SYMBOLIC TECHNIQUES FOR LOW ORDER LFT-MODELLING

Simon Hecker and Andras Varga

Institute of Robotics and Mechatronics,
German Aerospace Center, DLR-Oberpfaffenhofen,
D-82234 Wessling, Germany
simon.hecker@dlr.de, andras.varga@dlr.de

Abstract: Symbolic preprocessing techniques are very useful to obtain low order LFT-representations for parametric models. In this paper we give an overview about existing preprocessing methods and we present new techniques and enhancements of existing methods. All methods are implemented in the new version 2 of the LFR-toolbox and their capabilities are illustrated by a challenging aircraft parametric uncertainty modelling example.

Keywords: Robust Control, Linear Fractional Transformation (LFT), Parametric Uncertain Systems

1. INTRODUCTION

Many physical dynamical systems can be described by differential equations depending on parameters, which are either not exactly known (i.e., uncertain) or are time-varying. The design of robust controllers for such systems ensuring the stability and performance requirements for all allowable parameter variations and over the whole range of operating conditions is a highly complex task and can be addressed only by employing advanced synthesis techniques like μ (Zhou *et al.*, 1996) or the *linear parameter varying* (LPV) control. The key aspect of these linear system based synthesis approaches is employing special model uncertainty descriptions based on *linear fractional transformation* (LFT) (Zhou *et al.*, 1996).

Recall that for a partitioned matrix

$$M = \left[\begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array}\right] \in \mathbf{R}^{(p_1+p_2)\times(m_1+m_2)}$$

and $\Delta \in \mathbf{R}^{m_1\times p_1}$, the *upper LFT* is defined as

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

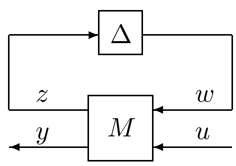

Fig. 1. LFT-Representation

and represents the input-output mapping between u and y after closing the *upper* loop in Fig. 1.

The following LFT-modelling problem is considered in this paper: given a $p_2{\times}m_2$ real matrix $G(\boldsymbol{\delta})$ depending rationally on k parameters grouped into the real vector $\boldsymbol{\delta} = (\,\delta_1,\ldots,\delta_k\,)$, find an equivalent LFT-representation (M,Δ) such that $G(\boldsymbol{\delta}) = \mathcal{F}_u(M,\Delta)$ with

$$\Delta = \operatorname{diag}(\delta_1 I_{r_1},\ldots,\delta_k I_{r_k}) \quad (2)$$

and $m_1 = p_1 = \sum_{i=1}^{k} r_i$. Here, m_1 represents the order of the LFT-representation (M,Δ) of $G(\boldsymbol{\delta})$.

Of paramount importance in many robust control design applications is to obtain *low order* LFT-representations of parametric system models. However, obtaining an LFT-representation of a multivariate rational model is essentially a multi-dimensional (n-D) realization problem (Bose, 1982) for which a minimal realization theory is still lacking. Therefore, in practice different order reduction techniques are employed in conjunction with an object oriented LFT-realization approach like that one described in (Terlouw *et al.*, 1993). Typically, exact or approximate order reduction based on numerical techniques (D'Andrea and Khatri, 1997; Varga and Looye, 1999) is performed as the final step of LFT-modelling, while the first step consists of a symbolic preprocessing which tries to find transformed parametric expressions which lead to lower order LFT representations via automated object-oriented LFT-realization procedures.

In this paper we present an overview of symbolic transformation methods which can be useful for generating low order LFT-representations and propose several new techniques and enhancements of existing symbolic preprocessing methods. The corresponding software tools have been implemented in the recent version 2 of the LFR-toolbox (Hecker *et al.*, 2004). The effectiveness of symbolic preprocessing is illustrated by a challenging aircraft uncertainty modelling example.

Before starting with the description of the symbolic methods, we briefly present an extension of the object-oriented LFT-realization described in (Terlouw *et al.*, 1993), which allows to represent arbitrary multivariate rational expressions in LFT-form. For example, $G(\delta) = 1/\delta$ has no LFT-representation with Δ of the form (2) because of singular M_{22} in the LFT-representation of δ. In practice, to avoid such difficulties, a symbolic normalization of parameters is performed first, which however often tends to significantly increase the order of the resulting LFT-representation (Cockburn and Morton, 1997). To prevent the need for such early normalization, the following inverse free formula for the inversion of an LFT-representation can be employed: $(\mathcal{F}_u(M,\Delta))^{-1} = \mathcal{F}_u(M_{inv}, \Delta_{inv})$, where

$$M_{inv} = \left[\begin{array}{cc|c} M_{22}+I_{p_2} & M_{21} & I_{p_2} \\ M_{12} & M_{11} & 0 \\ \hline -I_{p_2} & 0 & 0 \end{array}\right], \Delta_{inv} = \begin{bmatrix} I_{p_2} & 0 \\ 0 & \Delta \end{bmatrix} \quad (3)$$

This formula easily follows from the definition of a more general descriptor LFT-representation proposed in (Hecker and Varga, 2004). The new constant block I_{p_2} in Δ_{inv} can be interpreted as an additional dimension in a multidimensional system representation. Note that this block will vanish after a final normalization of the LFT-representation (Hecker *et al.*, 2004).

By employing systematically (3), together with the operations for addition/subtraction, multiplication and row/column concatenation for LFT-objects from (Terlouw *et al.*, 1993), we can directly generate LFT-representations of arbitrary rational matrices. Avoiding the preliminary symbolic normalization leads usually to LFT-representations of lower order.

2. SYMBOLIC PREPROCESSING

2.1 Problem statement

We discuss several symbolic transformation techniques which are potentially useful in obtaining low order LFT-realization of several classes of rational parametric matrices. Let $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_k)$ be a parameter vector with k components and we denote by $\boldsymbol{\delta}^{-1} = (\delta_1^{-1}, \ldots, \delta_k^{-1})$ the vector of reciprocal variables. We consider three classes of matrices depending of $\boldsymbol{\delta}$ for which symbolic transformation techniques are discussed: $\mathbb{R}[\boldsymbol{\delta}]^{m\times n}$ – the set of $m\times n$ matrices with multivariate polynomial entries ; $\mathbb{R}(\boldsymbol{\delta})^{m\times n}$ – the set of $m\times n$ matrices with multivariate rational entries; and $\mathbb{R}[\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}]^{m\times n}$ – the set of $m \times n$ matrices with multivariate Laurent polynomials entries. This last case is explicitly considered since many aircraft and automotive related uncertainty modelling problems are described in terms of such type of matrices.

A multivariate Laurent polynomial $g(\boldsymbol{\delta})$ has the expanded form

$$g(\boldsymbol{\delta}) = \sum_{r=1}^{l} c_r \delta_1^{n_{r,1}} \delta_2^{n_{r,2}} \ldots \delta_k^{n_{r,k}}, \qquad (4)$$

where c_r are real coefficients and $n_{r,1}, \ldots, n_{r,k} \in \mathbb{Z}$ for $r = 1, \ldots, l$ are integer exponents. We can associate to this polynomial the order of the LFT-realization which results when applying an object-oriented realization approach as in (Terlouw *et al.*, 1993; Hecker and Varga, 2004) to the polynomial in the above expanded form. This order is given by

$$\mathrm{ord}(g(\boldsymbol{\delta})) = \sum_{r=1}^{l} \sum_{s=1}^{k} |n_{r,s}|$$

where we assumed that negative and positive powers of indeterminates contribute in the same way to the order. This assumption is valid when employing inversions formulas with constant blocks as in (3) to realize the elementary reciprocal variables. If $g(\boldsymbol{\delta})$ is a general multivariate rational function of the form

$$g(\boldsymbol{\delta}) = \frac{a(\boldsymbol{\delta})}{b(\boldsymbol{\delta})}$$

where $a(\boldsymbol{\delta})$ and $b(\boldsymbol{\delta})$ are polynomials in expanded forms, the associated order is given by

$$\mathrm{ord}(g(\boldsymbol{\delta})) = \mathrm{ord}(a(\boldsymbol{\delta})) + \mathrm{ord}(b(\boldsymbol{\delta}))$$

We can also associate to an $m \times n$ rational matrix $G(\boldsymbol{\delta})$ with elements $g_{ij}(\boldsymbol{\delta})$ the total order

$$\mathrm{ord}(G(\boldsymbol{\delta})) = \sum_{i=1}^{m} \sum_{j=1}^{n} \mathrm{ord}(g_{ij}(\boldsymbol{\delta})) \qquad (5)$$

which corresponds to realize $G(\boldsymbol{\delta})$ element-wise using row and column concatenations via the object-oriented LFT-realization approach.

The practical importance of symbolic methods for generating low order LFT-realizations lies in the lack of a minimal realization theory for multidimensional systems. This is why, the role of symbolic preprocessing in building low order LFT-realizations of a given rational matrix $G(\boldsymbol{\delta})$ is to find equivalent representations of individual matrix elements, entire rows/columns or even the whole matrix which lead to LFT-representations of lower orders than given by (5). In the following subsections we present several transformation techniques which can be used for this purpose.

2.2 Single Element Conversions

Several conversion can be performed on single rational functions which can be useful to obtain equivalent representations which lead automatically to reduced order LFT-representations via an object oriented realization. These conversions can be performed either iteratively with respect to selected single variables or can be performed simultaneously for several variables in a specified order. Using such conversions, it is in principle possible to determine in each case a least achievable order of the corresponding LFT-realizations over all permutations of the variables. However, performing such exhaustive searches leads generally to combinatorial problems with exponential complexity. Therefore, unless the number of variables is small (say below 10), exhaustive searches are impracticable. In what follows we illustrate via examples some of possible conversions.

2.2.1. Horner Form. The conversion of a multivariate polynomial to a nested Horner form is useful for an efficient numerical evaluation of polynomials. This conversion can be also helpful to generate low order LFT-representations by applying it to the numerator and denominator polynomials of a rational function (Varga and Looye, 1999). For a multivariate rational function with k variables an exhaustive search for the least order of corresponding LFT-realizations involves $2k!$ conversions. This approach is effective especially when a few variables have significantly larger powers than the rest of variables. Thus, in the case of many variables, exhaustive searches are meaningful only for the few variables with the highest powers. For many variables, code generation techniques for optimized evaluation of polynomial/rational functions could be useful alternatives (Varga and Looye, 1999). The conversion to Horner form can be easily extended to multivariate Laurent polynomials as well as can be generalized to multivariate polynomial matrices.

Example 2.1.

$$g(\boldsymbol{\delta}) = \delta_1^3 + \delta_2\delta_1^2 - 4\delta_1^2 - 4\delta_1\delta_2 + 3\delta_1 + 3\delta_2$$

The realization of this element without any preprocessing would lead to an LFT-realization of order 12. Employing the conversion to Horner form allows to express $g(\boldsymbol{\delta})$ as

$$g(\boldsymbol{\delta}) = 3\delta_2 + (3 - 4\delta_2 + (-4 + \delta_2 + \delta_1)\delta_1)\delta_1$$

which leads to an LFT-realization of order 6. For this polynomial both orderings of variables lead to the same orders of the LFT-realizations.

2.2.2. Partial Fraction Decomposition. This conversion allows to represent a rational function in an additively decomposed partial fraction form where the individual terms have usually much simpler forms. The basic decomposition consists of factoring the denominator polynomial with respect to one selected variable. An iterative realization procedure can be easily devised by performing the basic decomposition with respect to a selected order of variables. In each step, the basic decomposition is performed on all the terms computed at a previous step. For a multivariate rational function with k variables an exhaustive search for the least order of corresponding LFT-realizations involves $k!$ conversions. The main difficulty of employing this conversion is the need to compute symbolically the roots of multivariate polynomials.

Example 2.2.

$$g(\boldsymbol{\delta}) = \frac{2\delta_1^2 - 7\delta_1 - 3\delta_2 + \delta_2\delta_1^2 + \delta_2^2\delta_1 - \delta_2^2 + 3}{\delta_1^3 + \delta_2\delta_1^2 - 4\delta_1^2 - 4\delta_1\delta_2 + 3\delta_1 + 3\delta_2}$$

The partial fraction decomposition leading to the least order is given by

$$g(\boldsymbol{\delta}) = \frac{1}{\delta_1 - 1} + \frac{\delta_2}{\delta_1 - 3} + \frac{1}{\delta_1 + \delta_2} \qquad (6)$$

This decomposition results for both orderings of variables to LFT-realizations of order 4 instead of the expected order of 24.

2.2.3. Continued-Fraction Form. The conversion to continued-fraction form is useful for an efficient numerical evaluation of rational expressions and can also be applied as symbolic preprocessing for rational expressions. This conversion is usually performed for a selected variable and the resulting coefficients depend generally of the rest of variables. Although nested representations involving the representation of coefficients in continuous-fraction form are in principle possible to be computed, this computation is however not straightforward and can be frequently replaced by conversions to Horner form. The main advantage of this conversion is that it can be performed for arbitrary rational functions. In particular, for any rational function with only one parameter, this conversion allows to obtain the least order LFT-realization.

Example 2.3. The continued-fraction form of

$$g(\boldsymbol{\delta}) = \frac{8\delta_1^2 - 8\delta_1 + 2\delta_3 + 2\delta_1\delta_2 - \delta_2 + 2}{4\delta_1^2 - 4\delta_1 + \delta_3 + 1}$$

with respect to δ_1 is given by

$$g(\boldsymbol{\delta}) = 2 + \frac{\delta_2}{2\left(\delta_1 - \frac{1}{2} + \frac{\delta_3}{4\left(\delta_1 - \frac{1}{2}\right)}\right)}$$

and allows to obtain an LFT-representation of least order 4 instead of expected order 11. Note that the conversion to partial fraction form has no effect for this example on the order of the realization.

2.3 Matrix Conversions

2.3.1. Morton's Method. Any rational matrix $G(\boldsymbol{\delta}) \in \mathbb{R}(\boldsymbol{\delta})^{p\times m}$ can be expressed as an affine combination

$$G(\boldsymbol{\delta}) = G_0 + \sum_{i=1}^{n} c_i(\boldsymbol{\delta})G_i \tag{7}$$

where G_i, $i = 0, 1, \ldots, n$ are constant matrices and $c_i(\boldsymbol{\delta})$ are multivariate rational functions. Let $G_i = L_iR_i$ be full rank factorizations of G_i, where $L_i \in \mathbb{R}^{p\times r_i}$ and $R_i \in \mathbb{R}^{r_i\times m}$. The method proposed in (Morton, 1985) constructs LFT-realizations of each term $L_i(c_i(\boldsymbol{\delta})I_{r_i})R_i$ which serve to immediately obtain an LFT-realization of the whole rational matrix $G(\boldsymbol{\delta})$. The main advantage of this method in obtaining low order LFT-realizations is that it exploits the fact that frequently the constant matrices G_i have non-full ranks and this leads to an overall lower order realization of the rational matrix.

2.3.2. Enhanced Tree Decomposition. An efficient technique applicable to multivariate polynomial matrices is the *tree-decomposition* (TD) based approach proposed in (Cockburn and Morton, 1997). This method exploits the structure of a polynomial matrix to break it down into sums and products of "simple" terms and factors for which low order LFT-realizations can be easily constructed.

The TD approach can be employed to construct LFT-realizations of general rational matrices represented in polynomial fractional forms. It is well-known that any rational matrix $G(\boldsymbol{\delta}) \in \mathbb{R}(\boldsymbol{\delta})^{m\times n}$ can be expressed as a right or left factorization $G(\boldsymbol{\delta}) = N(\boldsymbol{\delta})D^{-1}(\boldsymbol{\delta})$ or $G(\boldsymbol{\delta}) = \widetilde{D}^{-1}(\boldsymbol{\delta})\widetilde{N}(\boldsymbol{\delta})$, respectively, where $N(\boldsymbol{\delta})$, $D(\boldsymbol{\delta})$, $\widetilde{N}(\boldsymbol{\delta})$, $\widetilde{D}(\boldsymbol{\delta})$ are polynomial matrices. Then, from the LFT-realizations of compound polynomial matrices $[\, N^T(\boldsymbol{\delta})\ D^T(\boldsymbol{\delta})\,]^T$ or $[\, \widetilde{N}(\boldsymbol{\delta})\ \widetilde{D}(\boldsymbol{\delta})\,]$ we can easily determine the LFT-realization of $G(\boldsymbol{\delta})$ by direct formulas (Hecker and Varga, 2004).

The *enhanced tree decomposition* (ETD) method is an extension of the TD method to the more general case of multivariate Laurent polynomial matrices $G(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}) \in \mathbb{R}[\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}]$. The enhanced method formally substitutes each reciprocal variable δ_i^{-1} in the Laurent polynomial matrix $G(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1})$ by a new variable, say $\widetilde{\delta}_i$, and applies the standard TD method to the resulting polynomial matrix $G(\boldsymbol{\delta}, \widetilde{\boldsymbol{\delta}})$. Furthermore, Morton's technique is integral part of the ETD and is applied in all cases where affine combinations of the form (7) arise as intermediate results during the decomposition. Besides the resulting lower orders of LFT-realizations, the integration of Morton's approach leads usually to significant time savings. For example, in the case of the RCAM example presented in section 4, a reduction of about 20% of the LFT realization time has been achieved.

Example 2.4. Consider the multivariate Laurent polynomial matrix

$$G(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}) = \begin{bmatrix} \frac{1}{\delta_1} + \delta_1\delta_2 & \frac{1}{\delta_1} \\ \frac{1}{\delta_1} + \frac{\delta_2}{\delta_3^2} & \frac{1}{\delta_1} \end{bmatrix}$$

By applying the ETD method, we obtain the following decomposition

$$G(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}) = T_1(\delta_1^{-1}) + T_2(\delta_1, \delta_3^{-1})T_3(\delta_2)$$

where

$$T_1(\delta_1^{-1}) = \frac{1}{\delta_1}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\frac{1}{\delta_1}\begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$\begin{aligned} T_2(\delta_1, \delta_3^{-1}) &= \delta_1\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\delta_3^2}\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}\delta_1\begin{bmatrix} 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\frac{1}{\delta_3^2}\begin{bmatrix} 1 & 0 \end{bmatrix} \end{aligned}$$

$$T_3(\delta_2) = \delta_2\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\delta_2\begin{bmatrix} 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

This decomposition leads immediately to an LFT-realization of order 5. Note that performing the standard TD on $G(\boldsymbol{\delta}, \widetilde{\boldsymbol{\delta}})$ without employing Morton's method would lead to an order 6, while a direct LFT-realization of the original matrix would lead to order 9.

2.4 Variable Splitting Factorization

For multivariate Laurent polynomials, a *variable splitting* (VS) technique can be employed to express such a polynomial in factored form, where the factors contain disjoint subsets of $\boldsymbol{\delta}$ and $\boldsymbol{\delta}^{-1}$, respectively. It is easy to show that any Laurent polynomial $g(\boldsymbol{\delta},\boldsymbol{\delta}^{-1})$ can be expressed as a product

$$g(\boldsymbol{\delta},\boldsymbol{\delta}^{-1}) = v(\boldsymbol{\delta}_{s_1},\boldsymbol{\delta}_{s_2}^{-1})^T u(\boldsymbol{\delta}_{s_3},\boldsymbol{\delta}_{s_4}^{-1})$$

where $v(\boldsymbol{\delta}_{s_1},\boldsymbol{\delta}_{s_2}^{-1})$ and $u(\boldsymbol{\delta}_{s_3},\boldsymbol{\delta}_{s_4}^{-1})$ are vectors depending on the sets of parameters $\boldsymbol{\delta}_{s_1},\boldsymbol{\delta}_{s_2}$ and $\boldsymbol{\delta}_{s_3},\boldsymbol{\delta}_{s_4}$, respectively, with $\boldsymbol{\delta} = \boldsymbol{\delta}_{s_1} \cup \boldsymbol{\delta}_{s_3} = \boldsymbol{\delta}_{s_2} \cup \boldsymbol{\delta}_{s_4}$, $\boldsymbol{\delta}_{s_1} \cap \boldsymbol{\delta}_{s_3} = \boldsymbol{\delta}_{s_2} \cap \boldsymbol{\delta}_{s_4} = \emptyset$. Typically, one chooses one of the factors, say $v(\boldsymbol{\delta}_{s_1},\boldsymbol{\delta}_{s_2}^{-1})$, to have only entries expressed by multivariate Laurent monomials. This factorization can be easily extended to row/colums vectors, in which case one of the factors becomes a matrix.

The VS factorization allows to transform the initial realization problem into two realization problems but each with fewer variables. The effectiveness of this technique in conjunction with the ETD is shown in section 4.

Example 2.5. Consider

$$g(\boldsymbol{\delta},\boldsymbol{\delta}^{-1}) = \frac{\delta_1}{\delta_2}\delta_3 + \frac{\delta_1}{\delta_2}\delta_4 + \frac{\delta_2}{\delta_1}\delta_3$$

By choosing $\boldsymbol{\delta}_{s_1} = \boldsymbol{\delta}_{s_2} = \{\delta_1,\delta_2\}$ and $\boldsymbol{\delta}_{s_3} = \boldsymbol{\delta}_{s_4} = \{\delta_3,\delta_4\}$, we obtain the VS factorization as

$$g(\boldsymbol{\delta},\boldsymbol{\delta}^{-1}) = \left[\begin{array}{cc} \frac{\delta_1}{\delta_2} & \frac{\delta_2}{\delta_1} \end{array}\right] \left[\begin{array}{c} \delta_3+\delta_4 \\ \delta_3 \end{array}\right]$$

which yields an order 6 LFT-representation (instead the expected order 9) when employed in conjunction with the ETD technique.

Example 2.6. The VS approach can easily be extended to vectors. Consider the multivariate vector

$$g(\boldsymbol{\delta},\boldsymbol{\delta}^{-1}) = \left[\begin{array}{cc} \frac{\delta_1}{\delta_2}\delta_3 + \frac{\delta_1}{\delta_2}\delta_4 + \frac{\delta_2}{\delta_1}\delta_3 & \frac{\delta_1}{\delta_2}\delta_4^2 + \frac{\delta_2}{\delta_1}\delta_4 \end{array}\right]$$

The corresponding vector based VS factorization for $\boldsymbol{\delta}_{s_1} = \boldsymbol{\delta}_{s_2} = \{\delta_1,\delta_2\}$, $\boldsymbol{\delta}_{s_3} = \boldsymbol{\delta}_{s_4} = \{\delta_3,\delta_4\}$ is

$$g(\boldsymbol{\delta},\boldsymbol{\delta}^{-1}) = \left[\begin{array}{cc} \frac{\delta_1}{\delta_2} & \frac{\delta_2}{\delta_1} \end{array}\right] \left[\begin{array}{cc} \delta_3+\delta_4 & \delta_4^2 \\ \delta_3 & \delta_4 \end{array}\right]$$

This yields an order 7 LFT-representation when employed in conjunction with the ETD. ETD applied directly to $g(\boldsymbol{\delta},\boldsymbol{\delta}^{-1})$ leads to order 8.

3. SOFTWARE TOOLS

The symbolic preprocessing techniques described in section 2 have been implemented in Version 2 of the LFR-Toolbox (Hecker *et al.*, 2004). The main function to compute LFT-realizations from symbolic multivariate rational matrices is `sym2lfr`, which provides several options to perform symbolic preprocessing, as for example, choosing among methods, performing or not an exhaustive search for least order realization (the so called "try-hard" option), etc. Some conversions are automatically performed within `sym2lfr`, as for example, the calculation of a polynomially factorized representation for a general rational matrix. The output of the function is an LFT-representation obtained via an object oriented LFT-generation, that can be further processed (normalization, numerical order reduction) with other functions of the LFR-Toolbox. Most of the underlying algorithms for symbolic preprocessing are implemented in MAPLE, thus ensuring a highly efficient symbolic manipulation. These algorithms are executed via the Extended Symbolic Toolbox of MATLAB, providing an user friendly interface to the MAPLE symbolic kernel.

4. APPLICATION EXAMPLES

Example 4.1. The capability of the combined VS and ETD approach can be best illustrated by applying this technique to the most complicated single element $a_{29}(\boldsymbol{\delta},\boldsymbol{\delta}^{-1})$ of the state matrix $A(\boldsymbol{\delta},\boldsymbol{\delta}^{-1})$ of the extended parametric Research Civil Aircraft Model (RCAM) (Varga *et al.*, 1998), one of the most complicated parametric models existing in the literature.. The uncertain parameter vector is $\boldsymbol{\delta} = (m, V, x_{cg}, z_{cg})$, where m is the aircraft mass, V is the air-speed, and x_{cg}, z_{cg} are the x-axis and z-axis components of the center of gravity position, respectively. The VS factorization of $a_{29}(\boldsymbol{\delta},\boldsymbol{\delta}^{-1}) = v(\boldsymbol{\delta}_{s_1},\boldsymbol{\delta}_{s_2}^{-1})^T u(\boldsymbol{\delta}_{s_3},\boldsymbol{\delta}_{s_4}^{-1})$ with $\boldsymbol{\delta}_{s_1} = \boldsymbol{\delta}_{s_2} = \{m, V\}$ and $\boldsymbol{\delta}_{s_3} = \boldsymbol{\delta}_{s_4} = \{x_{cg}, z_{cg}\}$ yields

$$v(\boldsymbol{\delta}_{s_1},\boldsymbol{\delta}_{s_2}^{-1}) = \left[\begin{array}{c} \frac{V}{m} \\ \frac{m}{V^3} \\ \frac{1}{V} \end{array}\right], \quad u(\boldsymbol{\delta}_{s_3},\boldsymbol{\delta}_{s_4}^{-1}) = \left[\begin{array}{c} u_1 \\ u_2 \\ u_3 \end{array}\right]$$

with

$$\begin{aligned} u_1 &= -46.849 + 100.133x_{cg} - 14.2516z_{cg} \\ &\quad -0.2556x_{cg}^2 - 0.0710x_{cg}^2 z_{cg} \\ &\quad +1.1243x_{cg}z_{cg} \\ u_2 &= 0.0022 - 0.0103x_{cg} - 0.0073z_{cg} \\ &\quad +0.0017x_{cg}^2 - 0.0011x_{cg}^2 z_{cg} \\ &\quad +0.0063x_{cg}z_{cg} \\ u_3 &= -0.0828 + 0.3060x_{cg} + 1.5189z_{cg} \\ &\quad -0.0106x_{cg}^2 + 0.0183x_{cg}^2 z_{cg} \\ &\quad -0.2422x_{cg}z_{cg} \end{aligned}$$

The application of the ETD yields LFT-representations of order 6 for $v(\boldsymbol{\delta}_{s_1},\boldsymbol{\delta}_{s_2}^{-1})$ and order 5 for $u(\boldsymbol{\delta}_{s_3},\boldsymbol{\delta}_{s_4}^{-1})$ and finally an LFT-representation of order 11 is obtained for $a_{29}(\boldsymbol{\delta},\boldsymbol{\delta}^{-1})$. Remarkably, the resulting order 11 obtained exclusively by symbolic preprocessing is smaller than "least" orders about 15 achieved by combining various symbolic and numerical order reduction tools in (Varga and Looye, 1999), starting from initial realization of orders as large as 193.

Example 4.2. The parametric state space matrices $A(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}), B(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}), C(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}), D(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1})$ of the RCAM have only elements as Laurent polynomials in the indeterminates. We computed several LFT-representations of the system matrix

$$S(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}) = \begin{bmatrix} A(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}) & B(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}) \\ C(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}) & D(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1}) \end{bmatrix}$$

and the results are presented in Table 1, where for each specific symbolic preprocessing we give in the successive columns the resulting orders without and with additional numerical n-D order reduction (D'Andrea and Khatri, 1997).

Table 1. Orders of LFT-realizations for the extended RCAM example.

Symbolic Preprocessing	Order	Order (numerically reduced)
None	400	262
Single Element	307	158
TD	156	97
ETD	131	86
VS+ETD	71	65

Without any symbolic preprocessing an order of 262 can be achieved by using numerical order reduction. Using various symbolic techniques on single matrix elements followed by application of numerical n-D order reduction, an LFT representation of order 158 has been computed in (Varga and Looye, 1999). The TD algorithm for a polynomially factorized representation as proposed in (Cockburn and Morton, 1997) yields an LFT-model of order 156, which can be numerically reduced to order 97. The ETD yields an LFT-representation of order 131, which can be exactly reduced to order 86. By employing the combined VS and ETD approach in conjunction with the "try-hard" option, we obtained an LFT-representation of $S(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1})$ with order 71 and we were able to exactly reduce this LFT-representation to order 65, which is very close to the theoretical least order bound of 56. In this specific case, the VS factorization has been applied to the columns of $S(\boldsymbol{\delta}, \boldsymbol{\delta}^{-1})$ using the variable splitting $\boldsymbol{\delta}_{s_1} = \boldsymbol{\delta}_{s_2} = \{m, V\}$ and $\boldsymbol{\delta}_{s_3} = \boldsymbol{\delta}_{s_4} = \{x_{cg}, z_{cg}\}$. For each VS factorized column, the ETD has been employed.

5. CONCLUDING REMARKS

We presented several symbolic manipulation techniques which are potentially useful in obtaining low order LFT-representations of parametric matrices. Both ad-hoc as well as systematic methods have been discussed, and their capabilities have been illustrated via simple examples. All presented methods have been implemented via user-friendly interfaces in the newly developed Version 2 of the LFR-toolbox.

Low orders of the LFT-representations can also be achieved by employing numerical order reduction in a postprocessing phase. However, in contrast to the numerically sensitive order reduction techniques based on tolerance dependent rank decisions, symbolic preprocessing can be applied without any loss of accuracy (floating-point numbers are represented exactly in rational form). Therefore, symbolic preprocessing and numerical postprocessing can be seen as complementary tools which can be efficiently used to obtain low order LFT-realizations. The strength of this combination approach in obtaining low order LFT-realizations has been illustrated by applying these techniques to a challenging aircraft uncertainty modelling problem.

REFERENCES

Bose, N. K. (1982). *Applied Multidimensional Systems Theory*. Van Nostrand Reinhold Company.

Cockburn, J.C. and B.G. Morton (1997). Linear fractional representation of uncertain systems. *Automatica* **33**(7), 1263–1271.

D'Andrea, R. and S. Khatri (1997). Kalman decomposition of linear fractional transformation representations and minimality. *Proc. of the American Control Conference*. Albulquerque, New Mexico. pp. 3557–3561.

Hecker, S., A. Varga and J.F. Magni (2005). Enhanced LFR Toolbox for MATLAB. *Aerospace Science and Technology*, **8**.

Hecker, S. and A. Varga (2004). Generalized LFT-based representation of parametric uncertain models. *European Journal of Control*, **10**.

Morton, B. (1985). New applications of μ to real parameter variation problems. *Proc. Conference on Decision and Control*. Fort Lauderdale, Florida. pp. 233–238.

Terlouw, J., P. Lambrechts, S. Bennani and M. Steinbuch (1993). Parametric uncertainty modeling using LFTs. *Proc. American Control Conference*. San Francisco, CA. pp. 267–272.

Varga, A. and G. Looye (1999). Symbolic and numerical software tools for LFT-based low order uncertainty modeling. *Proc. CACSD'99 Symposium, Kohala Coast, Hawaii*.

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

Zhou, K., J. C. Doyle and K. Glover (1996). *Robust and Optimal Control*. Prentice Hall.