TR R323-99
April 1999

A methodology for multi-objective design assessment and flight control synthesis tuning

Hans-Dieter Joos

Aerospace Science and Technology, Elsevier, Paris

DLR Oberpfaffenhofen
Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Robotik & Systemdynamik
Entwurfsorientierte Regelungstechnik
D–82234 Wessling

Multi-Objective Design Assessment and Flight Control Synthesis Tuning

H.-D. Joos,
DLR Deutsches Zentrum für Luft- und Raumfahrt e.V.,
Institute of Robotics and System Dynamics, Oberpfaffenhofen

Abstract

Flight control law design is a multi-variable control p[illegible] t requirements from multiple disciplines have to be satisfied. Visibility both in controller structure and design procedure is necessary to successfully handle such a complex design task. Automated multi-objective synthesis tuning based on a visible goal-attainment strategy is proposed to solve the problem. Nonlinear parameter optimisation is applied to solve multi-variable synthesis parameter tuning. Visibility of the design is based on a unified valuation of requirements by formulation of quality functions. Comprehensible linguistic expressions and demand verbalisation are used to normalise the various criteria making design alternatives quantitatively comparable. This approach is illustrated by flight control law case studies.

1 Introduction

Aircraft flight control law design is a challenging task: It is intrinsically a multivariable control problem with multiple sensors and multiple actuators to cope with strong dynamics interaction. High dynamics performance and safety demands enforce a multitude of design requirements to be satisfied concurrently. Typically, for autopilots such requirements refer to mission performance (tracking error and disturbance rejection), stability robustness with respect to key system parameters (mass, centre of gravity, computation time delay) as well as unmodelled dynamics (e.g. gain and phase margins), ride quality (passenger and pilot comfort by bounds on allowable acceleration and minimum damping), safety (envelope safeguards), and control activity (power consumed by the controls, control rates). In case of piloted flight, handling criteria such as, e.g., the CAP criterion, the C* criterion, the Neal-Smith criterion, and Duda's OLOP criterion for avoiding PIO design flaws, have to be considered in addition.

Model based multivariable control theory offers various kinds of synthesis methods [30] to parameterise multivariable control laws, while intrinsically coping with multivariable dynamics integration. Here the control engineer does not tune the multiple control law gains and filter parameters directly as in classical control engineering, rather s/he has to tune some synthesis parameters instead. Depending on the synthesis method, synthesis parameters may be components of the system eigenstructure, weighting matrices in an integral criterion, or weighting filters to shape the frequency behaviour of loop transfer singular values or structured singular value μ.

Manual synthesis tuning, is the common practice in classical as well as advanced modern control engineering. But manual tuning lacks a systematic for handling multiple tuners simultaneously. Manual tuning is inclined to favour a control law structure or synthesis method which allows sequential or 'add-on' search one parameter at a row. This eventually leads to more complex system solutions than it ought to be. In particular those methods which use 'add-on' performance weighting filters tend to high order synthesis models including such filters. This yields control laws of high dynamics order with corresponding demands on embedded computer implementation. Examples, notably of μ-synthesis D-K iteration, are given in [30].

This paper deals with the *quantitative* aspects of control law design *complementing* the qualitative structural insight provided by control theory. A methodology is developed of *automated synthesis tuning* of all the free tuning parameters simultaneously, with proper *goal-attainment decision visibility* for the design engineer. Concurrent design assessment during the design decision iteration loop [29] yields the necessary information to interactively compromise for a best balanced result with respect to dynamic performance, control effort, and robustness multi operating conditions and parameter tolerances. Based on nowadays high desktop computing power and supported by a pertinent computer-aided control system design environment such as described in [13] or [36], this methodology paves the way to dynamical less complex and more robust control laws. The choice for a specific synthesis method then is guided only by how easy the demands of low controller complexity (state dimension, scheduling difficulty) can be satisfied by the pertaining control law structure, and how visible this structure is with respect to practical realisation considerations, such as actuator limits, elastic interactions, anti-windup/boundary control logic, necessary nonlinear terms, classical stability margins, and the like.

Automated synthesis tuning, in this paper is based on multi-objective goal attainment which is instrumented by nonlinear parameter optimisation methods under inequality constraints. Nonlinear programming [1], [9], search methods like pattern search [17], or guided random search techniques like evolutionary or genetic algorithms [11], are numerical algorithms which can be used. Assessment visibility is based on a unified valuation of requirements' satisfaction by upper bounds on positive (semi-) definite criteria. In particular, linguistic valuation as 'good', 'satisfactory', 'bad' is possible by a type of semi-definite criteria formulation. In case of conflicting requirements interactive compromise negotiation is possible by applying moving upper bounds on a pareto-optimal set of design alternatives. For given requirements pertinent criteria as a function of indicators in time- and frequency domain can be formulated by the design engineer in most natural mathematical terms such as the maximum function, approximated only internally by smooth mathematical expressions for well behaved numerical treatment. Obviously, a pre-progammed repository of established handling quality criteria as well as standard control stability and performance criteria can be developed and made available for ready use as a criteria tool box [5], [36].

This approach has been successfully applied to diverse, nontrivial control design problems, e.g. [13], and has gained quite some maturity over the last decade [14]. Especially, the approach has been applied [21], [24] to both of the GARTEUR robust flight control benchmark problems, the civil one and the military one, yielding results which compare most favourably with all the other design entries documented in [30]. In the GARTEUR industrial assessment [7] of the civil aircraft control design challenge, evaluating control performance and industrial suitability, the approach received highest overall ranking. This is attributed to the fact that any industrial control law structure can be used and given requirements can be dealt with most visibly in their natural mathematical description in time *and* frequency domain. The approach is especially well suited for 'incremental design' by re-using design experience based on a previously developed control law structure and a comparison set of previously achieved assessment results. Hence the approach covers the control engineering task most often encountered in industrial practice.

The paper is organised as follows:

The following Section 2 deals with the diverse aspects of design assessment, i.e. requirements of various kind and different performance indicators. Section 3 deals with comparative assessment by sharp and soft quality functions, visualisation of design alternatives in criteria space as well as the denotations of 'better' and 'best possible' (pareto-optimal) designs for requirements valuation and compromising. Section 4 addresses control law parameterisation by synthesis methods and multi-variable synthesis tuning by means of parameter optimisation. Examples are given for multi-objective evaluation and compromising in flight control law design. Lastly, section 5 deals with multiple model selection and robustness assessment for robust tuning by multi-model compromising.

2 Various aspects of control law evaluation

In flight control law design a multitude of different design requirements has to be dealt with. A documented example for this is the RCAM (Research Civil Aircraft Model) GARTEUR Robust Flight Control Design Challenge [28] which is specified by an extensive set of design requirements in time domain. This benchmark problem addresses design of an autopilot for the final approach of a transport aircraft. For simulation the six degrees of freedom mathematical aircraft model is supplemented with models for wind, turbulence and other external influences. The control law is required to be stability robust with respect to variations in speed, weight, centre of gravity position (both vertical and horizontal), time delays, nonlinearity, and engine failure. It is distinguished between performance specifications, like rise time, settling time, overshoot and cross coupling of airspeed and altitude responses; disturbance attenuation specifications, like path deviation in case of wind; and control requirements, like minimum control energy and control rates. These requirements have been deployed in [21] as quality functions (criteria and demands) shown in Table 1. The robustness requirements have been dealt with by applying these 18 quality functions in parallel to three design models with different worst-case parameter settings. Hence a total of 54 quality functions has been considered concurrently.

	Requirements	Mathematical Criteria	Demands
1	Altitude unit step: zero steady state error, settling time < 45s	$c = \int_{t_1}^{t_2} (h(t)-1)^2 dt$ $t_1 = 10s, \quad t_2 = 30s$	min
2	Altitude unit step: rise time < 12s	$c = t_1 - t_2$ $h(t_1) = 0.1, \quad h(t_2) = 0.9$	<12
3	Cross coupling altitude airspeed: for a step in commanded altitude of 30m, the peak value of the transient of the absolute error between V_A and commanded airspeed should be smaller than 0.5m/s	$c = \max_t \lvert V_A(t) \rvert$	<0.5/30
4	Airspeed unit step: zero steady state error, settling time < 45s	$c = \int_{t_1}^{t_2} (V_A(t)-1)^2 dt$ $t_1 = 10s, \quad t_2 = 30s$	min
5	Airspeed unit step: rise time < 12s	$c = t_2 - t_1$ $V_A(t_1) = 10s, \quad V_A(t_2) = 30s$	<12
6	Cross coupling airspeed altitude: for a step in commanded airspeed of 13m/s, the	$c = \max_t \lvert h(t) \rvert$	<10/13

	peak value of the transient of the absolute error between h and commanded h_c should be smaller than $10m$		
7	Altitude unit step: overshoot < 5%	$c = \max_t h(t)$	<1.05
8	Airspeed unit step: overshoot < 5%	$c = \max_t V_A(t)$	<1.05
9	Airspeed wind disturbance: for a wind step with amplitude of $13m/s$ there should be no deviation in the airspeed larger than $2.6m/s$ for more than $15s$	$c = \max_{t>15} \lvert V_A(t) \rvert$	<2.6
10	Altitude wind disturbance: no explicit specification given	$c = \int_0^{t_2} h^2(t)\, dt$ $t_2 = 30s$	min
	Control activity criteria, effort minimisation for:		
11	tailplane, altitude command	$c = \int_0^{t_2} u^2(t)\, dt$	min
12	throttle, altitude command		min
13	tailplane, airspeed command		min
14	throttle, airspeed command	$c = \int_0^{t_2} \dot{u}^2(t)\, dt$	min
15	throttle, wind step		min
16	throttle rate, wind step		min
17	Relative stability of eigenvalues ev_i: no explicit specification	$c = 1 - \min_i \left(\frac{-\mathrm{Re}(ev_i)}{\lvert ev_i \rvert} \right)$	<0.6
18	Absolute stability of eigenvalues ev_i: no explicit specification	$c = \exp(\max_i(\mathrm{Re}(ev_i)))$	<0.95

Table 1: *Requirements* [28] *and quality functions, i.e. mathematical criteria and demands, used in* [21] *for RCAM longitudinal control law design.*

The requirements and quality functions of Table 1 are typical for design demands of an autopilot control law, expressing step response characteristics and relative and absolute eigenvalue stability. Control laws for piloted flight in addition require avoidance of Pilot-In-the-loop Oscillation (PIO) design flaws and proper attainment of handling quality levels. For instance, the $C*$ handling qualities criterion [37] requires the $C(t)*$ pilot stick-step response to lie within a bounded region. Other types of criteria are defined for transfer functions, such as the CAP criterion [3], or they are based on frequency responses, e.g. the OLOP PIO criterion [4] and the Neal-Smith handling quality criterion [40].

Figure 1 shows the handling quality and eigenvalue stability indicators used for re-tuning the Nz-flight control law of the 'Aerospace Technology Demonstrator' (ATD) aircraft of Dasa-Airbus as modelled in [32]. These handling qualities are to be compromised quantitatively with respect to elevator control rate. The upper left diagram shows flight path angle and pitch angle which should follow the flight path angle induced by Nz-command. The diagram below shows the C*-response together with its bounds for level 1. Eigenvalues indicate transient decay and damping. The right column shows the PIO-indicators 'Phase Rate' and 'Open-Loop Onset Point' (OLOP), as well as the 'Neal-Smith' handling quality indicator.

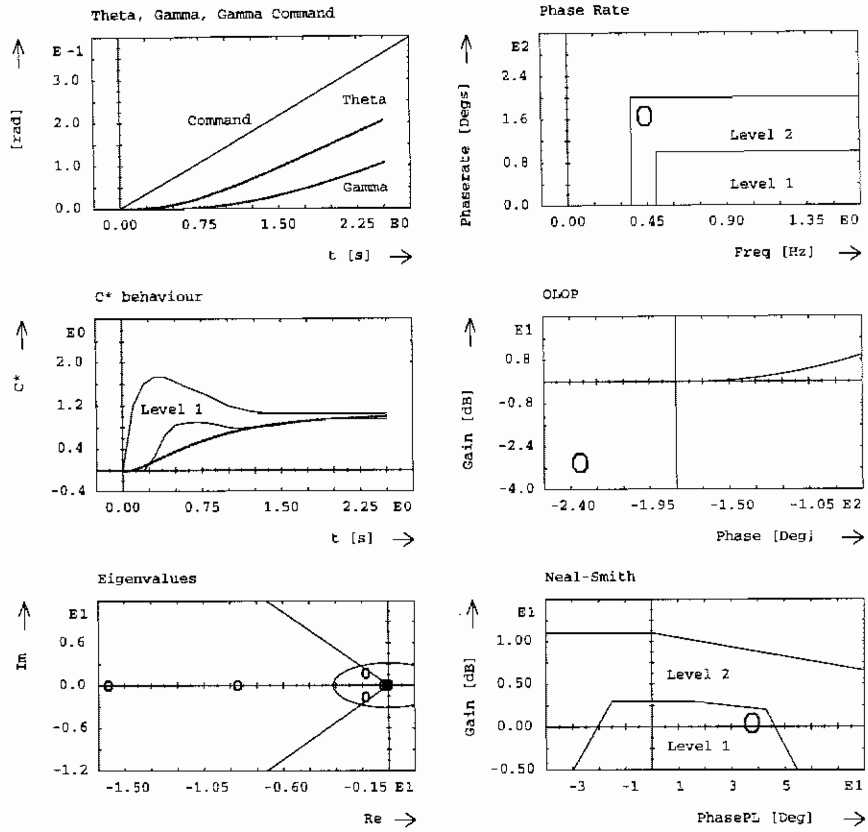

Figure 1: *Indicator functions and indicator points of stability and handling qualities.*

As the above examples indicate, computation of design quality functions requires the following three steps: First, analysis of the controlled system by time and frequency responses and eigenvalue computation. Second, computation of indicator functions or indicator points like the C* response or the Neal-Smith indicator point. Third, evaluation of a real-valued positive criterion which normalises the pertinent indicator to given bounds. Hence control law evaluation requires a computation chain with various types of analysis cases to be executed on different linear/nonlinear evaluation models with different model excitations by either deterministic or stochastic signals.

3 Comparative Assessment of Design Alternatives

3.1 Sharp and soft quality functions

Quality Function Deployment (QFD) of all design requirements is the prime paradigm in Concurrent Design Engineering [16] to achieve a well balanced product by a design process which needs no major re-design loops. It is the means to assess design satisfaction, to compare design alternatives, and to detect and negotiate design conflicts. A *quality function* is a tuple of an evaluation *criterion* $c(i)$ defined as a real-valued mathematical function of system performance indicators i, together with a pertinent *demand* d to evaluate performance satisfaction.

Without loss of generality, we can always formulate a quality function criterion as a real-valued function which assumes the smaller values the better the requirement is satisfied. Table 1 shows various examples. Then, design satisfaction can be valuated either by the demand that criteria values are lower than given upper bounds or that they are as low as possible. Table 1 also gives examples for such demands denoted either by '<' or 'min'. This allows *quality functions* to be written as

$$q_j := c_j / d_j \,, \tag{1a}$$

$q_j \le 1$: requirement j is satisfied (1b)

$q_j > 1$: requirement j is not satisfied (1c)

$q_j = \min$: requirement j has to be as good as possible, where $q_j \le 1$ is a satisfactory solution. (1d)

Besides strict stability boundaries, design requirements usually do not quantify sharp bounds which strictly separate 'good' from 'bad' in a gradual degradation of indicator values. Rather *intervals* of indicator values are addressed to qualify as 'satisfactory (good, level 1)', 'acceptable (level 2)' or 'not acceptable (bad, level 3)'. This can be taken care of mathematically by a suitable transformation of indicators as follows.

Any arbitrarily defined scalar indicator value i can be transformed to a quality function $q(i)$ by means of at most four 'good/bad' value definitions: $b_l < g_l < g_h < b_h$:

$$\begin{aligned} q(i) &\approx \max(L(i), 0, H(i)) \,, \\ L(i) &= (i - g_l)/(b_l - g_l) \,, \qquad b_l < g_l \\ H(i) &= (i - g_h)/(b_h - g_h), \qquad g_l < g_h < b_h \end{aligned} \tag{2}$$

This transformation is illustrated by Figure 2.

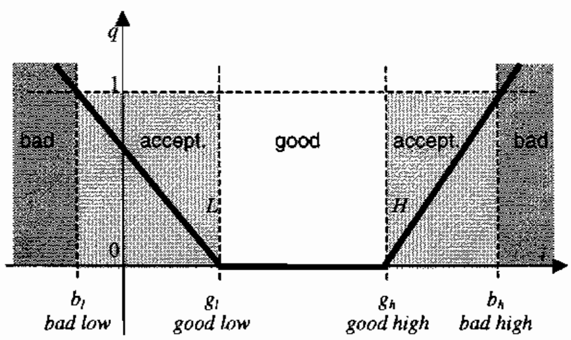

Figure 2: *Transformation of indicator values to interval quality functions.*

The transformation parameters b_l, g_l, g_h, b_h are to be chosen compliant with the following linguistic interpretation:

An indicator value i is considered to be

- 'satisfactory' or 'good' for values between g_l and g_u , i.e. $q << 1$
- 'acceptable' for values between b_l and g_l and between g_h and b_h , $q \le 1$
- 'not acceptable' or 'bad' for values smaller than b_l or greater than b_h, $q > 1$.

With this transformation, a general upper bound $d = 1$ defines a separation between 'acceptable' and 'not acceptable'. An additional feature of such a transformation is that all satisfactory indicator values are mapped to zero, i.e. to lowest possible criterion value. This is of advantage later, where multi objectives are handled by min-max optimisation.

Positive semi-definite interval criteria of this kind constitute 'soft' quality functions as distinguished to the 'sharp' quality functions with positive definite criteria listed in Table 1.

Compound quality functions can be formulated by using the maximum function

$$q(i_1,\cdots,i_j) = \max\{q_1(i_1),\cdots,q_k(i_1),q_{k+1}(i_2),\cdots,q_n(i_j)\}. \tag{3}$$

If the individual members of the maximum function (3) are soft quality functions as defined by (2), such a compound quality function can be formulated in terms of fuzzy logic as follows [22]:

(q has property s) if
(i_1 has property 1) AND (i_2 has property 2) AND … AND (i_j has property n) , (4)

where 'i_j has property …'.means that the indicator value i_j is 'good' or 'acceptable' with respect to its membership function.

As an example consider the stability indicator 'eigenvalue damping' ζ defined as

$$\zeta_i = -\mathrm{Re}\,\lambda_i / \sqrt{\mathrm{Re}^2\,\lambda_i + \mathrm{Im}^2\,\lambda_i}\ , \tag{5}$$

where values greater 0.7 (no matter how big) are considered as good, and values less than 0.3 are considered as bad. To transform ζ to a compliant criterion, the following 'good/bad' values are appropriate:

$$b_l = 0.3,\ g_l = 0.7,\ g_h = arbitrary,\ b_h = \infty\ ,$$

i.e. damping values greater than $b_l = 0.3$ are 'satisfactory' and greater than $g_l = 0{,}7$ are 'good'. Setting $b_h = \infty$ makes $H \equiv 0$, cf. (2), and the transformation looks as in Figure 3.

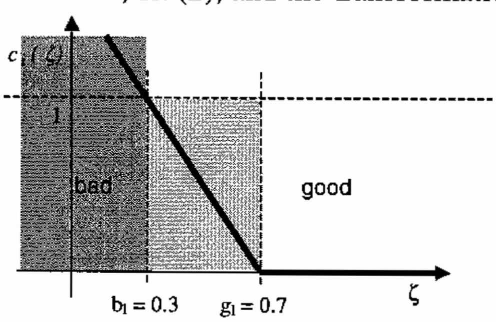

Figure 3: *Example: Transformation of a damping value α to positive semi-definite 'good-bad' criterion.*

Relation (4) can be used to take care of all eigenvalues simultaneously:
($c(\zeta)$ is satisfactory) if (c_1 is satisfactory) AND ... AND (c_n is satisfactory) which is equivalent to $\max\{c_1,\cdots,c_n\} < 1$.

The maximum function, as used in the transformation (2) or in Table 1, is often the natural form to describe indicators and quality functions mathematically. However, the maximum function is a non-smooth function whereas for numerical reasons often smoothness is required. For this purpose the maximum function can be numerically approximated [27] by a smooth function:

$$\max_k \{i_k\} = \lim_{\rho\to\infty} 1/\rho \ln\left(\sum_k \exp(\rho \cdot i_k)\right) \tag{6a}$$

$$= \max_k\{i_k\} + \lim_{\rho\to\infty} 1/\rho \ln\left(\sum_k \exp(\rho(i_k - \max_k\{i_k\})\right). \tag{6b}$$

Formula (6b) is used for numerical evaluation because there the exponent is always less or equal zero and hence unfavourably large values are avoided in evaluating the exponential functions. Summation is also numerically stable since all addends are positive and less than one. For a good approximation a suitable value for ρ is about 20.

If the maximum $\max_t\{i(t)\}$ of a continuous function $i(t)$ has to be taken, this function may be discretised as $i(t_k)$ and $\max_k\{i(t_k)\}$ can be smoothly approximated by formula (6b). An example is the C*-criterion where both the function *C*(t)* and the level 1 upper and lower bounds $C*_u(t)$, $C*_l(t)$ are continuous functions of time. The function

$$C(t)* = n_z(t) + \dot{q}(t)\frac{X\rho}{g} + q(t)\frac{U_m}{g} \tag{7}$$

is a linear combination of load factor n_z, pitch rate q and pitch acceleration $\dot{q}$; $X\rho$ is the distance between pilot seat and centre of gravity and U_m is an average velocity. Indicators for the C* criterion can be formulated by two maximum functions with discretised time t_k

$$i_{C*_u} = \max_k\{C*(t_k) - C*_u(t_k)\} + 1; \quad i_{C*_l} = \max_k\{C*_l(t_k) - C*(t_k)\} + 1 \,. \tag{8}$$

These indicators adopt a value greater than one for C* outside the bounds, equal to one if a C* value reaches the bounds or they are both less than one if the C*-response is completely inside the level 1 bounds. The *C** criterion then can be formulated as
(*C** is level 1) if $(i_{C*_u} \le 1)$ AND $(i_{C*_l} \le 1)$, cf. relation (4).

3.2 Conflicting requirements

Quality functions with positive 'the smaller the better' criteria and quality limiting upper bounds yield a most visible comparative satisfaction assessment of design alternatives. Define for all quality functions

$$\alpha := \max_j\{q_j\}, \quad j = 1,...,J. \tag{9}$$

Then requirements' satisfaction of a design alternative (II) is said to be *better* than of a design alternative (I) if $\alpha^{(II)} < \alpha^{(I)} \leq 1$ for all quality functions. In particular, a *best possible* design alternative is characterised by

$$\alpha^* = \min\{\alpha\} \ . \qquad (10)$$

The method of Kreisselmeier and Steinhauser [27] to achieve a best possible design alternative by vector optimisation is based on finding α^*.

Two requirements are ultimately *in conflict* if for the corresponding quality functions q_i, q_k,

$$q_i = q_k = \alpha^*, \qquad \alpha^* > 1 \ , \qquad (11)$$

which means that with the given upper bounds these requirements cannot be satisfied, and hence there is no satisfactory solution possible.

If there exists a satisfactory solution $\alpha^* \leq 1$, then q_i, q_k with the property

$$q_i = q_k = \alpha^*, \qquad \alpha^* \leq 1 \ , \qquad (12)$$

belong to the set of 'non-inferior' alternatives within the set of all satisfactory solutions.

This is illustrated for two criteria in Figure 4 which shows the set of admissible criteria values in 2-dimensional criteria space. Any point c of this set can be achieved. Accordingly the given criteria upper bounds d are also represented by a point in the criteria space. As defined above a design is considered to be satisfactory if $q_j \leq 1$, *i.e.* $c_j \leq d_j$. Hence all design alternatives in the dark subset are satisfactory. The highlighted border of the admissible criteria space represents the set of non-inferior or pareto-optimal solutions [38]. A member of this set characterises a compromise in the sense that an improvement in one criterion does cause degradation in the other criterion. The design objective is to generate pareto-optimal design alternatives and to negotiate best-possible compromise solutions based on user priorities.

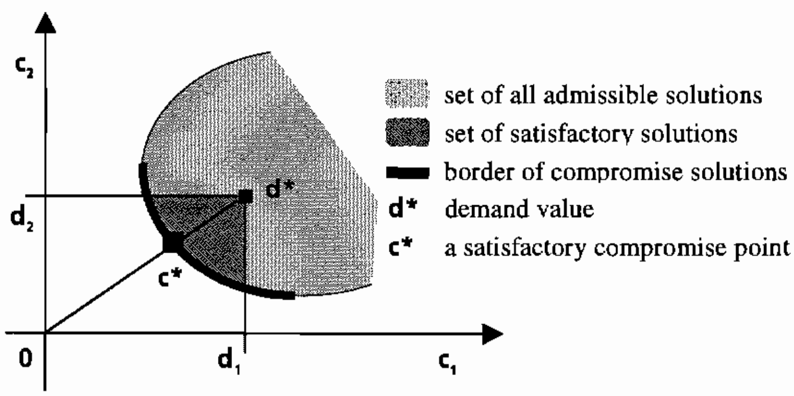

Figure 4: *Quality function satisfaction in 2-dimensional criteria space*

3.3 Visualisation of quality functions

Figure 4 visualises the set of two quality functions $\{c_1 / d_1\}$ and $\{c_2 / d_2\}$ in a 2-dimensional cartesian display. Since comparative design assessment requires to consider *many* quality functions simultaneously we need a *high dimensional display* which allows to show *different* design alternatives concurretly. The means for that is a display in 'parallel co-ordinates' [19], Figure 5. Each quality function is represented on one of the parallel co-ordinate axes and a polygonal line connects all quality functions for one design alternative. In Figure 5 each polynomial line represents one of the 5 different alternatives of the ATD control law re-tuning case study. Satisfaction assessment is most visible: all polygonal lines below a border line of value 1 indicate requirements satisfaction and values above this line indicate design deficiencies. This also allows to detect satisfaction conflicts among requirements: For conflicting criteria the polygonal lines are crossing. In Figure 5 a major satisfaction conflict can be immediately detected between 'elevator rate' and satisfaction of the C*-criterion. (The Phase-Rate criterion here is normalised to level 2 satisfaction.)

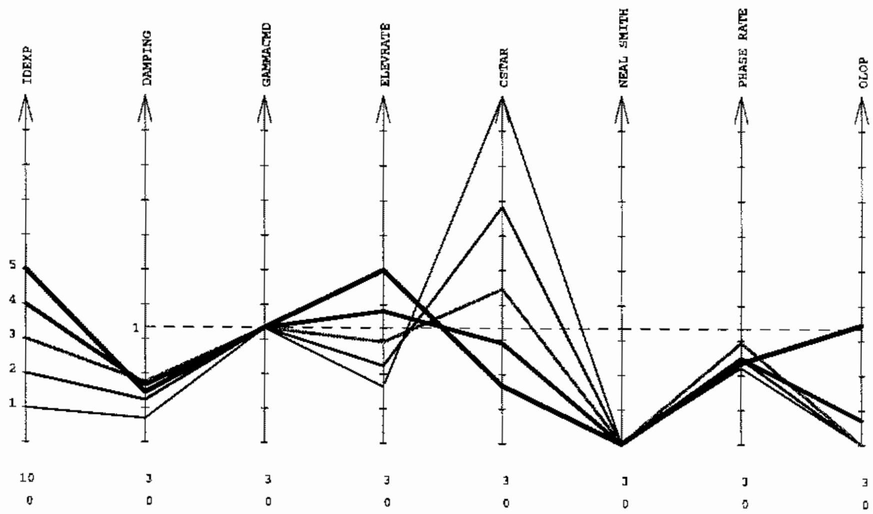

Figure 5: *Parallel co-ordinates display of five ATD control law re-tuning alternatives assessed by seven quality functions each. (The first co-ordinate orders the design alternatives 1...5.)*

Figure 6 shows the pertaining indicators of quality functions of Figure 5.

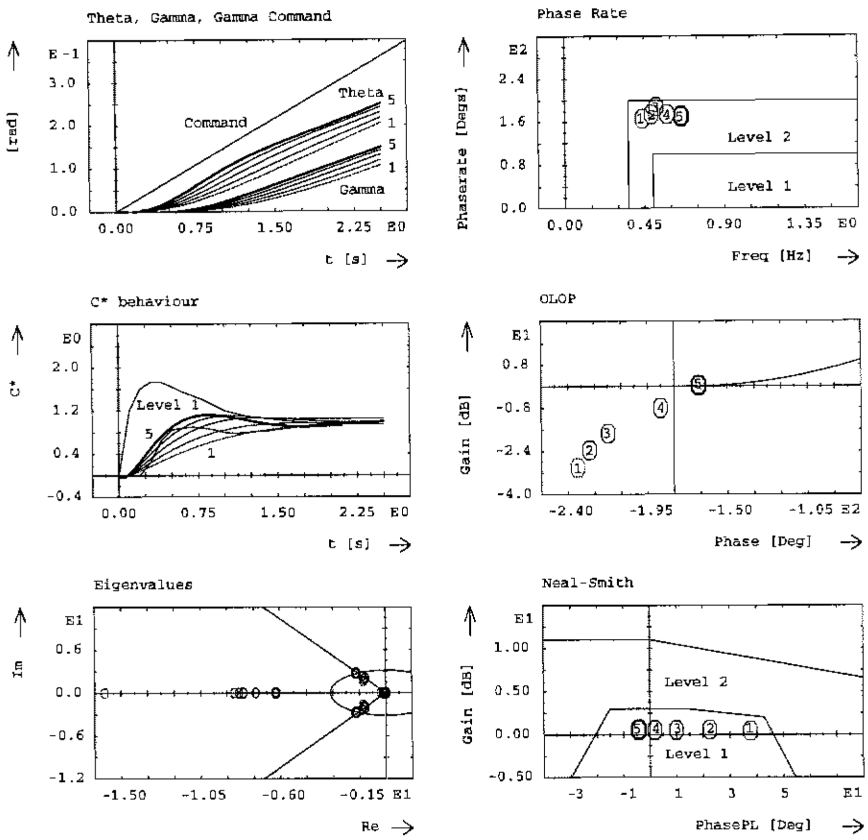

Figure 6: *Indicators corresponding to the quality functions displayed in Parallel Co-ordinates of Figure 5.*

4 Synthesis Tuning

4.1 Various ways of control law parameterisation

Control law synthesis consists of two activities: the development of a proper control law structure taking into account the available feedback- and actuation variables as well as implementation issues, and tuning of the control parameters to satisfy the control requirements as deployed by quality functions. The first activity belongs to the specific domain of control theory. The second activity belongs to the general domain of design engineering.

To cope with the given requirements the control law structure has to be chosen sufficiently rich in independent tuning parameters. This may necessitate design iterations to change the dynamics structure of an already 'best possibly' tuned control structure candidate. To develop a control law structure we may distinguish three basic approaches of control theory: the 'classical' PID error approach, the 'modern' model-based analytic approach, and the rule-based 'fuzzy control' approach.

The classical approach deals with proportional-, integral-, and derivative action on the control error and shaping of dynamic compensation filters to cope with feedback stability. The tuning parameters are the PID gains and filter parameters. An example is the structure and parameterisation of the TECS control law in [6]. Another example of direct control parameterisation is the 'Nz-law' of the ATD re-tuning study referred to in this paper.

The model-based analytic approach of modern control theory provides a broad spectrum of different synthesis methods [30]. They can be mainly classified as eigenstructure methods, as linear quadratic Gaussian (LQG) optimal control methods and as H-inf optimal loop shaping methods. Solvability requirements of the underlying mathematical synthesis problem induce a specific type of control law structure e.g. full state- or observer dynamics feedback. The synthesis formalism parameterises this control structure as a function of free synthesis-tuning parameters such as elements of the desired eigenstructure, parameters of positive definite weighting matrices in a quadratic form, or the parameters of weighting filters for loop shaping. Hence in this approach the control law is parameterised not explicitly but implicitly as a function of the synthesis tuning parameters. An example of aircraft lateral control law design in this view is dealt with in [24] using the Target Feedback Loop/Loop Transfer Recovery (TFL/LTR) synthesis method. Analytic methods of order reduction can be included in synthesis tuning to cope for bandwidth restrictions in digital control implementation [23].

The fuzzy control approach yields a nonlinear-gain feedback control structure specified linguistically by if-then rules on fuzzyfied error actuation variables. Tuning parameters are the scaling coefficients of membership functions and the weights among the rules. An application of this kind of tuning is described in [18] for the synthesis of a robust back-up stabilisation control law for aerodynamically unstable longitudinal flight.

Hence if the control law parameters are denoted by K and the tuning parameters are denoted by T then we may distinguish between direct control law parameterisation and indirect parameterisation via a synthesis algorithm $K = f(T)$:

classical control	$K = T$
analytical model-based control	$K = f(T, \textit{synthesis model})$
fuzzy control	$K = f(T, \textit{fuzzy control rules})$

Direct control law parameterisation is visible with respect to the control law structure. Indirect control law parameterisation via a synthesis algorithm is aimed to be visible with respect to some type of system property, e.g. stability and mode observability by eigenstructure assignment. In particular, indirect control law parameterisation via analytical model-based synthesis algorithms shows its value in multivariable control problems where proper dynamics integration with multiple sensors and multiple actuators must be achieved by strongly interacting control laws. Analytical restraints on the synthesis tuning parameters guarantee particular system properties to be satisfied intrinsically, such as system stability with respect to the synthesis model.

4.2 Tuning by parameter optimisation

Usually there are multiple parameters to be tuned simultaneously. Also, both synthesis parameters and additional control law parameters may have to be tuned concurrently if an analytical control law structure is augmented for specific systems engineering demands such as the need of additional notch filters. In view of the multitude of quality functions to be satisfied, manual sequential tuning of one parameter after another, most likely, is not very efficient neither in engineering time nor in the result which can be achieved. Hence an algorithmic tuning procedure is looked for, which can be used for *automated tuning of multiple, different-type parameters* to solve the quality function inequalities satisfaction problem. The way to proceed is to use nonlinear constrained parameter optimisation.

Note that the quality functions depend on the tuning parameters T, $q(T) = c(T)/d$, via the assessment computation chain according to Figure 7.

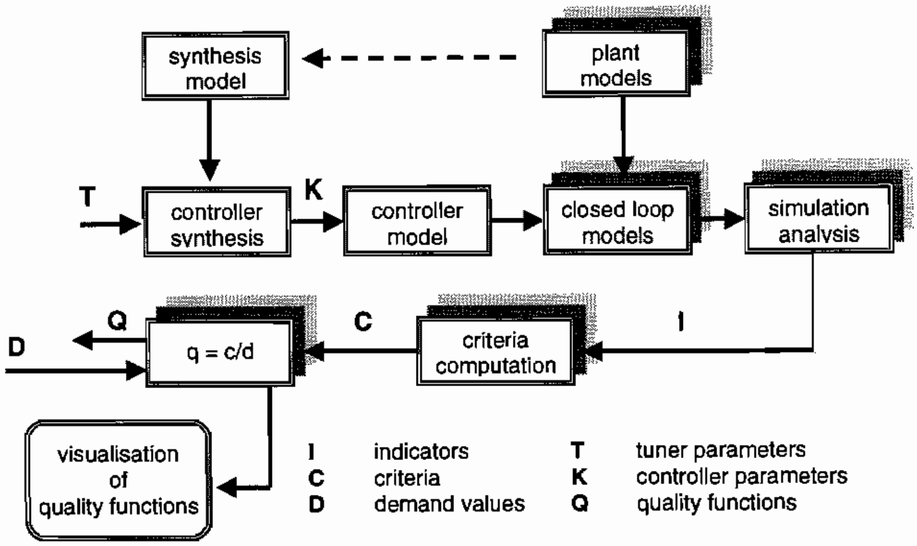

Figure 7: *Computation chain of control law assessment by quality functions, including parameter synthesis $K = f(T,$ synthesis model).*

Requirements' satisfaction demands to find a set of tuning parameter values T^f such that the set of quality function inequalities is satisfied, cf. (1b):

$$q_j(T^f) \leq 1, \text{for all} \quad j = 1,\ldots,I_j \ ,$$

while minimising the quality functions which should be as small as possible, cf. (1d):

$$q_k(T^f) = \min, \text{for all} \quad k = 1,\ldots,I_k \ .$$

This *feasibility problem* can be solved by a constrained minimisation problem with an auxiliary variable $\alpha \geq 0$:

$$\begin{aligned} &\min_{T,\alpha} \{\alpha(T)\} \\ &s.t. \quad q_k(T) \leq \alpha \\ &\qquad\ \ q_j(T) \leq 1. \end{aligned} \tag{13}$$

Any parameter value set T^f satisfying $q_j \leq 1$ is a so-called 'feasible' solution set which satisfies the design requirements. Moreover, if $T = T^*$ minimises (13) then T^* is a pareto-optimal solution [38] and vice versa.

The inequality-constrained parameter optimisation problem (13) can be solved by standard nonlinear programming methods. A well established class of algorithms to solve such problems is Sequential Quadratic Programming (SQP) [1], [9], [25], [39], [41], where the original optimisation problem is approximated by a sequence of quadratic programming problems. Such algorithms usually are designed in such a way that in the first steps they try to get a feasible solution and in the following steps they search for a minimum within the feasible solution set [41]. Hence if one is interested just in a feasible solution, one does not need to wait all the computations for convergence to a minimum.

The constrained minimisation problem (13) is equivalent [34] to a min-max vector optimisation problem

$$\min_{T} \max_{j} \{q_j(T)\} \qquad (14)$$
$$s.t. \quad q_j(T) \le 1.$$

This min-max approach together with the smooth max-approximation (6), is used in [27].

The constrained minimisation problem formulation (13) has some advantage over the min-max formulation (14) in that smoothness of quality function criteria is preserved. In addition, constrained minimisation algorithms allow box constraints on T to be appended.

In [22] it has been shown by example of an aircraft landing control law synthesis, that the min-max formulation (14) is the natural formulation if 'soft' fuzzy-type criteria (4) are used to describe design requirements. The resulting min-max optimisation problem (14) can be solved numerically as the equivalent constrained scalar optimisation problem (13) .

Parameter optimisation (13) or (14) serves to compute feasible or pareto-optimal tuning parameters $T = f(Q)$. It thereby closes the computation chain of Figure 7 yielding an automated tuning loop. Parameter optimisation of the type (13) is used in both the ANDECS-MOPS [13] and the CONDUIT [36] control design suites. The main computation effort is required for executing the synthesis algorithm and the various simulation and analysis cases which are needed to evaluate the quality functions for design assessment. Hence 'cheap' synthesis algorithms such as eigenstructure or Riccati equation algorithms should be favoured for control law parameterisation. Compared to the synthesis and analysis computation cost the optimiser's 'overhead' is only a few percent.

To handle many different quality functions in a flexible way, a Graphical User Interface (GUI) for steering the optimisation set up is helpful.

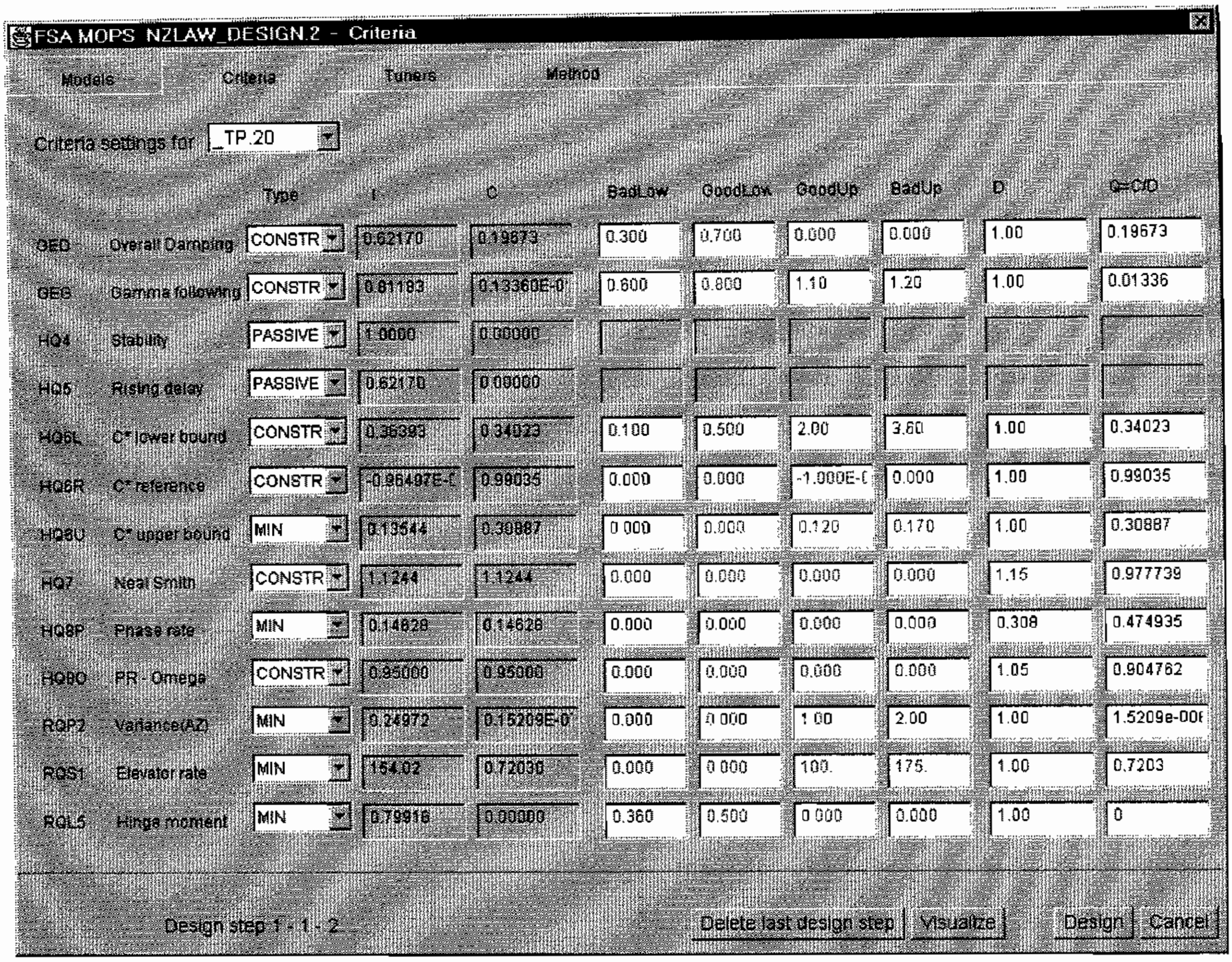

Figure 8: *Graphical user interface to set demand values for standard criteria in flight control.*

A flight control user interface of DLR's ANDECS-MOPS environment, which is used for the case studies described in this paper is shown in Figure 8. This GUI serves for input of criteria related data, like bad/good values, demand values, or the setting of a criterion as constraint. The criteria on display are standard design objectives in flight control, including handling qualities and PIO-criteria. In the lower left corner, the current design step (1-1-2) is indicated referring to the hierarchical design data structure depicted in [20].

4.3 Multiobjective Compromising

While in the previous section parameter optimisation is used for satisfying quality functions with respect to prespecified demands, in this section 'compromising' in criteria space is dealt with. This means the demands d for conflicting criteria c are now used as free *decision variables*. They serve to get some pareto-optimal criteria values improved at the expense of others. How much expense one is willing to pay can be specified by how much the demand and according inequality constraint is relaxed. Doing this interactively is called 'negotiating'. Having the means for quantitative negotiating on a pareto-optimal compromise solution set allows an objective design negotiation between designer and systems engineering manager, of what is possible at all with a chosen control law structure or synthesis method, or whether a change in control law structure becomes necessary to satisfy the given requirements.

Figure 1 shows the indicators resulting of design assessment of a given control law with parameters T^0. The C*-criterion is not satisfied and the PIO-criterion 'Phase Rate' is satisfied only within level 2, whereas the indicators for 'Neal-Smith' and 'OLOP' are of level

1. Some analysis showed that predominantly the C*-criterion is in conflict with the elevator control rate. Hence the question is to be investigated of how efficiently the C* criterion can be improved at the expense of control rate, while keeping all other quality functions to their achieved level of satisfaction. A systematic procedure is as follows: First a pareto optimum is computed starting with the given control gains T^0. I.e. C* and elevator-control rate are minimised by (13) until both quality functions become equal, cf. (11), constraining the other criteria to their level of satisfaction attained so far. Figure 9 shows criteria values marked by '0' for the start value T^0 and marked by '1' for the achieved pareto-optimal solution. Both C* and control rate have been reduced in value and hence both are improved.

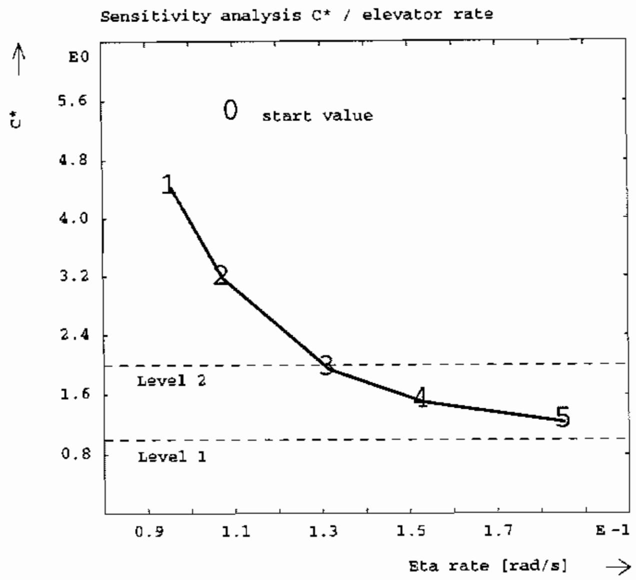

Figure 9: *Set of pareto-optimal compromise solutions in the C*/elevator-rate criteria plane.*

A further compromise point '2' in the pareto-optimal solution set is now sought as follows:

- C* is set to be minimised.
- The demand value for the control rate is relaxed to the value which had been required for the original control gains T^0 and this value is now used as upper bound constraint.

After only a few seconds of computation time, the optimiser increases control rate up to its upper bound and minimises C* as much as possible.

Sequentially relaxing the upper bound for control rate and repeating this procedure 3 times results in the compromise set of Figure 9, which depicts the border shape of Figure 4. Actually, the five tuning alternatives correspond to the ones represented in parallel co-ordinates in Figure 5, the according indicators and analysis function are those from Figure 6. The shape of the pareto-optimal compromise set visualised how worthwhile it is to increase admissible control rate as a trade off to C* . For example, an increase of elevator rate by 0.02 [rad/s] from its given level at tuning case 2 will cause an improvement of C^* performance of about 30% to achieve level 2 at point '3', whereas further increasing control rate yields no comparable improvement of C*.

4.4 Design iteration in control law structure

By example of the study it is now investigated whether and how a dynamically augmented control law can improve handling performance while keeping control rate at the level attained by a previous manual tuning. In the course of the tuning process the controller structure is augmented by a lead-lag filter to get more tuning parameters. We start with a given control law feedback structure

$$u^0 = (K_1 + K_1 K_2 / s)\, Nz + (K_3 + k(H))q \tag{15}$$

with free tuner parameters K_1, K_2, K_3 and with *k(H)* as fixed but height dependent pitch rate feedback gain. We augment this feedback law by a lead-lag filter with time constants T_1, T_2

$$u^1 = u^0\, (s + T_1)/(T_2 s + 1) \tag{16}$$

The following tuning steps have been performed, cf. the columns of Table 2: Note that all criteria are normalised in such a way that $c \le 1$ is 'level 1' and $c \le 2$ is 'level 2'.

1. Analysis of given design, which yields the start values for re-tuning.
2. The 3 parameter control law structure (15) are optimised to improve C*, and possibly 'Phase Rate' while all other quality functions are kept within their attained level of satisfaction. The result is that keeping control rate at the same level allows only improvement in C* while 'Phase Rate' is in conflict and degrades. The chosen setting of demand values and the resulting criteria values are listed in Table 2.
3. Since no major improvement is possible with control rate restricted to its given level, now the controller structure is augmented by a lead-lag filter, providing more design freedom by two additional gains. By the augmented structure now level 2 can be achieved for both C* and 'Phase Rate'.
4. Negotiating C* and 'Phase Rate' by allowing C* to attain the upper bound of level 2 allows 'Phase Rate' to attain level 1, while all other criteria are kept satisfied, in particular, and control rate is not increased with respect to design step 1 we have started with.

	1. Analyse to get start values T^0	2. Optimise to improve C*, phase rate 3 gains	3. Augment by lead lag dynamics to improve C*, phase rate 5 gains	4. Negotiate C* level 2 versus phase rate level 1 5 gains
Damping	satisfied	d=1, constraint c<1, satisfied	d=1, constraint c<1, satisfied	d=1, constraint c<1, satisfied
Gamma command	satisfied	d=1, constraint c<1, satisfied	d=1, constraint c<1, satisfied	d=1, constraint c<1, satisfied
Elevator rate	c=0.39	d=c, constraint c=d, satisfied	d=c, constraint c=d, satisfied	d=c, constraint c=d, satisfied
C*	c=5.84 unsatisfactory	d=α, minimise c=3.11 level 3	d=α, minimise c=1.75 level 2 satisfied	d=2, constraint c=2 level 2 satisfied
Neal Smith	c=0.80 Level 1 satisfied	d=1, constraint c=1 level 1 satisfied	d=1, constraint c=0.72 level 1 satisfied	α =1, constraint c=0.66 level 1 satisfied

Phase rate	c=1.57 Level 2 satisfied	d=α, minimise c=2.19 level 3	d=α, minimise c=1.67 level 2 satisfied	d=α, minimise c=0.97 level 1 satisfied
OLOP	satisfied	d=1, constraint satisfied	d=1, constraint satisfied	d=1, constraint satisfied
Tuners	$K_1 = 1.3$ $K_2 = 2.5$ $K_3 = 0$	$K_1 = 5.13$ $K_2 = 1.27$ $K_3 = 0$	$K_1 = 4.33$ $K_2 = 1.21$ $K_3 = 0$ $T_1 = 0.93$ $T_2 = 0.16$	$K_1 = 5.45$ $K_2 = 1.01$ $K_3 = 0.17$ $T_1 = 0.78$ $T_2 = 0.08$
Computational effort		9 iterations, 30 sec	10 iteration, 43 sec	4 iterations, 15 sec

Table 2: *ATD re-tuning with equal controller effort but alternative controller structures.*

The resulting control gain tuning history is also shown in Table 2. Manually tuning of the three or five control law parameters would have been quite a cumbersome and time consuming trial-and-error task since the appropriate parameter changes behave nonlinearly.

Computation times are measured on a Pentium 266 MHz PC and include dynamic online visualisation like Figure 5 and Figure 6 after each optimisation step to monitor the search iterations. The overall computational effort of these tuning experiments is small. Hence *interactive compromise negotiation* is possible.

5 Robust Control Laws by Multi-Model Tuning

Multi-quality functions' control law tuning by parameter optimisation is not restrained to a specified type of system model and disturbance description. A linear or nonlinear synthesis model for control law parameterisation can be used together with a set of linear/nonlinear assessment cases, i.e. evaluation models and disturbances. Different assessment cases can be dealt with concurrently and this can be used for 'multi-model robust' control law tuning. Especially, multi operating conditions and parameter tolerances can be coped with in this way. A main issue here is negotiating among different models' control performance.

5.1 Robustness to multi operating conditions

Systematic 'multi-model robust' flight control design, for the first time, was applied by Kreisselmeier and Steinhauser, with published results in [26]. By taking into account a number of extreme flight conditions simultaneously, as e.g. in Figure 10, a robust command- and stability augmentation control law *without gain scheduling* for an F4-c aircraft had been designed. Later in [12] a feasibility study of robust back-up stabilisation for the JAS 39 aircraft has been reported on. There, ten extreme flight conditions were taken care of concurrently, characterising the entire flight envelope from high aerodynamic instability in the landing phase up to well-behaved stable supersonic flight. A stability-robust CAP level 1 command- and stability augmentation has been achieved over the entire flight envelope by a fixed-gain, properly tuned, third order dynamics feedback filter from pitch rate to elevator. This was an astonishing result for flight control experts at that time, discovered only by systematic, quantitative compromising using the instrumental approach of multi-objective/multi-model vector optimisation [27].

Usually it turns out that besides a medium 'nominal' flight condition only these flight conditions in the concerned flight envelope have to be taken as 'active' in the multi model negotiating process which show 'worst' dynamic behaviour. In case of Figure 9 these might be conditions No 1, 2 and 5.

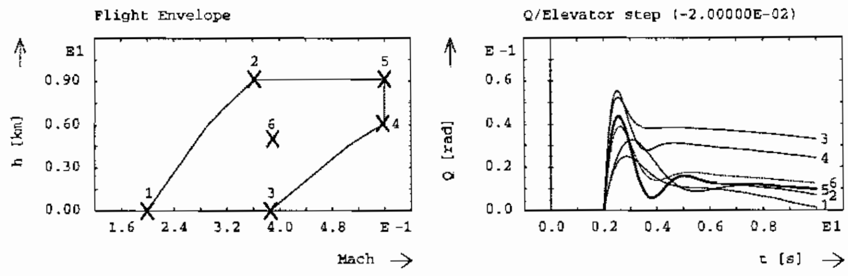

Figure 10: *Extreme flight conditions over the flight envelope used in multi-model robust control tuning.*

In general there exists no theory that guarantees stability or performance robustness across the range of operation, if only a finite number of operating points is considered simultaneously. It depends on the physical properties of the system to be designed, whether deficiencies can exist. If they exist, they have to be added to the set of worst-case operating points treated simultaneously by the multi-model approach. Of course, robustness of the controller 'around' an operating point can be enforced by appending suited robustness criteria (e.g. gain/phase margins) to the set of otherwise specified performance criteria. This has been demonstrated in a case study of the Eurofighter aircraft.

5.2 Robustness to parameter tolerances

Robustness to system parameter tolerances, so called structured parameter uncertainties is achieved by applying a common controller to a set of worst-case parameters system models out of the parameter tolerance band $p \in \mathrm{P}$.

The worst-case parameter model approach can be formalised as follows. Assume the design problem is completely determined by means of criteria c_i together with demands d_i, the tuners T and parameters p known to lie within the parameter range P. Then the robust design problem is equivalent to an infinite min-max problem:

$$\min_{T} \max_{i, p \in \mathrm{P}} \{c_i(T, p) / d_i\}$$

The multi-model approach now consists in a finite discretisation of this infinite problem to get an ordinary multi-objective optimisation problem:

$$\min_{T} \max_{i,j} \{c_i(T, p_j) / d_i\}, \quad p_j \in \mathrm{P}.$$

To find appropriate multi-model parameter sets p_j, one can proceed as follows [2]:

1. Examine the system properties for $T = T^0$ (e.g. the open-loop behaviour) with respect to parameter variations. This can be done by maximising quality functions over $p \in \mathrm{P}$.
2. Select nominal parameters and some parameter combinations for extreme "worst-case" system behaviour, e.g., the slowest and fastest dynamic response.

3. Perform a multi-model/multi-criteria tuning with the selected parameter combinations.
4. Perform a parametric assessment of the controlled system with optimised tuners to detect possible deficiencies within the parameter set $p \in P$. If severe deficiencies are detected, add a model with this particular parameter set to the multi-model set and re-tune the augmented synthesis problem.

This method has been successfully applied [21] in the GARTEUR RCAM-Robust Flight Control Design Challenge with large parameter tolerances in mass, center of gravity, and control implementation time delay. The resulting flight control law turned out to be the most robust one [15] of all design entries [30] if robustness is judged by the structured singular value μ.

5.3 Robust gain scheduling

Gain scheduling is common practice in flight control. The multi-model approach can help to reduce the number of controllers to be designed for gain scheduling. One way is to combine the multi-model approach of section 5.1 and 5.2:

If control scheduling is to be made with respect to, e.g., air speed, tune a set of controllers for different speed, while all other operating conditions like height, mass, center of gravity, and parameter tolerances in aerodynamic derivatives are robustly covered by the multi-model approach as described above.

Another scheduling approach is similar to [31], where the controller K is set up as a function of a scheduling parameter δ like

$$K = K_0 + \delta K_1 + \delta^2 K_2 .$$

The nominal controller is given for $\delta = 0$. Again the multi-model approach is applied to tune this controller for multiple values of δ covering the range of variation.

6 Conclusions and Outlook

This paper describes quality function deployment and its use in multi-objective control synthesis tuning, with application to flight control design. The main feature of this methodology is that the various kinds of design objectives can be taken into account in their most natural form and design alternatives can be assessed most visibly with respect to given requirements. Multi-objective synthesis tuning by min-max parameter optimisation allows interactive compromising in the set of what can be best-possibly achieved with a chosen control law structure.

The same kind of quality functions can be used for parametric robustness assessment to find deficiencies in stability and performance. By applying quality functions in a worst-case sense the same optimisation tools as for synthesis tuning can be used for detecting hidden design deficiencies.

The aim of the multi-objective tuning approach is to achieve solutions that satisfy all requirements concurrently. The achieved solutions are compromises between competing requirements. Design-tuning ends when the optimiser finds a satisfactory solution with agreed upon trade-offs. Local parameter optimisation techniques only find local pareto-optima. To find a global optimum optimisation of non-convex functions has to be applied. Guided random search like response surface techniques or evolutionary genetic algorithms have to be investigated to their usability for flight control synthesis tuning.

The fact that low-complexity controller structures can be chosen as basis for synthesis tuning is especially important in flight control, to avoid nowadays 'add-on' control systems complexity. The example of the ATD case study shows how optimal tuning can disclose a simple control law structure (with 3 tuning parameters) to be insufficient whereas a dynamically augmented structure (with 5 tuning parameters) can be tuned to satisfy all requirements. Hence multi-objective, multi-parameter optimisation may provide the quantitative clues for a design iteration in control law structure, as well.

Multi-objective synthesis tuning, for good reasons, is kept an iterative design technique, since in practice no single, ideal 'optimal' solution exists. Rather there is an infinite set of pareto-optimal compromise solutions. Systematic techniques to support the designer in finding these solutions and to support him in deciding to prefer between different design outcomes are described. Simultaneous visualisation of linked information in different displays which should be accessible via different assignment principles are very much assistant for the design process itself.

7 References

[1] M. Avriel. Nonlinear Programming: Analysis and Methods. Prentice-Hall Inc., Series in Automatic Computation, 1976.

[2] J. Bals, W. Fichter, and M. Surauer. Optimisation of magnetic and angular-momentum control for low earth orbit satellites. 3rd ESA International Conference on Spacecraft Guidance, Navigation and Control Systems, ESTEC, Noordwijk, NL, 1996.

[3] W. Bihrle. A Handling Qualities Theory for Precise Flight Path Control. RTD Technical Report Air Force Flight Dynamics Laboratory, 3, 1966.

[4] H. Duda. Flight control system design considering rate saturation. Aerospace Science and Technology, No. 4, pp. 265-275, 1998.

[5] G. Duus and H. Duda. HQML: Ein Programmpaket zur Flugeigenschaftsanalyse unter MATLAB. DLR report, IB 111-97/22, Institut für Flugmechanik Braunschweig, 1997.

[6] L. F. Faleiro and A. A. Lambregts. Analysis and Tuning of a 'Total Energy Control System' Control Law using Eigenstructure Assignment. This issue, 1999.

[7] C. Fielding and R. Luckner. The Industrial View. In [30], pp. 567-602, 1997.

[8] R. Finsterwalder. Algorithm Animation of Computational Chains. Beiträge zur graphischen Datenverarbeitung, Springer, 1992.

[9] R. Fletcher. Practical Methods of Optimization. Wiley, 1987.

[10] F. W. Gembicki and Y. Y. Haimes. Approach to Performance and Parameter Sensitivity Multiobjective Optimization: The Goal Attainment Method. IEEE Transaction on Automatic Control, AC-20, pp. 769-771, 1975.

[11] D. E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.

[12] G. Grübel and H.-D. Joos. Performance Robust Design via a Multi-Criteria/Multi-Model Approach – A Flight Control Case Study. Workshop on "Modelling, Robustness and Sensitivity Reduction in Control Systems, R.F. Curtain, NATO ASI Series F, Springer Verlag, vol. 34, pp.173-195, 1986.

[13] G. Grübel, J. Bals, R. Finsterwalder, H.-D. Joos and M. Otter. The ANDECS Framework for Computational System Dynamics Integration, Case Study: Applied Multidisciplinary

Dynamics Design Experimenting. Proceedings IFAC Conference on Integrated Systems Engineering, pp. 89 – 118, Baden-Baden, 1994.
[14] G. Grübel, J. Bals and H.-D. Joos. A CACSD Environment for Controller Tuning. Journal a, Vol. 38, No. 3, pp. 30-33, 1997.
[15] G. Grübel. An Other View of the Design Challenge Achievements. In [30], pp. 603-609, 1997.
[16] R. T. Haftka and J. Sobieszczanski-Sobieski. Multidisciplinary Aerospace Design Optimization: Survey of Recent Developements. AIAA, No. 96-0711, 1996.
[17] R. Hooke and T. A. Jeeves. "Direct Search" Solution of Numerical and Statistical Problems. J. ACM 8, pp. 212-229, 1961.
[18] H.-D. Joos, M. Schlothane and G. Grübel. Multi-Objective Design of Controllers with Fuzzy Logic Using the Control Engineering Computation Environment ANDECS. IEEE/IFAC Tucson, pp.75-82, 1994.
[19] A. Inselberg. The plane with parallel coordinates. In The Visual Computer, pp 69-91, 1985.
[20] H.-D. Joos. Informationstechnische Behandlung des mehrzieligen regelungstechnischen Entwurfs. Fortschrittberichte VDI, Reihe 20, Nr. 90, VDI Verlag, Düsseldorf, 1993.
[21] H.-D. Joos. Multi-Objective Parameter Synthesis (MOPS). In [30], pp. 199-217, 1997.
[22] K. H. Kienitz. Controller Design Using Fuzzy Logic – A Case Study. Automatica, Vol. 20, No. 2, pp. 549-554, 1993.
[23]K. H. Kienitz and G. Grübel. LQG/LTR Robust Flight Control Design with Enforced Controller Bandwidth Restriction. Proceedings of the AIAA-98-4435, 1998.
[24] K. H. Kienitz and G. Grübel. Multiobjective Target Feedback Loop / Loop Transfer Recovery (TFL/LTR) design with application to aircraft lateral control in high incidence regime. Aerospace Science and Technology, 1998.
[25] D. Kraft. A Software Package for Sequential Quadratic Programming. DFVLR-FB 88-28, 1988.
[26] G. Kreisselmeier and R. Steinhauser. Application of Vector Performance Optimization to a Robust Control Loop Design for a Fighter Aircraft. Intern. J. Control, vol. 37, No. 2, pp. 251-284, 1983.
[27] G. Kreisselmeier and R. Steinhauser. Systematic Control Design by Optimising a vector performance index. IFAC Symposium on Computer Aided Design of Control Systems, Zürich, pp. 113-117, Pergamon Press, 1979.
[28] P. Lambrechts, S. Benani, G. Looye and D. Moormann. The RCAM Design Challenge Problem Description. In [30], pp. 149-169, 1997.
[29] A. G. J. MacFarlane, G. Grübel and J. Ackermann. Future Design Environments for Control Engineering. Automatica, Vol. 25, pp. 165-176, 1989.
[30] J.F.Magni, S.Bennani and J.C. Terlouw. Robust flight control: a design challenge. Lecture notes in control and information sciences 224, Springer Verlag, 1997.
[31] J.-F. Magni. Multimodel Eigenstructure Assignment in Flight-Control Design. Aerospace Science and Technology, 3(2), 1999.
[32] D. Moormann, P. J. Mostermann, G. Looye. Object Oriented Computational Model Building of Aircraft Flight Dynamics and Systems. Aerospace Science and Technology, 3(2), 1999.
[33] E. Muir. The HIRM Design Challenge Problem Description. In [30], pp. 449-443, 1997.
[34] W. Murray and M. L. Overton. A projected Langragian algorithm for nonlinear minmax optimization. SIAM J.Sci.Stat.Comp. 1, pp 345-370, 1980.
[35] W. T. Nye, A. L. Tits. An Application-oriented, optimisation-based methodology for interactive design of engineering systems. Int. J. Control, Vol. 43, No. 6, pp 169-1721, 1986.

[36] M. B. Tischler, J. D. Colbourne, M. R. Morel, D. J. Biezad, W. S. Levine and V. Moldoveanu. CONDUIT – A New Multidisciplinary Integration Environment for Flight Control Development. AIAA-97-3773, pp. 1759-1781, 1997.
[37] H. N. Tobie, E. M. Elliot and L. G. Malcom. A new longitudinal handling qualities criterion. Proc. Of the National Aerospace Electronics Conference, pp. 93-99, Dayton, 1966.
[38] Y. Sawaragi, H. Nakayama and T. Tanino. Theory of Multiobjective Optimization. Academic Press Inc., Mathematics in Science and Engineering, Volume 176, Orlando Florida, 1985.
[39] K. Schittkowski. Nonlinear Programming Codes. Springer, Berlin, 1980.
[40] H. Wünnenberg (Ed.). Handling qualities of unstable augmented aircraft. AGARD-AR-279, 1991.
[41] J. L. Zhou, A. L. Tits, and C. T. Lawrence. User's Guide for FSQP Version 3.7: A FORTRAN Code for Solving Constrained Nonlinear (Minmax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints. University of Maryland, Systems Research Center, TR-92-107r2, 1992.