INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL, VOL. 7, 113–125 (1997)

MULTIDIMENSIONAL REALIZATION OF LARGE SCALE UNCERTAIN SYSTEMS FOR MULTIVARIABLE STABILITY MARGIN COMPUTATION

EVAN L. RUSSELL, CHRISTOPHER P. H. POWER AND RICHARD D. BRAATZ

[1]*Department of Chemical Engineering, University of Illinois, 600 South Mathews Avenue, Box C-3, Urbana, IL 61801-3792, U.S.A.*

SUMMARY

The prevailing framework for robust stability and performance analysis requires that the uncertain system be written as a linear fractional transformation of the uncertain parameters. This problem is algebraically equivalent to the problem of deriving the state space realization for a multidimensional transfer function matrix, for which a systematic algorithm was recently provided by Cheng and DeMoor.[1] In this work an algorithm is developed that reduces the dimension of the realizations while improving numerical accuracy, reducing computational expense, and reducing run-time memory requirements. Such improvements are required for the realization of large scale uncertain systems, which have large numbers of inputs, outputs, states, and/or uncertain parameters.

International Journal of Robust and Nonlinear Control, vol. 7, 113–125 (1997).
(No of Figures: 1 No of Tables: 3 No of Refs: 33)

Key words: multidimensional realization; multiparameter margin; gain scheduling; robust stability

1. INTRODUCTION

The prevailing framework for robust stability and performance analysis requires that the uncertain system be written as a linear fractional transformation of the uncertain parameters p, as defined by the leftmost block diagram in Figure 1.[2,3] The uncertain parameters are located on the main diagonal of the diagonal matrix $\Delta(\mathrm{p})$, with $M(s)$ being a transfer function matrix which is not a function of p. Other spatial or temporal operators, such as the Laplace transform variable s^{-1}, the temporal delay operator z, and spatial delay operators z_1 and z_2, can be treated algebraically like uncertain parameters. The middle and rightmost block diagrams show how this is done for the Laplace transform variable, where A,B,C and D are the state space matrices for the transfer function $M(s)$. Numerically efficient synthesis of robust controllers requires that this linear fractional transformation be a function of the state space matrices of the nominal open-loop system. This is true whether the controllers to be designed are fixed[4–6] or gain scheduled.[7,8]

Lu, Zhou and Doyle[9], and Cheng and DeMoor[1] have noted that deriving such a linear fractional transformation is algebraically equivalent to deriving the state space realization for a multidimensional transfer function matrix, a problem which has been studied for many years.[10,11] Though the solution for one-dimensional realization is described in undergraduate control textbooks[12] and the solutions for other special cases of the multidimensional realization problem have been derived,[13–15,6] the first systematic algorithm to solve this problem when the entries of the transfer

CCC 1049–8923/97/020113–13

Received 10 October 1995
Revised 13 April 1995

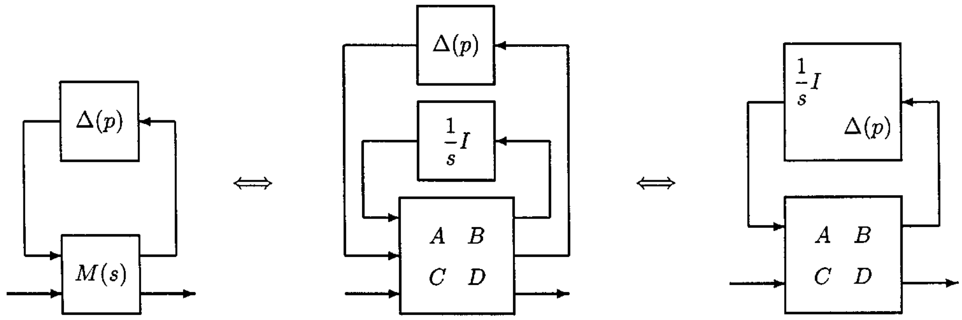

Fig. 1. Linear fractional transformations and multidimensional realizattions

function matrix are rational functions appears to be that developed by Cheng and DeMoor[1]. Though the resulting linear fractional transformations are usually not of minimal state dimension, this is not particularly a concern for the frequency-by-frequency robust stability and performance analysis[2,15] of linear time invariant uncertain systems with small numbers of inputs, outputs and parameters.

Our particular interest in multidimensional realizations stems from our interest in the application of robustness analysis to large scale processes, that is, processes which have large numbers of inputs, outputs, states and uncertain parameters. Numerical examples show that improvements to Cheng and DeMoor's algorithm[1] are required for the realization of large scale uncertain systems. In this work an algorithm is developed that reduces the dimension of the realizations while improving numerical accuracy, reducing computational expense, and reducing run-time memory requirements. Numerical examples illustrate the extent and nature of the improvements.

We would like to note that the interesting recent work by Beck and Doyle[16,17] on the calculation of minimal state space realizations does not preclude the algorithm presented in this paper, as their algorithm *reduces* the dimension of a *given realization*. Our algorithm can be used to provide input data for use by their algorithm,[16,17] with the end result being a minimal realization (in this case, 'minimal' refers to a realization which has the lowest dimension for each of the uncertain parameters; this is in contrast to the definition of 'minimal' typically used in the multidimensional realization literature[10,11]).

2. BACKGROUND

This section provides necessary definitions and poses the multidimensional realization problem in mathematical terms.

Multidimensional realization is the problem of determining the state space model for a system transfer function with multiple parameters.[10] The problem can be cast in the linear fractional transformation (LFT) framework. The LFT of $\Delta(p)$ with the coefficient matrix L is defined by the following expression[13]

$$F_u(L, \Delta(p)) = L_{22} + L_{21}\Delta(p)[I - L_{11}\Delta(p)]^{-1}L_{12} \tag{1}$$

where $F_u(L, \Delta(p))$ has dimension $n \times m$; L_{11}, L_{12}, L_{21} and L_{22} are complex or transfer function matrices with dimensions $r \times r$, $r \times m$, $n \times r$ and $n \times m$ respectively; and $\Delta(p)$ has the diagonal structure

$$\Delta(p) = \text{blockdiag}(p_1 I_{r_1}, \ldots, p_q I_{r_q}) \tag{2}$$

with $p = [p_1, \ldots, p_q]$ being a vector of uncertain parameters. The block structure vector (bs) is defined by

$$bs = [r_1, \ldots, r_q] \tag{3}$$

where r_i is the number of times p_i appears in $\Delta(p)$. Note that the dimensions of the L_{ij} are such that

$$L = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \tag{4}$$

is well-defined.

The leftmost diagram in Figure 1 is the block diagram representation for the LFT of $\Delta(p)$ with the coefficient matrix being a transfer function matrix $M(s)$, with s being the Laplace transform variable. Numerically efficient robust control synthesis requires knowledge of the state space matrices for $M(s)$, whether the controllers to be designed are fixed[4–6] or gain scheduled.[7,8] Since the transfer function for a state space realization (A, B, C, D) of $M(s)$ is given by $D + C(sI - A)^{-1}B = D + C(s^{-1}I)[I - A(s^{-1}I)]^{-1}B$[12], comparison with (1) shows that $M(s)$ is an LFT of $s^{-1}I$ with respect to the coefficient matrix

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{5}$$

This implies that s^{-1} can be treated algebraically like an uncertain parameter, as shown in the middle and rightmost block diagrams in Figure 1. Thus, the goal of multidimensional realization can be defined as determining a constant coefficient matrix L, and a diagonal matrix $\Delta(p)$ which contains all temporal and/or spatial operators and all the uncertain parameters, such that

$$M(p) = F_u(L, \Delta(p)) \tag{6}$$

is the given multidimensional rational transfer function matrix.

We will assume that $M(p)$ is well-posed, that is, that every element of $M(p)$ is bounded for $p = 0$. Then the rational matrix $M(p)$ can be written in the form

$$M(p) = \sum_{i=1}^{d} f_i(p) M_i \tag{7}$$

where M_i is a constant matrix and $f_i(p)$ is a rational function of the form

$$f_i(p) = \frac{\prod_{j=1}^{m} y_{ij}(p)}{\prod_{k=1}^{n} z_{ik}(p)}; \quad \prod_{k=1}^{n} z_{ik}(0) \neq 0 \tag{8}$$

where $y_{ij}(p)$ and $z_{ik}(p)$ are polynomials.

3. MULTIDIMENSIONAL REALIZATION ALGORITHM

The multidimensional realization algorithm is summarized in Appendix A. For brevity, we will discuss only the major features of the algorithm here.

The first step of the algorithm is to use matrices from LR decompositions of the M_i to construct the LFT of the rational matrix $M(p)$ in terms of the scalar transfer functions $f_i(p)$

$$M(p) = F_u(L_a, \Delta(f)) \tag{9}$$

where L_a is a constant matrix and

$$\Delta(f) = \text{blockdiag}(f_1(p)I_{a_1}, \ldots, f_d(p)I_{a_d}) \tag{10}$$

This step is similar to Cheng and DeMoor's step,[1] but it is much more efficient for most multi-dimensional realization problems. The role of Steps 3–7 is to calculate the LFTs for all $f_i(p)$ in terms of p. Step 8 collects these LFTs together to determine the LFT of $\Delta(f)$ in terms of the parameters p_j

$$\Delta(f) = F_u(L_b, \Delta(p)) \tag{11}$$

Step 9 combines (9) and (11) to obtain an overall LFT in the form (6). Step 10 successively employs minimal one-dimensional realization theory to each parameter p_j to arrive at the final realization.

The steps devised by Cheng and DeMoor[1] are indicated in Appendix A by the absence of an asterisk ($*$) by the step number. The modifications given by Steps 2–5 exploit structure between and within the transfer functions to provide realizations of lower order than provided by the original algorithm of Cheng and DeMoor[1]. Step 2 reduces the order of the realization when there is a term common to all the transfer function elements in a row. Letting n be the number of times a common term appears in a row, the block structure produced using the new algorithm does *not* grow with n while the block structure produced using Cheng and DeMoor's algorithm[1] grows as n. This step is motivated by problems in mechanical engineering, where the common term is a spring constant, mass, or similar term.[18] Step 3 applies to transfer function elements which contain a first order Padé time delay approximation, and reduces the order by replacing the realization obtained using Cheng and DeMoor's algorithm[1] with a minimal order realization similar to that given by Lundström.[19] Letting n correspond to the order of Pade's approximation, the block structure produced using the new algorithm grows as n while the block structure produced using Cheng and DeMoor's algorithm[1] grows as $(n^3/3) + n^2 + n$. This step is motivated by problems in chemical and mechanical engineering, where time delays are common.[20–23] Step 4 applies one-dimensional realization theory to factors that appear in the numerator and denominator which contain only one parameter. A comparison of the dimensional growth as the order of the problem size increases is illustrated in Example 1 from Table 2. Step 5 exploits the structure in which temporal or spatial operators typically appear in transfer functions. Letting the function to be realized be $f(p) = \alpha_n p^n + \alpha_{n-1} p^{n-1} + \cdots + \alpha_1 p + \alpha_0$, the block structure produced using the new algorithm grows as n while the block structure produced using Cheng and DeMoor's algorithm[1] grows as $n^2 + n$.

The new algorithm has been written in MATLAB code (available via world wide web[24]), which is just a step-by-step implementation of the algorithm given in Appendix A.

4. NUMERICAL EXAMPLES

The performance of Cheng and DeMoor's algorithm[1] is compared with our new algorithm, both with and without performing the one-dimensional reduction procedure (Step 10 of the algorithm, Appendix A), for several realization problems obtained from the open literature in Table 1. The corresponding block structure vectors are given in Table 2 with the coefficient matrices listed

Table 1. List of the examples

Example	Source reference	Transfer function $M(p)$	Parameter vector (p)
1	27	$\dfrac{\alpha_n p^n + \alpha_{n-1} p^{n-1} + \cdots + \alpha_1 p + \alpha_0}{\beta_m p^m + \beta_{m-1} p^{m-1} + \cdots + \beta_1 p + 1}$	p
2	15	$\begin{bmatrix} 0 & 1 & 0 \\ \dfrac{-(k_0 + w_k \delta_k)}{m_0 + w_m \delta_m} & \dfrac{-(c_0 + w_c \delta_c)}{m_0 + w_m \delta_m} & \dfrac{1}{m_0 + w_m \delta_m} \\ 1 & 0 & 0 \end{bmatrix}$	$[\delta_k, \delta_c, \delta_m]$
3	1	$\dfrac{(p_1 + 1)(p_2 + 1)/6}{s^2[s^4 + \left(\frac{5}{6} p_1 + \frac{3}{2} p_2 + \frac{7}{3}\right) s^2 + (p_1 + 1)(p_2 + 1)]}$	$[p_1, p_2]$
4	1	$\begin{bmatrix} 1 + \dfrac{p_2}{1 + 2p_1} & \dfrac{5 p_1 p_2}{1 + 6 p_1 p_2} \\ \dfrac{5 p_1 p_2}{1 + 6 p_1 p_2} & 1 + \dfrac{3p_1}{1 + 4p_2} \end{bmatrix}$	$[p_1, p_2]$
5	28	$\begin{bmatrix} \frac{1}{2}(\beta^2 + 2\alpha\beta\delta + \alpha^2\delta^2) & \beta + 2 + \alpha\delta \\ 1 & 0 \end{bmatrix}$	δ
6	29	$\begin{bmatrix} \dfrac{z^{-1} - s^{-1} z^{-1}}{1 + s^{-1}} & \dfrac{z^{-1} + s^{-1}}{1 - s^{-1} + z^{-1} - s^{-1} z^{-1}} \\ \dfrac{z^{-1} - s^{-1}}{1 + s^{-1} - z^{-1} - s^{-1} z^{-1}} & \dfrac{s^{-1} + s^{-1} z^{-1}}{1 + 2s^{-1} - z^{-1} - 2s^{-1} z^{-1}} \end{bmatrix}$	$[s^{-1}, z^{-1}]$
7	30	$\dfrac{2w^{-1} - z^{-1} + 1}{(1 - w^{-1})(1 + 3z^{-1})(1 + 2w^{-1} z^{-1})}$	$[w^{-1}, z^{-1}]$
8	31	$\dfrac{s^{-1} z^{-1} + s^{-2} z^{-2}}{1 + s^{-2} z^{-2}}$	$[s^{-1}, z^{-1}]$
9	32	$\dfrac{2z_1^{-2} z_2^{-1} + 6z_1^{-2} + 3z_1^{-1} + 6z_1^{-1} z_2^{-1} + z_2^{-1}}{2z_1^{-2} z_2^{-1} + 6z_1^{-2} + 6z_1^{-1} + 7z_1^{-1} z_2^{-1} + 4z_2^{-1} + 1}$	$[z_1^{-1}, z_2^{-1}]$
10	32	$\dfrac{-2z_1^{-2} z_2^{-1} + 6z_1^{-2} + 3z_1^{-1} + 6z_1^{-1} z_2^{-1} + z_2^{-1}}{-2z_1^{-2} z_2^{-1} + 6z_1^{-1} + 6z_1^{-1} + 5z_1^{-1} z_2^{-1} + 4z_2^{-1} + 1}$	$[z_1^{-1}, z_2^{-1}]$

in Appendix B. For all of these examples except Example 4, the dimension of the realization (dimension of the $\Delta(p)$ matrix) with the new algorithm before the one-dimensional reduction step (Column 3) is less than Cheng and DeMoor's algorithm[1] (Column 2). By comparing the total number of flops for the two algorithms (Columns 4 and 5), it is clear that reducing the order of the realization before the one-dimensional reduction step plays a key role in reducing the computational expense. In each case the number of flops is less than or comparable to Cheng and DeMoor's algorithm.[1] For Example 7, the number of flops with Cheng and DeMoor's algorithm[1] is over an order of magntiude larger than the number of flops used by the new algorithm.

Since reducing the order of the realization before the one-dimensional reduction step plays a key role in reducing the computational expense, it is reasonable to examine performing the one-dimensional reduction procedure at intermediate points in the algorithm (i.e., after Step 6 and Step 7) to reduce the computational expense. Correspondingly, we have developed a generalization[25] of Fan and Tits' model reduction scheme[33] to be used in these intermediate steps. These new schemes applied to Cheng and DeMoor's algorithm[1] are compared with Cheng and DeMoor's and the new algorithm using the problems described in the following.

Table 2. Block structure vectors and computational expense for the various algorithms

Example	C/D without 1-D reduction	New without 1-D reduction	C/D with 1-D reduction Number of flops	New with 1-D reduction Number of flops
1	$\frac{1}{2}(n^2+n+m^2+m)$	$\max(m,n)$	?	$\max(m,n)$
2	[1, 1, 6]	[1, 1, 1]	[1, 1, 1]	[1, 1, 1]
			19 980	6 036
3	[4, 4]	[2, 3]	[2, 1]	[2, 1]
			10 587	8 219
4	[6, 6]	[6, 6]	[4, 4]	[4, 4]
			71 101	70 641
5	4	3	2	2
			4 063	5 221
6	[12, 11]	[8, 10]	[6, 4]	[6, 4]
			244 908	185 394
7	[14, 14]	[3, 3]	[3, 2]	[3, 2]
			291 700	14 470
8	[5, 5]	[4, 4]	[3, 2]	[4, 3]
			24 242	25 346
9	[12, 6]	[4, 6]	[2, 1]	[2, 1]
			90 231	43 010
10	[11, 6]	[4, 5]	[4, 1]	[4, 2]
			75 082	35 964

Now we will investigate the performance of the algorithms on uncertain systems of larger dimensionality with varying sparseness. Let us consider the $n \times n$ banded transfer function matrix $M(p)^{mn}$ with a band of size m and the i, jth element given by

$$[M(p)]_{ij}^{mn} = \begin{cases} \dfrac{k_{ij}(-\frac{1}{2}\theta_{ij}s+1)}{(\tau_{ij}s+1)(\frac{1}{2}\theta_{ij}s+1)} & \text{for } |i-j| \leqslant m \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where $k_{ij} = k_{ij}^{o} + w_{k_{ij}}\delta_{k_{ij}}, \tau_{ij} = \tau_{ij}^{o} + w_{\tau_{ij}}\delta_{\tau_{ij}},\ \theta_{ij} = \theta_{ij}^{o} + w_{\theta_{ij}}\delta_{\theta_{ij}}, \tau_{ij}^{o} = \theta_{ij}^{o} = 1, w_{k_{ij}} = w_{\tau_{ij}} = w_{\theta_{ij}} = 0.1,\ \forall i, j$, and for the appropriate non-zero elements in the banded matrix,

$$k_{ij}^{o} = \begin{cases} 1 & \text{for } |i-j| \leqslant 0 \\ 0{\cdot}9 & \text{for } |i-j| \leqslant 1 \\ 0{\cdot}7 & \text{for } |i-j| \leqslant 2 \\ 0{\cdot}8 & \text{for } |i-j| \leqslant 3 \\ 1 & \text{for } |i-j| \leqslant 4 \end{cases} \tag{13}$$

The structure of $M(p)^{mn}$ is typical for modelling adhesive coaters and polymer extruders[20,23]. The parameter vector p is given by $p = [s^{-1}, \delta_{k_{11}}, \delta_{\tau_{11}}, \delta_{\theta_{11}}, \ldots, \delta_{k_{n,(n-m)}}, \delta_{\tau_{n,(n-m)}}, \delta_{\theta_{n,(n-m)}}]$.

Table 3 contains the computational results for these banded transfer function matrices with a variety of dimensions n and band sizes m. It is clear that by considering these results (number of flops, error estimates and realization dimension) the order of preference from most to least is: 1z) new algorithm, 2) Cheng and DeMoor's algorithm[1] with intermediate one-dimensional reduction steps, 3) Cheng and DeMoor's algorithm[1] with intermediate generalized Fan and Tits' reduction steps,[25] and 4) Cheng and DeMoor's algorithm.[1] Note that 3) and 4) contain comparable errors.

Table 3. Block structure vectors and computational results for the various algorithms

Band (m)	Dimension (n)		C/D 1-D	C/D FT/1-D	C/D 1-D/1-D	New 1-D
2	5	bs wo/1-D	$[247, 19 \times (3, 3, 5)]$	$[87, 19 \times (2, 2, 5)]$	$[76, 19 \times (2, 2, 1)]$	$[57, 19 \times (1, 1, 1)]$
		bs w/1-D	$[48, 19 \times (3_2, 2, 1)]$	$[48, 19 \times (2, 2, 1)]$	$[48, 19 \times (2, 2, 1)]$	$[43, 19 \times (1, 1, 1)]$
		Number of flops	9.48E+08	2.26E+08	8.49E+08	3.15E+07
		Error	2.29E−15	4.56E−15	1.57E−15	5.73E−16
2	6	bs wo/1-D	$[312, 24 \times (3, 3, 5)]$	$[110, 24 \times (2, 2, 5)]$	$[96, 24 \times (2, 2, 1)]$	$[72, 24 \times (1, 1, 1)]$
		bs w/1-D	$[60, 24 \times (3_2, 2, 1)]$	$[60, 24 \times (2, 2, 1)]$	$[60, 24 \times (2, 2, 1)]$	$[54, 24 \times (1, 1, 1)]$
		Number of flops	2.02E+09	4.47E+08	1.68E+08	6.30E+07
		Error	3.18E−15	5.30E−15	2.30E−15	8.60E−16
2	9	bs wo/1-D	No convergence	$[179, 39 \times (2, 2, 5)]$	$[156, 39 \times (2, 2, 1)]$	$[117, 39 \times (1, 1, 1)]$
		bs w/1-D		$[96, 39 \times (2, 2, 1)]$	$[96, 39 \times (2, 2, 1)]$	$[87, 39 \times (1, 1, 1)]$
		Number of flops		2.09E+09	7.07E+08	2.71E+08
		Error		9.54E−15	4.11E−15	1.39E−15
3	5	bs wo/1-D	$[299, 23 \times (3, 3, 5)]$	$[107, 23 \times (2, 2, 5)]$	$[92, 23 \times (2, 2, 1)]$	$[69, 23 \times (1, 1, 1)]$
		bs w/1-D	$[56, 23 \times (3_2, 2, 1)]$	$[56, 23 \times (2, 2, 1_2)]$	$[56, 23 \times (2, 2, 1)]$	$[51, 23 \times (1, 1, 1)]$
		Number of flops	1.78E+09	3.99E+08	1.53E+08	5.24E+07
		Error	2.62E−15	5.46E−15	2.82E−15	4.83E−16
3	7	bs wo/1-D	No convergence	$[173, 37 \times (2_3, 2, 5)]$	$[148, 37 \times (2, 2, 1)]$	$[111, 37 \times (1, 1, 1)]$
		bs w/1-D		$[88, 37 \times (2_3, 2, 1_2)]$	$[88, 37 \times (2, 2, 1)]$	$[81, 37 \times (1, 1, 1)]$
		Number of flops		1.71E+09	5.85E+08	2.24E+08
		Error		9.11E−15	3.01E−15	8.23E−16
4	6	bs wo/1-D	No convergence	$[160, 34 \times (2_3, 2, 5)]$	$[136, 34 \times (2, 2, 1)]$	$[102, 34 \times (1, 1, 1)]$
		bs w/1-D		$[80, 34 \times (2_3, 2, 1_2)]$	$[80, 34 \times (2, 2, 1)]$	$[74, 34 \times (1, 1, 1)]$
		Number of flops		1.53E+09	4.50E+08	1.70E+08
		Error		8.67E−15	3.54E−15	7.20E−16

2_3 = either 2 or 3 repetitions of the parameter, but usually 2.
C/D = realizations from Cheng and DeMoor's algorithm.[1]
1-D = one-dimensional reduction procedure (Step 10).
1-D/1-D = one-dimensional reduction procedure applied in the intermediate part of the algorithm.
FT/1-D = generalized Fan and Tits' reduction procedure[25] applied in the intermediate part of the algorithm.
No convergence = no convergence for the SVD during the one-dimensional reduction step.

It is important to highlight three key features of the results from Table 3. First, for the cases where the one-dimensional reduction procedure converges $[(m, n) \in \{(2, 5), (2, 6), (3, 5)\}]$, the dimension of $\Delta(p)$ matrices produced by the new algorithm (Column 7) is approximately half that produced by Cheng and DeMoor's[1] (Column 4). In fact, the realizations produced by the new algorithm are minimal for all of the parameters excluding s^{-1}. A careful look at the algorithm indicates that this will be true for all matrices with the structure of $M(p)^{mn}$ given by (12), irrespective of the values of m and n. Second, in each case Cheng and DeMoor's algorithm[1] required over an order of magnitude more flops to calculate the final realization than the new algorithm. Third, since the one-dimensional reduction step at the end of the algorithm did not converge for $[(m, n) \in \{(2, 9), (3, 7), (4, 6)\}]$, obtaining lower order intermediate realizations may be essential to using the reduction routine.

The run-time memory requirements for the four algorithms were tested on the same computer. The largest problem size that Cheng and DeMoor's algorithm[1] could realize was $m = n = 7$. Cheng and DeMoor's algorithm[1] with either intermediate reduction scheme could handle the $m = n = 9$ problem. On the other hand, the new algorithm could realize the $m = n = 10$ problem.

5. CONCLUSIONS

An algorithm which provides a multidimensional realization of reduced dimension is derived which exploits the structure of the transfer functions. For several realization problems of small dimension, both Cheng and DeMoor's[1] and our algorithm gave realizations of similar dimensions. For realization problems of larger dimension, the realizations (dimension of the $\Delta(p)$ matrix) resulting from our algorithm were of significantly lower order, were more accurate, were calculated with much less computation, and required less run-time memory.

Two ways to extend the algorithm are: 1) extend Step 4 to applying 2-D realization theory[11,26] in addition to the 1-D realization theory[12]; and 2) an additional final step, in which the multidimensional minimal realization algorithm of Beck and Doyle[16,17] is applied.

APPENDIX A. MULTIDIMENSIONAL REALIZATION ALGORITHM

The following is the multidimensional realization algorithm. The steps devised by Cheng and DeMoor[1] are indicated by the absence of an asterisk (∗) by the step number. The new steps which exploit the structure of the rational functions are indicated by an asterisk (∗) by the step number.

*Step 1**. Performing the LR decomposition on each M_i

$$M_i = L_i R_i^{\mathrm{T}} \tag{14}$$

allows us to construct matrices

$$U = [L_1, \ldots, L_d]; \quad V = [R_1, \ldots, R_d] \tag{15}$$

which satisfy

$$M(p) = U\Delta(f)V^{\mathrm{T}} \tag{16}$$

We see that $M(p)$ is an LFT in terms of the scalar transfer functions $f_i(p)$

$$M(p) = F_u(L_a, \Delta(f)) \tag{17}$$

where

$$L_a = \begin{bmatrix} 0 & V^{\mathrm{T}} \\ U & 0 \end{bmatrix}; \quad \Delta(f) = \mathrm{blockdiag}(f_1(p)I_{a_1}, \ldots, f_d(p)I_{a_d}) \tag{18}$$

The LFT's in the remaining steps are in terms of p. Steps 3 to 7 are used to calculate the LFT for all $f_i(p)$'s. All of these LFT's are combined in Step 8. The coefficient matrix of the LFT for $f_i(p)$ is represented by L_{f_i}. The vector bs_{f_i} referes to the block structure associated with $f_i(p)$, and bs with other subscripts are related similarly.

*Step 2**. If $M(p)$ contains a row with a common denominator $d(p)$, then $M(p)$ can be factored as

$$M(p) = \begin{bmatrix} I & 0 & 0 \\ 0 & \frac{1}{d(p)} & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \\ G \\ \\ \end{bmatrix}, \quad [G]_{jk} = \begin{cases} [M(p)]_{jk} & \text{for } j \neq r' \\ \prod_{j=1}^{m} y_{ij}(p) & \text{for } j = r' \end{cases} \tag{19}$$

where r' is the row with the common $d(p)$. Note that if there is more than one row with the same denominator on the same row, then the previous factorization can be performed successively for each row. If $M(p)$ contains at least one row with the same denominator on the same row, then

(i) obtain L and bs for each of the factored matrices from (19) by implementing steps 1 to 9;

(ii) determine L and bs for $M(p)$ by placing the two LFT's from step (i) in series and rearranging the block diagram;
(iii) implement step 10 on L and bs from step (ii).

*Step 3**. If $f_i(p)$ has a time delay which is approximated using the first order Padé, then $f_i(p)$ has the form $n(p)\{[(-(\theta_0 + w_\theta\delta_\theta)/2] + s^{-1})\}/\{[(\theta_0 + w_\theta\delta_\theta)/2] + s^{-1}\}$, where $n(p)$ is a rational function with the same form as $f_i(p)$ defined in (8). Then

(i) use the following LFT, where $p = [s^{-1}, \delta_\theta]$, for Padé's first order approximation from Reference 19;

$$L_{td} = \begin{bmatrix} \frac{-2}{\theta_0} & \frac{2w_\theta}{\theta_0^2} & \frac{2}{\theta_0} \\ 1 & \frac{-w_\theta}{\theta_0} & 0 \\ 2 & \frac{-2w_\theta}{\theta_0} & -1 \end{bmatrix}, \quad bs_{td} = [1, 1] \tag{20}$$

(ii) perform steps 4 to 7 to determine the LFT for $n(p)$;
(iii) place $\{[-(\theta_0 + w_\theta\delta_\theta)/2] + s^{-1}\}/\{[(\theta_0 + w_\theta\delta_\theta)/2] + s^{-1}\}$ and $n(p)$ in series and rearrange the block diagram to obtain L_{f_i};
(iv) set $bs_{f_i} = bs_n + bs_{td}$; and
(v) go to the beginning of step 3 with the next $f_i(p)$.

*Step 4**. If there exists $\prod_{j=1}^{a} y_{ij}(p)/\prod_{k=1}^{b} z_{ik}(p)$ as a factor of $f_i(p)$ in terms of only one parameter p_l, then

(i) obtain the LFT of $\prod_{j=1}^{a} y_{ij}(p_l)/\prod_{k=1}^{b} z_{ik}(p_l)$ by applying one-dimensional realization theory;[12]
(ii) repeat step (i) for each possible parameter; and
(iii) go to the beginning of step 5 with the $y_{ij}(p)$ and $z_{ik}(p)$ not used in step (i).

*Step 5**. This step factors a polynomial expression to possibly reduce the number of times each parameter appears in the expression. This step applies independently to all $y_{ij}(p)$ and $z_{ik}(p)$ not used in step 4. The factorization process is listed below:

(i) expand the expression and collect like terms;
(ii) isolate the terms in which the parameter present in the most number of terms p_j are present and factor p_j out of these terms, i.e., to give $p_j u(p) + w(p)$; and
(iii) apply step (ii) to $u(p)$ and $w(p)$ successively until complete.

Step 6. Any factored polynomial forms resulting from step 5 are trivially written in the form of a block diagram. Rearrange the block diagrams to determine the LFT expressions in terms of the parameters $(p_1, \ldots, p_q)$.

Step 7. The LFT expression for $f_i(p)$ is determined in the following manner:

(i) place all of the LFT's from step 4 obtained by one-dimensional realization theory and the LFT's for each of the $y_{ij}(p)$ from step 6 in series and obtain the LFT by rearranging the block diagram;
(ii) place all of the $z_{ik}(p)$ from step 6 in series and obtain the LFT by rearranging the block diagram;
(iii) place the LFT from step (ii) in parallel with -1, place this sum in the backward part of a feedback loop, and obtain the LFT by rearranging the block diagram; and
(iv) determine L_{f_i} and bs_{f_i} by placing the LFT's from steps (i) and (iii) in series and rearranging the block diagram.

Step 8. L_b' is constructed by placing all of the L_{f_i} as block diagonals of matrix. By reordering the variables L_b can be determined such that

$$\Delta(f) = F_u(L_b, \Delta(p)) \tag{21}$$

where

$$L_b = \begin{bmatrix} L_{b11} & L_{b12} \\ L_{b21} & L_{b22} \end{bmatrix} \tag{22}$$

and $bs = \sum_{i=1}^{d} a_i bs_{f_i}$ where a_i is the number of times $f_i(p)$ is repeated in $\Delta(f)$.

Step 9. The L in (6) is calculated by using the Redheffer Star Product,[13] where $L = S(L_b, L_a)$ is defined as

$$\begin{aligned} &S(L_b, L_a) \\ &= \begin{bmatrix} L_{b11} + L_{b12}L_{a11}(I - L_{b22}L_{a11})^{-1}L_{b21} & L_{b12}(I - L_{a11}L_{b22})^{-1}L_{a12} \\ L_{a21}(I - L_{b22}L_{a11})^{-1}L_{b21} & L_{a22} + L_{a21}L_{b22}(I - L_{a11}L_{b22})^{-1}L_{a12} \end{bmatrix} \end{aligned} \tag{23}$$

Plugging in L_a from (18) gives the simplified equation for L

$$L = \begin{bmatrix} L_{b11} & L_{b12}V^{\mathrm{T}} \\ UL_{b21} & UL_{b22}V^{\mathrm{T}} \end{bmatrix} \tag{24}$$

Step 10. Successively apply minimal one-dimensional realization theory to each parameter p_j until further reduction does not occur.

APPENDIX B. COEFFICIENT MATRICES FOR REALIZATIONS OF EXAMPLES 1–11

Example 1

See controllable and observable canonical forms in Reference 12.

Example 2

For $w_k = 1, w_c = 2, w_m = 3, k_0 = 4, c_0 = 5, m_0 = 6$,

$$L = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 \\ 0.5 & 0.5 & -0.5 & -2 & -2.5 & 0.5 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0.1667 & 0.1667 & -0.1667 & -0.6667 & -0.8333 & 0.1667 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Example 3

For $s = 1$,

$$L = \begin{bmatrix} -0.4231 & 0.2308 & 0.3997 & 0.03846 \\ 0 & 0 & 0 & 0.1667 \\ 0.03331 & 0.2443 & -0.57691 & 0.04071 \\ -0.4231 & 0.2308 & 0.3997 & 0.03846 \end{bmatrix}$$

Example 4

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -0.1730 & 0 & 8.444 & 0 & 0.5643 \\ 0 & -0.0168 & -0.1826 & 0 & -0.1157 & 0.1834 & -5.974 & 0.00376 & -3.521 & 0 \\ 0 & -0.1826 & -1.983 & 0 & 0.0107 & 1.991 & 0.5501 & 0.0408 & 0.3242 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.0168 & 0 & 0.8210 & 0 & -5.804 \\ -0.1991 & 0.0193 & -0.00178 & 2.048 & -3.999 & 0 & 0.0774 & 0 & 0 & 0 \\ 0.0132 & 0 & 0 & 0.0138 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0.00386 & 1 & -0.0917 & -0.0397 & 0.0774 & 0 & -0.00150 & 0 & 0 & 0 \\ -0.6450 & 0 & 0 & -0.6718 & 0 & 0 & 0 & 0 & 0.0205 & 0 \\ 0 & -0.0917 & -0.9958 & 0 & 0 & 0.9708 & 0 & 1.434 & 1 & 0 \\ 0.0498 & 0 & 0 & -0.5121 & 0.9724 & 0 & -1.433 & 0 & 0 & 1 \end{bmatrix}$$

Example 5

For $\alpha = 1$ and $\beta = 2$,

$$L = \begin{bmatrix} -0.1667 & 0.2357 & -0.5774 & 0.5774 \\ -0.1179 & 0.1667 & 0.8165 & 0.4082 \\ 0 & 2.449 & 2 & 4 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Example 6

$$L = \begin{bmatrix} -0.254 & 0.745 & -0.578 & 0.231 & 0.746 & 0.194 & 0.034 & -0.956 & 1.23 & -0.600 & 0 & 0 \\ 0.745 & -0.612 & 0.697 & -0.408 & 0.761 & 0.262 & -0.348 & 1.20 & 0.768 & 0.0066 & 0 & 0 \\ -0.059 & 0.196 & -0.181 & 0.018 & 0.095 & -0.900 & 0.076 & -0.183 & 0.249 & 0.765 & 0.859 & -0.240 \\ 0.285 & 0.303 & -0.098 & -0.194 & -0.027 & -0.248 & 0.368 & 0.167 & 0.412 & 0.298 & -0.291 & -1.18 \\ 0.714 & 0.447 & -0.0020 & -0.460 & -0.176 & 0.529 & 0.737 & 0.599 & 0.628 & -0.254 & 0.119 & 0.538 \\ -0.121 & 0.330 & -0.316 & 0.044 & 0.170 & -1.58 & 0.113 & -0.335 & 0.417 & 1.34 & -0.403 & 0.502 \\ 0 & 0 & -0.324 & 0.242 & -0.104 & 0.112 & 0.075 & -0.560 & -0.171 & -0.330 & -1.12 & 0.109 \\ 0 & 0 & 0.282 & -0.499 & 0.221 & -0.0079 & -0.196 & 0.862 & -0.312 & 0.0049 & -0.645 & 0.103 \\ 0 & 0 & 0.165 & 0.466 & -0.213 & -0.239 & -0.178 & -0.235 & -0.839 & 0.426 & 0.246 & 0.548 \\ 0 & 0 & 0.425 & 0.318 & -0.150 & -0.342 & -0.0038 & 0.023 & 0.357 & 0.902 & -0.522 & -0.104 \\ 0 & 0 & 0.018 & 0.165 & 1.81 & 0.452 & -0.231 & -0.012 & 1.77 & -0.573 & 0 & 0 \\ 0 & 0 & 0.085 & 0.045 & -0.359 & 2.52 & 0.105 & -0.044 & -0.449 & -2.30 & 0 & 0 \end{bmatrix}$$

Example 7

$$L = \begin{bmatrix} 1 & 0 & -1 & -0.7459 & 1.202 & -1 \\ -2 & 0 & 2 & -0.2073 & -3.458 & 2 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0.8496 & -1.119 & -0.8346 & 1.344 & -1.492 \\ 0 & 0.5274 & 1.802 & 1.344 & -2.165 & 2.403 \\ -1 & 0 & 1 & -0.1037 & -1.7209 & 1 \end{bmatrix}$$

Example 8

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -0.1426 & 1.726 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7662 & 0.6426 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -0.4544 & 0.7662 & 0 & -0.4544 & 0 & 0 & 0 \\ 0 & 0.5418 & 0.6426 & 0 & 0.5418 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1426 & -1.726 & 0 \end{bmatrix}$$

Example 9

$$L = \begin{bmatrix} -1\cdot009 & -5\cdot900 & 1\cdot288 & -0\cdot6978 \\ 0\cdot1630 & -4\cdot991 & 1\cdot232 & -0\cdot2062 \\ -0\cdot8217 & 14\cdot65 & -4 & 0\cdot4082 \\ -1\cdot119 & -10\cdot76 & 2\cdot449 & 0 \end{bmatrix}$$

Example 10

$$L = \begin{bmatrix} 0\cdot00665 & -0\cdot0393 & 0\cdot1343 & 0\cdot0185 & 0\cdot6928 & 1\cdot852 & -0\cdot0471 \\ -0\cdot0385 & 0\cdot2338 & -1\cdot394 & -6\cdot803 & 0\cdot1505 & -2\cdot383 & 0\cdot2782 \\ -0\cdot0146 & 0\cdot0866 & -0\cdot3236 & -0\cdot3536 & 0\cdot00211 & -0\cdot1246 & -0\cdot9504 \\ 0\cdot0219 & -0\cdot1178 & -0\cdot6619 & -11\cdot92 & 0\cdot0555 & -4\cdot834 & -0\cdot1307 \\ 0 & 0\cdot1626 & -0\cdot5835 & 26\cdot09 & -0\cdot2283 & 10\cdot09 & 0\cdot4683 \\ 0 & 0\cdot4601 & 0\cdot2426 & -8\cdot824 & 0\cdot0854 & -3\cdot772 & -0\cdot1751 \\ 0 & 0\cdot0132 & -1\cdot264 & -13\cdot73 & 0\cdot1219 & -5\cdot384 & 0 \end{bmatrix}$$

ACKNOWLEDGEMENTS

Yi Cheng and Bart DeMoor at the Department of Electrical Engineering, Katholieke Universiteit Leuven, Kardinaal Mercierlann 94, B-3001 Hevelee, Belgium are gratefully acknowledged for providing a copy of their code which provided the base code for our modifications. The scientific responsibility for the numerical examples; however, is assumed by the authors of this manuscript. This research was supported by the UIUC Research Board and the UIUC Office of the Vice Chancellor of Research.

REFERENCES

1. Cheng, Y. and B. DeMoor, 'A multidimensional realization algorithm for parametric uncertainty modeling and multiparameter margin problems', *Int. J. Control*, **60**, 789–807 (1994).
2. Doyle, J. C., 'Analysis of feedback systems with structured uncertainties', *IEE Proc. Part. D*, **129**, 242–250 (1982).
3. Safonov, M. G., 'Stability margins of diagonally perturbed multivariable feedback systems', *IEE Proc. Part D*, **129**, 251–256 (1982).
4. Packard, A.K. 'What's new with μ structured uncertainty in multivariable control', Ph.D. thesis, University of California Berkeley, 1988.
5. Doyle, J. C., K. Glover, P. P. Khargonekar and B. A. Francis, 'State-space solutions to standard H_2 and H_∞ control problems', *IEEE Trans. Automat. Control*, **34**, 831–847 (1989).

6. Zhou, K., J. C. Doyle and K. Glover, *Robust and Optimal Control*, Prentice-Hall, Englewood Cliffs, New Jersey, 1995.
7. Becker, G. and A. Packard, 'Robust performance of linear parametrically varying systems using parametrically-dependent linear dynamic feedback', *Systems and Control Letters*, **23**, 205–215 (1994).
8. Packard, A., 'Gain scheduling via linear fractional transformations', *Systems and Control Letters*, **22**, 79–92 (1994).
9. Lu, W.-M., K. Zhou and J. C. Doyle, 'Stabilization of linear systems under structured uncertainty', *Proc. of the Conference on Decision and Control*, Brighton, UK, 1991, pp. 1239–1244, IEEE, Piscataway, New Jersey.
10. Bose, N. K., 'Problems and progress in multidimensional system theory', *Proc. of the IEEE*, **65**, 824–840 (1977).
11. Kung, S. Y., B. Levy, M. Morf and T. Kailath, 'New results in 2-D systems theory, part II: 2-D state-space models—realization and the notations of controllability, observability, and minimality', *Proc. of the IEEE*, **65**, 945–861 (1977).
12. Kailath, T., *Linear Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1980.
13. Doyle, J. C., A. Packard and K. Zhou, 'Review of LFTs, LMIs, and μ', *Proc. of the Conference on Decision and Control*, Brighton, U.K., 1991, pp. 1227–1232, IEEE, Piscataway, New Jersey.
14. Packard, A. K., 'What's new with μ: structured uncertainty in multivariable control', Ph.D. thesis, University of California, Berkeley, 1988.
15. Packard, A. and J. C. Doyle, 'The complex structured singular value', *Automatica*, **29**, 71–109 (1993).
16. Beck, C., 'Minimality of uncertain systems and IQCs', *Proc. of the Conference on Decision and Control*, Florida, 1994, pp. 3068–3073, IEEE, Piscataway, New Jersey.
17. Beck, C. and J. Doyle, 'Realizations of uncertain systems and formal power series', *Proc. of the American Control Conference*, Seattle, Washington, 1995, pp. 1481–1485, IEEE, Piscataway, New Jersey.
18. Braatz, R. D., and M. Morari, 'Robust control for a noncolocated spring-mass system', *J. of Guidance*, *Control and Dynamics*, **15**, 1103–1110 (1992).
19. Lundstrom, P., 'Studies on robust multivariable distillation control, Ph.D. thesis, University of Trondheim, Trondheim, Norway, 1994.
20. Braatz, R. D., 'The current status of sheet and film process control', Proc. of the International Conference on Chemical Process Control V, Tahol City, California, 1996, pp. 1–4.
21. Braatz, R. D., 'Internal model Control', in *The Control Handbook*, CRC Press, Boca Raton, Florida, 1995, pp. 215–224.
22. Braatz, R. D., 'μ-sensitivities as an aid for robust identification', M.S. report, California Institute of Technology, Pasadena, California, 1991.
23. Featherstone, A. P. and R. D. Braatz, 'Control relevant identification of sheet and film processes', *Proc. of the American Control Conference*, Seattle, Washington, 1995, pp. 2692–2696.
24. Russell, E. L. and R. D. Braatz, 'MD—a code for multidimensional realization', Software package, University of Illinois, Urbana, www.cen.uiuc.edu/~braatz, 1996.
25. Russell, E. L., 'Multidimensional realization and model reduction of large scale uncertain systems', Technical report, (M.S. thesis), University of Illinois, Urbana, 1996.
26. Pinto, J. R. C. and P. E. Wellstead, 'Self-tuning filters and predictors for two-dimensional systems: parts I, II and III', *Int. J. Control*, **42**, 457–515 (1985).
27. Chen, C.-T., *Linear System Theory and Design*, Harcourt Brace College Publishers, Orlando, Florida, 1984.
28. Amann, N. and F. Allgower, 'μ-suboptimal design of a robustly performing controller for a chemical reactor', *Int. J. Control*, **59**, 665–687 (1994).
29. Haratani, N., 'Description and realization of separable-denominator two-dimensional transfer matrix', *IEICI Trans. Fundamentals*, **E75-A**, 806–812 (1992).
30. Manikopoulos, C. N., S. E. Mentzelopoulou and G. E. Antoniou, 'Minimal circuit and state space realization of three-term separable denominator 2-D filters', *Int. J. Electronics*, **71**, 723–731 (1991).
31. Gu, G., J. L. Aravena and K. Zhou, 'On minimal realization of 2-D systems', *IEEE Trans. Circuits and Systems*, **38**, 1228–1233 (1991).
32. Paraskevopoulos, P. N. and G. E. Antoniou, 'Minimal realization of two dimensional systems via a state space cyclic model', *Int. J. Electronics*, **74**, 491–511 (1993).
33. Fan, M. K. H. and A. L. Tits 'Characterization and efficient computation of the structured singular value', *IEEE Trans. on Automat. Control*, **AC-31**, 734–743 (1986).