions, called `sinr`, `sour`, `minr`, `mour`, are used by `l2siso` and `l2mimo` to perform the input and output operations concerning the reduced model.

Finally, to facilitate the use of MATLAB routines (which require that the polynomial coefficients be supplied according to descending powers of s), like the function `impulse`, included in the *Control system toolbox* and used in the example section, a special function, named `rever`, has been included in the package: it reverses the order of the denominator and numerator(s) coefficients and associates with the (common) denominator the numerator of interest.

5. Examples

The performance of the algorithm has been tested on a number of examples. In any case, the number of iterations required to find a (local) minimum of J has been small even if the initial guess had not been close to it and the stopping criterion very stringent (typically, less than 10 iterations for original systems of order 6–8 with less than 3 inputs and/or outputs). However, if the distribution of the original system poles and zeros is very spread and/or quasi-cancellations of pole-zero pairs occur, the problem may become ill-conditioned and the number of iterations increases: for instance, the number of iterations needed to find an 8th-order approximant for a 17th-order system with the above characteristics has been greater than 40. Clearly, since different local minima can be present, a suitable number of starting points should be considered.

5.1. *Example 1*

The first example, which refers to an SISO system, is taken from [19] and has also been considered in [9] in connection with pole retention techniques.

The original 8th-order transfer function is:

$$f(s) = \frac{n(s)}{d(s)}$$

with

$$\begin{aligned} n(s) &= 4.026610 \cdot 10^4 + 1.853269 \cdot 10^5 s + 2.215650 \cdot 10^5 s^2 + 1.224091 \cdot 10^5 s^3 \\ &\quad + 3.632059 \cdot 10^4 s^4 + 5.975406 \cdot 10^3 s^5 + 5.137200 \cdot 10^2 s^6 + 1.8 \cdot 10 s^7 \\ &= 18(s+0.32)(s+2.45-j0.53)(s+2.45+j0.53)(s+5-j0.65) \\ &\quad (s+5+j0.65)(s+5.89)(s+7.43), \end{aligned}$$

$$\begin{aligned} d(s) &= 40320 + 109584s + 118124s^2 + 67284s^3 + 22449s^4 + 4536s^5 \\ &\quad + 546s^6 + 36s^7 + s^8 \\ &= (s+1)(s+2)(s+3)(s+4)(s+5)(s+6)(s+7)(s+8). \end{aligned}$$

As already oberved, the poles of the optimal models tend to occur near the poles of the full-order system, which gives support to the suggestion made in [9] of choosing a suitable set of original poles, i.e., the dominant ones, as a convenient starting point. In fact, this choice has led in a very small number of iterations to the L_2-optimal models. However the same solutions have been reached in less than 8 iterations adopting initial guesses with all coefficients equal to 1. In the sequel, we give the models of order from 5 to 1 with the corresponding minimum index values obtained in this way (the values for the models of order 7 and 6 are extremely small and comparable with the effects of rounding errors).

$$\begin{aligned} m_v(s) &= 17.999968(s+4.880298)(s+2.355235-j0.5977134) \\ &\quad (s+2.355235+j0.5977134)(s+0.3199891), \\ c_v(s) &= (s+7.773184)(s+3.866904)(s+2.616948)(s+2.114325) \\ &\quad (s+0.9992669), \\ J_v &= 5.468801\cdot 10^{-11}; \end{aligned}$$

$$\begin{aligned} m_{iv}(s) &= 18.00051(s+5.546283)(s+1.860185)(s+0.3205420), \\ c_{iv}(s) &= (s+7.868239)(s+4.733892)(s+1.562191)(s+1.024225), \\ J_{iv} &= 5.799822\cdot 10^{-8}; \end{aligned}$$

$$\begin{aligned} m_{iii}(s) &= 17.98747(s+3.502370)(s+0.3130394), \\ c_{iii}(s) &= (s+7.453438)(s+2.864725)(s+0.9287173), \\ J_{iii} &= 2.520804\cdot 10^{-5}; \end{aligned}$$

$$\begin{aligned} m_{ii}(s) &= 17.78501(s+0.2596419), \\ c_{ii}(s) &= (s+6.662265)(s+0.7330904), \\ J_{ii} &= 5.590181\cdot 10^{-3}; \end{aligned}$$

$$\begin{aligned} m_i(s) &= 18.61081, \\ c_i(s) &= s+8.220893, \\ J_i &= 6.202763\cdot 10^{-1}. \end{aligned}$$

Note that the values of J are appreciably smaller than those for the best pole-retaining models given in [9].

The impulse responses of the original system and of the reduced models of order 2 and 1 are shown in figure 1: the second-order model reproduces the original

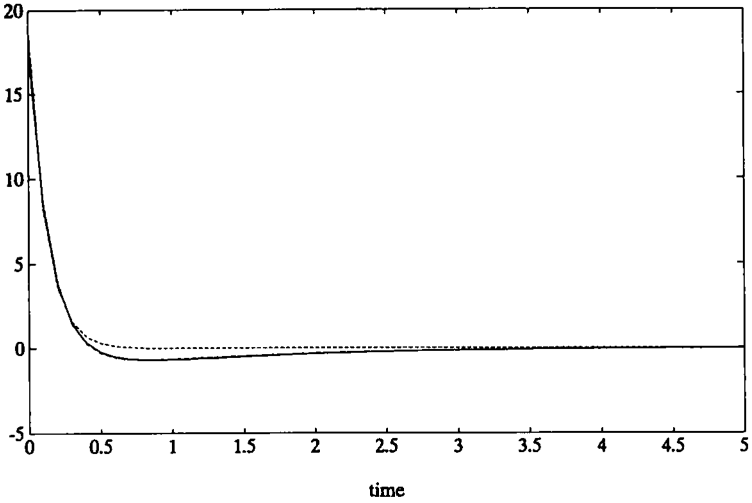

Figure 1. Impulse response of the original system of example 1 (solid line) and of the reduced models of order 2 (dashed-dotted line) and 1 (dashed line).

behaviour very satisfactorily. The responses of the other higher-order reduced models are not represented because they practically coincide with the original one.

5.2. *Example 2*

This second example refers to an SIMO system with 2 outputs which has first been studied in [11]. It has also been considered in [7] where attention concentrates on the reduction in balanced and modal coordinates. The original 6th-order transfer matrix is

$$F(s) = \frac{\begin{bmatrix} n_{11}(s) \\ n_{21}(s) \end{bmatrix}}{d(s)}$$

with

$$\begin{aligned} n_{11}(s) = {} & -(5.892792 \cdot 10^5 + 1.703558 \cdot 10^7 s + 1.933526 \cdot 10^5 s^2 \\ & + 2.980877 \cdot 10^4 s^3 + 1.045770 \cdot 10^2 s^4 + 3.479562 s^5), \end{aligned}$$

$$\begin{aligned} n_{21}(s) = {} & -(1.705535 \cdot 10^7 + 3.172180 \cdot 10^5 s + 3.439612 \cdot 10^4 s^2 \\ & + 3.649765 \cdot 10^2 s^3 + 8.133429 s^4 + 5.232000 \cdot 10^{-2} s^5), \end{aligned}$$

$$\begin{aligned} d(s) = {} & 2.664275 \cdot 10^5 + 5.843032 \cdot 10^5 s + 2.371008 \cdot 10^6 s^2 \\ & + 2.790969 \cdot 10^4 s^3 + 4.577306 \cdot 10^3 s^4 + 1.766590 \cdot 10 s^5 + s^6. \end{aligned}$$

In order to find a 4th-order model, we have arbitrarily started the suggested algorithm from the following model:

$$m_{11}^{(0)}(s) = -(5 + 20s + s^2 + 0.01s^3),$$
$$m_{21}^{(0)}(s) = -(1 + 10s + s^2 + 0.1s^3),$$
$$c^{(0)}(s) = 1 + 10s + 100s^2 + 1000s^3 + s^4,$$

which has turned out to be very far from the optimal solution. Nevertheless, after 7 iterations only, the following model has been obtained:

$$m_{11}(s) = -(9.849753 \cdot 10^2 + 2.851144 \cdot 10^4 s + 7.639196 \cdot 10 s^2 + 3.705600 s^3),$$
$$m_{21}(s) = -(2.854926 \cdot 10^4 + 2.981424 \cdot 10^2 s + 7.890070 s^2 + 5.217566 \cdot 10^{-2} s^3),$$
$$c(s) = 4.459679 \cdot 10^2 + 9.744784 \cdot 10^2 s + 3.960059 \cdot 10^3 s^2 + 1.296605 \cdot 10 s^3 + s^4,$$
$$J = 6.096262 \cdot 10^{-3}.$$

Note that the reduced model practically retains the four modes with the largest absolute values of the residues, i.e., those corresponding to the poles at about $-6.3 \pm j62$ and $-0.12 \pm j0.31$.

The impulse responses of the reduced model are almost coincident with those of the original system and, therefore, are not represented.

5.3. *Example 3*

This third example refers to an MIMO system with 2 inputs and 2 outputs. It has been considered in [15] in connection with an error minimization technique with fixed poles. The original system is:

$$F(s) = \frac{\begin{bmatrix} n_{11}(s) & n_{12}(s) \\ n_{21}(s) & n_{22}(s) \end{bmatrix}}{d(s)}$$

with

$$n_{11}(s) = 102 + 133s + 63s^2 + 13s^3 + s^4,$$
$$n_{12}(s) = 1581 + 2163.5s + 1109.5s^2 + 264.5s^3 + 28.5s^4 + s^5,$$
$$n_{21}(s) = 4 + 41.3s + 13.1s^2 + s^3,$$
$$n_{22}(s) = 62 + 644.15s + 244.35s^2 + 28.6s^3 + s^4,$$
$$d(s) = 50 + 212.5s + 318s^2 + 223s^3 + 81s^4 + 14.5s^5 + s^6$$
$$= (s+5)(s+4)(s+2-j)(s+2+j)(s+1)(s+0.5).$$

In this case too, by starting from arbitrary initial points (e.g., all coefficients equal to 1), the algorithm arrives in about 10 iterations at the following model with $\deg\{c(s)\} = 2$:

$$G(s) = \frac{\begin{bmatrix} m_{11}(s) & m_{12}(s) \\ m_{21}(s) & m_{22}(s) \end{bmatrix}}{c(s)}$$

with

$$\begin{aligned} m_{11}(s) &= 1.041851 - 5.720402 \cdot 10^{-3}s, \\ m_{12}(s) &= 16.15166 + 9.447782 \cdot 10^{-1}s, \\ m_{21}(s) &= 1.196381 + 8.155153 \cdot 10^{-2}s, \\ m_{22}(s) &= 1.812015 + 1.503528s, \\ c(s) &= 5.196188 \cdot 10^{-1} + 1.469527s + s^2. \end{aligned}$$

The corresponding index value turns out to be $J \simeq 3.13$, whereas the value for the model obtained in [15], which retains the poles at -0.5 and -1, is $J \simeq 8.6$. Observe again that the L_2-optimal model poles are close to the original system poles chosen in [15].

Figure 2 shows the impulse responses corresponding to each i/o pair of the original system and of the reduced model. It is clearly seen that the responses at output 1 to an impulse applied to inputs 1 and 2 are approximated very well, whereas the approximation of the responses at output 2 to an impulse applied to

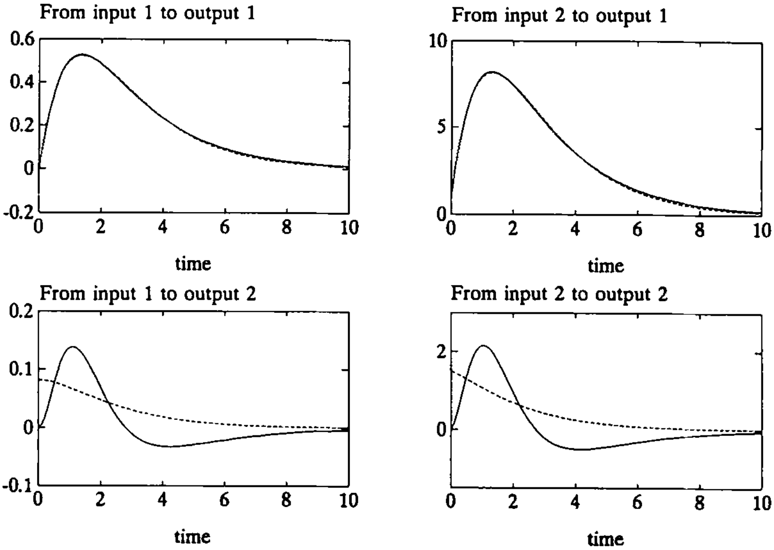

Figure 2. Inpulse reponses of the original system of example 3 (solid lines) and of the reduced model (dashed lines).

inputs 1 and 2 are poorer, even if the deviations are magnified by the different scales adopted. It might be said that the better approximation of the response characterized by the highest peak (from input 2 to output 1) is obtained at the expense of a worse approximation of the other responses.

6. Conclusions

In this paper we have considered the problem of computing a reduced-order model of form (2) for an original higher-order multivariable linear system with transfer matrix (1) in such a way that the squared L_2 norm (4) of the approximation error (3) is minimal. To this purpose, a numerical algorithm has been developed (section 3) which is based on the compact form (7)–(8) of the first-order necessary conditions of optimality. It is characterized by remarkable computational simplicity compared to the other available techniques.

The algorithm described in section 4 has been implemented on a PC using standard MATLAB functions. It is available in library NUMERALGO of *netlib*.

The program has successfully been tested on a variety of examples, three of which have been discussed in section 5.

Appendix

A1. *Algorithm convergence*

For simplicity, we shall limit attention to the case of SISO systems for which the reduced transfer *function $g(s)$* has simple *real* poles; the extension to the general case would entail a considerable increase in notation without changing the essential features of the problem.

With the above assumption, we have

$$g(s) = \sum_{i=1}^{\rho} \frac{r_i}{s+p_i}, \quad r_i \in \mathbb{R}, p_i \in \mathbb{R}_+, \tag{24}$$

where r_i is the (scalar) residue at pole $-p_i$, and the index to be minimized can be expressed as

$$\begin{aligned} J(\mathbf{r},\mathbf{p}) &= \frac{1}{2\pi j}\int_{-j\infty}^{+j\infty} [f(s)-g(s)][f(-s)-g(-s)]\,\mathrm{d}s \\ &= \mathbf{r}^{\mathrm{T}}P_{11}\mathbf{r} - 2\mathbf{r}^{\mathrm{T}}\mathbf{f}(\mathbf{p}) + \|f\|^2, \end{aligned}$$

where $f(s)$ is the original transfer function, $\mathbf{r}^{\mathrm{T}} = [r_1, r_2, \dots, r_\rho], P_{11} = \{(p_i+p_j)^{-1}\}, i,j = 1,2,\dots,\rho, \mathbf{p}^{\mathrm{T}} = [p_1, p_2, \dots, p_\rho], \mathbf{f}^{\mathrm{T}}(\mathbf{p}) = [f(p_1), f(p_2), \dots, f(p_\rho)]$.

It is easy to verify that the associate $2\rho \times 2\rho$ Hessian matrix $H(\mathbf{r},\mathbf{p})$ can be written as

$$H(\mathbf{r},\mathbf{p}) = 2\begin{bmatrix} I & O \\ O & R \end{bmatrix} \overline{H}(\mathbf{r},\mathbf{p}) \begin{bmatrix} I & O \\ O & R \end{bmatrix}$$

with $R = \text{diag}\{r_i\}, i = 1,2,\ldots,\rho$, and

$$\overline{H}(\mathbf{r},\mathbf{p}) = \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} - B_2R^{-1} \end{bmatrix} = P - \begin{bmatrix} O & O \\ O & B_2R^{-1} \end{bmatrix}, \tag{25}$$

where P_{12} and P_{22} are *symmetric* matrices given by $P_{12} = \{-(p_i + p_j)^{-2}\}$ and $P_{22} = \{2(p_i + p_j)^{-3}\}, i,j = 1,2,\ldots,\rho$, respectively, and $B_2 = \text{diag}\{(\partial/\partial p_i)[f'(p_i) - g'(p_i)]\}, i = 1,2,\ldots,\rho$.

Clearly, $H(\mathbf{r},\mathbf{p})$ is positive (negative) semidefinite if $\overline{H}(\mathbf{r},\mathbf{p})$ is. Now, P_{11} is positive definite for $p_i > 0, i = 1,2,\ldots,\rho$, so that $\overline{H}(\mathbf{r},\mathbf{p})$ cannot be negative semidefinite and $J(\mathbf{r},\mathbf{p})$ does not exhibit maxima: this property (often overlooked) is of course a consequence of the fact that the restriction of the objective function to the linear space obtained by fixing the values of the poles is quadratic and thus convex.

However, by setting $\partial J(\mathbf{r},\mathbf{p})/\partial\mathbf{r}$ to zero, which is a necessary condition for optimality, we form a set of ρ equations linear in the components of $\mathbf{r}$ (whose coefficient matrix is precisely P_{11}). These components can thus be expressed as functions of $\mathbf{p}$, i.e., $\mathbf{r} = \mathbf{r}(\mathbf{p})$. Replacing $\mathbf{r}$ by $\mathbf{r}(\mathbf{p})$ in $J(\mathbf{r},\mathbf{p})$, we obtain a function of $\mathbf{p}$ only:

$$\hat{J}(\mathbf{p}) = J(\mathbf{r}(\mathbf{p}),\mathbf{p}),$$

whose derivative with respect to $\mathbf{p}$ must also be zero at a stationary point. Of course, $\hat{J}(\mathbf{p})$ may well exhibit maxima which are therefore saddle points for $J(\mathbf{r},\mathbf{p})$.

The equation (11) describing the generic step of the iterative procedure in the SISO case corresponds to a set of interpolation conditions: in fact, it simply means that the *current* error $e^{(h+1)}(s) = f(s) - g^{(h+1)}(s)$ must exhibit zeros of multiplicity 2 at the *opposites* $p_k^{(h)}, k = 1.2.\ldots,\rho$, of the poles of the function $g^{(h)}(s)$ computed in the preceding step, in other words, $g^{(h+1)}(s)$ must *interpolate* the given function $f(s)$ at the above points with intersection number 2. Specifically, we have:

$$f(p_k^{(h)}) - g^{(h+1)}(p_k^{(h)}) = 0, \quad k = 1,2,\ldots,\rho, \tag{26}$$

$$f'(p_k^{(h)}) - g'^{(h+1)}(p_k^{(h)}) = 0, \quad k = 1,2,\ldots,\rho. \tag{27}$$

Unlike the methods of descent type, the heuristics under consideration need not a priori ensure the attracting nature of the minima of $\hat{J}$ and the repelling nature of its maxima. Therefore, it is necessary to investigate the convergence of the algorithm. To this purpose, we shall refer to the parameters p_k and r_k of $g(s)$, which clearly identify the same function as the corresponding numerator and denominator coefficients.

Set (26,27) can be rewritten in vector form as:

$$\Psi_{\mathrm{I}}(\mathbf{p}^{(h)}, \mathbf{p}^{(h+1)}, \mathbf{r}^{(h+1)}) = 0, \tag{28}$$

where

$$\mathbf{p}^{(i)} = [p_1^{(i)}, p_2^{(i)}, \ldots, p_\rho^{(i)}]^{\mathrm{T}}, \quad i = h, h+1,$$

$$\mathbf{r}^{(h+1)} = [r_1^{(h+1)}, r_2^{(h+1)}, \ldots, r_\rho^{(h+1)}]^{\mathrm{T}}.$$

Equation (28) implicitly defines a function Ψ_{E} that supplies the values of $\mathbf{r}^{(h+1)}$ and $\mathbf{p}^{(h+1)}$ from those of $\mathbf{p}^{(h)}$, i.e.,

$$\begin{bmatrix} \mathbf{r}^{(h+1)} \\ \mathbf{p}^{(h+1)} \end{bmatrix} = \Psi_{\mathrm{E}}(\mathbf{p}^{(h)}) = \begin{bmatrix} \Psi_{\mathrm{E1}}(\mathbf{p}^{(h)}) \\ \Psi_{\mathrm{E2}}(\mathbf{p}^{(h)}) \end{bmatrix},$$

where Ψ_{E1} and Ψ_{E2} are vectors of dimension ρ.

By assuming the invertibility of the Jacobian $\partial\Psi_{\mathrm{I}}/\partial\Psi_{\mathrm{E}}$ (which is guaranteed in the neighbourhood of the considered critical points, as we shall see later), the derivative of Ψ_{E} with respect to $\mathbf{p}^{(h)}$ can be expressed as:

$$\frac{\mathrm{d}\Psi_{\mathrm{E}}}{\mathrm{d}\mathbf{p}^{(h)}} = -\left[\frac{\partial\Psi_{\mathrm{I}}}{\partial\Psi_{\mathrm{E}}}\right]^{-1} \cdot \frac{\partial\Psi_{\mathrm{I}}}{\partial\mathbf{p}^{(h)}},$$

where, taking into account the partial fraction expansion (24) of $g(s)$, we have

$$\frac{\partial\Psi_{\mathrm{I}}}{\partial\Psi_{\mathrm{E}}} = \begin{bmatrix} \overline{P}_{11} & \overline{P}_{12} \\ \overline{P}_{12} & \overline{P}_{22} \end{bmatrix} \begin{bmatrix} I & O \\ O & \overline{R} \end{bmatrix} \tag{29}$$

with

$$\overline{P}_{11} = \{(p_i^{(h)} + p_j^{(h+1)})^{-1}\}, \quad \overline{P}_{12} = \{-(p_i^{(h)} + p_j^{(h+1)})^{-2}\},$$

$$\overline{P}_{22} = \{2(p_i^{(h)} + p_j^{(h+1)})^{-3}\}, \quad \overline{R} = \operatorname{diag}\{r_i^{(h+1)}\}, \quad i,j = 1,2,\ldots,\rho,$$

and

$$\frac{\partial\Psi_{\mathrm{I}}}{\partial\mathbf{p}^{(h)}} = \begin{bmatrix} \overline{B}_1 \\ \overline{B}_2 \end{bmatrix}$$

with

$$\overline{B}_1 = \operatorname{diag}\left\{\frac{\partial}{\partial p_i^{(h)}}[f(p_i^{(h)}) - g^{(h+1)}(p_i^{(h)})]\right\}$$

and

$$\overline{B}_2 = \operatorname{diag}\left\{\frac{\partial}{\partial p_i^{(h)}}[f'(p_i^{(h)}) - g'^{(h+1)}(p_i^{(h)})]\right\}, \quad i = 1,2,\ldots,\rho.$$

By dropping the bars from the symbols that refer to the situation in which $\mathbf{p}^{(h+1)} = \mathbf{p}^{(h)} = \mathbf{p}$ and $\mathbf{r}^{(h+1)} = \mathbf{r}^{(h)} = \mathbf{r}$ (stationary point), and taking into account that $B_1 = O$ and

$$\frac{\partial \Psi_{\mathrm{I}}}{\partial \mathbf{p}} := \left.\frac{\partial \Psi_{\mathrm{I}}}{\partial \mathbf{p}^{(h)}}\right|_{\mathbf{p}^{(h)}=\mathbf{p}} = \begin{bmatrix} O \\ B_2 \end{bmatrix},$$

after simple manipulations, from the above equations we obtain

$$\frac{\partial \Psi_{\mathrm{E2}}}{\partial \mathbf{p}} := \left.\frac{\partial \Psi_{\mathrm{E2}}}{\partial \mathbf{p}^{(h)}}\right|_{\mathbf{p}^{(h)}=\mathbf{p}} = R^{-1}(P_{22} - P_{12}P_{11}^{-1}P_{12})^{-1}B_2. \tag{30}$$

Let us note that:

(i) matrix R is invertible under the assumption that all residues r_i are different from zero, i.e., the order of $g(s)$ is precisely ρ, and

(ii) the invertibility of P_{11} and $P_{22} - P_{12}P_{11}^{-1}P_{12}$ descends from the fact that the $2\rho \times 2\rho$ matrix P in (25) is the positive definite matrix of the quadratic form:

$$\int_0^\infty \left[\sum_{i=1}^{\rho} x_i \mathrm{e}^{-p_i t} - t\left(\sum_{i=1}^{\rho} x_{\rho+i} \mathrm{e}^{-p_i t}\right)\right]^2 \mathrm{d}t, \quad p_i > 0.$$

As is known, an "equilibrium" point $\mathbf{p}$ is repelling for the considered algorithm if at least one eigenvalue of (30) has magnitude greater than 1, and it is attracting if all its eigenvalues, which remain unchanged under the similarity transformation:

$$R\frac{\partial \Psi_{\mathrm{E2}}}{\partial \mathbf{p}}R^{-1} = (P_{22} - P_{12}P_{11}^{-1}P_{12})^{-1}B_2R^{-1}, \tag{31}$$

have magnitude less than 1.

According to a classic result of matrix analysis, the positive or negative (semi)-definite character of the block symmetric matrix $\overline{H}(\mathbf{r}, \mathbf{p})$ in (25) depends both on that of P_{11}, which is positive definite, and on that of its Schur complement:

$$-\Sigma = P_{22} - B_2R^{-1} - P_{12}P_{11}^{-1}P_{12}. \tag{32}$$

Now, matrix Σ in (32) is obtained for the value 1 of parameter λ from the pencil:

$$K - \lambda S,$$

where

$$K = B_2R^{-1}, \quad S = P_{22} - P_{12}P_{11}^{-1}P_{12}.$$

Since this pencil is regular because S is positive definite, there exists a (nonsingular) transformation [6, pp. 310–314]:

$$\mathbf{y} = Z\mathbf{z},$$

by which the quadratic form:

$$\mathbf{y}^{\mathrm{T}}(K - \lambda S)\mathbf{y}$$

is transformed into

$$\mathbf{z}^{\mathrm{T}}(\Lambda - \lambda I)\mathbf{z},$$

where I is the identity matrix of order ρ and Λ is the $\rho \times \rho$ (real) *diagonal* matrix formed from the solutions λ_i of

$$\det(K - \lambda S) = 0$$

(characteristic values of the pencil).

At a saddle point of J, which may correspond to either a saddle point or a maximum of $\hat{J}$, matrix Σ is neither positive nor negative semidefinite. Therefore, at least one eigenvalue of $\Lambda - I$ is positive or, equivalently, one characteristic value λ_i is greater than 1.

Since $\det(K - \lambda S) = 0$ if and only if

$$\det(S^{-1}K - \lambda I) = 0$$

(recall that $\det S \neq 0$), at least one eigenvalue of (31) or (30) is greater than 1, which in turn implies that the maxima and saddle points of $\hat{J}$ are (generically) repelling.

According to a similar argument, it is possible to see that at a point of minimum every eigenvalue of (30) is less than 1. Of course, this is not enough to ensure convergence and, in fact, it is possible to design examples where some eigenvalues are less than -1. In the many cases we have considered, however, this has seldom occurred and corresponded to very flat local minima, which means that many reduced models are characterized by almost the same value of the index and perform equally well from the practical point of view. This typically occurs when one wants to approximate an original second-order system exhibiting an underdamped oscillatory mode by means of a first-order model.

Let us finally make a remark about the stability of the iterates. It may happen that an unstable $g(s)$ satisfies the interpolation conditions (5) and (6) (that are no longer necessary for optimality in this case). If this $g(s)$ is revealed by the iterative procedure, it must be discarded. On the other hand, the Hurwitz property of all the iterates $c^{(h)}(s)$ is not necessary for the algorithm to converge to a Hurwitz $c(s)$, even if for continuity arguments there exists a neighbourhood of every attracting (stable) minimum from which this is reached through Hurwitz iterates only.

A2. *Invertibility of the coefficient matrix and its submatrices*

By eliminating the auxiliary variables $\mathbf{q}_{1k}^{(h+1)}$ and $\mathbf{q}_2^{(h+1)}$, the linear set of equations (12) of section 3 has been reduced to equations (17) and (21) which can be represented in compact form as:

$$\Xi \bar{\mathbf{x}} = \mathbf{t}, \tag{33}$$

where

$$\Xi = \begin{bmatrix} \Gamma & 0 & \dots & 0 & \Theta_1 \\ 0 & \Gamma & \dots & 0 & \Theta_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & \Gamma & \Theta_\mu \\ \Pi_1 & \Pi_2 & \dots & \Pi_\mu & \Phi \end{bmatrix} \in \mathbb{R}^{(\mu+1)\rho\times(\mu+1)\rho}, \tag{34}$$

$$\bar{\mathbf{x}} = [\mathbf{m}_1^{(h+1)}, \mathbf{m}_2^{(h+1)}, \dots, \mathbf{m}_\mu^{(h+1)}, \mathbf{c}^{(h+1)}]^{\mathrm{T}} \in \mathbb{R}^{(\mu+1)\rho},$$

$$\mathbf{t} = [\mathbf{t}_1^{\mathrm{T}}, \mathbf{t}_2^{\mathrm{T}}, \dots, \mathbf{t}_{\mu+1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{(\mu+1)\rho}.$$

In the previous section A1, it has been shown that the Jacobian (29) is nonsingular in the neighbourhood of the considered critical points of order ρ, so that the implicit function theorem ensures the existence and uniqueness of the function supplying the values of the poles and residues in such neighbourhood. Since there is a one-to-one correspondence between this parametrization and the set of numerator and denominator coefficients (entries of $\bar{\mathbf{x}}$), equation (33) also admits a unique solution and, consequently, the coefficient matrix Ξ is nonsingular.

In order to use the simplified version of the algorithm described in section 3, submatrix Γ should be nonsingular too. This is only generically true. However, the invertibility of Γ is guaranteed when $\rho < \mu = m_i \cdot m_o$. In fact, if the rank of Γ were equal to $\rho - 1$, the rank of Ξ would be at most $\sigma = \mu(\rho - 1) + 2\rho$ because of the structure of (34). Now, σ cannot match the actual (full) rank of Ξ, i.e., $(\mu + 1)\rho$, for $\rho < \mu$; it follows that in this case Γ must be nonsingular. For higher values of ρ, Γ can be singular only for a set of zero measure in its parameter space.

References

[1] P.R. Aigrain and A.M. Williams, Synthesis of n-reactance networks for desired transient response, J. Appl. Phys. 20 (1949) 597–600.

[2] K.J. Åström, *Introduction to Stochastic Control Theory* (Academic Press, New York, 1970).

[3] L. Baratchart, Recent and new results in rational L_2 approximation, in: *Modelling, Robustness and Sensitivity Reduction in Control Systems*, ed. R.F. Curtain (Springer, Berlin, 1987) pp. 119–126.

[4] L. Baratchart, M. Cardelli and M. Olivi, Identification and rational L_2 approximation: a gradient algorithm, Automatica 27 (1991) 413–418.

[5] A.E. Bryson and A. Carrier, Second-order algorithm for optimal model order reduction, J. Guidance Control Dynam. 13 (1990) 887–892.

[6] F.R. Gantmacher, *The Theory of Matrices*, Vol. 1 (Chelsea, New York, 1977).

[7] W. Gawronski and J.N. Juang, Model reduction for flexible structures, Control Dyn. Syst. 36 (1990) 143–222.

[8] D.C. Hyland and D.S. Bernstein, The optimal projection equations for model reduction and the relationships among the methods of Wilson, Skelton and Moore, IEEE Trans. Auto. Contr. AC-30 (1985) 1201–1211.

[9] W. Krajewski, A. Lepschy, G.A. Mian and U. Viaro, On model reduction by L_2-optimal pole retention, J. Franklin Inst. 327 (1990) 61–70.

[10] W. Krajewski, A. Lepschy and U. Viaro, Optimality conditions in multivariable L_2 model reduction, J. Franklin Inst. 330 (1993) 431–439.

[11] C.P. Kwong, Optimal chained aggregation for reduced order modelling, Int. J. Control 35 (1982) 965–982.

[12] L. Meier and D.G. Luenberger, Approximation of linear constant systems, IEEE Trans. Auto. Contr. AC-12 (1967) 585–587.

[13] R.N. Mishra and D.A. Wilson, A new algorithm for optimal reduction of multivariable systems, Int. J. Control 31 (1980) 443–466.

[14] B.C. Moore, Principal component analysis in linear systems: controllability, observability, and model reduction, IEEE Trans. Auto. Contr. AC-26 (1981) 17–32.

[15] S. Mukherjee and R.N. Mishra, Reduced order modelling of linear multivariable systems using an error minimization technique. J. Franklin Inst. 325 (1988) 235–245.

[16] M. Olivi and S. Steer, Approximation rationelle en norme L^2 des systèmes dynamiques, APII 24 (1990) 481–510.

[17] E. Rosencher, Approximation rationelle des filtres à un ou deux indices: une approche Hilbertienne, Thèse de docteur-ingénieur, Univ. Paris IX-Dauphine (1978).

[18] G. Ruckebush, Sur l'approximation rationelle des filtres, Rapport n. 35, CMA Ecole Polytechnique, Paris (1978).

[19] Y. Shamash, Linear system reduction using Padé approximation to allow retention of dominant modes, Int. J. Control 21 (1975) 257–272.

[20] J.T. Spanos, M.H. Milman and D.L. Mingori, A new algorithm for L_2 optimal model reduction, Automatica 28 (1992) 897–909.

[21] J.L. Walsh, *Interpolation and Approximation by Rational Functions in the Complex Domain* (AMS, Providence, RI, 1935).

[22] D.A. Wilson, Optimum solution of model-reduction problem, Proc. IEE 117 (1970) 1161–1165.

[23] D.A. Wilson, Model reduction for multivariable systems, Int. J. Control 20 (1974) 57–64.

[24] D. Žigić, L.T. Watson, E.G. Collins, Jr. and D.S. Bernstein, Homotopy methods for solving the optimal projection equations for the H_2 reduced order model problem, Int. J. Control 56 (1992) 173–191.