Symbolic and numerical software tools for LFT-based low order uncertainty modeling

A. Varga and G. Looye

German Aerospace Center
DLR - Oberpfaffenhofen
Institute of Robotics and System Dynamics
D-82234 Wessling, Germany.
Fax: +49-8153-28-1441, E-mail: {andras.varga,gertjan.looye}@dlr.de

Abstract

One of the main difficulties in applying modern control theories for designing robust controllers for linear uncertain plants is the lack of adequate models describing structured physical model uncertainties. We present a systematic approach for the generation of uncertainty models described by *linear fractional transformations* (LFTs) and report on recently developed symbolic and numerical software to assist the generation of low order LFT-based uncertainty models. The kernel of the symbolic software is a Maple library for generation and manipulation of LFT models. Additional numerical tools for order reduction of LFT models are based on MATLAB and FORTRAN implementations of numerically reliable algorithms. Three examples of uncertainty modeling of aircraft dynamics illustrate the capabilities of the new software to solve high order uncertainty modeling problems.

1. Introduction

To apply robust control design methodologies or to perform robustness analysis it is important to have adequate descriptions of model uncertainties. An important class of modeling uncertainties are the *real parametric uncertainties* present in virtually every physical model. Nonlinear *parametric uncertain models* (PUMs) are often given in the form

$$\begin{array}{rcl} \dot{x}(t) & = & f(x(t), u(t), p) \\ y(t) & = & g(x(t), u(t), p), \end{array} \qquad (1)$$

where x, u, y are the state-, input- and output–vectors respectively, and p is a vector of model parameters. Such models which explicitly depend on the real physical parameters in p are well suited for nonlinear simulations and parametric studies (e.g., worst-case parameter analysis).

The nonlinear PUMs of the general form (1) have found little use in design methodologies for robust controllers. However, since parametric dependencies are explicit, these models can serve to generate linearized parameter dependent uncertainty models of the form

$$\begin{array}{rcl} \dot{\widetilde{x}} & = & A(p)\widetilde{x} + B(p)\widetilde{u} \\ \widetilde{y} & = & C(p)\widetilde{x} + D(p)\widetilde{u}, \end{array} \qquad (2)$$

where $\widetilde{x}$, $\widetilde{u}$ and $\widetilde{y}$ are small variations of the state-, input- and output-vectors with respect to their equilibrium values. The model (2) is a linear PUM describing locally the effects of parametric uncertainties. We can assume that any non-rational parametric expressions in the elements of the state space model matrices $A(p)$, $B(p)$, $C(p)$ and $D(p)$ can be replaced by polynomial or rational approximations. Furthermore, to account for dependencies of the entries of the system matrices on particular equilibrium points (or trim conditions), it is possible to enlarge the parameter vector p by adding to it some components of the equilibrium point vectors.

The linear rational PUM of form (2) can be converted to a so-called *linear fractional transformation* (LFT) model, which is the required uncertainty representation to be used within modern linear robust control design methodologies like μ-synthesis [16]. This conversion is essentially a multidimensional (n-D) state space realization problem. Since this problem is not even theoretically completely solved, there are no general procedures to compute minimal n-D realizations. In practice, adhoc procedures involving a lot of heuristic are used to construct LFT models and often the resulting LFT models have orders which are too high to be of practical use. Therefore, generating low order LFT models and reducing the order of LFT models are issues of paramount importance for the successful usage of modern synthesis techniques as μ-analysis and μ-synthesis.

In this paper we present a systematic approach for generation of lower order uncertainty models described by LFTs starting from linear PUMs of form (2). Furthermore, we describe recently developed symbolic and numerical software tools for generation and reduction of LFT models. The core of our software is a symbolic computation library implemented in Maple to generate low order LFT models. This is complemented by several numerical MATLAB-tools for order reduction of LFT models. These latter tools are based on robust numerical software available in the control library SLICOT [5]. We present three examples of increasing complexities illustrating the capabilities of the new software on basis of the linear PUM of the *Research Civil Aircraft Model* (RCAM) derived in [15]. This model was used in the GARTEUR robust flight control benchmark problem [11].

2. Generation of low order LFT models

In what follows we sketch very briefly the approach to convert a parametric description of a linear system of the form (2) into an LFT-based uncertainty description. Recall first that for a partitioned matrix M

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \in \mathbb{R}^{(r_1+r_2)\times(q_1+q_2)}$$

and for $\Delta \in \mathbb{R}^{q_1\times r_1}$, the *upper* LFT $\mathcal{F}_u(M,\Delta)$ is defined by the feedback-like formula [10]

$$\mathcal{F}_u(M,\Delta) = M_{22} + M_{21}(I - \Delta M_{11})^{-1}\Delta M_{12}.$$

Any uncertainty in a parameter p_i expressed as $p_i \in [\,\underline{p}_i,\ \overline{p}_i\,]$ can be transcribed in a normalized form $p_i = p_{i0} + s_{i0}\delta_i$ with $|\delta_i| \le 1$, $p_{i0} = (\underline{p}_i + \overline{p}_i)/2$ and $s_{i0} = (\overline{p}_i - \underline{p}_i)/2$. This local parameter uncertainty can be then expressed as an elementary upper LFT

$$p_i = \mathcal{F}_u\left(\begin{bmatrix} 0 & s_{i0} \\ 1 & p_{i0} \end{bmatrix}, \delta_i\right). \tag{3}$$

Since all elements of matrices A, B, C and D are rational functions in parameters p_i, $i = 1,\ldots,q$, the structured parametric uncertainties at the components level can be transformed into structured parametric uncertainties at the level of system matrices by using the basic coupling formulas for LFT models [10]. Specifically, products, sums, divisions of individual variables, can be directly represented by series, parallel or feedback couplings of the corresponding elementary LFTs of form (3). The same is true for the corresponding operations with more general rationally dependent parametric expressions. Finally, elementary matrix constructs like row and column concatenations, or diagonal stacking are immediately expressible by equivalent LFT constructs. Thus, for all system matrices, LFT uncertainty models can be readily generated using elementary LFT operations.

It follows that if we write the state space description (2) in the form

$$\begin{bmatrix} \dot{\widetilde{x}} \\ \widetilde{y} \end{bmatrix} = S(p)\begin{bmatrix} \widetilde{x} \\ \widetilde{y} \end{bmatrix} = \begin{bmatrix} A(p) & B(p) \\ C(p) & D(p) \end{bmatrix}\begin{bmatrix} \widetilde{x} \\ \widetilde{y} \end{bmatrix}$$

then we can express $S(p)$ as an LFT

$$S(p) = \mathcal{F}_u\left(\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}, \Delta\right), \tag{4}$$

where the diagonal matrix

$$\Delta = diag[\,\delta_1 I_{n_1}, \delta_2 I_{n_2}, \cdots, \delta_q I_{n_q}\,]$$

has on its diagonal the normalized uncertainty parameters δ_1, δ_2, ..., δ_q. Note that

$$S_{22} = \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix},$$

where A_0, B_0, C_0 and D_0 are the *nominal* system matrices (for all δ_i set to zero). The order of the LFT uncertainty description of the system (2) is $n_\Delta = \sum_{i=1}^q n_i$, where n_i is the order of the block in Δ corresponding to the uncertain parameter p_i.

A systematic procedure to generate LFT-based uncertainty descriptions has been described in [12]. However, the LFT models obtained by using the accompanying MATLAB tools relying on this approach (the PUM Toolbox), have typically very large orders and numerically sensitive order reductions are necessary to be performed both during LFT generation as well as after obtaining the LFT model. The interlaced symbolic and numeric manipulations are hazardous since they involve many tolerance dependent rank decisions. A single erroneous rank decision could lead to useless results. In what follows we discuss several enhancements of the approach of [12] with the help of symbolic manipulations of rational functions and we describe alternative order reduction tools for LFT models which successfully complement the symbolic computations.

Since the construction of minimal order LFT descriptions of rational functions or matrices (i.e., with least n_Δ) is theoretically still open, we concentrated our efforts on implementing n-D realization methods which tend to minimize the orders of generated LFTs. The basic computation in our approach is the low order realization of multivariate polynomials and rational functions. For polynomials, it is easy to see that the minimal order of LFT realization is directly related to the minimal number of operations (additions and multiplications) required to evaluate the polynomial. Two approaches have been considered for generating low order LFTs for polynomials. The first approach exploits common expressions in polynomials and thus provides an "optimal" algorithm to evaluate a multivariate polynomial with the "least" number of operations. The implementation of this approach suffers from the fact that operations with constants are also counted, and thus the method does not always ensure the least operation count. The second approach is along the lines of methods used in [6] to evaluate multivariate polynomials using nested Horner schemes. In many cases, the Horner scheme leads to less operations than the optimization method, and thus to lower order LFTs. Frequently, these methods are able to generate LFT realizations which are almost minimal. The construction of LFT realizations of rational functions amounts to generate low order LFT realizations for the quotient of numerator and denominator polynomials. To ensure low order realizations of rational functions, symbolic partial order reduction can be additionally performed (see below).

The above approach ensures basically that individual elements of system matrices are realized with almost minimal order LFTs. However, the assembled LFTs for the whole matrices can be of much higher order than the minimal realization because of possible common expressions among the matrix elements. Thus, our approach could be complemented at matrix level by additional heuristic as proposed in [7], [4], [8] to extract common expressions in the underlying system matrices or to arrive at factored or additively decomposed expressions. All these methods often lead to substantially lower order LFT realizations, because the source of non-minimality in the generated LFTs is in most cases of structural nature.

Order reduction of LFT models is important to obtain lower order LFT realizations if the original realization is not minimal. For a recent survey of existent methods see [2]. A possible numerical approach is the exact model reduction technique for LFT systems as proposed in [3]. This procedure extends the exact *Balance & Truncate* (B&T) model reduction approach of 1-D discrete systems to n-D systems. From computational point of view it involves the seeking of minimum rank non-negative definite block diagonal solutions of two Lyapunov-type *linear matrix inequalities* (LMIs). It is presently questionable if this approach can be turned into an efficient procedure to derive minimal realizations of LFT descriptions, since no efficient procedures exist to find singular structured solutions of large LMIs. On the other side, this approach can be potentially employed to generate lower order LFT approximations by using balancing related multi-dimensional truncation techniques [3].

An effective alternative approach for exact order reduction of high order LFTs is to use block-diagonal similarity transformation matrices of the form $T = \mathrm{diag}\,[T_1, \ldots, T_q]$, which commute with the uncertainty structure of Δ (i.e. $T\Delta = \Delta T$), to remove uncontrollable/unobservable parts. Recently, a minimal realization procedure based on n-D controllability/observability forms has been proposed in [9]. This approach appears to be well suited to be turned into a reliable numerical procedure. Significant reductions of order can often also be often by using sequential 1-D reduction techniques based on orthogonal controllability/observability canonical forms computed separately for each block of Δ [12]. Alternatively, sequential 1-D order reduction procedures for standard discrete-time systems can be employed to obtain approximate low order LFT models. The numerical examples presented in this paper confirm the potential usefulness of this approach in obtaining low order approximate LFT models.

One additional particular aspect worth to be mentioned in this context is the possibility to exploit that frequently the S_{11} matrix in the resulting LFT description (4) has most of its eigenvalues equal to zero (all eigenvalues are equal to zero if $S(p)$ is a polynomial matrix in p). Since many of these eigenvalues are uncontrollable or unobservable, a special order reduction procedure can be devised to remove in a first step only those uncontrollable/unobservable eigenvalues which are zero. This can be done easily by using numerical singular value decompositions on the block rows/columns of the LFT realization or even symbolic LU-like rank revealing decompositions. Since not all locally uncontrollable/unobservable zero eigenvalues can be removed in this way, the procedure, although effective in practice, does not always provide the maximum achievable reduction.

3. Software Tools

A library of basic functions for symbolic generation and manipulation of LFT models, called `lftlib`, has been implemented in the computer algebra language Maple. The operations are performed on LFT objects which can be either a parametric matrix or an LFT model defined by a quadruple of matrices and a variables list. Thus all matrix-matrix, matrix-LFT and LFT-LFT operations can be performed transparently using the same basic functions. The functions allow the generation and manipulation of *symbolic* LFT models. The main advantage of this feature is the possibility to construct LFT realizations only for subsets of parameters. This allows the usage of a unique original model even if different uncertain parameter sets are considered. This feature also supports experiments for an incremental generation of LFT models, by successively generating LFT realizations for distinct subsets of uncertain parameters. In contrast, approaches based on MATLAB (e.g., the PUM-software [12]) always require the definition of the complete uncertain parameter set and the specification of numerical values for the rest of parameters, since the manipulated LFT descriptions always have to be numerical.

The library `lftlib`[1] contains over 20 functions grouped in several categories:

[1] The authors thank Dr. D. Kaesbauer from DLR-Oberpfaffenhofen for expert help in implementing the first versions of several routines of this library.

- Functions for generation of LFT models
  - for multivariate polynomials
  - for rational functions
  - for rational matrices
- Functions for basic LFT manipulations
  - LFT couplings (e.g., series, parallel)
  - row/colums concatenations
  - inverse/dual of an LFT model
  - symbolic order reduction
  - lower-upper LFT conversions
  - LFT to rational matrix conversion
  - similarity transformation
  - variable substitution
- Auxiliary functions
  (e.g., sorting, scaling, interface to MATLAB).

The most basic function of the library generates an LFT model for a multivariate polynomial, converted previously to a special form by an expression evaluation function. This conversion can be done either by the common subexpression optimization function `optimize` or by the conversion function `convert` (with the `horner` flag set to obtain a nested Horner-form of multivariate polynomials). The expression evaluations scheme to be employed as well as the option for order reduction can be specified by the user with help of global variables. The effects of these options on obtaining lower order LFT models is illustrated in Example 1 of the next section.

The general approach which we pursued for implementing numerical software for order reduction of LFT models was to ensure: (1) high efficiency by exploiting structural features of the problem, (2) numerical robustness by preventing unnecessary failures, (3) enhanced numerical accuracy by using numerical reliable algorithms, and (4) increased user-friendlyness by using an appropriate computational environment. The efficiency of computations, numerical robustness, and high accuracy of results have been achieved mainly by using FORTRAN based robust implementations of algorithms for all basic computations. Languages like FORTRAN or C++ allow exploiting all structural features of the problems with the lowest computational effort and memory usage. This strongly contrasts with the intrinsic limitations of popular user-friendly computational environments (e.g., MATLAB or MATRIX_X), which basically rely on an interpretational mode of work. Probably the best possible trade-off results by combining the user-friendly operation of MATLAB with the facilities to use external functions (so called *mex*-functions) implemented in FORTRAN. For basic computations as system scaling, minimal realization or order reduction, *mex*-functions have been implemented based on FORTRAN routines from the LAPACK-based [1] public domain control library SLICOT[2] [5]. These *mex*-functions have been further used to implement reliable and efficient n-D order reduction software in MATLAB.

For numerical order reduction of LFT models two alternative MATLAB-based software tools are available. For order reduction based on n-D controllability/observability staircase forms [9], the original n-D systems software[3] has been enhanced by adding an LFT scaling facility (using a SLICOT-based *mex*-function) and replacing the call to the $O(n^4)$ complexity MATLAB function `cotrbf` by a call to an efficient $O(n^3)$ complexity FORTRAN based *mex*-function for controllability staircase form computation. For the sequential 1-D order reduction we implemented a special *m*-function calling optionally two *mex*-functions for minimal realization based on 1-D controllability/observability staircase forms [13], or the discrete-time B&T approach for order reduction using enhanced accuracy square-root and balancing-free algorithms [14]. The usage of this latter approach allows to compute approximate LFT models in a similar fashion as in the case of standard systems.

Essential for the application of numerical techniques based on computing numerically sensitive controllability staircase forms or even on balancing related model reduction techniques is that the underlying LFT models are well-scaled. It is important to emphasize that the LFT models resulting from symbolic manipulations have often a very wide range of values. For instance, the norms of matrices of LFT realizations in the examples of the next section were sometimes as high as 10^{32} or even higher. It is clear that without adequate scaling procedures all numerical computations on these models are hopeless. The MATLAB-based scaling procedure available in the *Control Toolbox* was not able to achieve significant norm reductions and therefore the numerical reduction of LFT models was not possible. In contrast, with the SLICOT-based *mex*-functions it was possible to reduce the norm of all underlying matrices in the LFT realizations below 10. This spectacular reduction of numerical ranges allowed to obtain very accurate results even for very high order LFT models.

4. Examples

In this section we present several examples illustrating the generation of LFT uncertainty models. All employed symbolic models are available as Maple codes on the web[4].

Example 1. This example is intended to illustrate the effectiveness of different options available in the symbolic software to achieve lower order LFT models for a

[2] `ftp://wgs.esat.kuleuven.ac.be/pub/WGS/SLICOT/`
[3] `http://www.mae.cornell.edu/Raff/software/md/md.html`
[4] `http://www-er.df.op.dlr.de/~varga/lft/`

single rational function. This computation is basic for generation of LFT models starting from a given linear PUM (2). As an example we take the most complicated term $a_{29}(p)$ in the $A(p)$ matrix of the extended envelope RCAM [15]. The four uncertain parameters in the RCAM are: the mass m, two components of the position of center of gravity X_{cg} and Z_{cg}, and the trimmed air speed V_A. The expression of $a_{29}(p)$ is

$$a_{29} = 0.061601 \frac{\tilde{a}_{29}}{C_w V_A}$$

where $C_w = \frac{mg}{\frac{1}{2}\rho V_A^2 S}$ and

$$\begin{aligned} \tilde{a}_{29} = \; & 1.6726\, X_{cg}\, {C_w}^2 Z_{cg} - 0.17230\, {X_{cg}}^2 C_w \\ & -3.9324\, X_{cg}\, C_w\, Z_{cg} - 0.28903\, {X_{cg}}^2 {C_w}^2 Z_{cg} \\ & -0.070972\, {X_{cg}}^2 Z_{cg} + 0.29652\, {X_{cg}}^2 C_w\, Z_{cg} \\ & +4.9667\, X_{cg}\, C_w - 2.7036\, X_{cg}\, {C_w}^2 \\ & +0.58292\, {C_w}^2 - 0.25564\, {X_{cg}}^2 - 1.3439\, C_w \\ & +100.13\, X_{cg} - 14.251\, Z_{cg} - 1.9116\, {C_w}^2 Z_{cg} \\ & +1.1243\, X_{cg}\, Z_{cg} + 24.656\, C_w\, Z_{cg} \\ & +0.45703\, {X_{cg}}^2 {C_w}^2 - 46.850 \end{aligned}$$

The used uncertainty normalizations are

$$\begin{aligned} m &= 125000 + 25000\delta m \\ X_{cg} &= 0.23 + 0.08\delta X_{cg} \\ Z_{cg} &= 0.105 + 0.105\delta Z_{cg} \\ V_A &= 80 + 10\delta V_A \end{aligned} \tag{5}$$

where δm, δX_{cg}, δZ_{cg} and δV_A are the normalized uncertain parameters.

The following table contains the resulted block structures for $\Delta = \{\delta m I_{n_1}, \delta X_{cg} I_{n_2}, \delta Z_{cg} I_{n_3}, \delta V_A I_{n_4}\}$ and the total order n_Δ of Δ for different settings of optimization and order reduction options in Maple:

Optimization	Reduction	$\{n_1, n_2, n_3, n_4\}$	n_Δ
None	No	$\{49, 72, 36, 136\}$	293
`optimize`	No	$\{19, 6, 1, 83\}$	109
`horner`	No	$\{3, 6, 9, 41\}$	59
`optimize`	Yes	$\{4, 4, 1, 18\}$	27
`horner`	Yes	$\{3, 2, 3, 18\}$	26

These results show that the use of symbolic preprocessing functions `optimize` and `convert` (with `horner` option) allows a substantial reduction of the initial orders of generated LFTs. In combination with partial order reduction, this reduction is particularly effective and leads to a more than 10 times reduction of the orders of the symbolically generated LFTs.

The least achieved block orders starting from the last two entries in the above table have been obtained with numerical 1-D/n-D reductions and have block structures $\{4, 4, 1, 5\}$ and $\{3, 2, 3, 7\}$, respectively. Further, an approximate LFT model with block structure $\{3, 3, 1, 3\}$ has been obtained by using B&T model reduction for a tolerance of 10^{-4} on the Hankel-singular values. The corresponding approximation $\bar{a}_{29}$ of a_{29} shows a very good agreement over the whole range of values of uncertainties. For a sample of 10^4 uniformly generated values of the normalized parameters, the maximum relative error in $\bar{a}_{29}$ satisfied

$$\max \left| \frac{a_{29} - \bar{a}_{29}}{a_{29}} \right| \approx 6 \cdot 10^{-5}.$$

Example 2. This example illustrates the generation of locally valid LFT models for a realistic, but relatively simple nonlinear aircraft model RCAM [11]. The linear PUM model has been obtained by symbolic linearization of the nonlinear RCAM in symmetric longitudinal flight. The uncertain parameters are those in Example 1 and the same uncertainty normalization (5) has been used. The initial LFT realization of the system matrix $S(p)$ obtained using symbolic realization with `optimize` and with partial order reductions has the block structure $\{43, 6, 8, 26\}$ and total order $n_\Delta = 83$. Repeated numerical 1-D order reductions led to a reduced LFT model with a block structure $\{8, 4, 4, 18\}$ and $n_\Delta = 34$. The same block structure has been obtained by using n-D order reduction. By using the discrete B&T model reduction technique with a tolerance of 0.01 on the Hankel singular values an LFT model with block structure $\{8, 2, 2, 10\}$ and $n_\Delta = 22$ has been obtained. Monte-Carlo analysis indicates a .006% agreement of the approximate LFT model with the original system matrix $S(p)$. It is interesting to note that for this model, the original LFT model generated with the method of [12] has order larger than 200.

Example 3. This example relies on the linear parametric model in [15] generated for the RCAM. This model has a substantially increased complexity because of additional parameter fitting performed on individual elements of system matrices. The purpose of this fitting was to extend the validity of the linear PUM over the whole flight envelope of RCAM. The initial LFT realization of the system matrix $S(p)$ obtained using symbolic realization with `optimize` and with partial order reductions performed on matrix element level has the block structure $\{80, 28, 10, 189\}$ and order $n_\Delta = 307$. Partial order reductions applied on matrix level led to a block structure $\{73, 21, 9, 158\}$ and order $n_\Delta = 261$. Numerical 1-D (based on discrete B&T method) and n-D order reductions led to reduced LFT models with block structures $\{33, 18, 7, 66\}$ and $\{39, 20, 8, 91\}$, respectively, and the corresponding orders $n_\Delta = 124$ and $n_\Delta = 158$. By using the discrete B&T model reduction technique with a tolerance of 0.001 on the Hankel singular values an LFT model with block structure $\{27, 11, 5, 38\}$ and $n_\Delta = 81$ has been obtained. Monte-Carlo analysis indicates a .001% agreement of the approximate LFT model with the original system matrix $S(p)$.

5. Conclusions

We presented an overview of low order LFT generation for parametric uncertain linear models. Due to lack of general minimal realization procedures for n-D systems, ad-hoc methods based on symbolic LFT realization methods as well as on reliable numerical order reduction tools are essential for obtaining low order LFT models. A collection of symbolic Maple-based and numeric MATLAB-based software tools has been implemented which allows the generation of low order LFT models. Realistic aircraft model examples illustrated the capabilities of this software to solve high complexity practical problems.

The symbolic approach for LFT generation has important advantages. Since all computations are performed symbolically (floating-point numbers are represented exactly in rational form), there is no accuracy loss during generating the LFT models. Further, symbolic LFT models with respect to a subset of parameters can be easily constructed. Finally, symbolic manipulation techniques open clear perspectives to further improve the LFT generation process by including more involved techniques at the level of whole matrices, like extracting common row/column expressions or using multivariate factorization/decomposition techniques.

Numerical tools are also necessary to further reduce the order of generated LFT models or even to generate approximate low order LFT models. Since the order of realistic LFT models is usually relatively high, the numerical software must be robust and accurate. Moreover, of fundamental importance for employing numerical sequential 1-D or n-D order reduction software is the need for good LFT scaling software. The proposed approach relying on using reliable numerical software implemented in FORTRAN within the user-friendly environment MATLAB is a very promising way to efficiently solve complex computational problems for systems, like the generation of low order LFT-based physical uncertainty models.

References

[1] E. Anderson, Z. Bai, J. Bishop, J. Demmel, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK User's Guide, Second Edition.* SIAM, Philadelphia, 1995.

[2] C. Beck and R. D'Andrea. Computational study and comparisons of LFT reducibility methods. In *Proc. American Control Conference*, 1998.

[3] C. Beck, J. Doyle, and K. Glover. Model reduction of multi-dimensional and uncertain systems. *IEEE Trans. Autom. Control*, 41:1466–1477, 1996.

[4] C. Belcastro, B. C. Chang, and R. Fischl. A matrix approach to low-order uncertainty modeling of real parameters. In *Proc. IFAC 13th Triennial World Congress, San Francisco, CA*, pp. 297–302, 1996.

[5] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT – a subroutine library in systems and control theory. In B. N. Datta, Ed., *Applied and Computational Control, Signals and Circuits*, vol. 1, Birkhäuser, 1998.

[6] J. Carnicer and M. Gasca. Evaluation of multivariate polynomials and their derivatives. *Mathematics of Computation*, 54:231–243, 1990.

[7] Y. Cheng and B. De Moor. A multidimensional realization algorithm for parametric uncertainty modeling problems and multiparameter margin problems. *Int. J. Control*, 60:3022–3023, 1994.

[8] J. C. Cockburn and B. G. Morton. On linear fractional representations of systems with parametric uncertainty. In *Proc. IFAC 13th Triennial World Congress, San Francisco, CA*, pp. 315–320, 1996.

[9] R. D'Andrea and S. Khatri. Kalman decomposition of linear fractional transformation representations and minimality. In *Proc. American Control Conference*, 1997.

[10] J. C. Doyle, A. Packard, and K. Zhou. Review of LFTs, LMIs, and μ. In *Proc. of 30th CDC, Brighton, England*, pp. 1227–1232, 1991.

[11] P. Lambrechts, S. Bennani, G. Looye, and D. Moormann. The RCAM Design Challenge Problem Description. In J.-F. Magni, S. Benani, and J. Terlouw, Eds., *Robust Flight Control, A Design Challenge*, vol. 224, *Lecture Notes in Control and Information Science*, pp. 341–359, Springer-Verlag, London, 1997.

[12] J. Terlouw, P. Lambrechts, S. Bennani, and M. Steinbuch. Parametric uncertainty modeling using LFTs. In *Proc. of AIAA GNC Conf., Hilton, South Carolina*, 1992.

[13] A. Varga. Numerically stable algorithm for standard controllability form determination. *Electron. Lett.*, 17:74–75, 1981.

[14] A. Varga. Numerical methods and software tools for model reduction. In I. Troch and F. Breitenecker, editors, *Proc. of 1st MATHMOD Conf., Viena*, vol. 2, pp. 226–230, 1994.

[15] A. Varga, G. Looye, D. Moormann, and G. Grübel. Automated generation of LFT-based parametric uncertainty descriptions from generic aircraft models. *Mathematical and Computer Modelling of Dynamical Systems*, 4:249–274, 1998.

[16] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice Hall, 1996.