ELSEVIER

Journal of Process Control 11 (2001) 543–552

JOURNAL OF PROCESS CONTROL

www.elsevier.com/locate/jprocont

Comparison of theoretical and computational characteristics of dimensionality reduction methods for large-scale uncertain systems

Rudiyanto Gunawan, Evan L. Russell, Richard D. Braatz *

Department of Chemical Engineering, University of Illinois at Urbana-Champaign, 600 South Mathews Avenue, Box C-3, Urbana, IL 61801, USA

Abstract

Synthesizing optimal controllers for large scale uncertain systems is a challenging computational problem. This has motivated the recent interest in developing polynomial-time algorithms for computing reduced dimension models for uncertain systems. Here we present algorithms that compute lower dimensional realizations of an uncertain system, and compare their theoretical and computational characteristics. Three polynomial-time dimensionality reduction algorithms are applied to the Shell Standard Control Problem, a continuous stirred-tank reactor (CSTR) control problem, and a large scale benchmark problem, where it is shown that the algorithms can reduce the computational effort of optimal controller synthesis by orders of magnitude. These algorithms allow robust controller synthesis and robust control structure selection to be applied to uncertain systems of increased dimensionality.

Keywords: Model reduction; Uncertain systems; Robust control

1. Introduction

The prevailing framework for robust controller synthesis requires that the uncertain system be written as a linear fractional transformation (LFT) of the uncertainties, as defined by the block diagram in Fig. 1 [16,42]. The block diagram of the uncertain system has been rearranged so that the uncertainties are located on the main diagonal of the block-diagonal matrix Δ, and $G(s)$ is a transfer function which is not a function of the uncertainties Δ. Weights are incorporated into $G(s)$ so that the norm of the perturbation matrix Δ is bounded by one.

The goal of robust control synthesis is to design an internally stabilizing controller that provides the best worst-case performance for all plants described by the uncertainty description. It has been shown that the synthesis of a robust optimal controller is *NP-hard* [9,10,45], which implies that it is highly unlikely that there exists a polynomial-time algorithm that can compute the robust optimal controller [20,33]. Except in those cases where inherent structural characteristics of a process can be exploited [18,22], the computational expense of robust controller synthesis has limited its applicability to rather simple systems. The 3-state CSTR studied by [1] provides an illustrative example of the computational difficulties in handling more realistic systems. The uncertain system representation for the reactor involved thirteen real parametric perturbations, with dimensions ranging from 1×1 to 11×11 for each perturbation. The computational expense of computing a robust optimal controller was too great to be applied directly to this system [1]. Note that most chemical processes in industry consist of more than one unit, have more than 3 states, and have more than 13 uncertain parameters.

Another issue is the quality of the robust controller computed for large scale systems using off-the-shelf software. Although off-the-shelf software packages usually compute nearly optimal robust controllers for uncertain systems of relatively low dimensionality [2,12], the controllers tend to become further from global optimality as the system dimensionality is increased [22]. Given the time an engineer has invested in modeling the process and quantifying the model uncertainties, it seems rather wasteful to use that information to design an unnecessarily conservative controller.

One strategy for computing robust controllers for processes with high dimensionality would be to develop better algorithms for robust controller design. Several researchers are currently working on this challenging problem [21,48,46]. It will be several years before algorithms suitable

* Corresponding author. Tel.: +1-217-333-5073; fax: +1-217-333-5052.

E-mail address: braatz@uiuc.edu (R.D. Braatz).

0959-1524/01/$ - see front matter
PII: S0959-1524(00)00024-X

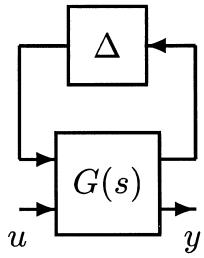

Fig. 1. Uncertain system where the input–output mapping is written as a linear fractional transformation of a block-diagonal matrix Δ.

for large scale systems will be available, even accounting for the yearly doubling of computer speeds. Another approach is to reduce the dimensionality of the uncertain system (more specifically, the number of states in $G(s)$ and the dimensionality of Δ in Fig. 1) before applying robust controller design software. While this task of dimensionality reduction has been thoroughly investigated for the case where there is no model uncertainty [23,28], there are relatively few such techniques for uncertain systems [4,15,38].

The purpose of this paper is to compare the theoretical and computational characteristics of algorithms developed for the dimensionality reduction of uncertain systems. First we summarize the framework for representing uncertain systems. Second, the dimensionality reduction algorithms are compared theoretically and computationally. Three polynomial-time algorithms are compared by application to the Shell Standard Control Problem, a CSTR control problem, and a benchmark problem taken from the literature. Some conclusions are made as to the effectiveness of the best available algorithms for application to large scale systems.

2. Background

The general framework for robustness analysis and synthesis requires that the uncertainty (the set of possible plants) be modeled as norm-bounded perturbations Δ_i on the nominal system $G(s)$ (see Fig. 1). The vector u represents all inputs to the open loop system, including manipulated, disturbance, and noise variables. The vector y represents all outputs of interest, whether measured or unmeasured, controlled or uncontrolled. This framework is sufficiently general to be used to represent processes for purposes beyond classical feedback control, such as reference prefiltering, inferential control, and feedforward control.

Through weights each perturbation is normalized so that

$$\|\Delta_i\|_\infty \equiv \sup_\omega \bar{\sigma}(\Delta_i) \leqslant 1, \tag{1}$$

where $\bar{\sigma}(\Delta_i)$ is the maximum singular value of the matrix Δ_i. The perturbation Δ_i is complex for representing unmodeled dynamics, and real for representing parametric uncertainty. Here the perturbations are assumed to be linear time invariant, although many of the dimensionality reduction algorithms considered here apply to more general classes of perturbations.

The set of real numbers will be denoted by $\mathcal{R}$, the set of complex numbers by $\mathcal{C}$, and the $r \times r$ identity matrix by I_r. The perturbations, which may occur at different locations in the system, are collected in the block-diagonal matrix Δ.

$$\Delta \equiv \left\{ \mathrm{diag}\left\{ \delta_1^r I_{r_1}, \cdots, \delta_k^r I_{r_k}, \delta_{k+1}^c I_{r_k+1}, \cdots, \delta_m^c I_{r_m}, \right.\right.$$
$$\left.\left. \Delta_{r_m+1}, \cdots, \Delta_{r_l} \right\} | \delta_i^r \in \mathcal{R}, \delta_i^c \in \mathcal{C}, \Delta_{r_i} \in \mathcal{C}^{r_i \times r_i}, \sum_{i=1}^{l} r_i = n \right\}, \tag{2}$$

and the system is arranged to match the diagram in Fig. 1. The nominal system $G(s)$ in Fig. 1 is determined by the nominal model, and the size and nature of the uncertainty. Off-the-shelf programs construct $G(s)$ and the structure of Δ directly from the system description [2,12,41].

The linear fractional transformation (LFT) is the mapping between u and y, which can be written in terms of Δ and G as (for brevity, the Laplace transform variable is suppressed)

$$F_u(G, \Delta) = G_{22} + G_{21}\Delta(I - G_{11}\Delta)^{-1}G_{12}, \tag{3}$$

where

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}. \tag{4}$$

with the horizontal and vertical lines representing the partitioning of the G matrix into submatrices.

The goal of robust controller design is to compute the controller that provides the best worst-case closed loop performance of all plants in the uncertainty set [17]. Algorithms for computing the controller that globally optimizes this objective for certain classes of uncertainties are currently under development [21,47,46].The most popular method for robust controller synthesis is commonly referred to as *DK-iteration* [17], which produces a sub-optimal controller. For the linear time invariant perturbations considered here, the number of states in the resulting controller is equal to the sum of the number of states of $G(s)$ and a complex function of the structure and dimension of Δ and the dynamics of $G(s)$ [2]. Repeated scalar perturbations cause the greatest amount of computations and the greatest increase in the number of controller states — the more times the scalar is repeated, the larger the number of controller states.

To produce lower order controllers with less computational expense and which are closer to global optimality, it is essential to reduce the dimensionality of the repeated scalar perturbations, and so most dimensionality reduction efforts have focussed primarily on perturbations of this type. Such algorithms can also aid in the computation of estimates of the best performance achievable by any controller that is also robust to model uncertainties. Such estimates are useful for robust control structure selection, which is the problem of selecting the best measured and manipulated variables to use for control purposes while taking model uncertainties into account [7,25]. Such dimensionality reduction algorithms would allow tools for robust control structure selection to be applied to processes of increased dimensionality (larger numbers of inputs, outputs, and/or states).

Dimensionality reduction algorithms for uncertain systems are also applicable to general multidimensional systems, for example, where there are multiple spatial and/or temporal transform variables. These multidimensional model representations can be used to describe many large scale processes of industrial interest. In particular, two-dimensional models can be used to describe the simultaneous machine and cross directional behavior of sheet and film processes [8,50]. These connections are described in more detail elsewhere [26,41].

3. Dimensionality reduction methods for uncertain systems

This section presents dimensionality reduction algorithms for uncertain systems.

3.1. Multidimensional SVD

A polynomial-time algorithm was developed for reducing the dimension of large scale uncertain systems using structured singular value decompositions (SVDs) to remove subspaces that have a negligible effect on the overall input-output mapping of the uncertain system [38]. This multidimensional SVD algorithm provided significant dimensionality reduction in several numerical systems as well as a large scale paper machine control system.

However, the original multidimensional SVD algorithm was applicable to matrices, not to transfer functions. Because of this, the algorithm was appropriate for reducing the computational requirements associated with robustness margin computation — not for reducing the dimensionality of LFTs used for controller synthesis. A generalization of the multidimensional SVD algorithm to LFTs was developed and applied to a series of numerical examples [39]. A strength of the generalized multidimensional SVD algorithm over many other dimensionality reduction algorithms was its ability to handle full-block uncertainties. A weakness was that it rarely outperformed the successive multidimensional realization algorithm (discussed next) when applied to repeated scalar perturbations.

3.2. Successive multidimensional realization

The state space realization of a transfer function $G(s) = C(sI - A)^{-1}B + D$ is algebraically equivalent to an LFT with the Laplace transform variable s^{-1} treated as a parameter δ_1^c in (2) [32,41]. One-dimensional state space minimal realization algorithms are described in undergraduate controls textbooks [11,23] and implemented in off-the-shelf control software [43]. Such algorithms transform a given realization into a minimal realization, where minimal refers to the state space representation of a transfer function with the smallest possible number of states (each state is represented by a $1/s$ in Fig. 2). The basic idea of the minimal realization algorithms are to remove states that are uncontrollable and/or unobservable, as these do not affect the overall input-output mapping. Implementations of one-dimensional minimal realization algorithms that can handle systems with thousands of states are currently available in off-the-shelf software packages.

Any one-dimensional minimal realization algorithm can be applied to reduce the dimension of repeated scalar perturbations as well as the states in a linear fractional transformation [24,41]. The key idea is that the algebraic problem of reducing the dimension of a repeated scalar perturbation is equivalent to the algebraic problem of reducing the number of s^{-1} terms required by a state space system to describe a given transfer function (in Fig. 2). In other words, any repeated scalar perturbations δ_i can be treated in exactly the same manner as the inverse of the Laplace transform variable (see Fig. 3). Given any LFT in terms of uncertain parameters as in (2), a transfer function can be defined by treating any repeated scalar perturbations as the inverse of the Laplace transform variable. Then, a one-dimensional minimal realization algorithm is applied to the aforementioned transfer function, reducing the dimensionality of the chosen repeated scalar perturbation. This procedure is performed for each real and complex repeated uncertain parameter, one at a time. Although the resulting lower dimension uncertain system may not be minimal [15], our experience is that the algorithm can provide substantial dimensionality reductions.

Our main implementation of the successive multidimensional realization algorithm uses the program MINREAL in the Control System Toolbox [43] as the implementation of the one-dimensional minimal realization algorithm. The program MINREAL calls the subroutines *ctrbf* and *obsvf* to transform the system into controllability and observability Kalman canonical

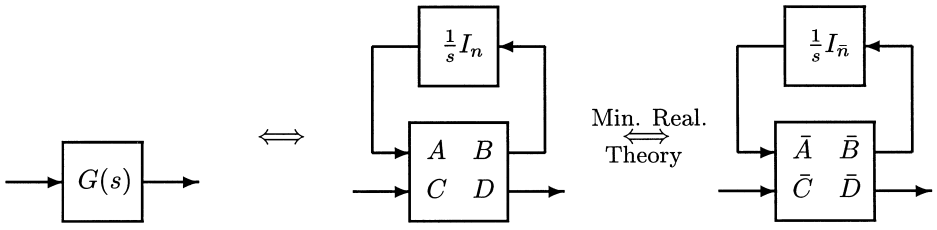

Fig. 2. Representation of a transfer function in terms of a linear fractional transformation of the inverse of the Laplace transform variable.

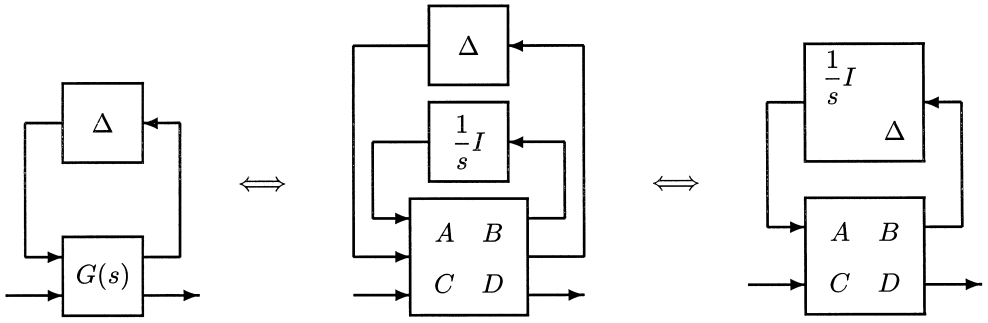

Fig. 3. The Laplace transform variable s^{-1} can be treated algebraically exactly like a repeated scalar perturbation.

forms by implementing the Staircase algorithm of [36]. The algorithm procedure is similar to the Gaussian elimination or the Hessenberg reduction [37] in which submatrices of the state space matrices A, B and C are transformed by unitary transformations into Kalman canonical form. During the procedure of the Staircase algorithm, linear dependency of rows of some matrices needs to be determined. This is done by estimating the rank of the matrix, which is the number of the singular values that are larger than a given tolerance. Once the system is in Kalman canonical form, the program MINREAL eliminates the states that are uncontrollable and/or unobservable. A MATLAB program that implements the succesive multidimensional realization algorithm can be downloaded from the authors' website [40].

There are few considerations regarding implementation that are worth discussing. In our main implementation of the successive multidimensional realization algorithm, a tolerance is used in the one-dimensional minimal realization algorithm to determine the ranks of some matrices, which determines which states are reachable and observable. Our implementation sets the tolerance as the square root of the floating point relative accuracy of MATLAB (i.e. 2.22×10^{-16}). As pointed out by Moore [28], it is possible to construct numerical examples where a nonzero tolerance used in a one-dimensional minimality algorithm can result in a large error in the input-output mapping of the reduced system. While we have not experienced such sensitivities for LFTs naturally constructed for real processes, the fact that such examples exist indicates that it is important to numerically estimate the worst-case error of the input–output mapping of the reduced system, as is done in the Examples section, to ensure that such sensitivities have not occurred. If such a sensitivity does occur in a particular problem, then it is suggested to either not reduce the dimensionality for the subsystem that is causing the sensitivity, or replace minreal with an alternative one-dimensional reduction method such as balanced truncation (e.g. implementations include BALREAL in the Control Systems Toolbox, [43] or SYSBAL in the μ-Analysis and Synthesis Toolbox [2]). Similar observations and comments hold for the multidimensional Kalman decomposition discussed next.

3.3. Multidimensional Kalman decomposition

Another approach is to define the output of Δ as a generalized state, and to perform a decomposition of the LFT in terms of reachable and observable, unreachable and observable, reachable and unobservable, and unreachable and unobservable portions of the generalized state [15]. This is a generalization of the notion of the Kalman decomposition as used in one-dimensional minimal realization theory [23]. All states that are unreachable and/or unobservable are truncated, as such states have no effect on the overall input-output mapping.

By including the perturbations and all temporal operators such as the delay operator or $1/s$ as part of a generalized Δ, multidimensional Kalman decomposition is derived from the perspective of a generalized state space equation

$$\begin{aligned} x &= \Delta(Ax + Bu) \\ y &= Cx + Du \end{aligned} \tag{5}$$

where x is the generalized state. Insert the state transformation $x = Tz$ into the generalized state space equation to give

$$\begin{aligned} z &= T^{-1}\Delta(ATz + Bu) \\ y &= CTz + Du \end{aligned} \tag{6}$$

Provided the matrix T commutes with the generalized Δ:

$$\Delta T = T\Delta, \tag{7}$$

the generalized state space equation can be written as

$$\begin{aligned} z &= \Delta\left(T^{-1}ATz + T^{-1}Bu\right) \\ y &= CTz + Du \end{aligned} \tag{8}$$

This is a direct generalization of the similarity transformation for one-dimensional systems to multidimensional systems. In the one-dimensional case where $\Delta = (1/s)I$, the matrix T is full block. In the multi-dimensional case, the generalized Δ is block-diagonal and the matrix T is block-diagonal so as to commute with Δ. Hence, for each subblock of Δ which is a repeated scalar perturbation, the corresponding subblock of T is a full matrix.

Multidimensional Kalman decomposition computes a similarity transformation that transforms the generalized state x into z which is decomposed into reachable and observable, unreachable and observable, reachable and unobservable, and unreachable and unobservable parts:

$$x = Tz = [T_{ro}\ T_{r\bar{o}}\ T_{\bar{r}o}\ T_{\bar{r}\bar{o}}]\begin{bmatrix} z_{ro} \\ z_{r\bar{o}} \\ z_{\bar{r}o} \\ z_{\bar{r}\bar{o}} \end{bmatrix} \tag{9}$$

$$\begin{aligned} z = \begin{bmatrix} z_{ro} \\ z_{r\bar{o}} \\ z_{\bar{r}o} \\ z_{\bar{r}\bar{o}} \end{bmatrix} &= \begin{bmatrix} \Delta_{ro} & & & \\ & \Delta_{r\bar{o}} & & \\ & & \Delta_{\bar{r}o} & \\ & & & \Delta_{\bar{r}\bar{o}} \end{bmatrix} \\ &\left(\begin{bmatrix} A_{11} & 0 & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ 0 & 0 & A_{33} & 0 \\ 0 & 0 & A_{43} & A_{44} \end{bmatrix}\begin{bmatrix} z_{ro} \\ z_{r\bar{o}} \\ z_{\bar{r}o} \\ z_{\bar{r}\bar{o}} \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ 0 \\ 0 \end{bmatrix}u\right) \\ y &= [C_1\ 0\ C_3\ 0]\begin{bmatrix} z_{ro} \\ z_{r\bar{o}} \\ z_{\bar{r}o} \\ z_{\bar{r}\bar{o}} \end{bmatrix} + Du. \end{aligned} \tag{10}$$

The dimensionality of Δ and the generalized state z is reduced by truncating parts of the system that are not both reachable and observable:

$$\begin{aligned} z_{ro} &= \Delta_{ro}(A_{11}z_{ro} + B_1u) \\ y &= C_1z_{ro} + Du \end{aligned} \tag{11}$$

This is a direct generalization of minimality reduction using the one dimensional Kalman decomposition.

The similarity transformation T is constructed by repeated calls to a one-dimensional Kalman decomposition algorithm. The implementation of this algorithm by [15] first decomposes the state into its reachable and unreachable parts and then into its observable and unobservable parts. The one-dimensional Kalman decompositions are implemented using the subroutines *ctrbf* and *obsvf* discussed earlier. Unlike the implementation of successive multidimensional realization which truncates the system after each application of one-dimensional minimal realization algorithm, the truncations in multidimensional Kalman decomposition are only performed twice — once after each decomposition.

In the case where the perturbations are general non commutative operators, multidimensional Kalman decomposition gives an LFT with minimal dimensionality. Although the resulting lower dimension LFT is not necessarily minimal for linear time invariant perturbations, this algorithm can still lead to very large dimensionality reductions.

Like successive multidimensional realization, the computational steps involve repeated application of a one-dimensional minimal realization algorithm. The matrices to which the one-dimensional algorithm is applied are different. It can be shown that the multidimensional Kalman decomposition and the successive multidimensional realization algorithms are not equivalent (see example in Section 4.4), and that successive multidimensional realization cannot produce a reduced system of lower dimensionality than that produced by multidimensional Kalman decomposition [14]. A MATLAB program which implements the algorithm can be downloaded from a website [13].

As mentioned in the previous section, the worst-case error in the input-output mapping should be computed to ensure that the reduced dimension model accurately captures the input–output dynamics.

3.4. Multidimensional balanced truncation

Multidimensional balanced truncation is a dimensionality reduction algorithm in which structured grammians are used to compute a balanced realization for the uncertain system, and then generalized states which have a small effect on the overall input-output mapping are truncated [4]. The structured grammians are matrices $X \geqslant 0$ and $Y \geqslant 0$ that solve

$$\begin{aligned} A^*XA - X + C^*C &\leqslant 0, \\ AYA^* - Y + BB^* &\leqslant 0, \end{aligned} \tag{12}$$

where X and Y are restricted to have a block diagonal structure so as to commute with the generalized Δ, and A, B, and C are the generalized state space matrices for the system. Computing X and Y is known as a linear matrix inequality (LMI) feasibility problem. The LMI feasibility problem is a convex problem which can be solved in polynomial-time using interior point methods [31].

Generalized Hankel singular values, which are defined to be the square roots of the eigenvalues of XY, are used to determine which states in the balanced realization can be truncated while having a small effect on the overall input–output mapping. These singular values have a direct association with the degree of reachability or observability — the smaller the singular values, the weaker the degree. In the implementation of this algorithm, the singular values that are lower than a given tolerance (that is, a certain degree of reachability or observability) are truncated. The most computationally expensive step of the algorithm is computing the structured grammians that satisfy the pair of linear matrix inequalities (12). Although computing the structured grammians is a polynomial-time calculation [5], it is much more expensive than the successive multidimensional realization and multidimensional Kalman decomposition algorithms.

There are close theoretical relationships between multidimensional balanced truncation and multidimensional Kalman decomposition [3]. In fact, Kalman-like decomposition structures can be constructed for uncertain systems using either algorithm. An advantage of multidimensional Kalman decomposition is that it can handle unstable uncertain systems directly, whereas multidimensional balanced truncation does not. A major advantage of the multidimensional balanced truncation is that it can reduce the dimensionality for systems in which the exact reduction algorithms cannot, in other words, it is a model reduction method [28]. Rigorous upper bounds have been derived on the difference between the input-output mapping of the original uncertain system and the reduced system [4,49].

A weakness of any inexact open loop dimensionality reduction algorithm is that such algorithms ignore the *control relevancy* of the reduced model, that is, the closed loop performance requirements are not used to ensure that the open loop model errors have a small effect on the achievable closed loop performance. It is well-known that some open loop model errors can have a negligible effect on the achievable closed loop performance while other model errors of similar magnitude can have a huge effect [35]. However, multidimensional balanced truncation is expected to give good closed loop results as long as only those states corresponding to very small Hankel singular values are truncated. A MATLAB program implementing multidimensional balanced truncation is available [4].

The implementation of multidimensional balanced truncation by [4] uses the program MINCX in the LMI Control Toolbox [19], which implements a potential reduction interior point method [30]. This class of interior point methods is based on explicit Lyapunov functions (i.e. the potential), where the accuracy of the solution can be estimated from the reduction of the potential during the algorithm [31]. Both our experience [44] and a detailed study by Mittelmann [27] indicates that the MINCX program usually converges, but it is much slower than most semidefinite programming codes. Readers interested in algorithms, implementations, and comparisons for LMI solvers are referred to the references [27,44].

4. Examples

The effectiveness of successive multidimensional realization (SR), multidimensional Kalman decomposition (KD), and multidimensional balanced truncation (BT) is explored using three examples based on real systems. Another example is used to show a difference between the SR and KD algorithms.

4.1. Shell standard control problem

The first example is the model for a heavy oil fractionator provided by Shell Oil [34]:

$$P=\left[\begin{array}{ccc}
\frac{(4.05+2.11\delta_1)e^{-27s}}{50s+1} & \left(\frac{(1.77+0.39\delta_2)e^{-28s}}{60s+1}\right) & \frac{(5.88+0.59\delta_3)e^{-27s}}{50s+1} \\
\frac{(5.39+3.29\delta_1)e^{-18s}}{50s+1} & \frac{(5.72+0.57\delta_2)e^{-14s}}{60s+1} & \frac{(1.20+0.12\delta_4)e^{-27s}}{45s+1} \\
\frac{(3.66+2.29\delta_1)e^{-2s}}{9s+1} & \frac{(1.65+0.35\delta_2)e^{-20s}}{30s+1} & \frac{(1.44+0.16\delta_5)e^{-27s}}{40s+1} \\
\frac{(5.92+2.34\delta_1)e^{-11s}}{12s+1} & \frac{(2.54+0.24\delta_2)e^{-12s}}{27s+1} & \frac{(6.90+0.89\delta_3)e^{-15s}}{40s+1} \\
\frac{(4.13+1.71\delta_1)e^{-5s}}{8s+1} & \frac{(2.38+0.93\delta_2)e^{-7s}}{19s+1} & \frac{(1.52+0.13\delta_4)e^{-15s}}{25s+1} \\
\frac{(4.06+2.39\delta_1)e^{-8s}}{13s+1} & \frac{(4.18+0.35\delta_2)e^{-4s}}{33s+1} & \frac{(1.83+0.13\delta_5)e^{-15s}}{20s+1} \\
\frac{(4.38+3.11\delta_1)e^{-20s}}{33s+1} & \frac{(4.42+0.73\delta_2)\mathrm{e}^{-22s}}{44s+1} & \frac{(5.53+0.67\delta_3)e^{-2s}}{40s+1} \\
 & & \frac{1.16+0.08\delta_4}{11s+1} \\
 & & \frac{1.27+0.08\delta_5}{6s+1} \\
 & & \frac{(8.10+0.32\delta_3)e^{-2s}}{20s+1} \\
 & & \frac{1.73+0.02\delta_4}{5s+1} \\
 & & \frac{1.79+0.04\delta_5}{19s+1} \\
 & & \frac{(6.23+0.30\delta_3)e^{-2s}}{10s+1} \\
 & & \frac{1.31+0.03\delta_4}{2s+1} \\
 & & \frac{1.26+0.02\delta_5}{22s+1} \\
 & & \frac{(6.53+0.72\delta_3)e^{-s}}{9s+1} \\
 & & \frac{1.19+0.08\delta_4}{19s+1} \\
 & & \frac{1.17+0.01\delta_5}{24s+1} \\
 & & \frac{7.20+1.33\delta_3}{19s+1} \\
 & & \frac{1.14+0.18\delta_4}{27s+1} \\
 & & \frac{1.26+0.18\delta_5}{32s+1}
\end{array}\right] \tag{13}$$

Each gain has uncertainty associated with it, and the uncertainties in each column are correlated.

There are many methods to construct a block diagram as shown in Fig. 1 [2,41]. An approach similar to Section 11.2.5 of [29] gives

$$G(s) = \begin{bmatrix} 0 & \tilde{R} \\ \hat{P}(s) & \hat{P}(s)\tilde{I} \end{bmatrix}, \ \Delta = \mathrm{diag}\{\delta_1 I_7, \cdots, \delta_5 I_7\}, \tag{14}$$

where

$$\hat{P}(s) = \begin{bmatrix} \tilde{P}_1 \tilde{P}_2 \cdots \tilde{P}_5 \end{bmatrix}^{7\times 35}, \tilde{P}_j = \begin{bmatrix} \tilde{P}_{1j} & & & \\ & \tilde{P}_{2j} & & \\ & & \ddots & \\ & & & \tilde{P}_{7j} \end{bmatrix}^{7\times 7} \tag{15}$$

$$\tilde{R} = \begin{bmatrix} R_1 & & & \\ & R_2 & & \\ & & \ddots & \\ & & & R_5 \end{bmatrix}^{35\times 5}, \ R_j = \begin{bmatrix} r_{1j} \\ r_{2j} \\ \vdots \\ r_{7j} \end{bmatrix}^{7\times 1}, \tag{16}$$

$$\tilde{I} = \begin{bmatrix} \underline{1} & & & \\ & \underline{1} & & \\ & & \ddots & \\ & & & \underline{1} \end{bmatrix}^{35\times 5}, \ \underline{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}^{7\times 1}, \tag{17}$$

$r_{ij} = w_{ij}/K_{ij}$, K_{ij} is the nominal steady-state gain (e.g. $K_{11} = 4.05$), w_{ij} is the uncertainty weight (e.g. $w_{11} = 2.11$), and $\hat{P}$ is the nominal plant transfer function [P in Eq. (13) with $\delta_k = 0$]. With using a third-order Pade approximation for the time delays, the number of states in $G(s)$ using this construction was 97. A balanced realization algorithm (BALMR from the Robust Control Toolbox [12]) was used to reduce the states to 88.

All three algorithm implementations were able to reduce the total dimension of Δ to five 1×1 real independent scalars, which is the lowest possible dimension (this can be shown by exploiting structural considerations when constructing the LFT [41]). The SR, KD, and BT algorithms implemented in MATLAB and run on a Sun Ultra 2200 took 1, 2 and 54 s, respectively. The difference in running times between the SR and KD algorithm implementations is mostly due to the way that similarity transformations are handled in the software implementation of the KD algorithm downloadable from the web [13]. A re-implementation of the KD algorithm to remove unnecessary similarity transformations would result in software of similar computational speed as our implementation of the SR algorithm [40]. To quantify the accuracy of the algorithm implementations, the worst-case error in the overall input–output mapping was estimated:

$$E(\omega) \equiv \|\Delta\|_\infty^{\max \leqslant 1} \bar{\sigma}\left\{ F_u(G(j\omega), \Delta) - F_u\left(\bar{G}(j\omega), \bar{\Delta}\right)\right\} \tag{18}$$

where $\bar{G}$ and $\bar{\Delta}$ are the matrices for the reduced uncertain system, and ω is the frequency, and the maximum was computed over 1000 values for Δ selected randomly from a uniform distribution for each uncertain parameter (Fig. 4). The results indicate that all three algorithms produced the lowest dimension LFTs with negligible error for all frequencies. Note that the maximum singular value bounds the error in each element of the input-output mapping from u to y (Fig. 1), so each element is also captured with high fidelity.

It is also instructive to compare the computational expense of designing a robust optimal controller for the Shell Standard Control problem for the original and the reduced LFTs. Designing a suboptimal robust controller for the original LFT is beyond the limit of existing DK-iteration software [2]. On the other, off-the-shelf DK-iteration software [2] applied to the reduced dimension LFT would converge in a manner of minutes on a Sun Sparcstation 20 or a Pentium III personal computer.

4.2. CSTR model

This is a variation on the uncertain model of a 3-state continuous-stirred tank reactor (CSTR) model studied by [1]. The process is the acid-catalyzed cylopentadiene production in an aqueous solution using electrophylic hydration. The uncertain system has 7 repeated perturbations with $\Delta = \mathrm{diag}\{\delta_1 I_3, \delta_2 I_3, \delta_3 I_4, \delta_4, \delta_5, I_4\delta_6 I_3, \delta_7\}$.

The implementations of SR, KD, and BT algorithms gave a reduced system with block structure $\bar{\Delta} = \mathrm{diag}\{\delta_1 I_3, \delta_2 I_3, \delta_3 I_3, \delta_4, \delta_5, \delta_6, \delta_7\}$. The number of degrees of freedom for the D-scales used in DK-iteration for the reduced system is nearly half the number required in the original problem. The worst case error estimates indicate that the reduced dimension models are highly accurate in capturing the overall input-output dynamics (see Fig. 5). Based on our computational experience solving such problems, DK-iteration would require approximately an order-of-magnitude more time to compute a suboptimal controller for the original system than for the reduced dimension system.

4.3. Benchmark problem

This is a benchmark problem which describes the phugoid motion of a DHC2-Beaver aircraft [24]. The original LFT for this system had 2 states with $\Delta = \mathrm{diag}\{\delta_1 I_{88}, \delta_2 I_{63}\}$. Such an LFT is too large to be used for robust controller synthesis using existing commercial software [2,6,19]. The SR and KD algorithms applied to the phugoid problem produced a system with two states and $\bar{\Delta} = \mathrm{diag}\{\delta_1 I_5, \delta_2 I_7\}$, which is an order of

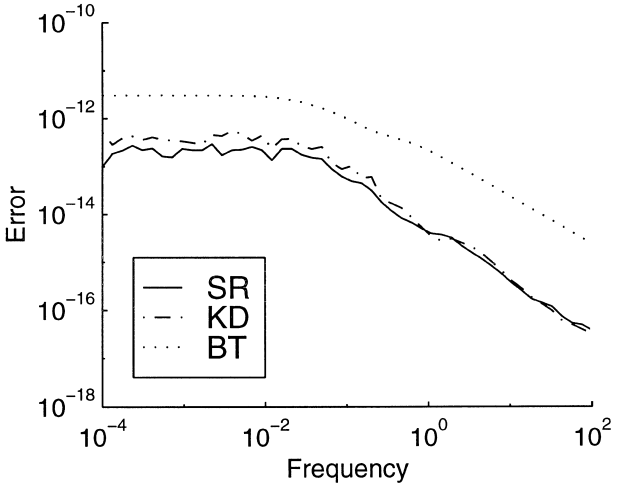

Fig. 4. Estimate of worst-case input–output error for Example 1 using multidimensional balanced truncation (BT), successive multidimensional realization (SR), and multidimensional Kalman decomposition (KD) algorithms.

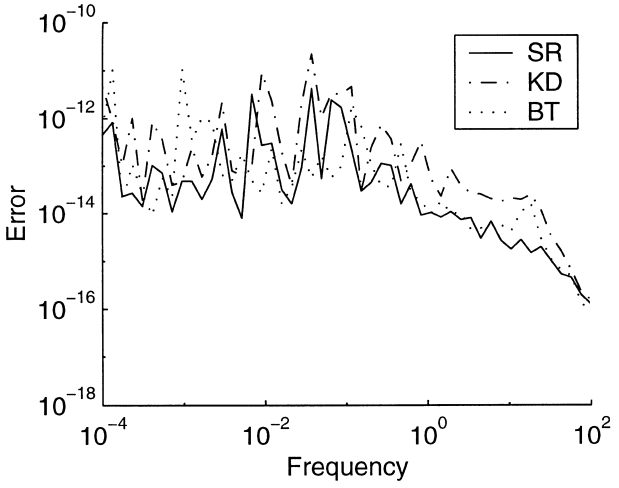

Fig. 5. Estimate of worst-case input–output error for Example 2 using multidimensional balanced truncation (BT), successive multidimensional realization (SR), and multidimensional Kalman decompositon (KD) algorithms.

magnitude reduction in the dimensionality of Δ. The implementation of BT algorithm was too computationally expensive to be applied directly to the phugoid problem. An estimate of the worst-case error in the input–output mapping of the reduced model obtained from the SR and KD algorithm implementations is plotted in Fig. 6. The errors in the reduced dimension systems are negligible. Existing software can be used to synthesize robust controllers for the reduced dimension LFTs [6,19].

Note that it was reported in [24] that the SR algorithm reduced the dimensionality of the original LFT to two states with $\bar{\Delta} = \mathrm{diag}\{\delta_1 I_{6,} \delta_2 I_7\}$. The incorrect result in [24] is likely due to an imprecise implementation of their minimal realization algorithm.

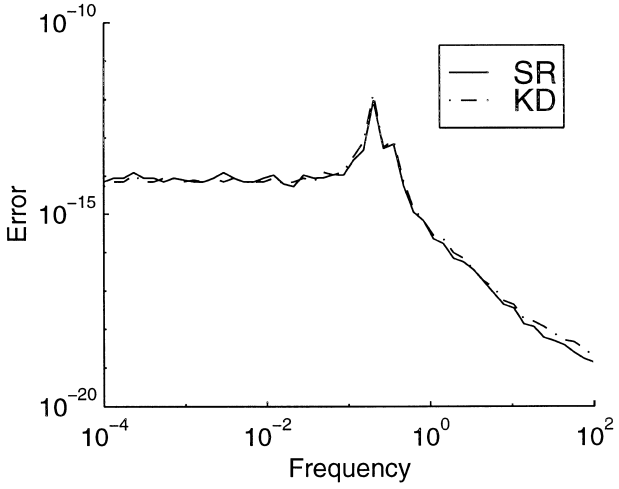

Fig. 6. Estimate of worst-case input–output error for Example 3 using successive multidimensional realization (SR) and multidimensional Kalman decomposition (KD) algorithms.

4.4. SR and KD comparison

The SR and KD algorithms gave the same level of dimensionality reduction for the three examples. However, as noted in Section 3.3, the two algorithms are in fact different. To illustrate this, consider the following system [14]:

$$G(s) = \left[\begin{array}{cc|c} 1 & 1 & 0 \\ 1 & 1 & 0 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right], \quad \Delta = \mathrm{diag}\{\delta_1, \delta_2\}. \tag{19}$$

The generalized states (that is, the outputs of δ_1 and δ_2) are both unreachable and observable since the input u does not enter the system and the output y is exactly equal to the generalized state vector. Since the generalized states are unreachable, the implementation of KD algorithm correctly eliminates both generalized states. However, the implementation of SR algorithm did not eliminate either generalized state. The reason is that reachability of a state in the SR algorithm is determined not only from the input but also from the output of other perturbations. In this example, the first state is reachable from the second state and vice versa. Hence, the one-dimensional realization algorithms used in SR cannot remove either state. The KD algorithm correctly removes the states because it can determine the unreachability directly from the input.

5. Conclusions

This paper provided an overview of algorithms for the dimensionality reduction of uncertain systems. The successive multidimensional realization (SR), multidimensional

Kalman decomposition (KD), and multidimensional Balanced Truncation (BT) algorithms can greatly reduce the dimension of systems with real and complex repeated diagonal uncertainties. The dimensionality reduction algorithms allow robust controller synthesis and robust control structure selection to be performed on uncertain systems of higher dimension. MATLAB programs implementing these algorithms are publicly available.

The KD and SR algorithms gave the same dimensionality reduction with similar computational expense for three uncertain systems taken from the literature. The fact that examples can be constructed in which the KD algorithm produces a lower dimensional realization than the SR algorithm, but not vice versa, motivates the use of the KD algorithm for exact dimensionality reduction. The BT algorithm is significantly more computationally expensive than the KD and SR algorithms, but it can reduce the dimensionality for systems in which the exact reduction algorithms cannot. The BT algorithm will directly benefit from continued advances in LMI algorithms and software implementations, so that the difference in computational expense between the algorithms will likely shrink over the next five years.

Acknowledgements

The authors would like to thank Dr. Raffaelo D'Andrea and Dr. Carolyn L. Beck for providing the implementations of multidimensional Kalman decomposition and multidimensional balanced truncation, and Dr. N. Amann and Dr. Frank Allgower for giving us the LFT for the uncertain model for the CSTR example.

References

[1] N. Amann, F. Allgower, μ-Suboptimal design of a robustly performing controller for a chemical reactor, Int. J. Control 59 (1994) 665–687.
[2] G.J. Balas, J.C. Doyle, K. Glover, A.K. Packard, R.S.R. Smith, μ-Analysis and Synthesis Toolbox (μ-Tools): MATLAB functions for the Analysis and Design of Robust control Systems, The Math Works, Natick, Massachusetts, 1992.
[3] C. Beck, R. D'Andrea, Minimality, controllability and observability for uncertain systems. In Proceedings of the American Control Conference, IEEE Press, Piscataway, NJ 1997, pp. 3130–3135.
[4] C. Beck, J.C. Doyle, K. Glover, Model reduction of multidimensional and uncertain systems, IEEE Trans. on Auto. Control 41 (1996) 1466–1477.
[5] S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, Linear Matrix Inequalities in System and Control Theory, vol. 15 of Studies in Applied Mathematics, SIAM, Philadelphia, PA, 1994.
[6] S. Boyd, S. Wu, A Parser/Solver for Semidefinite Programs With Matrix Structure: User's guide, Stanford University, Stanford, California, 1996.
[7] R.D. Braatz, J.H. Lee, M. Morari, Screening plant designs and control structures for uncertain systems, Comp. & Chem. Eng. 20 (1996) 463–468.
[8] R.D. Braatz, B.A. Ogunnaike, A.P. Featherstone, Identification, estimation, and control of sheet and film processes, In Proceedings of the IFAC World Congress, Elsevier Science, Tarrytown, NY, 1996, pp. 319–324.
[9] R.D. Braatz, E.L. Russell, Robusteness margin computation for large scale systems, Comp. & Chem. Eng. 23 (1999) 1021–1030.
[10] R.D. Braatz, P.M. Young, J.C. Doyle, M. Morari, Computational complexity of μ calculation, IEEE Trans. on Auto. Control 39 (1994) 1000–1002.
[11] C.-T. Chen, Linear system theory and design, Harcourt Brace College Publishers, Orlando, FL, 1984.
[12] R.Y. Chiang, M.G. Safonov, Robust Control Toolbox: For Use with MATLAB, The Math Works, Natick, MA, 1992.
[13] R. D'Andrea, Model reduction of LFT. http://www.mae.cornell.edu/raff/software/md/md.html, 1998. Computer software.
[14] R. D'Andrea, 1999. Private communication.
[15] R. D'Andreas, S. Khatri, Kalman decomposition of linear fractional transformation representations and minimality. In Proceedings of the American Control Conference, IEEE Press, Piscataway, NJ 1997, pp. 3557–3561.
[16] J.C. Doyle, Analysis of feedback systems with structured uncertainities, IEE Proc. Pt. D 129 (1982) 242–250.
[17] J.C. Doyle, Structured uncertainity in control system design, in: Proceedings of the IEEE Conference on Decision and Control, IEEE Press, Piscataway, NJ,1985, pp. 260-265.
[18] A.P. Featherstone, R.D. Braatz, Control–oriented modeling of sheet and film processes, AIChE J. 43 (1997) 1989–2001.
[19] P. Gahinet, A. Nemirovskii, A.J. Laub, M. Chilali, LMI Control Toolbox User's Guide, The Math Works, Natick, MA, 1995.
[20] M.R. Garey, D.S. Johnson, Computers and Intractibility: A Guide to NP-Completeness, W.H. Freeman and Company, New York, 1983.
[21] K.-C. Goh, M.G. Safonov, J.H. Ly, Robust synthesis via bilinear matrix inequalities, Int. J. of Robust and Nonlinear Control 6 (1996) 1079–1095.
[22] M. Hovd, R.D. Braatz, S. Skogestad, SVD controllers for H_{2-}, $H_{\infty-}$, and μ-optimal control, Automatica 33 (1996) 433–439.
[23] T. Kailath, Linear Systems, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[24] P. Lambrechts, J. Terlouw, S. Bennani, M. Steinbuch, Parametric uncertainity modeling using lft's, in: Proceedings of the American Control Conference, IEEE Press, Piscataway, NJ, 1993. pp. 267–272.
[25] J.H. Lee, R.D. Braatz, M. Morari, A. Packard, Screening tools for robust control structure selection, Automatica 31 (1995) 229–235.
[26] W.-M. Lu, K. Zhou, J.C. Doyle, Stabilization of linear systems: an LFT approach, IEEE Trans. on Auto. Control 41 (1996) 50–65.
[27] H.D. Mittelmann, Several SDP-codes on problems from SDPLIB, 1999. ftp://plato.la.asu.edu/pub/sdplib.txt.
[28] B.C. Moore, Principle component analysis in linear systems: controllability, observability, and model reduction, IEEE Trans. on Auto. Control 26 (1981) 17–32.
[29] M. Morari, E. Zafiriou, Robust Process Control, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[30] A. Nemirovskii, P. Gahinet, The projective method for solving linear matrix inequalities, in: Proceedings of the American Control Conference, IEEE Press, Piscataway, NJ, 1994, pp. 840–844.
[31] Y. Nesterov, A. Nemirovskii, Interior Point Polynomial Algorithms in Convex Programming, vol. 13 of Studies in Applied Mathematics, SIAM, Philadelphia, PA, 1994.
[32] A.K. Packard, What's New with μ: Structured Uncertainity in Multivariable Control, PhD thesis, University of California, Berkeley, 1988.
[33] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[34] D.M. Prett, M. Morari, (Eds.), Shell Process Control Workshop, Butterworth, Stoneham, MA, 1987.
[35] D.E. Rivera, M. Morari, Control-relevant model reduction problems for SISO H_2, H_∞, and μ-controller synthesis, Int. J. of Control 46 (1987) 505–527.
[36] H.H. Rosenbrock, State Space and Multivariable Theory, John Wiley, New York, 1970.
[37] H.H. Rosenbrock, Computer-Aided Control System Design, Academic Press, London, 1974.
[38] E.L. Russell, R.D. Braatz, Model reduction for the robustness margin computation of large scale uncertain systems, Comp. & Chem. Eng. 22 (1998) 913–926.
[39] E.L. Russell, R.D. Braatz, Reduced models for uncertain systems. LSSRL Technical Memo UIUC-LSSRL 99-002, University of Illinois, Urbana, IL 61801-3792, 1999.
[40] E.L. Russell, R. Gunawan, R.D. Braatz, Successive Multidimensional Realization Algorithm. University of Illinois, Urbana, http://brahms.scs.uiuc.edu/~erp/lssrl/software/sr.m, 1999. Computer software.
[41] E.L. Russell, C.P.H. Power, R.D. Braatz, Multidimensional realizations of large scale uncertain systems for multivariable stability margin computation, Int. J. of Robust and Nonlinear Control 7 (1997) 113–125.
[42] M.G. Safonov, Stability margins of diagonally perturbed multivariable feedback systems, IEE Proc. Pt. D 129 (1982) 251–256.
[43] THE MATHWORKS — Control System Toolbox User's Guide, The Mathworks, Natick, Massachusetts, 1998.
[44] T. Togkalidou, R.D. Braatz, A bilinear matrix inequality approach to the robust nonlinear control of chemical processes, in: Proceedings of the American Control Conference, IEEE Press, Piscataway, NJ, in press.
[45] O. Toker, H. Ozbay, On the NP-hardness of the purely complex mu computation, analysis/synthesis, and some related problems in multidimensional systems, in: Proceedings of the American Control conference, IEEE Press, Piscataway, NJ, 1995, pp. 447–451.
[46] J.G. VanAntwerp, R.D. Braatz, A tutorial on linear and bilinear matrix inequalities, J. of Process Control, in press.
[47] J.G. VanAntwerp, R.D. Braatz, N.V. Sahinidis, Globally optimal robust control for systems with nonlinear time-varying perturbations, Comp. & Chem. Eng. 21 (1997) S125–S130.
[48] J.G. VanAntwerp, R.D. Braatz, N.V. Sahinidis, Globally optimal robust process control, J. of Process Control 9 (1999) 375–383.
[49] W. Wang, J. Doyle, C. Beck, K. Glover, Model reduction of LFT systems, in: Proceedings of the IEEE conference on Decision and Control, IEEE Press, Piscataway, NJ, 1991.
[50] P.E. Wellstead, M.B. Zarrop, W.P. Heath, A.P. Kjaer, X. Troyas, Two dimensional methods for machine direction and cross direction estimation and control, in: Control Systems '96 Preprints, Montreal, Quebec, April 30–May 2, 1996, CPPA, pp. 5–8.