Mathematical and Computer Modelling of Dynamical Systems 1387-3954/98/0404-249$12.00
1998, Vol. 4, No. 4, pp. 249-274

Automated Generation of LFT-Based Parametric Uncertainty Descriptions from Generic Aircraft Models

A. VARGA*, G. LOOYE, D. MOORMANN, G. GRÜBEL

ABSTRACT

A computer assisted modelling methodology is developed for the generation of linearized models with parametric uncertainties described by Linear Fractional Transformations (LFTs). The starting point of the uncertainty modelling is a class of generic nonlinear aircraft models with explicit parametric dependence used for simulation purposes. The proposed methodology integrates specialized software tools for object-oriented modelling, for simulation, and for numerical as well as symbolic computations. The methodology has many generic features being applicable to similar nonlinear model classes.

Key words: linear fractional transformations, nonlinear systems modelling, software tools, uncertainty modelling.

1 INTRODUCTION

The dynamical behaviour of many lumped-parameter processes, and in particular of a flying aircraft, can be described by non-linear dynamic system models of the form

$$\begin{aligned} E(x(t),\, p)\dot{x}(t) &= F(x(t),\, u(t),\, p) \\ y(t) &= G(x(t),\, u(t),\, p) \end{aligned} \tag{1}$$

where x, u, y are the state-, input- and output–vectors respectively, and p is a vector of model parameters. The matrix $E(x(t),\, p)$ is structurally non-singular and thus can be inverted if necessary. A generic aircraft model can be seen as the interconnection of several dynamical and static subsystems describing different parts of the aircraft dynamics and of the interactions of aircraft with its flight environment. Generic models are useful for obtaining dynamical models of particular aircraft by appropriately choosing the various component subsystems and the corresponding parameter sets. For a particular aircraft model

*DLR - Oberpfaffenhofen, German Aerospace Center, Institute for Robotics and System Dynamics, P.O.B. 1116, D-82230 Wessling, Germany. E-mail: andreas.varga@dlr.de

different instantiations can be obtained with respect to various flight conditions and parameter values.

The structured singular value (also called μ) was introduced to study linear models with structured parametric uncertainties [4], described by *linear fractional transformations* (LFTs) [5, 22]. The evolving μ-analysis and synthesis methodology represents a powerful tool for analysis of robustness and for robust control synthesis (see for instance [6] for a quite complex practical synthesis example). One of the main hurdles in using the μ-analysis and synthesis techniques is the need to develop LFT-based parametric uncertainty descriptions for the underlying plant models. Obtaining such descriptions is usually a very tedious, error-prone modelling task and can involve tremendous symbolic manipulations even for relatively few parameters. Therefore, the availability of tools to automate the generation of LFT-based parametric uncertainty descriptions is important to complement existing μ-analysis and synthesis software.

In this article we discuss the automatic generation of LFT-based parametric uncertainty models starting from a generic nonlinear aircraft-dynamics model. The main problem addressed is how to approximate *all* linearizations of a nonlinear model of the form (1) over *all* flight conditions and *all* parameter values by a unique linear time-invariant state space model with explicit parametric uncertainties expressed as LFTs. The proposed computer assisted modelling methodology integrates many specialized software tools as, for instance, tools for object-oriented modelling to generate particular aircraft models, tools for numerical computations to determine equilibrium points or to reduce the order of LFT descriptions, as well as tools for symbolic computations to perform system linearizations and to generate LFT-based uncertainty descriptions. Note that, although the starting point of our discussion was a particular class of models, the main steps of the proposed computer assisted methodology are of generic value being applicable to similar model classes encountered in many practical applications. As an example, we will discuss the generation of an LFT-based uncertainty description for the *Research Civil Aircraft Model* (RCAM) [16] developed within the GARTEUR *Action Group on Robust Flight Control*. The RCAM example served as a benchmark problem for the GARTEUR robust control design challenge [12].

Physical models of generic aircraft-dynamics have been developed in [16] by using the object-oriented modelling environment Dymola [7, 8]. The key aspect of modelling with Dymola is the possibility to obtain nonlinear dynamic models which exhibit *explicit* parametric dependencies. This aspect led us to the idea to investigate the possibility of using the generic Dymola models as starting points to generate LFT-based parametric uncertainty models in a form directly suited for μ-analysis [1, 10]. Such an approach involves basically two steps. In the first step, linear models exhibiting implicit or explicit parameter dependencies are generated. To be further tractable, these dependencies must be expressed exclusively by rational functions in the parameters. In the second step, LFT-based parametric descriptions are generated from these rational parametric representations by using multidimensional state space realization algorithms. A new approach combining a method based on parametric study [18] and the numerical and symbolic linearization method of [20] is proposed in this article.

The organization of the article is as follows. In chapter 2 we present the main aspects of the development of generic aircraft models using the object oriented modelling tool Dymola. Then we discuss in chapter 3 a combined numerical and symbolic linearization approach of the nonlinear aircraft dynamics model for the purpose of generating LFT-based parametric uncertainty descriptions. A semi-automatic procedure has been developed based on one- or two-dimensional data fitting to obtain rational parametric expressions for each matrix element which are valid over the whole flight envelope. In chapter 4 we describe the generation of LFT for system matrices with elements given by rational expressions of parameters. Appendices A summarizes the system variables, while in Appendix B we give the complete rational parameter-dependent system matrices used to generate the LFT uncertainty description for RCAM.

2 DEVELOPMENT OF GENERIC MODELS

2.1 General model building aspects

A generic model based approach for aircraft-dynamics modelling has been developed in [16]. This approach is based on the object-oriented modelling environment Dymola [7, 8] and is a versatile methodology to generate nonlinear aircraft models primarily for simulation purposes. Such models can describe the whole dynamics of a particular aircraft in all flight conditions. Moreover models corresponding to particular flight conditions and/or emphasizing only parts of the aircraft dynamics (e.g. longitudinal dynamics) can be easily generated. The explicit parametric dependencies present in the generated models allows to perform various types of parametric studies for simulation, optimization, or robustness analysis purposes. A main advantage of the object oriented modelling approach is that, it allows to describe models of aircraft dynamics in the notation of aircraft physics (i.e. of flight mechanics). In contrast to this approach, control system oriented model building tools like Simulink [15] use signal flows or input-output block diagrams.

Fig. 1 shows the main physical subsystems describing the different physical parts of the aircraft and the different phenomena influencing the aircraft aerodynamics.

A simplifying feature of such an approach to aircraft modelling is that each model component is described in its own coordinate system. Thus gravity,

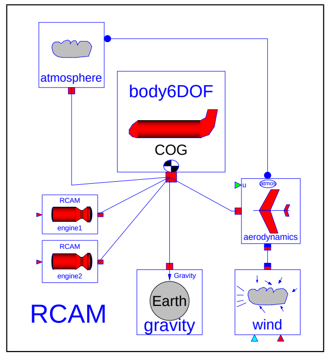

Fig. 1. Object diagram of RCAM

wind, and atmosphere are conveniently described in an earth related coordinate system, aerodynamics in a wind coordinate system, and engines in a body-fixed coordinate system. This is achieved by using additional objects to describe coordinate transformations and an object to describe the relationship between velocity, wind, and airspeed. All necessary components are grouped into an aircraft objects library as that presented in Fig. 2. Different representations of one component may be present to allow model building of different complexity or various functionalities

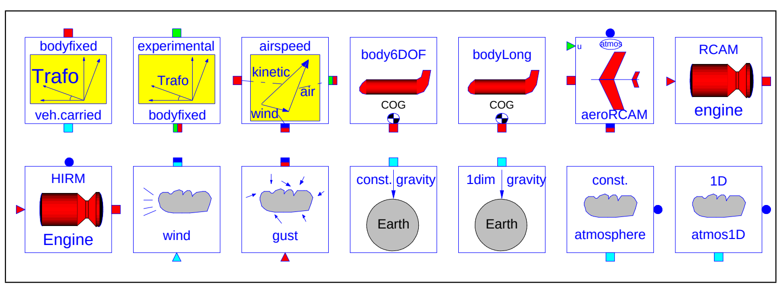

Fig. 2. Aircraft model library

An additional feature in object oriented modelling of physical systems is the encapsulation of objects. The internal implementation of details (e.g. of the aerodynamics) are not visible, when viewing the RCAM object model as depicted in Fig. 1. By encapsulation, the implementation of an object can be changed without affecting the functionality of the entire model. More details of aircraft modelling are discussed in [16].

Dymola has a graphical interface which allows an interactive generation of models by component aggregation. The resulting model contains both graphical as well as textual information. The graphical information shows the constituent blocks, the hierarchy of blocks, and the interconnections between blocks. The textual information describes all the underlying mathematical details of a block and is provided in the form of ordinary mathematical and boolean equations.

The symbolic equation handler of Dymola generates a state space model of the form (1) from the parameter instantiated equations of each object, together with the equations derived from the interconnection structure. The equations are sorted and solved according to the specified inputs and outputs. Equations, which are formulated in an object but not needed for the specified configuration, are removed automatically. The result is a mathematical model with a minimum number of equations for the specified task.

2.2 Software tools

From the graphical and textual information, Dymola can automatically generate efficient code for different simulation environments, like Simulink, ACSL or ANDECS_DSSIM. The code for Simulink can be an m-file or a cmex-file. Fortran or C code can be exported in the DSblock neutral simulation-model format [17], to be used in any other simulation run-time environment capable of importing Fortran or C models. This facility, in particular, is used for the ANDECS control design optimization environment [11].

To obtain the LFT based uncertainty description from the RCAM example depicted in Fig. 1, a MATLAB code based Simulink model and a C code model have been generated. The MATLAB code generated as an m-file allows the conversion to a symbolic model which is processable by symbolic computation tools. The generated C code model is a cmex-file which is linked directly to Simulink and mainly serves for a fast trimming of the nonlinear model in various equilibrium points. For an increased computational performance, Dymola could also generate Fortran subprograms to be used by the ANDECS_DSSIM simulation module [11], or by DSSTAT to compute equilibrium points or DSLIN to perform numerical linearizations.

2.3 RCAM example

In the following we present some particular features of the RCAM example in Fig. 1. The state vector x, control input vector u and measurable output vector y have dimensions 12, 5, and 15 respectively. The physical meaning of the components of the state, input and output vectors is given in Appendix A. The parameter vector p in the nonlinear model (1) is defined as $p = (m, X_{cg}, Y_{cg}, Z_{cg})$, where m is the mass of the aircraft and (X_{cg}, Y_{cg}, Z_{cg}) represent the three coordinates of the center of gravity of the aircraft. These parameters have the following ranges of variation:

$$\begin{array}{lcl} m & \in & [\,100000\,kg,\ 150000\,kg\,] \\ X_{cg} & \in & [\,0.15\bar{c},\ 0.31\bar{c}\,] \\ Y_{cg} & \in & [\,-0.03\bar{c},\ 0.03\bar{c}\,] \\ Z_{cg} & \in & [\,0,\ 0.21\bar{c}\,], \end{array}$$

where $\bar{c} = 6.6\,m$ is the mean aerodynamic chord [2]. The nominal values of parameters used throughout this article are $m_{nom} = 120000\,kg$, $X_{cgnom} = 0.23\bar{c}$, $Y_{cgnom} = 0$, $Z_{cgnom} = 0.0\bar{c}$.

In the next chapters we present as an example the generation of the LFT-based uncertainty description for the linearized RCAM in symmetric horizontal flight defined by constant air speed $V_A = 80\ m/s$ and $y_{cg} = 0$. Note that no parameter variation will be included with respect to the parameter y_{cg}, but we will consider V_A as an additional uncertain parameter with the following range of variation:

$$V_A \in [\,1.23 V_{stall}\ , 90\,m/s\,].$$

V_{stall} is defined from the following equilibrium relation: $mg = \frac{1}{2}\rho V_{stall}^2 C_{L_{max}}$, where $C_{L_{max}} = 2.75$ and ρ is the density of air.

3 PARAMETER DEPENDENT LINEARIZATION

3.1 General aspects

As already mentioned, our aim is to obtain a linear time-invariant state space model with explicit parameter dependencies, which satisfactorily approximates *all* linearizations of the nonlinear model (1) over *all* flight conditions and *all* parameter values. For this purpose there are several approaches possible with various degrees of conservativeness.

The first approach at hand is to repeatedly perform numerical linearizations of the nonlinear model (1) in several equilibrium points $(x^{(i)}, u^{(i)})$, $i = 1, \ldots, N$, corresponding to particular values $p^{(i)}$ of the physical parameters. The resulting linear models form a so called *multi-model state description*

$$\begin{array}{rcl} \delta\dot{x}(t) & = & A_i \delta x(t) + B_i \delta u(t) \\ \delta y(t) & = & C_i \delta x(t) + D_i \delta u(t) \end{array}, \quad i = 1, \ldots, N\ , \tag{2}$$

where A_i, B_i, C_i, D_i, for $i = 1, \ldots, N$, are constant real matrices of appropriate dimensions. For each entry in one set of the above system matrices it is easy to determine the corresponding lower and upper bounds. For instance, let

A^{min} and A^{max} be the matrices containing the lower and upper bounds for the entries of state matrices. Then all (i, j) entries can be replaced by a single entry of the form $a_{ij} = a^0_{ij} + s_l p^A_l$, where $a^0_{ij} = (a^{min}_{ij} + a^{max}_{ij})/2$ is the *nominal* value, $s_l = (a^{max}_{ij} - a^{min}_{ij})/2$ is the *slope* corresponding to element (i, j). If $s_l \neq 0$, then p^A_l is an uncertain parameter which varies between -1 and 1. The index l corresponds to the column stacked vector representation of the state matrix. We can attach such a parametric uncertainty to each varying entry of the matrices A_i and finally we can define the parameter dependent state matrix

$$A(p_A) = A^0 + p^A_1 A^1 + \cdots + p^A_{k_A} A^{k_A},$$

where A^0 has the entries a^0_{ij} and A^l is a rank one matrix of the form $A^l = s_l e_i e^T_j$, with e_i being the i-th column of the identity matrix. Analogously, we can define the matrices $B(p_B)$, $C(p_C)$ and $D(p_D)$, and we can replace the multi-model (2) by an *affine parameter-dependent representation* of the form

$$\begin{aligned} \delta\dot{x}(t) &= A(p_A)\delta x(t) + B(p_B)\delta u(t) \\ \delta y(t) &= C(p_C)\delta x(t) + D(p_D)\delta u(t). \end{aligned} \tag{3}$$

Note that all parameters appearing in the above defined linear system are artificially introduced and thereby reflect the uncertainties of the physical parameters $p_1, \ldots, p_k$ only implicitly. Although the parametric linear model (3) certainly covers all possible linearizations arising from the nonlinear model (1), this representation of physical uncertainties is conservative because of ignoring possible joint parameter dependencies in the model. Thus, using this modelling approach as the basis for a μ-analysis or synthesis may lead to difficulties in designing robust controllers or in assessing their robustness [18].

A second approach to obtain a linear parametric representation suitable to generate an LFT-based uncertainty description is by using symbolic linearization in a nominal flight condition. For nominal values of the model parameters p_{nom} it is possible to compute numerically an equilibrium point of the system $\{\bar{x}, \bar{u}\}$ satisfying the the system of nonlinear equations

$$0 = F(\bar{x}, \bar{u}, p_{nom}) . \tag{4}$$

Let $\bar{F}(\delta x, \delta u, p)$ and $\bar{G}(\delta x, \delta u, p)$ be

$$\begin{aligned} \bar{F}(\delta x, \delta u, p) &:= [E(\bar{x} + \delta x, p)]^{-1} F(\bar{x} + \delta x, \bar{u} + \delta u, p) \\ \bar{G}(\delta x, \delta u, p) &:= G(\bar{x} + \delta x, \bar{u} + \delta u, p) \end{aligned} \tag{5}$$

corresponding to "the right hand sides " of the ordinary differential equations arising from (1). By symbolic linearization of the nonlinear model (1) in the neighbourhood of an equilibrium point $\{\bar{x}, \bar{u}\}$ we obtain a linear time-invariant model of the form

$$\begin{aligned} \delta\dot{x} &= A(p)\delta x + B(p)\delta u \\ \delta y &= C(p)\delta x + D(p)\delta u, \end{aligned} \tag{6}$$

where the system matrices are given by

$$\begin{aligned} A(p) &= \left.\frac{\partial \bar{F}(\delta x, \delta u, p)}{\partial(\delta x)}\right|_{\substack{\delta x=0\\ \delta u=0}}, \quad B(p) = \left.\frac{\partial \bar{F}(\delta x, \delta u, p)}{\partial(\delta u)}\right|_{\substack{\delta x=0\\ \delta u=0}}, \\ C(p) &= \left.\frac{\partial \bar{G}(\delta x, \delta u, p)}{\partial(\delta x)}\right|_{\substack{\delta x=0\\ \delta u=0}}, \quad D(p) = \left.\frac{\partial \bar{G}(\delta x, \delta u, p)}{\partial(\delta u)}\right|_{\substack{\delta x=0\\ \delta u=0}}. \end{aligned} \tag{7}$$

We can assume that any non-rational parametric expressions in the elements of the state space model matrices are replaced by polynomial or rational approximations. Thus the model (6) represents a linearized system arising from the nonlinear model (1) in a *rationally parameter-dependent representation* with the matrices $A(\cdot)$, $B(\cdot)$, $C(\cdot)$ and $D(\cdot)$ having only entries which are rational functions of the physical parameters $p_1, \ldots, p_k$. Notice that, if necessary, it is possible to enlarge the parameter vector p with some components of the equilibrium point vectors in order to account for dependencies of the entries of the system matrices on particular linearization points.

The main advantage of this approach is that it allows an exact description of joint parametric dependencies in the model, and thus it can be used for an accurate modelling of the parametric uncertainties. The main limitation of this approach follows from the fact that the elements of the system matrices depend on the equilibrium point and on the nominal parameter values for which the linearization was performed. Thus, the resulting approximate linear model (6) is valid only in a small neighbourhood around the linearization point and therefore, such a model is generally not appropriate to be used in a μ-analysis and synthesis methodology if large parametric variations are involved, or if the controller to be designed must be robust in all flight conditions.

Obtaining a unique linear model, based on an LFT description of the uncertainties, which covers all flight conditions and all possible parameter variations, is obviously a difficult model building task. In what follows we offer a kind of compromise solution to this problem, by combining the two approaches mentioned above in such a way that the conservativeness of the first approach is reduced and the validity of the obtained model (6) is increased over the whole range of flight conditions and parameter variations.

The method which we propose can be seen as an updating procedure of an already existing model of the form (6). By using symbolic linearization in an arbitrary trimming point, it is possible to figure out which entries of the system matrices depend explicitly on the components of the trimming point vectors $\bar{x}$ and $\bar{u}$. Entries with no explicit dependence on state and input components don't need any corrections. However, the entries with explicit dependence on state and input components need to be corrected to take into account the dependence on trimming conditions. If the trimming could be performed symbolically by solving the nonlinear equations (4) for $\bar{x}$ and $\bar{u}$, we would obtain these vectors as explicit functions of parameters p and possibly of some additional trimming parameters. Thus we could obtain explicit expressions of all elements of the system matrices as (possibly non-rational) functions of system and trimming parameters. Unfortunately, for complicated models like that of an aircraft, symbolic trimming is not generally feasible. Thus, in general a multi-dimensional curve fitting for the elements of the system matrices is necessary to find approximate rational or polynomial relations on the trimming parameters.

To perform parameter fitting, the linearization data obtained for the multi-model representation (2) can be employed. For particular entries sometimes it is possible to use parameter fitting with respect to a reduced set of parameters only. For instance, in the case of an aircraft, textbook information (see [2]) provides useful hints to guess the proper form of the parametric dependencies. This approach, however, is strongly limited by the availability of reliable general purpose multi-dimensional parameter fitting procedures. Fortunately, in the case of RCAM, all necessary parameter fitting for the aircraft model can be done by using one- and two-dimensional parameter fitting.

The proposed approach is significantly more complicated than the two approaches previously discussed. Besides repeated linearizations, it involves also multi-dimensional curve fittings to update selected entries of the system matrices obtained for the nominal flight conditions and nominal parameter values.

3.2 Software tools

To determine the matrices of the multi-model (2) we performed repeated trimming and numerical linearizations for various flight conditions and parameter values by using the Simulink C-mex model generated by Dymola.

For the generation of the matrices of the linearized model (6) we can start with the Simulink m-file generated for RCAM by Dymola and convert it first to a Maple readable form. Note that Maple [9] is a very popular and powerful symbolic computation tool, having a command syntax which is very similar to the syntax of MATLAB commands. A standard interface between MATLAB and Maple is provided from both MATLAB side as well as from Maple side. By using symbolic manipulations within Maple (differentiation, polynomial and rational approximation), symbolic expressions of the system matrices (6) can be obtained and can then be used to generate LFT-based parametric uncertainty models. Here we discuss only some details of the conversion to the Maple format.

The Simulink m-file is actually a text file written in the MATLAB language devised to evaluate “the right hand sides “ in (5) of the ordinary differential equations, which arise from (1) for a given equilibrium point $\{\bar{x}, \bar{u}\}$. The well structured layout of the MATLAB interface allows the automatic processing of this text file to transform it into a source file conforming with the syntax of Maple. The only real difficulty is caused by the presence of *if-then-else* constructs in the generated MATLAB code. However, because all decisions implied by these constructs are well defined in the computed equilibrium points, all *if-then-else* constructs can be eliminated. Note that the resulting symbolic model (5) is only valid in the neighborhood of the equilibrium points. The automatic elimination of *if-then-else* constructs was done with the help of a special MATLAB script, which simultaneously executes and writes the executed code to an external file. Then a simple text processing routine written in MATLAB converts this file into a Maple readable form. These two conversion steps produce explicit Maple readable symbolic expressions for $\bar{F}(\delta x,\, \delta u,\, p)$ and $\bar{G}(\delta x,\, \delta u,\, p)$, to serve further for symbolic evaluation of the matrices of the linearized model.

For one- and two-dimensional parameter fitting, the PUM-toolbox [13] provides an interactive fitting routine which allows also to introduce an additional uncertain parameter to account for the approximation error. For the RCAM example, all parameter fitting can be carried out by a single routine, `fit_rcam.m`. In this routine, first all linearizations are read and all required data are collected. Next the fitting routine of PUM is called for each parameter to be approximated by a polynomial. The resulting polynomials are stored in the PUM-database as specially structured matrices. Since these polynomials need to be substituted in the parametric linear model in Maple, a function is available to translate these matrices into polynomials in string-format, readable by Maple. Using this function, `fit_rcam.m` writes the symbolic expressions of the polynomials into a Maple script. When executed, this script loads the obtained expressions of the polynomials and calculates symbolically the matrix elements of the parametric state-space model which need to be replaced. This replacement of the matrix elements depending on the trimming point takes place by calling the above Maple script after executing the symbolic linearization step.

3.3 RCAM example (continued)

For the RCAM example, let us consider the (7,7) element in the state matrix A. A large set of linearization has been created to cover the whole flight envelope for steady symmetric horizontal flight. From a global search over these linearizations, the maximum and minimum values of the (7,7) element are found as: $a_{77}^{min} = -0.0489$ and $a_{77}^{max} = -0.0193$. Thus to cover all possible values, we can express a_{77} as

$$a_{77} = -0.0341 + 0.0148\delta a_{77},$$

where δa_{77} is an uncertain parameter which may take any value between -1 and 1. As mentioned before, the major disadvantage of this approach is that this perturbation element has no direct physical meaning. Following this approach for each varying element in the matrices, we obtain as many parameters as there are varying elements, while in reality the elements change in a certain pattern due to joint variations of a limited number of physical parameters.

The symbolic linearization with Maple, at a fixed trim condition defined by constant air speed V_A and symmetric horizontal flight, gives the following expression for the (7,7) element of A:

$$a_{77} = -48.784\frac{V_A}{m} \tag{8}$$

Unfortunately the expression (8) of a_{77} is not valid over the whole flight envelope because generally a_{77} depends on the trimming point. However this expression is still a very good approximation in a small neighbourhood of the equilibrium point for small deviations of the parameters from their nominal values. The situation is quite different for other elements, as for instance the (1,1) element of A, which do not depend on the trimming point. For example, the resulting rational expression

$$a_{11} = -1900.1\frac{V_A}{m} \tag{9}$$

is valid with a very good accuracy over all flight conditions involving trimmed airspeed.

Now we show how to find improved rational parametric expressions for those elements of the system matrices which depend on the trimming point. As an example we consider again a_{77} and try to determine a rational expression for it, which covers all possible linearizations. From a textbook approximation we find (neglecting effects of the Mach-number) the following expression for a_{77}, as the X_u-element in a linear approximation of aircraft equations of motion [2]:

$$a_{77} = X_u = -\bar{q}\frac{S}{m}\frac{2}{V_A}(C_D - \alpha C_L) = -\frac{V_A}{m}\rho S(C_D - \alpha C_L), \tag{10}$$

where $\bar{q} = \frac{1}{2}\rho V_A^2$ is the dynamic pressure, C_D is the drag coefficient, C_L is the lift coefficient and α is the angle of attack. All these values correspond to the chosen trimming conditions. We immediately see that X_u not only depends on mass and speed, but also on the trimmed aerodynamic state of the aircraft. Only the V_A/m relation has been recognized by symbolic linearization at a fixed trimming point.

We can put the expression for a_{77} in the form

$$a_{77} = -\frac{g}{\frac{1}{2}\rho S}\frac{\rho S(C_D - \alpha C_L)}{C_W V_A} = -0.061601\frac{\tilde{a}_{77}}{C_W V_A}, \tag{11}$$

where $C_W = \frac{mg}{\rho/2 V_A^2 S}$ and $\tilde{a}_{77} = \rho S(C_D - \alpha C_L)$. For an aircraft the aerodynamic parameters α, C_D and C_L depend on the parameters mass, speed and center of gravity location. In fact it is known that $\tilde{a}_{77} = f(C_W, X_{cg})$, where $f(C_W, X_{cg})$ depends at most quadratically of C_W and X_{cg}. However in the case of $\tilde{a}_{77}$ it was possible to obtain a quite accurate approximation with the second order polynomial in C_W

$$\tilde{a}_{77} = 1.5667\, {C_w}^2 - 16.241\, C_w + 65.449.$$

Similar and even more simpler dependencies can be obtained for the analogous subexpression $\tilde{a}_{79}$, $\tilde{a}_{97}$ and $\tilde{a}_{99}$ of the elements a_{79}, a_{97} and a_{99}, respectively.

The good approximation by a second order fit can be observed from Fig. 3, where the curve in the middle represents the actual fit.

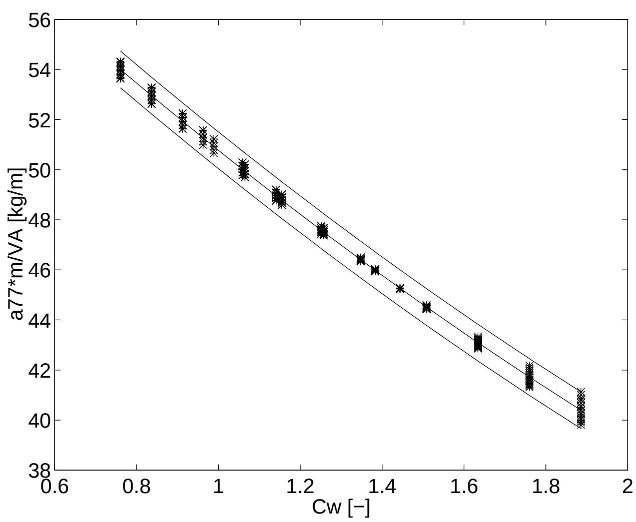

Fig. 3. Polynomial fit of $a_{77} * m/V_A$ with C_W

In this figure we immediately see, that for each C_W there is some vertical variation. This is mainly caused by the variation of the horizontal center of gravity location, X_{cg}. The computed second order approximation is sufficiently accurate to even allow extrapolation outside the flight envelope. The drawn bounds give an indication of the resulting approximation error if the dependence on X_{cg} is neglected. However, the largest deviation is within 2% of the corresponding function value and thus the influence of X_{cg} is almost negligible.

Lower and upper bounds are drawn below and above the fitted curve, representing minimal translations of this curve such that all data points lie in between. If the fitted curve is the function $f_0(C_W)$, then we can cover all data points by a function of the form

$$f(C_W) = f_0(C_W) + s_f \delta_f$$

where s_f is an appropriate scaling and δ_f is a new parameter which varies between -1 and 1. Although this approximation seems a bit conservative, especially at lower values of C_W, in this way we can, with an additional uncertain parameter, also cover the fitting errors and thus all possible linearizations. Note however, that δ_f is an uncertain parameter without any direct physical interpretation. Before we make use of such a parameter, we could try to get a better expression for f_0 by two-dimensional curve-fitting with C_W and X_{cg}. The magnitude of s_f will then reduce to a very small value. On the other hand, we don't want the expression to become too complicated. For this reason, we decided to keep the one-dimensional fit by neglecting the influence of X_{cg} on this entry. Substitution of $m = 120000\,kg$ and $V_A = 80\,m/s$ gives $a_{77} = -0.03252$, which is equal to the value resulting by numerical linearization at identical conditions.

The single parameter approximation is not valid for other entries like a_{27}, a_{29}, a_{38}, b_{72} or α. a_{38}, b_{72} and α can be approximated accurately using a two-dimensional fitting with respect to C_W and X_{cg}. For two elements, a_{27} and a_{29}, a three-dimensional fitting was determined, each element being determined in the form $f(C_W, X_{cg})Z_{cg} + c$ with c a constant. For each of these two elements we determined first $f(C_W, X_{cg})$ by using a two-dimensional parameter fitting with a fixed Z_{cg} and then we performed a one-dimensional fit with Z_{cg} as free parameter to determine c. The expressions of all the resulting matrices depending rationally on the parameters are given in Appendix B.

At this point another issue arises. That is the choice of parameters in the description of the linear model. Naturally we choose m, X_{cg}, Z_{cg} and V_A. However, it will be clear that this will lead to a huge Δ-block in the LFT-realisation, even after order reduction. Each occurrence of a parameter in a polynomial, multiplied with its degree, adds an extra element to the Δ-block. Where $C_W \sim m/V_A^2$ occurs, one element is added for m and two for V_A.

We can do something about this by choosing C_W and V_A as independent parameters, so that each occurrence of C_W adds only one element to the $\Delta - block$.

4 GENERATION OF LFT DESCRIPTIONS

4.1 General aspects

In this chapter we present an approach to convert parametric description of a linear system of the forms (3) or (6) into LFT-based uncertainty descriptions. Because an affinely dependent function is a particular class of rational function, the approach presented in this chapter is applicable for both types of models. Note however, that for the affine parameter-dependent representation (3), a more efficient approach is to use the special techniques described in [23].

We assume that the elements of the matrices of the linearized system (6) depend only on the parameters p_i, $i = 1, \ldots, q$. (As mentioned before, the vector p optionally can include some components of the equilibrium point ($\bar{x}$, $\bar{u}$).) Any uncertainty in a parameter p_i expressed as $p_i \in [\,\underline{p_i},\ \bar{p}_i\,]$ can be transcribed in a normalized form $p_i = p_{i0} + s_{i0}\delta p_i$ with $|\delta p_i| \leq 1$, $p_{i0} = (\underline{p_i} + \bar{p}_i)/2$ and $s_{i0} = (\bar{p}_i - \underline{p_i})/2$. This local parameter uncertainty is then expressed as an elementary upper LFT

$$p_i = \mathcal{F}_u\left(\left[\begin{array}{cc} 0 & s_{i0} \\ 1 & p_{i0} \end{array}\right], \delta p_i\right).$$

Recall that for a partitioned matrix M

$$M = \left[\begin{array}{cc} M_{11} & M_{12} \\ M_{21} & M_{22} \end{array}\right] \in \mathbb{R}^{(p_1+p_2)\times(q_1+q_2)}$$

and for $\Delta \in \mathbb{R}^{q_1\times p_1}$, the *upper* LFT $\mathcal{F}_u(M, \Delta)$ is defined as

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

If all elements of matrices A, B, C and D are rational functions in parameters p_i, $i = 1, \ldots, q$, then the parametric uncertainties at the components level can be transformed to structured uncertainties at the level of system matrices by using the properties of LFTs [5]. Thus for the system matrices, LFT uncertainty models can be generated in the form

$$\begin{array}{rclrcl} A(p) & = & \mathcal{F}_u\left(\left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & A_0 \end{array}\right], \Delta_A\right), & B(p) & = & \mathcal{F}_u\left(\left[\begin{array}{cc} B_{11} & B_{12} \\ B_{21} & B_0 \end{array}\right], \Delta_B\right), \\ C(p) & = & \mathcal{F}_u\left(\left[\begin{array}{cc} C_{11} & C_{12} \\ C_{21} & C_0 \end{array}\right], \Delta_C\right), & D(p) & = & \mathcal{F}_u\left(\left[\begin{array}{cc} D_{11} & D_{12} \\ D_{21} & D_0 \end{array}\right], \Delta_D\right), \end{array}$$

where Δ_A, Δ_B, Δ_C and Δ_D are diagonal matrices having on the diagonals the normalized uncertainty parameters δp_1, δp_2, $\ldots$, δp_q. Notice that A_0, B_0,

C_0 and D_0 are the *nominal* values of the respective matrices (for all δp_i set to zero).

Procedures to generate LFT-based uncertainty descriptions have been proposed in [19] and [3]. The resulting LFT-based parametric descriptions are generally non-minimal. The construction of minimal order descriptions is essentially a multidimensional minimal realization problem, which even for the 2-D case is a difficult problem to solve. Without special concern for minimization of the orders of the generated LFTs, the resulting LFTs have usually a much higher order than the minimal order. An ad-hoc procedure suggested in [19] can be used for reducing the order of individual repeated blocks. The procedure essentially solves 1-D minimal realizations problems for each repeated block. Although there is no guarantee for minimality, this procedure is apparently effective on many practical examples. A more involved approach based on model reduction techniques for LFT systems can be also used [21]. The use of the latter procedure involves the solution of Lyapunov-type *linear matrix inequalities* (LMIs).

4.2 Software tools

For the approach proposed in [19] MATLAB implementations are also available in the PUM software package [13]. Our approach is based on the recently developed interface between the Maple symbolic parametric description and the PUM software. The Maple procedure `maple2pum` was implemented to generate a MATLAB script file which allows the complete automation of the generation of the LFT representations for the system matrices with rational entries. This script simply performs all operations which are usually manually done to run the PUM software, that is, loading the symbolic information about each rational matrix entry into the PUM database MUNCMOD, running PUM to generate the LFT description and order reduction, and performing optionally the update or storing of the generated LFT model. The main problem with the usage of the PUM software is the usually high order of the generated LFT models. Although some improvement of the original software has been performed to lower the orders of generated intermediary LFTs, the final LFTs have sometimes excessively large orders which lead to numerical difficulties during order reduction or in using the μ-analysis software.

An alternative approach which potentially leads to lower order LFT models is to use Maple instead of MATLAB, to generate the LFT-based parametric descriptions. This approach is attractive because the matrices of the linearized model are already available symbolically in Maple and thus further manipulations of systems matrices can easily be performed to reduce the order of generated LFTs. For instance, the separation of common expressions in several elements can drastically reduce the resulting orders. Such an approach can be efficiently combined with the alternative realization procedure proposed in [3]. Another possibility to exploit is the optimization feature present in Maple to minimize the number of multiplications in evaluating polynomials in several variables. The usage of this feature, in conjunction with the generation of LFT descriptions within Maple, leads implicitly to the minimization of the orders of all generated intermediary LFT descriptions of the individual entries of the system matrices. This in turn contributes to reduce the order of the global LFT description for the entire matrices.

4.3 RCAM LFT models

We generated three LFT-based uncertainty models of RCAM, starting from the rational system matrices presented in Appendix B by using appropriate substitutions. The generated models correspond to a symmetric horizontal flight at constant air speed of $V_A = 80\,m/s$. In the first model the air speed value was assumed exactly known (no uncertainty on air speed). This model served mainly for the post-design assessment of the stability robustness of RCAM controllers [14]. More involved LFT descriptions have been obtained by adding the air speed as an uncertain parameter. The second model was obtained by taking C_W as uncertain parameter instead of m. The advantage of this model is its lower order in comparison with the third model, where m and V_A are used as uncertain parameters. The orders of blocks of the LFT models are summarized in the following table:[1]

Parameters	m	C_W	X_{cg}	Z_{cg}	V_A	Order of Δ
Model I	17	0	15	3	0	35
Model II	0	43	19	5	23	90
Model III	50	0	41	8	204	303

Note that the obtained orders correspond almost certainly to non-minimal realizations of the corresponding LFTs. However, this redundancy doesn't practically influence the usability of the generated models for robustness analysis and robust synthesis purposes.

5 CONCLUSION

It has been shown that a computer aided generation of LFT-based parametric uncertainty descriptions from a generic aircraft-dynamics model with explicit parametric uncertainties is possible by using various specialized numerical and

[1] LFT-models available on request (mat-file), please contact first author.

symbolic software tools. The resulting models are well suited to be immediately used in available μ-analysis and synthesis software tools [1]. The proposed approach is also of generic value, being applicable to similar model classes encountered in practice. To problems with *non-explicit* parameter dependencies, as for example those with parameters defined by *table look-up* procedures, the proposed approach is applicable provided rational approximations can be found to replace all non-analytical functional dependencies. Interesting open problems are the minimal realization and the order reduction of LFT representations.

A. NOMENCLATURE FOR RCAM: STATES, INPUTS, OUTPUTS, PARAMETERS

The nomenclature used and detailed information on RCAM can be found in [12]. The following tables summarize this nomenclature, as it is used for the formulation of the RCAM. In these tables, F_E denotes the earth-fixed reference frame, F_B denotes the body-fixed reference frame, F_V denotes the vehicle-carried vertical frame, and F_M denotes the measurement reference frame (see [12]).

The state variables of the nonlinear RCAM are expressed in SI units and are defined in Table 1. In this table, 'CoG' denotes 'Centre of Gravity'.

The control inputs to the model are given in Table 2. The outputs from the model are given in SI units and are shown in Table 3. Only the model outputs labeled as 'measured' can be assumed to be available as inputs to the controller that is to be designed. The 'simulation' outputs are only intended to be used for evaluation and should not be used in the final controller.

The parameters used in RCAM are given in Table 4. The parametric changes for RCAM are defined in Table 5.

Table 1. States definitions

Symbol			Name	Unit
p	x(1)	=	roll rate (in F_B)	rad/s
q	x(2)	=	pitch rate (in F_B)	rad/s
r	x(3)	=	yaw rate (in F_B)	rad/s
ϕ	x(4)	=	roll angle (Euler angle)	rad
θ	x(5)	=	pitch angle (Euler angle)	rad
ψ	x(6)	=	heading angle (Euler angle)	rad
u_B	x(7)	=	x component of inertial velocity in F_B	m/s
v_B	x(8)	=	y component of inertial velocity in F_B	m/s
w_B	x(9)	=	z component of inertial velocity in F_B	m/s
x	x(10)	=	x position of aircraft CoG in F_E	m
y	x(11)	=	y position of aircraft CoG in F_E	m
z	x(12)	=	z position of aircraft CoG in F_E	m

Table 2. Inputs definitions

Symbol			Name	Unit
δ_A	u(1)	=	aileron deflection	*rad*
δ_T	u(2)	=	tailplane deflection	*rad*
δ_R	u(3)	=	rudder deflection	*rad*
δ_{TH_1}	u(4)	=	throttle position of engine 1	*rad*
δ_{TH_2}	u(5)	=	throttle position of engine 2	*rad*

Table 3. Outputs definitions

Symbol			Name	Unit
Measured				
q	y(1)	=	pitch rate (in F_B) = x(2)	*rad/s*
n_x	y(2)	=	horizontal load factor (in F_B) = $\frac{F_x}{mg}$	-
n_z	y(3)	=	vertical load factor (in F_B) = $\frac{F_z}{mg}$	-
w_V	y(4)	=	z component of inertial velocity in F_V	*m/s*
z	y(5)	=	z position of aircraft CoG in F_E = x(12)	*m*
V_A	y(6)	=	air speed	*m/s*
V	y(7)	=	total inertial velocity	*m/s*
β	y(8)	=	angle of sideslip	*rad*
p	y(9)	=	roll rate (in F_B) = x(1)	*rad/s*
r	y(10)	=	yaw rate (in F_B) = x(3)	*rad/s*
ϕ	y(11)	=	roll angle (Euler angle) = x(4)	*rad*
u_V	y(12)	=	x component of inertial velocity in F_V	*m/s*
v_V	y(13)	=	y component of inertial velocity in F_V	*m/s*
y	y(14)	=	y position of aircraft CoG in F_E = x(11)	*m*
χ	y(15)	=	inertial track angle	*rad*
Simulation				
ψ	y(16)	=	heading angle (Euler angle) = x(6)	*rad*
θ	y(17)	=	pitch angle (Euler angle) = x(5)	*rad*
α	y(18)	=	angle of attack	*rad*
γ	y(19)	=	inertial flight path angle	*rad*
x	y(20)	=	x position of aircraft CoG in F_E = x(10)	*m*
n_y	y(21)	=	lateral load factor (in F_B)= $\frac{F_y}{mg}$	-

Table 4. Parameters definitions

Symbol		Name	Default	Unit
Mass Parameters				
m	=	aircraft total mass	120 000	kg
X_{cg}	=	x position of the CoG in F_M	0.23 $\bar{c}$	m
Y_{cg}	=	y position of the CoG in F_M	0	m
Z_{cg}	=	z position of the CoG in F_M	0	m
Aerodynamic Parameters				
$\bar{c}$	=	mean aerodynamic chord	6.6	m
S	=	wing planform area	260.0	m^2

Table 5. Possible parameter choices in RCAM

Parameter		Bounds				
m	:	100 000 kg	$<$	m	$<$	150 000 kg
X_{cg}	:	0.15 $\bar{c}$	$\leq$	X_{cg}	$\leq$	0.31 $\bar{c}$
Y_{cg}	:	-0.03 $\bar{c}$	$\leq$	Y_{cg}	$\leq$	0.03 $\bar{c}$
Z_{cg}	:	0.0 $\bar{c}$	$\leq$	Z_{cg}	$\leq$	0.21 $\bar{c}$
V_A	:	1.23 V_{stall}	$\leq$	V_A	$\leq$	90 m/s

In the above table V_{stall} is defined from the following equilibrium relation:

$$mg = \frac{1}{2}\rho V_{stall}^2 C_{L_{max}},$$

where $C_{L_{max}} = 2.75$ and ρ is the air density.

B. RATIONAL PARAMETER-DEPENDENT SYSTEM MATRICES

$$
A(p) = \left[\begin{array}{cccccccccccc}
-117.05\,\frac{1}{C_w\,V_A} & 0 & 50.807\,\frac{1}{C_w\,V_A} & 0 & 0 & 0 & 0 & \frac{-2.2278-0.054189\,X_{cg}+2.5880\,Z_{cg}}{C_w\,V_A} & 0 & 0 & 0 & 0 \\
0 & \frac{0.70528\,Z_{cg}-96.507+24.879\,X_{cg}}{C_w\,V_A} & 0 & 0 & 0 & 0 & 0.061601\,\frac{\tilde{a}_{27}}{C_w\,V_A} & 0 & 0.061601\,\frac{\tilde{a}_{29}}{C_w\,V_A} & 0 & 0 & 0 \\
4.8192\,\frac{1}{C_w\,V_A} & 0 & -48.116\,\frac{1}{C_w\,V_A} & 0 & 0 & 0 & 0 & 0.061601\,\frac{\tilde{a}_{38}}{C_w\,V_A} & 0 & 0 & 0 & 0 \\
1.0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.0004 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{-1.9860\,\tilde{b}_{72}-1.0\,{V_A}^2\,\alpha\,C_w}{C_w\,V_A} & 0 & 0 & -9.8061 & 0 & -0.061601\,\frac{\tilde{a}_{77}}{C_w\,V_A} & 0 & -0.061601\,\frac{\tilde{a}_{79}}{C_w\,V_A} & 0 & 0 & 0 \\
V_A\,\alpha & 0 & -V_A & 9.8061 & 0 & 0 & 0 & -15.697\,\frac{1}{C_w\,V_A} & 0 & 0 & 0 & 0 \\
0 & \frac{-241.25+0.0040000\,C_w\,V_A+{V_A}^2\,C_w}{C_w\,V_A} & 0 & 0 & -9.8100\,\alpha & 0 & -0.061601\,\frac{\tilde{a}_{97}}{C_w\,V_A} & 0 & -0.061601\,\frac{\tilde{a}_{99}}{C_w\,V_A} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.000043244 & 0 & 1 & 0 & \alpha & 0 & 0 & 0 \\
0 & 0 & 0 & -V_A\,\alpha & 0 & V_A & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -V_A & 0 & -\alpha & 0 & 1 & 0 & 0 & 0
\end{array}\right]
$$

$B(p) =$

$$\begin{bmatrix}
\frac{-0.97053}{C_w} & 0 & \frac{0.33355+0.008129\,X_{cg}-0.38821\,Z_{cg}}{C_w} & 301.18\,\frac{1}{C_w\ {V_A}^2} & -301.18\,\frac{1}{C_w\ {V_A}^2} \\
0 & \frac{0.02188\,Z_{cg}-2.9935+0.77170\,X_{cg}}{C_w} & 0 & \frac{2152.8+7478.4\,Z_{cg}}{C_w\ {V_A}^2} & \frac{2152.8+7478.4\,Z_{cg}}{C_w\ {V_A}^2} \\
\frac{-0.02032}{C_w} & 0 & \frac{-0.41990+0.15568\,X_{cg}-0.008129\,Z_{cg}}{C_w} & 5768.4\,\frac{1}{C_w\ {V_A}^2} & -5768.4\,\frac{1}{C_w\ {V_A}^2} \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & -0.00077\,\frac{\tilde{b}_{72}\,V_A}{C_w} & 0 & \frac{72517.0}{C_w\ {V_A}^2} & \frac{72517.0}{C_w\ {V_A}^2} \\
0 & 0 & 2.3545\,\frac{1}{C_w} & 0 & 0 \\
0 & -7.48317\,\frac{1}{C_w} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

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

$$\begin{aligned}
\tilde{a}_{27} &= 2.1451\,X_{cg}\ {C_w}^2 Z_{cg} + 0.058556\,X_{cg}\ C_w\ Z_{cg} - 20.291\,X_{cg}\ C_w + 1.1425\,X_{cg}\ {C_w}^2 \\
&\quad -0.90635\ {C_w}^2 - 9.5334 + 9.2389\ C_w + 18.030\,X_{cg} - 5.7399\,Z_{cg} - 5.6075\ {C_w}^2 Z_{cg} \\
&\quad -0.97164\,X_{cg}\ Z_{cg} + 5.7418\ C_w\ Z_{cg} \\
\tilde{a}_{29} &= 1.6726\,X_{cg}\ {C_w}^2 Z_{cg} - 0.17230\,{X_{cg}}^2 C_w - 3.9324\,X_{cg}\ C_w\ Z_{cg} - 0.28903\,{X_{cg}}^2 {C_w}^2 Z_{cg} - 46.850 \\
&\quad -0.070972\,{X_{cg}}^2 Z_{cg} + 0.29652\,{X_{cg}}^2 C_w\ Z_{cg} + 4.9667\,X_{cg}\ C_w - 2.7036\,X_{cg}\ {C_w}^2 + 0.58292\ {C_w}^2 \\
&\quad -0.25564\,{X_{cg}}^2 - 1.3439\ C_w + 100.13\,X_{cg} - 14.251\,Z_{cg} - 1.9116\ {C_w}^2 Z_{cg} + 1.1243\,X_{cg}\ Z_{cg} \\
&\quad +24.656\ C_w\ Z_{cg} + 0.45703\,{X_{cg}}^2 {C_w}^2 \\
\tilde{a}_{38} &= 0.096425\,{X_{cg}}^2 C_w - 0.086069\,{X_{cg}}^2 + 1.6082\,X_{cg}\ C_w - 16.591\,X_{cg} - 7.0577\ C_w + 18.418 \\
\tilde{a}_{77} &= 1.5667\ {C_w}^2 - 16.241\ C_w + 65.449 \\
\tilde{a}_{79} &= -201.39\ C_w + 121.84 \\
a_{97} &= 144.91\ C_w + 171.66 \\
\tilde{a}_{99} &= 24.355\ {C_w}^2 + 6.0937\ C_w + 962.75 \\
\alpha &= -0.041088\,X_{cg}\ C_w - 0.0053886\,X_{cg} + 0.17559\ C_w - 0.16287 \\
\tilde{b}_{72} &= 4.9092\,X_{cg}\ C_w + 0.73956\,X_{cg} - 21.270\ C_w + 19.721
\end{aligned}$$

$$C(p) = \left[\begin{array}{cccccccccccc}
0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -0.20245\,\frac{\tilde{b}_{72}}{C_w\ V_A} & 0 & 0 & 0.00040155 & 0 & -0.0062794\,\frac{\tilde{a}_{77}}{C_w\ V_A} & 0 & -0.0062794\,\frac{\tilde{a}_{79}}{C_w\ V_A} & 0 & 0 & 0 \\
0 & \frac{-24.593+0.00040000\,C_w\ V_A}{C_w\ V_A} & 0 & 0 & 0 & 0 & -0.0062794\,\frac{\tilde{a}_{97}}{C_w\ V_A} & 0 & -0.0062794\,\frac{\tilde{a}_{99}}{C_w\ V_A} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -V_A & 0 & -\alpha & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & V_A^{-1} & 0 & 0 & 0 & 0 \\
1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.000043244 & 0 & 1 & 0 & \alpha & 0 & 0 & 0 \\
0 & 0 & 0 & -V_A\,\alpha & 0 & V_A & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 \\
0 & 0 & 0 & -\alpha & 0 & 1 & 0 & V_A^{-1} & 0 & 0 & 0 & 0
\end{array}\right]$$

$$D(p) = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & -0.0000785\frac{V_A\tilde{b}_{72}}{C_w} & 0 & \frac{7392.15}{C_w\ V_A^2} & \frac{7392.15}{C_w\ {V_A}^2} \\
0 & -\frac{0.76281}{C_w} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

REFERENCES

1. G. Balas, J. Doyle, K. Glover, A. Packard, and R. Smith. *μ-Analysis and Synthesis Toolbox 2.0.* The MathWorks Inc., Natick, MA, 1993.
2. R. Brockhaus. *Flugregelung.* Springer-Verlag, Berlin, 1994.
3. 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.
4. J. C. Doyle. Analysis of feedback systems with structured uncertainties. *IEE Proceedings*, 129, Part D:242–250, 1982.
5. J. C. Doyle, A. Packard, and K. Zhou. Review of LFTs, LMIs, and μ. In *Proc. of 30th CDC, Brighton, England*, pages 1227–1232, 1991.
6. J.C. Doyle, K. Lenz, and A. Packard. Design examples using μ-synthesis: space shuttle lateral axis FCS during reentry. In R. F. Curtain, editor, *Modelling, Robustness, and Sensitivity Reduction in Control Systems*, volume F-34 of *NATO ASI Series*, pages 127–163. Springer Verlag Berlin, 1987.
7. H. Elmquist. Object-oriented modeling and automatic formula manipulation in Dymola. Scandinavian Simulation Society SIMS'93, Konsberg, Norway, June 1993.
8. H. Elmquist. *Dymola – Dynamic Modeling Language. User's Manual.* Dynasim AB, 1994.
9. B. W. Char *et al. Maple V Language Reference Manual.* Springer-Verlag, 1991.
10. P. Gahinet, A. Nemirovski, A. Laub, and M. Chilali. *LMI Control Toolbox User's Guide.* The MathWorks Inc., Natick, MA, 1995.
11. G. Grübel, H.-D. Joos, M. Otter, and R. Finsterwalder. The ANDECS design environment for control engineering. In *Prepr. of 12th IFAC World Congress, Sydney, Australia*, 1993.
12. P. Lambrechts, S. Benanni, G. Looye, and D. Moormann. The RCAM Design Challenge Problem Description. In J.-F. Magni, S. Bennani, and J. Terlouw, editors, *Robust Flight Control, A Design Challenge*, volume 224 of *Lecture Notes in Control and Information Science*, pages 341–359. Springer-Verlag, London, 1997.
13. P. F. Lambrechts and J. C. Terlouw. *A Matlab Toolbox for Parametric Uncertainty Modeling.* Philips Research Eindhoven and NLR Amsterdam, 1992.
14. G. Looye, D. Moormann, A. Varga, and S. Bennani. Post-design stability robustness assessment of the RCAM controller design entries. Technical Publication TP-088-35, Group for Aeronautical Research and Technology in Europe. Technical report, GARTEUR-FM(AG-08), 1997.
15. Mathworks: *SIMULINK Dynamic System Simulation Software,* The MathWorks Inc., Natick, MA, 1992.
16. D. Moormann. Aircraft Physical Modelling Leading to Automatic Code Generation. Chapter 3 in Robust Flight Control Design Challenge. Problem Formulation and Manual: the Research Civil Aircraft Model (RCAM), GARTEUR AG-08 TP-088-3; also: The High Incidence Research Model (HIRM), GARTEUR AG-08 TP-088-4. Technical report, DLR Oberpfaffenhofen, 1996.
17. M. Otter. DSblock: A neutral description of dynamic systems. Technical report tr r81-92, DLR Oberpfaffenhofen, May 1992.
18. J. Schuring and R. M. P. Goverde. A μ-Synthesis Approach (2). In J.-F. Magni, S. Bennani, and J. Terlouw, editors, *Robust Flight Control, A Design Challenge*, volume 224 of *Lecture Notes in Control and Information Science*, pages 341–359. Springer-Verlag, London, 1997.
19. J. Terlouw, P. Lambrechts, S. Bennani, and M. Steinbuch. Parametric uncertainty modeling using LFTs. In *Proc. of AIAA GNC Conf., Hilton, South*

Carolina, 1992.
20. A. Varga, D. Moormann, D. Kaesbauer, and G. Grübel. From generic aircraft models towards LFTs based parametric uncertainties descriptions. In F. Breitenecker and I. Husinsky, editors, *Proc. 1995 EUROSIM Conference, Vienna, Austria*, pages 409–414. Elsevier, Amsterdam, 1995.
21. W. Wang, J. Doyle, and C. Beck. Model reduction of LFT systems. In *Proc. 30th CDC, Brighton, England*, pages 1233–1238, 1991.
22. P. M. Young, M. P. Newlin, and J. C. Doyle. μ analysis with real parametric uncertainty. In *Proc. 30th CDC, Brighton, England*, pages 1251–1256, 1991.
23. K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice Hall, 1996.