JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 31, No. 4, July–August 2008

Improved μ-Analysis Results by Using Low-Order Uncertainty Modeling Techniques

Simon Hecker*
DLR, German Aerospace Center, 82234 Wessling, Germany

DOI: 10.2514/1.33050

The paper describes some recent enhancements of symbolic factorization techniques for multivariate Laurent polynomial matrices, which are very effective as a preprocessing step for generating linear fractional representations of low order. The symbolic methods are applied to realize a linear fractional representation of full accuracy and probably minimal order for a very complex linear parametric representation of a Research Civil Aircraft Model. The μ-analysis results obtained with the very accurate and low-order linear fractional representation coincide with the robust stability analysis results obtained from a computationally demanding, optimization-based, worst-case search. Furthermore, it is shown that the quality of the μ-analysis results depends on the accuracy and order of the underlying model, i.e., the tightest upper and lower bounds are obtained using the most accurate and least-order linear fractional representation.

I. Introduction

MANY physical systems can be represented as a linear parametric state-space system of the form

$$\dot{x} = A(\delta)x + B(\delta)u \qquad y = C(\delta)x + D(\delta)u \tag{1}$$

where $A(\delta)$, $B(\delta)$, $C(\delta)$, and $D(\delta)$ are real matrices depending rationally on k parameters grouped in the vector $\delta = (\delta_1, \ldots, \delta_k)$. In aircraft or automotive models, this parameter vector may include the mass or the position of the center of gravity, which are usually not exactly known (i.e., are uncertain) or time varying. To analyze, if the system given in Eq. (1) is robustly stable for all allowable parameter variations is not a trivial task and may be performed using exhaustive search techniques like parameter gridding or Monte Carlo analysis. Alternatively, one may use more efficient techniques like μ-analysis coming from modern robust control theory [1], where a parameter gridding can be avoided. For the application of these modern robust control techniques, it is necessary to transform Eq. (1) into a special uncertain system description called linear fractional representation (LFR) [1].

LFR realization problem: Given the $p_2 \times m_2$ real matrix

$$G(\delta) = \begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix} \tag{2}$$

Find matrices M and Δ such that

$$G(\delta) = \mathcal{F}_u(M, \Delta) = M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12} + M_{22} \tag{3}$$

where

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \in \mathbb{R}^{(p_1+p_2)\times(p_1+m_2)} \tag{4}$$

and

$$\Delta = \operatorname{diag}(\delta_1 I_{r_1}, \ldots, \delta_k I_{r_k}) \tag{5}$$

with order r, defined as

$$r = p_1 = \sum_{i=1}^{k} r_i$$

Representing a parameter-dependent matrix as an LFR is basically equivalent to a multidimensional realization problem [2]. No unique realization (M, Δ) exists and a theory to find a realization of minimal order is still lacking. However, to improve the accuracy and to reduce the computational effort for the LFR-based robust control techniques [1], it is of paramount importance to use LFRs of *high accuracy and least possible orders*. One of the most promising approaches to generate low-order LFRs is to apply a three step procedure: 1) symbolic preprocessing of $G(\delta)$ [3–6], 2) object-oriented LFR realization [7,8], 3) numerical order reduction [8,9].

For a given parametric matrix $G(\delta)$, the symbolic preprocessing allows one to find equivalent representations of this matrix, which automatically lead to lower order LFRs, when employed in conjunction with the object-oriented LFR realization approach.

The object-oriented LFR realization method starts with the generation of elementary LFR objects for the uncertain parameters δ_i, i.e.,

$$\delta_i = \mathcal{F}_u(M_i, \delta_i) = \mathcal{F}_u\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \delta_i\right), \qquad i = 1, \ldots, k \tag{6}$$

Then, basic formulas for the interconnection of LFRs are employed to finally realize an LFR for $G(\delta)$. As an example, the formula for the addition of two LFRs is given by

$$\mathcal{F}_u\left(\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix}, \Delta_1\right) + \mathcal{F}_u\left(\begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix}, \Delta_2\right) = \mathcal{F}_u\left(\begin{bmatrix} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ C_1 & C_2 & D_1 + D_2 \end{bmatrix}, \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix}\right) \tag{7}$$

and corresponding formulas for multiplication, inversion, and row/column concatenation can be found in [7]. The method is very flexible and can easily be automated. However, a blindfold application of this method may yield LFRs of larger order than the least possible one.

Example 1: Using Eq. (7), an LFR $\mathcal{F}_u(M, \Delta)$ for $G(\delta) = \delta_1 + \delta_2$ is obtained as

Presented at the AIAA Guidance, Navigation, and Control Conference 2007, Hilton Head Island, SC, 20–23 August 2007; received 25 June 2007; revision received 17 October 2007; accepted for publication 2 November 2007.

*Institute of Robotics and Mechatronics, Oberpfaffenhofen.

$$M = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \qquad \Delta = \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix} \tag{8}$$

This LFR has the minimal order of two. However, applying the preceding construction to the expression $G(\delta) = \delta_1 + \delta_1$, one may realize it as an LFR of order two with M given in Eq. (8) and $\Delta = \mathrm{diag}(\delta_1, \delta_1)$. Obviously, a first-order LFR for $G(\delta) = \delta_1 + \delta_1$ is possible starting from an equivalent expression

$$G(\delta) = 2\delta_1 = \mathcal{F}_u\left(\begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix}, \delta_1\right) \tag{9}$$

From this very simple example, it can be seen that the trivial symbolic simplification of $G(\delta) = \delta_1 + \delta_1$ to $G(\delta) = 2\delta_1$ allows us to reduce the LFR order by one when the object-oriented LFR realization technique is directly applied. Therefore, it is clear that the resulting order of the generated LFR depends crucially on the way the expressions underlying the LFR realization are evaluated, and the role of symbolic preprocessing is to find simpler evaluation schemes of rational expressions and matrices which finally lead to LFRs of lower order. Several ad hoc or systematic symbolic preprocessing techniques, ranging from simple polynomial factorizations to more complex matrix decomposition algorithms, are available. In several practical examples, the symbolic preprocessing appears to be the most important step in generating low-order LFRs.

The final step in the LFR realization procedure is to apply numerical order reduction. To some level, these methods allow a further separation and truncation of nonminimal parts of the LFR, or one may calculate a sufficiently accurate numerical approximation of the original LFR. However, compared with the symbolic preprocessing, there are two main drawbacks. First, the uncertain parameters δ_i are considered as noncommuting operators [9], which drastically limits the order reduction capability. Second, these techniques are always based on tolerance-dependent rank decisions, which introduce approximation errors in the resulting reduced-order LFRs. This will be shown in Sec. III.C.

In this paper, some enhancements of a recently introduced symbolic preprocessing technique called variable splitting (VS) factorization [3] are presented. When applied to parametric matrices, these enhancements yield factors of smaller dimension, which generally allow one to generate LFRs of lower order using the object-oriented LFR realization approach. Using this symbolic preprocessing technique, an LFR of possibly minimal order is generated for the Research Civil Aircraft Model (RCAM) [10], which is one of the most complicated parametric models available in literature. The robust stability analysis for 12 controllers performed with this LFR coincide with the results obtained from a computationally demanding global search approach, which was applied in the GARTEUR (Group for Aeronautical Research and Technology in Europe) project on robust flight control [10]. This approach combines parameter gridding and optimization-based worst-case search to find parameter combinations that yield worst-case, closed-loop stability properties and takes about 5.5 h for each controller, whereas the LFR-based analysis takes only some minutes. Furthermore, it is shown that the quality of the LFR-based stability analysis depends on the accuracy and the order of the underlying LFR, i.e., the smallest gap between the upper and lower bounds calculated during μ-analysis is obtained when the highest accuracy and least-order LFR is used.

II. Enhanced Variable Splitting Factorization

A. Definitions

Let $\delta = (\delta_1, \ldots, \delta_k)$ be a parameter vector and denote by $\delta^{-1} = (\delta_1^{-1}, \ldots, \delta_k^{-1})$ the vector of reciprocal variables. Two classes of matrices depending on δ are considered, for which the VS factorization can be directly applied: $\mathbb{R}[\delta]^{m\times n}$ (the set of $m \times n$ matrices with multivariate polynomial entries), and $\mathbb{R}[\delta, \delta^{-1}]^{m\times n}$ (the set of $m \times n$ matrices with multivariate Laurent polynomial entries). This last case is explicitly considered because many aircraft and automotive related parametric models are described in terms of such matrices. Note that for general rational parametric matrices $G(\delta)$, one can always find a left (or right) (Laurent) polynomial factorization $G(\delta) = D^{-1}(\delta)N(\delta)$ [or $G(\delta) = \tilde{N}(\delta)\tilde{D}^{-1}(\delta)$] such that the VS factorization can be applied to the concatenated matrix $[\,N(\delta) \quad D(\delta)\,]$ (or $[\,\tilde{N}(\delta)^T \quad \tilde{D}(\delta)^T\,]^T$).

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

$$g(\delta) = \sum_{r=1}^{l} c_r \delta_1^{n_{r,1}} \delta_2^{n_{r,2}} \ldots \delta_k^{n_{r,k}} \tag{10}$$

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 LFR which results when applying the object-oriented LFR realization approach, as described in [7,8], to the polynomial in the preceding expanded form. This order is given by

$$\mathrm{ord}\,[g(d)] = \sum_{r=1}^{l} \sum_{s=1}^{k} |n_{r,s}| \tag{11}$$

where we assumed that negative and positive powers of indeterminates contribute in the same way to the order. This assumption is valid when employing the generalized LFR [7] to realize the reciprocal variables.

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

$$\mathrm{ord}\,[G(\delta)] = \sum_{i=1}^{m} \sum_{j=1}^{n} \mathrm{ord}[g_{ij}(\delta)] \tag{12}$$

which corresponds to realize $G(\delta)$ elementwise using row and column concatenations via the object-oriented LFR realization approach.

The role of symbolic preprocessing in building low-order LFRs of a given $G(\delta)$ is to find equivalent representations of individual matrix elements, entire rows/columns or even the whole matrix, which lead to LFRs of lower order than given by Eq. (12). Note, in the following, it is always assumed that the LFRs are generated using the object-oriented LFR realization approach.

B. Calculation of a Lower Bound for the LFR Order

The LFR order given in Eq. (12) is obtained after direct application of the object-oriented LFR realization to the parametric matrix $G(\delta)$. Symbolic preprocessing may help to find LFRs of lower orders but, as already mentioned, there exists no solution for the generation of a minimal order LFR for a multivariate parametric matrix [2]. Furthermore, for a given parametric matrix $G(\delta)$, there exists no way to calculate to minimal achievable LFR order. However, to quantify the complexity of an LFR, which is obtained after application of symbolic preprocessing, object-oriented LFR realization, and exact numerical order reduction [9], it is important to have at least a lower bound for the achievable minimal LFR order.

A simple procedure to determine a lower bound for the LFR order of a polynomial parametric matrix $G(\delta)$ is to determine for each parameter δ_i the maximum power m_{δ_i} over all matrix entries $g_{i,j}(\delta)$. A lower bound is then given by [4]

$$\sum_{i=1}^{k} m_{\delta_i}$$

As an example, a lower bound of s is obtained for the parametric vector $G(\delta) = [\delta_1^s \;\; \delta_1^{s-1} \;\; \ldots \;\; \delta_1]$, where the maximum power of δ_1 over all vector entries is s. In this case the, lower bound is exact and describes the minimal LFR order for this parametric vector. However, the same lower bound is obtained for the parametric matrix $G(\delta) = \mathrm{diag}(\delta_1^s, \delta_1^{s-1}, \ldots, \delta_1)$, where the minimal LFR order is $s(s+1)/2$. This shows that for parametric matrices, this method may result in a very bad estimate for the minimal LFR order, which comes from the fact that the structural information for the occurrence of the parametric expressions in the matrix is not considered.

To overcome this problem, the following procedure to calculate a more accurate lower bound for the minimal LFR order is proposed:

1) Set counter $i = 1$.

2) Substitute all parameters in $G(\delta)$, except δ_i, with random values resulting in the one-parametric matrix $G_i(\delta_i)$.

3) Construct a minimal-order LFR $(M_i, \Delta_i = \delta_i I_{m_i})$ for $G_i(\delta_i)$.

4) If $i < k$ then increment i and go to step 3, otherwise, go to step 5.

5) The lower bound is given by

$$m = \sum_{i=1}^{k} m_i$$

The proposed procedure can be easily implemented and, for practical examples, it yields quite good estimates (see Sec. III and [7]). Note that for one-parametric systems, a minimal-order LFR can always be generated (step 3 of the preceding procedure). For the aforementioned matrix $G(\delta) = \mathrm{diag}(\delta_1^s, \delta_1^{s-1}, \ldots, \delta_1)$, the lower bound will now be exactly the minimal LFR order. The following example shows that, in some cases, a gap between the lower bound calculated with proposed procedure and the exact minimal LFR order cannot be avoided, and, to the best of the author's knowledge, there exists no method to exactly calculate the minimal achievable LFR order.

Example 2: Consider the parametric vector $G(\delta) = [\delta_1\delta_2 \;\; \delta_1 + \delta_2]$. In this case, the procedure yields a lower bound of $m = 2$. However, there will exist no LFR of order less than three.

C. Variable Splitting Factorization: Scalar Case

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

$$g(\delta, \delta^{-1}) = v(\delta_{s_1}, \delta_{s_2}^{-1})^T u(\delta_{s_3}, \delta_{s_4}^{-1}) \tag{13}$$

where $v(\delta_{s_1}, \delta_{s_2}^{-1})$ and $u(\delta_{s_3}, \delta_{s_4}^{-1})$ are vectors, depending on the parameter vectors δ_{s_1},δ_{s_2} and δ_{s_3},δ_{s_4}, respectively. The parameter index sets $s, s_1, \ldots, s_4$ are chosen such that $s = (1, \ldots, k)$, $s = s_1 \cup s_3 = s_2 \cup s_4$, $s_1 \cap s_3 = s_2 \cap s_4 = \emptyset$, and δ_s denotes a parameter vector including all parameters δ_i with $i \in s$.

Typically, one chooses one of the factors, say $v(\delta_{s_1}, \delta_{s_2}^{-1})$, to have only entries expressed by multivariate Laurent monomials. The VS factorization allows one to transform the initial realization problem into two realization problems, but each with fewer variables. This technique is beneficial for cases where scalar techniques, such as transformation to Horner form, do not allow a splitting of variables.

Example 3: Consider

$$g(\delta) = \delta_1 + \delta_1\delta_2 + \delta_2 \tag{14}$$

Using scalar factorizations as transformation to Horner form, one obtains either $g(\delta) = \delta_1(1 + \delta_2) + \delta_2$ or $g(\delta) = \delta_1 + (\delta_1 + 1)\delta_2$. Both representations allow one to reduce the resulting LFR order from four to three.

By choosing $\delta_{s_1} = \delta_1$ and $\delta_{s_3} = \delta_2$, we obtain the VS factorization as

$$g(\delta) = [\delta_1 \;\; 1]\begin{bmatrix} 1 + \delta_2 \\ \delta_2 \end{bmatrix} \tag{15}$$

which allows an additional decomposition, such that

$$g(\delta) = [\delta_1 \;\; 1]\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}\delta_2 + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) \tag{16}$$

yielding an LFR of minimal order two.

In the following, the VS factorization for the scalar case, which was already presented in [3], will be extended to allow an application to vectors and matrices.

D. Variable Splitting Factorization: Vector Case

The VS factorization can be simply extended to parametric row vectors of the form

$$g_{\mathrm{vec}} = [g_1(\delta, \delta^{-1}) \;\; \ldots \;\; g_m(\delta, \delta^{-1})] \tag{17}$$

as

$$\begin{aligned} g_{\mathrm{vec}} &= V^T U \\ &= \begin{bmatrix} v_1(\delta_{s_1,1}, \delta_{s_2,1}^{-1}) \\ \vdots \\ v_m(\delta_{s_1,m}, \delta_{s_2,m}^{-1}) \end{bmatrix}^T \begin{bmatrix} u_1(\delta_{s_3,1}, \delta_{s_4,1}^{-1}) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & u_m(\delta_{s_3,m}, \delta_{s_4,m}^{-1}) \end{bmatrix} \end{aligned} \tag{18}$$

where $v_i(\delta_{s_1,i}, \delta_{s_2,i}^{-1})$ and $u_i(\delta_{s_3,i}, \delta_{s_4,i}^{-1})$, $i = 1, \ldots, m$, are parametric column vectors.

A limitation of the simple vector extension of the VS factorization is that the factor U in Eq. (18) does not allow one to exploit any dependencies and common factors of the $u_i(\delta_{s_3,i}, \delta_{s_4,i}^{-1})$ for a further decomposition, which may yield a further reduction of the resulting LFR order. This can be simply improved by applying the following *condensation algorithm*, which condenses the factors V and U of Eq. (18) in cases where elements of V only differ by a constant factor $c \in \mathbb{R}$. Therefore, we introduce a matrix $P_{ij,c}$, which is an identity matrix of appropriate dimension, where the element p_{ij}, $i \neq j$ is substituted by a constant $c \in \mathbb{R}$ and the jth row is removed. A left multiplication of U with $P_{ij,c}$ yields $\tilde{U}$, where c times the jth row of U is added to the ith row of U and the jth row of U is removed. As an example, consider $U = \mathrm{diag}(u_1, u_2, u_3)$. Left multiplying U with $P_{12,4}$ yields

$$P_{12,4}U = \begin{bmatrix} 1 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} u_1 & 0 & 0 \\ 0 & u_2 & 0 \\ 0 & 0 & u_3 \end{bmatrix} = \begin{bmatrix} u_1 & 4u_2 & 0 \\ 0 & 0 & u_3 \end{bmatrix} \tag{19}$$

After the application of Algorithm 1, the condensed version of U may now have more than one nonzero entry in a row. If these entries have common factors, a further reduction of the resulting LFR order may be achieved.

Example 4: Consider the VS factorization of the form in Eq. (18) for the parametric row vector g_{vec} given by

$$\begin{aligned} g_{\mathrm{vec}}(\delta) &= [g_1(\delta)|g_2(\delta)] = [2\delta_1\delta_2 + 2\delta_2|\delta_1\delta_2 + \delta_1\delta_3 + \delta_2] \\ &= \begin{bmatrix} v_1(\delta) \\ v_2(\delta) \end{bmatrix}^T \begin{bmatrix} u_1(\delta) & 0 \\ 0 & u_2(\delta) \end{bmatrix} = \begin{bmatrix} 2\delta_1 \\ 2 \\ \delta_1 \\ 1 \end{bmatrix}^T \begin{bmatrix} \delta_2 & 0 \\ \delta_2 & 0 \\ 0 & \delta_2 + \delta_3 \\ 0 & \delta_2 \end{bmatrix} \end{aligned} \tag{20}$$

where an application of the enhanced tree decomposition (ETD) [3] yields

Algorithm 1 Condensation$_{\mathrm{vec}}(V, U)$

```
1: L = length(V)
2: if L > 1 then
3:   for i = 1 to L − 1 do
4:     for j = (i + 1) to L do
5:       if V(j) = cV(i) then
6:         Remove V(j) from V
7:         U = P_{ij,c} U
8:         L = L − 1
9:       end if
10:    end for
11:  end for
12: end if
```

Algorithm 2 Condensation$_{\text{Mat}}(\bar{V}, \bar{U})$

```
L = rowlength(Ū)
if L > 1 then
  for i = 1 to L − 1 do
    for j = (i + 1) to L do
      if Ū(j) = cŪ(i) then
        Remove row Ū(j) from Ū
        V̄ = V̄P̄_{ij,c}
        L = L − 1
      end if
    end for
  end for
end if
```

$$g_{\text{vec}}(\delta) = \left(\delta_1 \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T + \begin{bmatrix} 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}^T \right) \left(\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_2 & 0 \\ 0 & \delta_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \delta_3 \\ 0 & 0 \end{bmatrix} \right) \tag{21}$$

allowing one to obtain an LFR of order four instead of order eight for a direct realization of g_{vec} without symbolic preprocessing. However, applying Algorithm 1 after generating the VS factorization of the form in Eq. (18) yields

$$g_{\text{vec}}(\delta) = \begin{bmatrix} 2\delta_1 \\ 2 \end{bmatrix}^T \begin{bmatrix} \delta_2 & 0.5(\delta_2 + \delta_3) \\ \delta_2 & 0.5\delta_2 \end{bmatrix} \tag{22}$$

and further application of the ETD yields

$$g_{\text{vec}}(\delta) = \begin{bmatrix} 2\delta_1 \\ 2 \end{bmatrix}^T \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} \delta_2 [1 \;\; 0.5] + \begin{bmatrix} 0 & .5\delta_3 \\ 0 & 0 \end{bmatrix} \right) \tag{23}$$

allowing one to obtain an LFR of minimal order three.

E. Variable Splitting Factorization: Matrix Case

The VS factorization for vectors can be simply extended to parametric matrices as

$$G = \begin{bmatrix} g_{\text{vec},1} \\ \vdots \\ g_{\text{vec},p} \end{bmatrix} = \bar{V}\,\bar{U} = \begin{bmatrix} V_1^T & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & V_p^T \end{bmatrix} \begin{bmatrix} U_1 \\ \vdots \\ U_p \end{bmatrix} \tag{24}$$

The matrix $\bar{V}$ has only one nonzero element in each column. In analogy to the vector case of the VS factorization, a condensation of the matrices $\bar{V}$ and $\bar{U}$ in Eq. (24) may be possible. The condensed version of matrix $\bar{V}$ may have more than one nonzero entry in a column, which can be further exploited to reduce the resulting LFR order. The condensation of $\bar{V}$ and $\bar{U}$ is performed using Algorithm 2, where $\bar{P}_{ji,c}$ is an identity matrix of appropriate dimension, where the element $\bar{p}_{ji}$, $i \neq j$ is substituted by a constant $c \in \mathbb{R}$ and the jth column is removed. The ith row of $\bar{U}$ is denoted as $\bar{U}(i)$.

III. Robust Stability Analysis for the Research Civil Aircraft Model

In the GARTEUR Action Group 8 on robust flight control [10], one of the benchmark problems was the design of an autopilot for the final segments of a landing approach for a fictitious transporter aircraft, which is referred to as RCAM. The control laws had to be robust with respect to uncertain parameters, time delays, nonlinearities, and engine failures. For comparability, we keep the numbers 11, 12, 13, 14, 15, 16, 18, 19, 21, 22, 24, and 25, as given in [10], to denote the 12 controllers that are investigated in the following.

The linear, rational parametric system matrices $A(\delta)$, $B(\delta)$, $C(\delta)$, and $D(\delta)$ of the RCAM can be found in [10] and represent one of the most complicated parametric models available in literature. The model has 5 inputs, 12 states, and 15 measured outputs. The uncertain parameter vector δ of the RCAM is given as $\delta = (m, V, X_{\text{cg}}, Z_{\text{cg}})$, where m is the mass of the aircraft, V is the airspeed, and the horizontal and vertical position of the center of gravity are given by X_{cg} and Z_{cg}, respectively. The corresponding ranges of variation are summarized in Table 1.

Table 1 Parameter ranges of variation

Parameter	Range	Nominal value	Unit
m	[100,000, 150,000]	125,000	kg
V	[70, 90]	80	m/s
X_{cg}	[1, 2.05]	1.525	m
Z_{cg}	[0, 1.39]	0.695	m

The overall structure for robust stability analysis is presented in Fig. 1, where r is the reference signal, y is the output, and K is the controller, which is chosen from the set of 12 linearized controllers.

For each input $[u(1)$–$u(5)]$ of the RCAM, an actuator model is defined and the dynamic transfer matrix of the block G_a is given by

$$G_a = \begin{bmatrix} \frac{6.667}{s+6.667} & 0 & 0 & 0 & 0 \\ 0 & \frac{6.667}{s+6.667} & 0 & 0 & 0 \\ 0 & 0 & \frac{3.333}{s+3.333} & 0 & 0 \\ 0 & 0 & 0 & \frac{0.6667}{s+0.6667} & 0 \\ 0 & 0 & 0 & 0 & \frac{0.6667}{s+0.6667} \end{bmatrix}$$

Furthermore, an uncertain time delay $\tau[0.05, 0.1]$ s is assumed at each actuator input. The delay is approximated with a first-order Padé filter as

$$e^{-\tau s} \approx \frac{2 - \tau s}{2 + \tau s}$$

and is reasonably accurate up to a frequency of ± 10 rad/s. Because τ is identical for each actuator input, the uncertainty matrix Δ_τ of the LFR (M_τ, Δ_τ) is given by $\Delta_\tau = \tau I_5$. In addition, a small extra complex perturbation is added. As it is assumed that $\|\Delta_c\|_\infty = \|\text{diag}(\delta_{c_1}, \delta_{c_2}, \delta_{c_3}, \delta_{c_4}, \delta_{c_5})\|_\infty \leq 1$, the perturbation $0.01\Delta_c$ is only 1% and may, for example, account for unmodeled dynamics or gain and phase variations at the input of the actuators. However, the main motivation for the introduction of these small complex uncertainties is that the computation of the lower μ bounds becomes tractable [11]. The final interconnection structure for analysis is shown in Fig. 2, where

$$\Delta_T = \text{diag}(mI_{r_1}, VI_{r_2}, X_{\text{cg}}I_{r_3}, Z_{\text{cg}}I_{r_4}, \tau I_5, \delta_{c_1}, \delta_{c_2}, \delta_{c_3}, \delta_{c_4}, \delta_{c_5}) \tag{25}$$

with m, V, X_{cg}, Z_{cg}, $\tau \in \mathbb{R}$, and $\delta_{c_1}, \delta_{c_2}, \delta_{c_3}, \delta_{c_4}, \delta_{c_5} \in \mathbb{C}$.

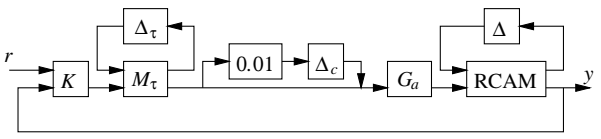

Fig. 1 Detailed structure for robust stability analysis for RCAM.

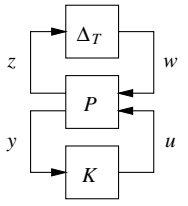

Fig. 2 Compact structure for robust stability analysis for RCAM.

In [10], the PUM (parametric uncertainty modeling) toolbox for MATLAB [12] was employed to develop LFRs for the RCAM. The PUM toolbox supports object-oriented LFR realization and one-dimensional repeated order reduction [8]. Using this tool, two LFRs with the following block structures $\Delta = \mathrm{diag}(mI_{r_1}, VI_{r_2}, X_{\mathrm{cg}}I_{r_3}, Z_{\mathrm{cg}}I_{r_4})$ and total order $r = \sum r_i$ were realized (see Table 2).

For robust stability analysis using μ analysis, the total order of model 2 was too large, which led to numerical problems using the available μ-analysis software. Therefore, in [10], it was decided to perform the stability analysis for RCAM only for fixed nominal speed $V = V_{\mathrm{nom}} = 80$ m/s. Substituting the uncertain parameter V with its nominal value in the parametric system matrices allowed the authors to realize model 1, which was of order 35. However, the results from stability analysis were overly optimistic, as the variation in speed was not considered. This could be deduced as, in parallel to the μ analysis, an optimization-based, worst-case search considering also variations in speed was applied. For some controllers that were developed in [10], the μ analysis indicated robust stability, whereas the worst-case search detected parameter combinations that resulted in unstable closed-loop systems.

To overcome these problems, the recent version 2 of the LFR-toolbox [13] is employed to generate new, low-order LFRs for the RCAM. The toolbox offers a variety of symbolic preprocessing techniques, including also the previously described enhanced VS factorization for vectors and matrices. The final LFR includes all uncertainties of the RCAM and is of reasonable order for robust stability analysis using μ.

A. LFR Model Realization for the Element a_{29}

As a starting point, the effectiveness of the proposed methods for the realization of low-order LFRs will be demonstrated by comparing different symbolic preprocessing, which are applied to the most complicated element, a_{29}, of the parametric system matrix $A(\delta)$. The expression of a_{29} can be put into the form $a_{29} = 0.061601(\tilde{a}_{29}/C_w V)$, where

$$
\begin{aligned}
C_w &= \tfrac{mg}{(1/2)\rho V^2 S} \\
\tilde{a}_{29} &= 1.6726X_{\mathrm{cg}}C_w^2Z_{\mathrm{cg}} - 0.17230X_{\mathrm{cg}}^2C_w - 3.9324X_{\mathrm{cg}}C_wZ_{\mathrm{cg}} \\
&\quad - 0.28903X_{\mathrm{cg}}^2C_w^2Z_{\mathrm{cg}} - 0.070972X_{\mathrm{cg}}^2Z_{\mathrm{cg}} + 0.29652X_{\mathrm{cg}}^2C_wZ_{\mathrm{cg}} \\
&\quad + 4.9667X_{\mathrm{cg}}C_w - 2.7036X_{\mathrm{cg}}C_w^2 + 0.58292C_w^2 - 0.25564X_{\mathrm{cg}}^2 \\
&\quad - 1.3439C_w + 100.13X_{\mathrm{cg}} - 14.251Z_{\mathrm{cg}} - 1.9116C_w^2Z_{\mathrm{cg}} \\
&\quad + 1.1243X_{\mathrm{cg}}Z_{\mathrm{cg}} + 24.656C_wZ_{\mathrm{cg}} + 0.45703X_{\mathrm{cg}}^2C_w^2 - 46.850
\end{aligned}
$$

and $S = 260$ m^2 (wing planform area), $g = 9.81$ m/s^2, and $\rho = 1.225$ kg/m^3 (air density).

Without any symbolic preprocessing, an LFR on the order of 69 could be generated using the object-oriented LFR realization procedure based on the generalized LFR [7]. In Table 3, the results obtained using three different symbolic preprocessing techniques are presented, where for each specific symbolic method, the resulting total orders without and with additional exact numerical n-dimensional order reduction [9] are given in the successive columns. Here, TD denotes the structured tree decomposition [4] and ETD the enhanced tree decomposition [3].

The total orders r without further exact numerical order reduction are ranging from 31 to 11. The ETD technique, which is especially suited for Laurent polynomial expressions, clearly outperforms the TD technique, which is applied to a polynomially factorized representation of a_{29} as proposed in [4].

Table 2 Orders of LFRs realized in GARTEUR AG 08

Model	$\{r_1, r_2, r_3, r_4\}$	r
1	$\{17, 0, 15, 3\}$	35
2	$\{47, 109, 30, 7\}$	193

The best result is obtained by the VS + ETD approach. The VS factorization of $a_{29}(\delta) = v(\delta_{s_1}, \delta_{s_2}^{-1})^T u(\delta_{s_3}, \delta_{s_4}^{-1})$ with $\delta_{s_1} = \delta_{s_2} = \{m, V\}$ and $\delta_{s_3} = \delta_{s_4} = \{X_{\mathrm{cg}}, Z_{\mathrm{cg}}\}$ yields

$$
v(\delta_{s_1}, \delta_{s_2}^{-1}) = \begin{bmatrix} \frac{V}{m} \\ \frac{m}{V^3} \\ \frac{1}{V} \end{bmatrix}, \qquad u(\delta_{s_3}, \delta_{s_4}^{-1}) = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}
$$

with

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

and the application of the ETD yields LFRs on the order of six for $v(\delta_{s_1}, \delta_{s_2}^{-1})$ and order five for $u(\delta_{s_3}, \delta_{s_4}^{-1})$, resulting in an LFR on the order of 11 for $a_{29}(\delta)$. The order 11 LFR is also very close to the lower bound for the LFR order for a_{29}, which is nine ($\{r_1, r_2, r_3, r_4\} = \{2, 4, 2, 1\}$). We conjecture that 11 is already the minimal order.

B. LFR Realization for the Full Research Civil Aircraft Model

The parametric state-space matrices $A(\delta)$, $B(\delta)$, $C(\delta)$, $D(\delta)$ of the RCAM are given in [10] and have only elements as Laurent polynomials in the indeterminates. Three LFRs for the concatenated matrix

$$
S(\delta) = \begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix}
$$

are computed and the results are presented in Table 4. Without symbolic preprocessing, an LFR with order 400 ($\{108, 201, 69, 22\}$) is obtained using the object-oriented LFR realization approach based on the generalized LFR.

Again, the best result is obtained by employing the combined VS and ETD approach. The resulting LFR of $S(\delta)$ is on the order of 66 and it is possible to exactly reduce this LFR to order 65, which is very close to the theoretical least-order bound of 56, calculated with the procedure described in Sec. II.B. In this specific case, the VS factorization has been applied to $S(\delta)$ using $\delta_{s_1} = \delta_{s_2} = \{X_{\mathrm{cg}}, Z_{\mathrm{cg}}\}$ and $\delta_{s_3} = \delta_{s_4} = \{m, V\}$.

Remark 1: The VS factorization requires one to define the sets δ_{s_i}, $i = 1, \ldots, 4$, describing the "parameter splitting." To obtain the best result (in terms of least LFR order), one has to check all possible combinations of parameter splittings, which is supported by a "try-hard" option in the LFR toolbox. For the RCAM, this requires 120 distinct applications of the VS + ETD approach. With an average computation time of 60 s for each decomposition, the whole approach takes about 2 h on an Intel T2500 2.0 GHz running MATLAB 7.1 under Windows XP. The TD and ETD methods must be applied only once, with a calculation time of 95 and 74 s, respectively.

C. Accuracy of Low-Order LFRs

For the very complex parametric model given by the RCAM, the VS + ETD approach allows one to obtain an LFR, where a further

Table 3 Reduced LFR model orders for element a_{29}

Preprocessing	r	$\{r_1, r_2, r_3, r_4\}$	r, red.	$\{r_1, r_2, r_3, r_4\}$, red.
TD	31	$\{5, 13, 6, 7\}$	21	$\{3, 11, 4, 3\}$
ETD	23	$\{3, 6, 8, 6\}$	14	$\{3, 6, 3, 2\}$
VS + ETD	11	$\{2, 4, 3, 2\}$	11	$\{2, 4, 3, 2\}$

Table 4 LFR orders for RCAM

Preprocessing	r	$\{r_1, r_2, r_3, r_4\}$	r, red.	$\{r_1, r_2, r_3, r_4\}$, red.
TD	137	{35, 61, 28, 13}	107	{35, 50, 17, 5}
ETD	109	{27, 45, 26, 11}	91	{24, 38, 21, 8}
VS + ETD	66	{16, 30, 15, 5}	65	{16, 30, 14, 5}

exact numerical order reduction is almost unnecessary, as the order of 66 can only be reduced to 65. For the a_{29} element, no additional numerical reduction is possible. These are remarkable results as the symbolic preprocessing can be applied without any loss of accuracy (floating-point numbers are represented exactly in rational form), whereas numerical order reduction is always based on tolerance-dependent rank decisions. Therefore, it is always beneficial for the accuracy of the resulting LFR if no numerical order reduction must be applied. To illustrate this, the accuracies of the low-order LFRs obtained with the combined VS and ETD approach are compared with the accuracies of numerically reduced LFRs obtained with the robust control toolbox 3.0.1 of MATLAB. This toolbox offers two kinds of numerical reduction called "basic" and "full." The basic method is based on the one-dimensional order reduction method of [8] and the MATLAB function `sminreal` is repeatedly called to detect structurally one-dimensional unobservable/uncontrollable parts. The full method is also based on the one-dimensional order reduction principle, but performs one-dimensional balancing and truncation, where all system parts with Hankel singular values less than 1e-16 are truncated.

Table 5 presents the accuracies of the different LFRs. The accuracy is derived as follows: substitute the symbolic vector δ in the symbolic matrix $S(\delta)$ with random numerical values δ_{rand} and subtract the numerical matrix $S(\delta)|_{\delta=\delta_{\text{rand}}}$ from the related numerical upper LFT $S'(\delta)|_{\delta=\delta_{\text{rand}}}\mathcal{F}_u(M, \Delta)|_{\delta=\delta_{\text{rand}}}$ for 200 random parameterizations δ_{rand} and the maximum of the 2-norms of the 200 samples is taken as accuracy, i.e.,

$$e = \max_i(\|S(\delta)|_{\delta=\delta_{\text{rand},i}} - S'(\delta)|_{\delta=\delta_{\text{rand},i}}\|_2), \qquad i = 1, \ldots, 200$$

The notation "n-d" ("Appr") in Table 5 means exact n-dimensional (approximately one-dimensional) order reduction and using the functions `minlfr` (`redlfr1`) of the LFR toolbox.

The numerical reduction method basic has almost no effect on the order of the LFRs. The method full, which is applied after every step during the object-oriented LFR realization, yields a large loss of accuracy with an error e of more than 3%. On the other hand, the LFRs obtained after symbolic preprocessing are almost of full accuracy, $e = 7\text{e-}18$ for a_{29} and $e = 6\text{e-}14$ for $S(\delta)$, which is very important to have reliable LFRs for the application of robust controller synthesis and stability/performance analysis.

It is interesting to see that if one may be confident with the low accuracy obtained by the full method, almost the same accuracy can be obtained by applying numerical approximation to the LFRs obtained after symbolic preprocessing. However, in this case, it is possible to further reduce the order to 7 and 48 for a_{29} and $S(\delta)$, respectively. Hence, for $S(\delta)$, an LFR with 13 times better accuracy and only 51% of the order could be realized using advanced symbolic preprocessing techniques.

Table 6 Robust stability analysis results for RCAM

Controller	μ upper bound	μ upper bound from [10]	ζ_{worst}	$\zeta_{\text{skew}\mu}$
12	0.50	0.36	0.26	0.25
16	0.59	0.35	0.21	0.19
14	0.74	0.57	0.08	0.055
22	0.82	0.39	0.15	0.12
13	0.85	0.51	0.04	0.035
15	0.98	0.44	——	——
25	0.99	0.65	0.05	——
24	0.99	0.94	−0.03	——
11	1.15	0.49	−0.06	——
18	1.26	0.83	−0.04	——
19	1.94	1.36	−0.13	——
21	2.28	1.53	−0.18	——

D. Results of μ-Analysis for the RCAM LFR with Order 65

The results of the μ analysis for controlled RCAM, using the very accurate LFR of order 65, are summarized in Table 6. The robust control toolbox of MATLAB [11] is used to perform the μ analysis.

For comparison, the μ analysis and the optimization-based, worst-case search results from [10] are also included. The worst-case search repeatedly calculates the eigenvalues of the linearized closed-loop system within an optimization-based parameter search directed to determine the minimum damping ζ_{worst} over all admissible uncertain parameter values. Each calculation of the least damping involves the trimming of the nonlinear open-loop system and the linearization of the closed-loop system.

In every case, the μ analysis from [10] led to lower values for the maximum of the μ upper bound. These more optimistic results come from the fact that the variation in airspeed V was not considered. Therefore, the μ-analysis results from [10] indicate robust closed-loop stability for the controllers 11, 18, 24, which was proven to be wrong from the worst-case search. The new μ-analysis results, involving the accurate, low-order LFR, including variations in V, mainly coincide with the worst-case search results. The closed-loop system with controller 12 yields the lowest μ upper-bound value and also the best damping value. The controllers 11 and 18 are clearly indicated to not fulfill the robust closed-loop stability test. For the closed-loop system with controller 24, a μ upper bound of 0.99 is obtained, which is at the robust stability boundary and practically indicates an unstable system, as already shown from the worst-case search.

For direct comparison of μ-analysis results with the worst damping results provided in [10], we also calculated a worst damping for the controllers 12, 13, 14, 16, and 22 using the functionalities of the skew μ toolbox for MATLAB [14]. The results are given in the last column of Table 6 and the values for $\zeta_{\text{skew}\mu}$ are slightly below the values ζ_{worst}, which exactly meets the expectations, as the skew μ results should serve as a guaranteed upper bound for the worst damping.

For verification of the μ-analysis results, the worst-case parameter combination obtained from the μ lower-bound calculation of the closed-loop system with the 24 controller was extracted. For the RCAM, these parameter values were at the vertices of the uncertain parameter domain, i.e., the normalized parameter values were either 1 or −1. Therefore, we generated two additional sets of uncertain

Table 5 Accuracy of LFRs for RCAM

Symbolic model	Symbolic preprocessing	Numerical reduction	r	$\{r_1, r_2, r_3, r_4\}$	e
a_{29}	——	basic	61	{12, 30, 18, 1}	1e-17
a_{29}	——	full	12	{4, 3, 4, 1}	4e-4
a_{29}	VS + ETD	——	11	{2, 4, 2, 3}	7e-18
a_{29}	VS + ETD	appr	7	{2, 2, 2, 1}	1e-4
$S(\delta)$	——	basic	395	{108, 201, 69, 17}	6e-14
$S(\delta)$	——	full	94	{24, 42, 19, 9}	3.3e-2
$S(\delta)$	VS + ETD	——	66	{16, 30, 15, 5}	6e-14
$S(\delta)$	VS + ETD	n-d	65	{16, 30, 14, 5}	6e-14
$S(\delta)$	VS + ETD	appr	48	{13, 22, 11, 2}	2.5e-3

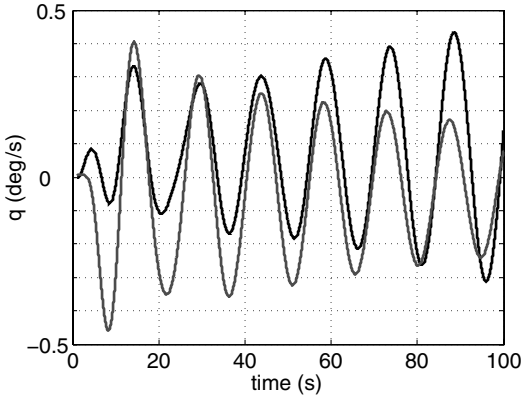

Fig. 3 Nonlinear simulation of RCAM.

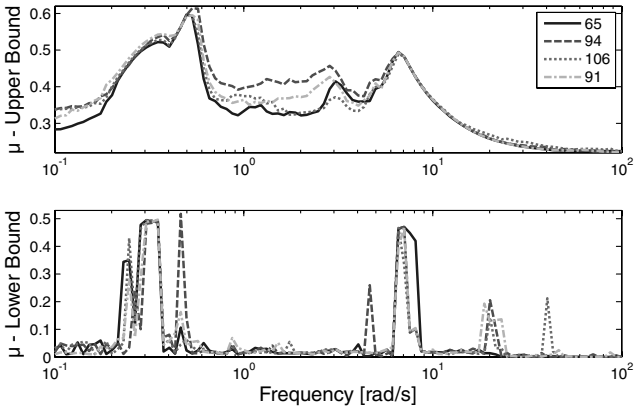

Fig. 4 μ upper and lower bounds for controller 16.

parameter values: one set, where these values were multiplied with 1.05 and another, where we multiplied these values with 0.95. With these two sets, a nonlinear simulation of the RCAM for a pitch rate response to an elevator step was performed (see Fig. 3) and one can clearly see that the black curve (corresponding to the worst-case parameter set multiplied with 1.05) shows an unstable behavior, whereas the light gray curve (corresponding to the worst-case parameter set multiplied with 0.95) shows a stable system. Hence, the μ-analysis result of 0.99 for this controller, which indicates that the closed-loop system is close to the stability boundary, could be verified using nonlinear simulation.

E. Comparison of μ-Analysis Results for LFRs of Different Order and Accuracy

As already mentioned in Sec. III.B, the calculation of the least-order LFR for the RCAM takes about 2 h using the VS + ETD approach, which is quite long compared to the 95 s and 74 s required for the TD and ETD approaches, respectively. The same holds for the slightly inaccurate (3.3% model error), numerically reduced LFR of order 94, which was obtained within several seconds using the robust control toolbox. However, to justify the effort required to compute a least-order LFR, we will compare the quality of the closed-loop μ-analysis results obtained with the RCAM LFRs of order 107 (TD), 91 (ETD), 94 (robust control toolbox), and 65 (VS + ETD). Therefore, for each controller, the closed-loop μ upper and lower bounds are calculated for each of the four LFRs along a grid of 100 logarithmically equally spaced frequency points between 0.1 rad/s and 100 rad/s. As a result, we obtain the vectors U_p, $L_p \in \mathbb{R}^{100}$, $p = 65, 91, 94, 107$, describing the upper- and lower-bound values for each of these LFRs, respectively. Furthermore, the required computation time T_p, $p = 65$, 91, 94, 107 for the μ analysis is calculated. For comparison, the following criteria are defined.

1)

$$\delta S_u(p) = \left(\sum_{i=1}^{100} U_p(i) - \sum_{i=1}^{100} U_{65}(i)\right) / \sum_{i=1}^{100} U_{65}(i)$$

which describes the relative increase of the sum of the upper-bound values compared with the sum of the upper-bound values of the least-order LFR.

2)

$$\delta S_l(p) = \left(\sum_{i=1}^{100} L_{65}(i) - \sum_{i=1}^{100} L_p(i)\right) / \sum_{i=1}^{100} L_{65}(i)$$

which describes the relative decrease of the sum of the lower-bound values compared with the sum of the lower-bound values of the least-order LFR.

3)

$$\delta P_l(p) = [\max_{i=1,\ldots,100} L_{65}(i) - \max_{i=1,\ldots,100} L_p(i)] / \max_{i=1,\ldots,100} L_{65}(i)$$

which describes the relative decrease of the upper-bound peak value compared with the upper-bound peak value of the least-order LFR.

4)

$$\delta T(p) = T_p / T_{65}$$

which describes the increase of computation time compared with the computation time required for the least-order LFR.

In Table 7, the criteria values are presented. For almost each controller, the worst upper-bound values are obtained using the inaccurate LFR of order 94. For some controllers, the upper-bound values are almost 9% larger than the values obtained with the accurate, least-order LFR. This shows that accuracy of the LFRs is important to calculate a tight upper bound. Furthermore, it can be clearly seen that among the accurate LFRs with orders 107, 91, and 65, a smaller LFR order yields smaller/tighter values for the upper bound. Note that the variation in the peak values of the upper bounds were only up to 2% and, therefore, these values are not presented separately.

Table 7 Comparison of μ-analysis results for different LFRs and controllers

	11	12	13	14	15	16	18	19	21	22	24	25
$\delta S_u(94)$	5.6%	5.3%	8.2%	4.5%	8.8%	8%	6%	7.6%	5.1%	8.7%	3%	7.2%
$\delta S_u(107)$	4.3%	3.7%	4%	3.5%	6.5%	2.7%	4.7%	5%	2.9%	3.6%	4%	1.5%
$\delta S_u(91)$	2%	3.2%	2.7%	2.2%	4.4%	4.4%	1.8%	1.7%	2.5%	2.7%	1.5%	2%
$\delta S_l(94)$	−6%	19%	18%	16%	7.1%	7.5%	28%	40%	13%	20%	−7.3%	63%
$\delta S_l(107)$	9.5%	7.0%	25%	23%	39%	9.8%	52%	19%	5.0%	3.8%	−0.2%	25%
$\delta S_l(91)$	−3.6%	6.7%	28%	6.2%	16%	9.2%	36%	13%	4.5%	20%	10%	12%
$\delta P_l(94)$	−33%	8.4%	0.3%	0.3%	−1.6%	−4.4%	35%	0.1%	0.0%	35%	−2.7%	58%
$\delta P_l(107)$	0.0%	8.2%	0.3%	0.0%	−1.6%	0.0%	37%	0.1%	0.0%	26%	2.0%	68%
$\delta P_l(91)$	−48%	8.5%	0.4%	25%	0.0%	0.0%	37%	0.0%	0.2%	−1.5%	−6.0%	−20%
$\delta T(94)$	2.5	2.3	2.5	2.5	2.1	2.1	2.2	2.3	1.9	2.2	2.6	2.2
$\delta T(107)$	2.9	3.2	3.7	3.2	3.2	3.0	3.3	3.0	2.9	3.4	3.3	3.5
$\delta T(91)$	2.1	1.9	2.2	2.3	1.9	1.9	2.0	2.0	1.8	2.1	2.3	2.2

Except for controller 11 and 24, the improvements in the lower-bound calculation are even more remarkable and the lower-bound values for the least-order LFR are up to 63% larger/tighter. In most cases, the peak value of the lower bound is also much larger.

The average computation time for the μ analysis using the least-order LFR is around 300 s. This time directly depends on the order of the LFR and, for the LFR with order 107, the computation time goes up to 1016 s (factor 3.7 for controller 13).

An example of the μ upper and lower bounds for the RCAM LFRs with orders 65, 91, 94, and 107, together with controller 16, are shown in Fig. 4.

IV. Conclusions

An enhancement of the variable splitting factorization was introduced, which allows an efficient application of this scalar technique to parametric vectors and matrices. Especially, the application of the condensation algorithms allows one to further exploit parametric dependencies and common factors in the matrix elements, which generally yields linear fractional representations of lower order when applied in combination with the object-oriented LFR realization approach. Furthermore, a simple algorithm for the calculation of a tight lower bound for the minimal LFR order was presented, which allows one to evaluate the order of the generated LFRs.

Using this enhanced symbolic preprocessing technique, an LFR of full accuracy and almost minimal order was generated for one of the most complicated parametric aircraft models available in literature. The robust stability μ-analysis results obtained with this LFR coincide with the results obtained from a computationally demanding, optimization-based, worst-case search, which could also be verified using nonlinear simulation. This shows that efficient methods for low-order LFR realization are of paramount importance for successful application of μ analysis, which then can be seen as a fast complement to time-consuming industrial approaches, such as Monte Carlo analysis or parameter gridding.

Furthermore, it could be shown that the quality of the μ analysis depends on the accuracy and the order of the underlying LFR. The smallest gap between the μ upper and lower bounds is obtained for the most accurate and least-order LFR.

References

[1] Zhou, K., Doyle, J. C., and Glover, K., *Robust and Optimal Control*, Prentice–Hall, Upper Saddle River, NJ, 1996.

[2] Bose, N. K., *Applied Multidimensional Systems Theory*, Van Nostrand Reinhold, New York, 1982.

[3] Hecker, S., and Varga, A., "Symbolic Manipulation Techniques For Low Order LFT-Based Parametric Uncertainty Modelling," *International Journal of Control*, Vol. 79, No. 11, 2006, pp. 1485–1494. doi:10.1080/00207170600725644

[4] Cockburn, J., and Morton, B., "Linear Fractional Representation of Uncertain Systems," *Automatica*, Vol. 33, No. 7, 1997, pp. 1263–1271. doi:10.1016/S0005-1098(97)00049-6

[5] Marcos, A., Bates, D. G., and Postlethwaite, I., "Symbolic Matrix Decomposition Algorithm for Reduced Order Linear Fractional Transformation Modelling," *Automatica*, Vol. 43, No. 7, 2007, pp. 1211–1218. doi:10.1016/j.automatica.2006.12.031

[6] Morton, B., "New Applications of μ to Real Parameter Variation Problems," *Proceedings of the Conference on Decision and Control*, Inst. of Electrical and Electronics Engineers, New York, 1985, pp. 233–238.

[7] Hecker, S., and Varga, A., "Generalized LFT-Based Representation of Parametric Uncertain Models," *European Journal of Control*, Vol. 10, No. 4, 2004, pp. 326–337. doi:10.3166/ejc.10.326-337

[8] Terlouw, J., Lambrechts, P., Bennani, S., and Steinbuch, M., "Parametric Uncertainty Modeling Using LFTs," *Proceedings of the American Control Conference*, Inst. of Electrical and Electronics Engineers, New York, 1993, pp. 267–272.

[9] D'Andrea, R., and Khatri, S., "Kalman Decomposition of Linear Fractional Transformation Representations and Minimality," *Proceedings of the American Control Conference*, Inst. of Electrical and Electronics Engineers, New York, 1997, pp. 3557–3561. doi:10.1109/ACC.1997.609484

[10] Magni, J., Bennani, S., and Terlouw, J., *Robust Flight Control: A Design Challenge*, No. 224 in Lecture Notes in Control and Information Sciences, Springer, New York, 1997.

[11] Balas, G., Chiang, R., Packard, A., and Safonov, M., *Robust Control Toolbox 3.0.1*, MathWorks, Natick, MA, March 2005.

[12] Lambrechts, P., and Terlouw, J., *Matlab Toolbox for Parametric Uncertainty Modeling*, Philips Research Eindhoven and National Aerospace Lab., Amsterdam, 1992.

[13] Hecker, S., Varga, A., and Magni, J., "Enhanced LFR-Toolbox for Matlab," *Aerospace Science and Technology*, Vol. 9, No. 2, 2005, pp. 173–180. doi:10.1016/j.ast.2004.12.001

[14] Ferreres, G., and Biannic, J., "Skew Mu Toolbox (SMT) for Robustness Analysis," *Proceedings of the International Symposium on Computer Aided Control Systems Design (CACSD)*, Inst. of Electrical and Electronics Engineers, New York, 2004. doi:10.1109/CACSD.2004.1393894