

SRC TR 85-25

**Interaction, Specification
Refinement, and Tradeoff
Exploration in Optimization-Based
Design of Engineering Systems**

by

A.L. Tits and Z. Ma

INTERACTION, SPECIFICATION REFINEMENT, AND TRADEOFF EXPLORATION IN OPTIMIZATION-BASED DESIGN OF ENGINEERING SYSTEMS

A. L. Tits and Z. Ma

Electrical Engineering Department and Systems Research Center, University of Maryland College Park, MD 20742, USA

Abstract The classical nonlinear programming problem formulation is often too rigid to adequately describe an engineering design problem. Recently a design methodology was developed, based on a problem formulation featuring three types of specifications: hard constraints, soft constraints and objectives. The corresponding mathematical problem is a sequence of constrained 'minimax' optimization problems. Such minimax problems highlight the competition between various specifications and facilitate tradeoff exploration. Accordingly, the new methodology emphasizes designer's intuition and man-machine interaction.

In this paper, we demonstrate the crucial role of interaction in an engineering design context. As meaningful interaction is greatly facilitated if suitable sensitivity information is available, we then investigate ways of computing such information and of graphically conveying it to the user.

Keywords Computer-aided system design; man-machine systems; interactive computation; optimization-based design; multiobjective optimization; nonlinear programming.

BACKGROUND AN APPLICATION-ORIENTED DESIGN METHODOLOGY

Optimization techniques have been applied successfully to numerous design problems in various branches of engineering such as integrated circuits (see, e.g., work by Brayton, Hachtel and Sangiovanni-Vincentelli (1981)), control systems (see, e.g., work by Davison and Ferguson (1982) and by Gustafson and Desoer (1983)), earthquake resistant structures (see, e.g., work by Bhatti, Pister and Polak (1979)), or digital filters (see, e.g., work by Deczky (1972)). However, in many cases, the mathematical problem solved by the optimization algorithm may be remote from the real-world problem the designer is facing. This is due to the rigidity of the classical nonlinear programming problem which can be stated as

$$\min\{ f(x) \mid g(x) \leq 0 \}, \quad (1)$$

where $f(x)$ is the cost or objective function to be minimized and $g(x)$ represents several inequality constraints and where x is the vector of design parameters. While this formulation does encompass the general idea of optimizing some design objective while meeting various design specifications, it fails to take into account several important characteristics of a large class of design problems.

First, it is rarely the case that a single objective has to be optimized. In most applications, various objectives compete against each other and a compromise has to be reached. Amalgamating several objectives into a single cost function has the disadvantage, particularly acute in an interactive environment, of hiding the physical significance of these objectives.

Second, the above mathematical formulation (1) does not accept any violations of the constraints g . In design applications, constraints specifications are often relatively flexible and thus better put in words than in numbers. Hence, moderate violation of a constraint should be acceptable by the optimization algorithm; often, this will permit it to achieve a better value of the objective function(s). Notice that the constraint formulation

$$g(x) \leq 0$$

is particularly inadequate since it gives no way of estimating the importance of a given constraint violation.

Third and more generally, formulation (1) expresses only partially the knowledge a designer has about his problem. Some of this knowledge, built on experience and physical intuition,

is often impossible to express numerically. Also it is often difficult, if not impossible, for a designer to provide a priori precise and definite design specifications, since engineering specifications are often revised in view of the design obtained.

Nye and Tits (1986) proposed a methodology that gets around some of the difficulties just mentioned. This methodology emphasizes designer *intuition* and man-machine *interaction*. A fundamental idea in this methodology is the partition of the various specifications attached to engineering design problems into three categories: hard constraints, soft constraints, and objectives. An *objective* is a quantity that the designer would like to see as small (large) as possible; example: the gain of an operational amplifier. A *soft constraint* is a quantity that the designer would like to see smaller (larger) than some threshold, or, if this cannot be achieved, as close as possible to this threshold; example: the stability margin of a control system. A *hard constraint* is a quantity that the designer requires to be below (above) some threshold, any violation being unacceptable; example: a resistance value must be non-negative.

Obviously the various specifications or, at least, the various objectives and soft constraints are competing against each other. It is then necessary to be able to meaningfully compare the values of various specifications for a given x -vector value, i.e., to define the *normalized* value of a specification. In the approach proposed by Nye and Tits (1986), the driving idea is to follow as closely as possible the viewpoint of the designer by involving quantities, expressed in the physical units of his particular application, that call on his intuitive knowledge of the problem, thus facilitating his task of expressing the relative importance of the various specifications. This leads to the concept of *good* and *bad* value that the designer has to specify for each of the design specifications, based on the idea he has of a good or bad design. These good and bad values should be chosen according to the following *uniform satisfaction/dissatisfaction rule* for choosing the good and bad values: having all of the various objectives and soft constraints achieve their corresponding good values should provide the same level of 'satisfaction' to the designer for each, while achieving the bad values should provide the same level of 'dissatisfaction'. If $f_g^j(x)$ is the raw (unnormalized) value of f^j at x and if f_g^j and f_b^j are the corresponding good and bad values, the normalized value $f_n^j(x)$ is defined as

$$f_n^j(x) \triangleq \frac{f_g^j(x) - f_g^j}{f_b^j - f_g^j}. \quad (2)$$

To take into account the above considerations, it is then suggested by Nye and Tits (1988) to formulate the design problem as a sequence of three (or fewer) *constrained minimax* problems, i.e., problems of the form

$$\min_x \{ \max_{j \in m} f^j(x) \mid g^j(x) \leq 0 \ \forall j \in l \}, \quad (3)$$

where

$$m \triangleq \{ 1, 2, \dots, m \}$$

$$l \triangleq \{ 1, 2, \dots, l \}$$

and where all function values are normalized as shown above. The first problem of the sequence corresponds to the case when no initial x is known that satisfies all the hard constraints. In this case, the f^j 's are the hard constraints and $l=0$ (no constraints). The idea is to try to satisfy the hard constraints as soon as possible, since any design violating these would be of little value. As soon as a design parameter x is obtained that satisfies all the hard constraints, the second problem in the sequence is activated. This second problem corresponds to the case when there is a design parameter vector available which satisfies all hard constraints but some of the soft constraints or objectives do not achieve their good value. In this case, the f^j 's are the values of the objectives and soft constraints and the g^j 's are the hard constraints. Here the idea is to improve the "worst" achievement among objectives and soft constraints, which are treated in an identical manner. If and when a design parameter is obtained that achieves the good values of all the soft constraints and objectives, while still satisfying the hard constraints, the third problem in the sequence is activated. This third problem corresponds to the case when there is a design parameter vector available at which all hard constraints are satisfied and soft constraints and objectives at least reach their good value. In this case the f^j 's are the objectives and the g^j 's are the (hard and soft) constraints. The idea here is to improve the objective values only, while keeping the constraints satisfied.

In the sequel, we will focus on a single phase of this possibly 3-phase design process. This could be either the second or the third phase, but we will think of it as the *final* phase of the particular design considered.

NEED FOR INTERACTION

It has often been argued that any manual intervention of the user in an optimization process is unnecessary if the problem is correctly posed and the optimization algorithm is sound. In this section, we show that, in an engineering design environment, interaction between user and optimization process is generally crucial.

At this point, the choice for a minimax formulation may seem arbitrary, i.e., it may not be clear that solving problem (3) can result in a design that the user will consider 'best'. Since the original problem is essentially a constrained multiobjective problem, let us consider the well known concept of *Pareto optimality*. For the time being, assume that all the objective functions are to be *minimized*.

Definition. A point $x^* \in \mathbb{R}^n$ is a *local Pareto point* with respect to objective functions f^1, f^2, \dots, f^m and constraints g^1, g^2, \dots, g^l if, for $j=1, \dots, l$, $g^j(x^*) \leq 0$, and there exists a positive ρ such that, for any $x \in B(x^*, \rho)$,

$$f^j(x) < f^j(x^*) \text{ for some } j \in m$$

implies

$$f^k(x) > f^k(x^*) \text{ for some } k \in m$$

or

$$g^k(x) > 0 \text{ for some } k \in l.$$

#

Clearly, any point $x \in \mathbb{R}^n$ which is not a Pareto point cannot be considered as optimal.

¹ Since we assume that the designer knows exactly what he means by optimal, these curves can be thought of as precisely defined.

The following simple Fact (straightforward extension of Theorem 13.2 of Brayton and Spence (1980)) relates the concept of Pareto point to constrained minimax optimization.

Fact. Suppose x^* is a local Pareto point with respect to objective functions f^1, \dots, f^m and constraints g^1, \dots, g^l , and suppose that $f^j(x^*) > 0$ for all $j \in m$. Then there exist positive numbers w_1, \dots, w_m such that x^* is a local solution for the constrained minimax problem

$$\min_x \{ \max_{j \in m} w_j f^j(x) \mid g^j(x) \leq 0 \}. \quad (4)$$

#

In the following corollary, we do not assume any more that all objectives are to be minimized (i.e., possibly some of them are to be *maximized*), and that all optimal values are positive.

Corollary. Suppose x^* is a local Pareto point with respect to objective function f^1, \dots, f^m and constraints g^1, \dots, g^m . Then there exists (nonunique) scalars $\alpha^1, \dots, \alpha_m$ and β_1, \dots, β_m such that x^* is a local solution for the constrained minimax problem

$$\min_x \max_{j \in m} \left\{ \frac{f^j(x) - \alpha_j}{\beta_j} \mid g^j(x) \leq 0 \right\}. \quad (5)$$

#

Thus, no matter which Pareto point the designer feels is optimal, there exists a set of good and bad values such that, with the normalization (2), problem (3) will have x^* as a local solution. The question, however, is how to determine a suitable set of good and bad values. To the designer, the design problem may not be equivalent to a constrained minimax problem in any obvious way. Thus the originally chosen good and bad values will likely not yield the desired Pareto point. The designer may then wish to *interactively modify* these good and bad values, guided by their application-related interpretation, so as to 'steer' the minimax solution to the desired solution. This in fact amounts to tradeoff exploration.

Suppose now that the designer is able to think of his design problem as one of minimizing the highest degree of dissatisfaction (i.e., maximizing the lowest degree of satisfaction) among those associated with the various specifications. Thus, suppose he is able to give, at the outset, precise and definite good and bad values (according to the uniform satisfaction/dissatisfaction rule). Thus, if all the good values are exactly achieved at the solution of the constrained minimax problem, the designer will be entirely satisfied. Consider Fig. 1 where designer satisfaction is plotted versus normalized specification value for each of the design objectives¹. After a minimax optimization has been performed, a typical situation would be one where several normalized specifications (the *competing* specifications) have achieved a value such as p on Fig. 1, while lower values have been achieved for others.

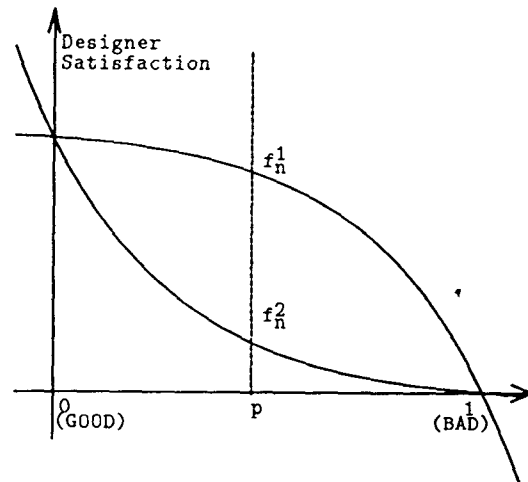


Fig. 1. Designer Satisfaction Versus Normalized Specification Values.

Obviously, in view of the likely *nonlinearity* of the various curves, the degrees of satisfaction corresponding to the various competing objectives will probably not be equal at this point. In other words, the resulting design will probably still not be considered optimal by the designer. A typical action would then be to *interactively* modify the good values to make them more realistic.

The methodology proposed by Nye and Tits (1986) does rely on interaction between designer and optimization process. Suitable information on the current performance of the design is communicated graphically to the designer whenever requested. The graphical display, which has been implemented in the DELIGHT (Nye and co-workers, 1981, Nye, 1983) system, is referred to as *Pcomb* (for performance comb, from its shape). A glance at this display allows the designer to compare achievements of the current design with respect to his various specifications. As a solution is approached, the designer can decide to revise the emphasis he had placed on some of the currently competing specifications, through suitable modification of corresponding good and bad values. The *Pcomb* display has proven very effective in academic as well as industrial environment (Nye and co-workers, 1983; Fan and co-workers, 1985a). However, it has appeared that a desirable piece of information was lacking from this display: *what change can be expected in the tradeoff solution if the emphasis of the various specifications is modified, i.e. if some of the good and bad values are altered.* The availability of such sensitivity information would help the designer in exploring tradeoffs in a more economical way, by discarding unpromising alternatives without having to perform costly simulations.

In the sequel, we show how such sensitivity information can be computed and efficiently displayed in the framework of the methodology of Nye and Tits (1986)

ESTIMATING THE SOLUTION

Suppose that, after running an optimization process for a few iterations, one wishes to explore tradeoff designs corresponding to various settings of the good and bad values. Clearly, the first step is to estimate what the optimal design will be if no such changes are performed. This should be obtained at low CPU cost.

When solving an optimization problem arising from an engineering design problem, most of the CPU time is typically spent in system simulations. Thus ideally one would like to obtain an estimate of the solution, from the current iterate, without performing any additional simulations. Second to best would be to only perform simulations that will anyway be needed for the next optimization iteration, whether or not some good or bad values are modified. The latter will be achieved provided that estimation of the solution requires function and gradient evaluations *at the current point only*. The former will be achieved if, in addition, gradients are already available. Such is practically the case when the system under consideration is described by a set of differential or difference equations, since for such systems, sensitivities are obtained as almost free by-products of function evaluations².

An obvious candidate formula for obtaining an estimate of the solution to a nonlinear programming problem is the sequential quadratic programming (SQP) iteration. In case of the constrained minimax problem (3), if the current iterate is x , this iteration can be written as (Han, 1981)

$$\begin{aligned} \min_{(d,\delta)} \delta + \frac{1}{2} \langle d, Bd \rangle \\ \text{s.t. } f^j(x) + \langle \nabla f^j(x), d \rangle \leq \delta \text{ for all } j \in m \quad (6) \\ g^j(x) + \langle \nabla g^j(x), d \rangle \leq 0 \text{ for all } j \in l, \end{aligned}$$

where B is some positive definite estimate of the Hessian of

² Thus, in such problems, most of the CPU time is spent in the stepsize computation

the Lagrangian. Let (d_0, δ_0) solve (6). Then the estimated solution vector for (3) is

$$x_0 = x + d_0.$$

The corresponding Kuhn-Tucker multipliers are estimate of the Kuhn-Tucker multipliers at the solution of (3). δ_0 is an estimate of the optimal value of (3). Note that, equivalently, (6) can be written as

$$\begin{aligned} \min_d \frac{1}{2} \langle d, Bd \rangle + \max_{j \in m} \{ f^j(x) + \langle \nabla f^j(x), d \rangle \} \quad (7) \\ \text{s.t. } g^j(x) + \langle \nabla g^j(x), d \rangle \leq 0 \text{ for all } j \in l. \end{aligned}$$

It is well-known that if B is a good enough approximation to the Hessian of the Lagrangian at the solution of (3) the SQP iteration yields a *superlinear* local rate of convergence. Thus it does make sense to take x_0 as the desired estimate.

The following easily proven proposition gives a simple geometric interpretation for the solution of (7).

Proposition. For any $v \in \mathbb{R}^n$, let us define

$$\|v\|_B \triangleq \langle v, Bv \rangle^{\frac{1}{2}}.$$

If d_0 solves (7) and $\|d_0\|_B = \rho$, then d_0 solves

$$\min_{\|d\|_B \leq \rho} \max_{j \in m} (f^j(x) + \langle \nabla f^j(x), d \rangle)$$

$$\text{s.t. } g^j(x) + \langle \nabla g^j(x), d \rangle \leq 0 \text{ for all } j \in l.$$

Thus x_0 solves a first order approximation to problem (3) constrained to the ball of radius ρ corresponding to norm $\|\cdot\|_B$ #

TRADEOFF EXPLORATION

In solving the quadratic program (6), one has identified objectives and constraints likely to be active at the solution of (3)—they are the objectives and constraints that are active for the quadratic program. We will denote these index sets respectively by $J_f(x)$ and $J_g(x)$, if the current point is x . When investigating the sensitivity of the optimal cost of (3) to variations in good and bad values, an attractive option is to assume that this active set will remain invariant. There are several reasons for doing so. First, in most cases, this assumption will hold true, if changes in good and bad values are small. Second, without this assumption, it would be necessary to solve a quadratic program similar to (6) for each of the sets of good and bad values one is interested in. This may well be prohibitive in an interactive environment. If one makes use of the assumption, one merely has to solve linear systems.

Thus, suppose that the good value f_g^k of f^k , $k \in J_f(x^*)$, is modified to \bar{f}_g^k , so that $f^k(x)$ becomes

$$\bar{f}^k(x) = \frac{f_r^k(x) - \bar{f}_g^k}{f_b^k - \bar{f}_g^k}.$$

Its gradient becomes

$$\nabla \bar{f}^k(x) = \beta \nabla f^k(x)$$

with

$$\beta = \frac{f_b^k - f_g^k}{f_b^k - \bar{f}_g^k}.$$

(If instead the bad value f_b^k is modified to \bar{f}_b^k , one has

$$\bar{f}^k(x) = \frac{f_r^k(x) - f_g^k}{\bar{f}_b^k - f_g^k}.$$

and the expression for $\nabla \bar{f}^k(x)$ is still valid, with β given by

$$\beta = \frac{f_b^k - f_g^k}{\bar{f}_b^k - f_g^k}.$$

An estimate of the new solution will then be obtained by solving the quadratic program

$$\begin{aligned} & \min_{(d, \delta)} \delta + \frac{1}{2} \langle d, Bd \rangle \\ \text{s.t. } & \bar{f}^k(x) + \beta \langle \nabla f^k(x), d \rangle \leq \delta \\ & f^j(x) + \langle \nabla f^j(x), d \rangle \leq \delta \text{ for all } j \neq k \\ & g^j(x) + \langle \nabla g^j(x), d \rangle \leq 0 \text{ for all } j \in I. \end{aligned}$$

If we do assume that the active sets remain invariant, we obtain

$$\begin{aligned} & \min_{(d, \delta)} \delta + \frac{1}{2} \langle d, Bd \rangle \\ \text{s.t. } & \bar{f}^k(x) + \beta \langle \nabla f^k(x), d \rangle = \delta \\ & f^j(x) + \langle \nabla f^j(x), d \rangle = \delta \text{ for all } j \in J_f(x), j \neq k \\ & g^j(x) + \langle \nabla g^j(x), d \rangle = 0 \text{ for all } j \in J_g(x). \end{aligned}$$

For simplicity of notation, we will now assume that all objectives and constraints are active.

The latter quadratic program can be solved explicitly. Writing the first order necessary conditions of optimality gives the linear system

$$B\bar{d} + \bar{N}\bar{\lambda} + M\bar{\mu} = 0$$

$$\sum_j \bar{\lambda}^j = 1$$

$$\bar{N}^T \bar{d} \begin{bmatrix} \delta \\ \bar{\lambda} \\ \bar{\mu} \\ \delta \end{bmatrix} = -\bar{f}(x_k)$$

$$M^T \bar{d} = -g(x_k),$$

where the columns of \bar{N} and M are the gradients of the active objectives and constraints (with $\beta \nabla \bar{f}^k(x^*)$ as the k th column of \bar{N}) and where the bars indicate that we are now considering the modified problem. Equivalently, the linear system above can be written as

$$\begin{bmatrix} B & \bar{N} & M & 0 \\ \bar{N}^T & 0 & 0 & -e \\ M^T & 0 & 0 & 0 \\ 0 & -e^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{d} \\ \bar{\lambda} \\ \bar{\mu} \\ \delta \end{bmatrix} = \begin{bmatrix} 0 \\ -\bar{f}(x) \\ -g(x) \\ -1 \end{bmatrix}, \quad (8)$$

where e is the m -vector with all entries equal to 1. This system can be solved for $\bar{\delta}$, the estimated minimax value at the solution of the modified problem. Although this requires solving a linear system of equations for each selected set of good and bad values, computation can be kept within reasonable bounds if one only considers instances where a single good or bad value is modified. This is because, in such cases, the matrix in (8) differs from the matrix in the nominal system

$$\begin{bmatrix} B & N & M & 0 \\ N^T & 0 & 0 & -e \\ M^T & 0 & 0 & 0 \\ 0 & -e^T & 0 & 0 \end{bmatrix} \begin{bmatrix} d_0 \\ \lambda_0 \\ \mu_0 \\ \delta_0 \end{bmatrix} = \begin{bmatrix} 0 \\ -f(x) \\ -g(x) \\ -1 \end{bmatrix} \quad (9)$$

only by a matrix of rank 2. The Cholesky factorization for (8) can then be obtained from that for (9) at moderate cost (see, e.g., work by Gill and co-workers (1974)).

If $(\bar{f}_g^k - f_g^k)$ is small, (8) can be simplified. Denoting by $A(\beta)$ the matrix in the left hand side of (8), subtracting (9) from (8), and expanding to the first order in $1-\beta$, one gets, since $A(\beta) - A(1)$ is linear in $1-\beta$,

$$A(1) \begin{bmatrix} \bar{d} - d_0 \\ \bar{\lambda} - \lambda_0 \\ \bar{\mu} - \mu_0 \\ \bar{\delta} - \delta_0 \end{bmatrix} + (A(\beta) - A(1)) \begin{bmatrix} d_0 \\ \lambda_0 \\ \mu_0 \\ \delta_0 \end{bmatrix} = \begin{bmatrix} 0 \\ -(\bar{f}(x) - f(x)) \\ 0 \\ 0 \end{bmatrix} \quad (10)$$

Note that, since $A(1)$ already appears in system (9), system (10) can be readily solved, without the need to perform any extra factorization.

Further simplification can be achieved if x is close to the solution of the original problem. Let us first rewrite the second and third block-equations in (10) as

$$\langle \nabla f^k(x), \bar{d} - d_0 \rangle + \bar{\delta} - \delta_0 = (1-\beta) \nabla f^k(x)^T d_0 - (\bar{f}^k(x) - f^k(x))$$

$$\langle \nabla f^j(x), \bar{d} - d_0 \rangle + \bar{\delta} - \delta_0 = 0 \quad j \neq k$$

$$\langle \nabla g^j(x), \bar{d} - d_0 \rangle = 0$$

Under our last assumption, d_0 is small. Since $1-\beta$ is also small, the first term in the right hand side of the first equation can be neglected. Since, from (9),

$$Bd_0 + \sum_j \lambda_0^j \nabla f^j(x) + \sum_j \mu_0^j \nabla g^j(x) = 0,$$

(10) yields

$$\sum_j \lambda_0^j (\delta - \delta_0) = Bd_0 + \lambda_0^k (\bar{f}^k(x) - f^k(x)),$$

i.e., in view of the last equation in (9),

$$\bar{\delta} - \delta_0 = Bd_0 + \lambda_0^k (\bar{f}^k(x) - f^k(x)) \quad (11)$$

Thus, as could be expected, the sensitivity of the optimal cost to changes in good or bad values is directly related to certain multipliers.

Relations (8), (10) and (11) give three different ways of defining $\delta - \delta_0$, the estimated variation in the minimax solution of (3) if a good or bad value is modified. In the next section, we propose a way of displaying this information to the designer.

THE ECOMB GRAPHICAL DISPLAY

Although similar in appearance to the Pcomb display (alluded to at the end of Section 2; see the work by Nye and Tits (1986) for more detail), the Ecomb (or estimation comb) has an entirely different interpretation. Beside displaying the estimated values of the various specifications at the solution of the original problem, it conveys information about the effect modifications of the various good values may have on the overall minimax solution.³ While we are convinced that, after a short learning period, the designer will feel comfortable in interpreting the Ecomb display, we admit that its complexity may throw of the first time reader. Thus a detailed explanation is in order. As a support to this explanation, consider the Ecomb display of Fig. 2, obtained while designing a flight controller for an F14 aircraft (see the work by Fan and co-workers (1985a, 1985b) for details on this problem).

For each specification, identified with a name, the *original* good and bad values are printed under the 'GOOD' and 'BAD' headings and, as in the Pcomb, a horizontal line indicates by the position of its tip the corresponding current normalized value. The intersection of this horizontal line with the vertical 'G' (for good) line marks the normalized value of 0 (corresponding to the good value) and the intersection with the vertical 'B' (for bad) line marks the normalized value of 1 (corresponding to the bad value). The horizontal line at the top of the display, 'Estimated Minimax', indicates the estimated optimal value of (3) for the current set of good and bad values, computed as described in an earlier section. Under the heading 'ESTIMATE', the Ecomb displays, for each specification, the estimated raw (unnormalized) value

³ Similar information could be displayed about the effect of changes in bad values. We feel however that the designer will generally favor adjusting good values.

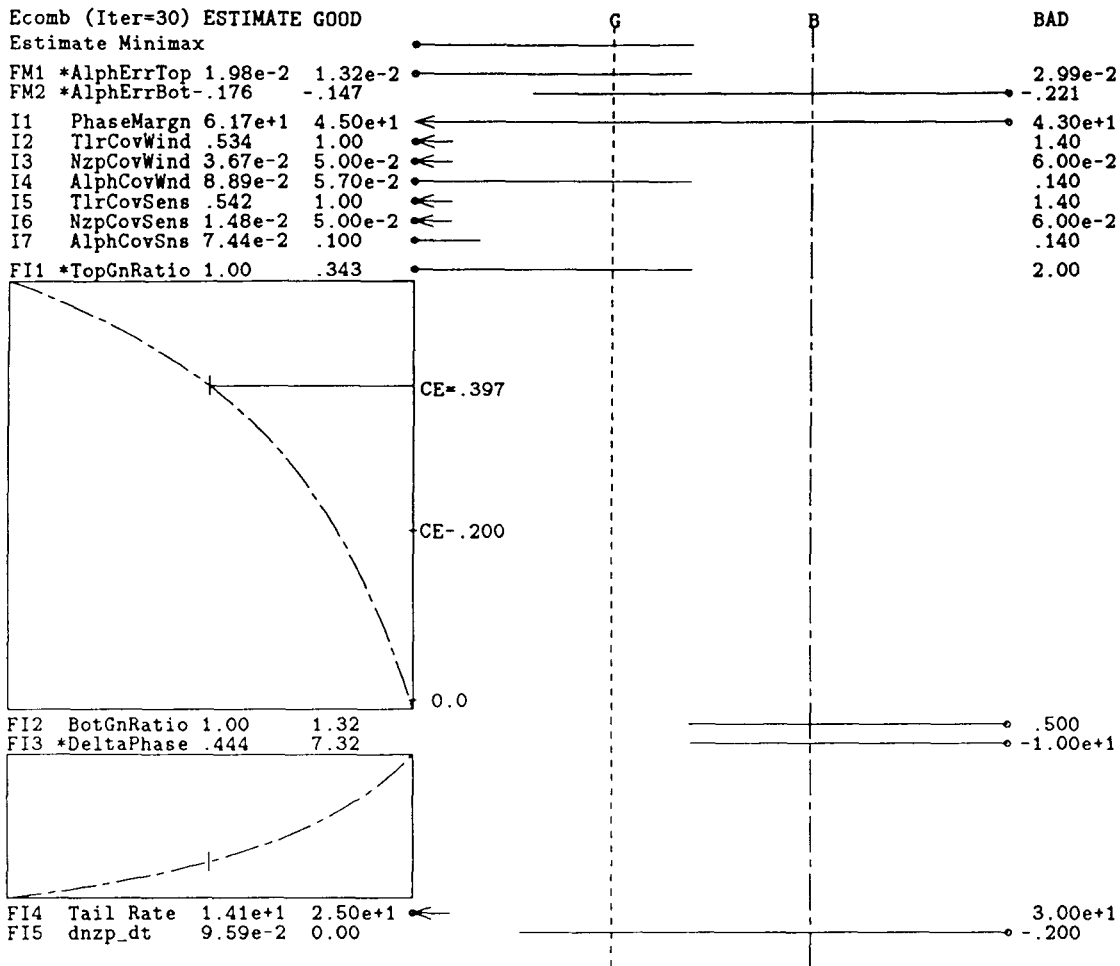


Fig. 2. The Ecomb Graphical Display.

that might be reached by that specification under the current set of good and bad values.

The estimated effect of *modifying a given good value* by various amounts, discussed in the previous section, is indicated by a set of curves displayed just below the name of the corresponding specification. Each one of these plots indicates in ordinate the estimated (normalized) optimal value of (3) expected to be reached if the good value of the specification under consideration is set to the corresponding value in abscissa—assuming that no other specification has its good or bad value modified. The current good value is exactly half way between the extremities of the horizontal axis. The horizontal scale is linear and the rightmost point corresponds to an increase of the good value by a quarter of the difference between the current good value and the current bad value. The vertical scale is also linear, is identical on all the plots, and is indicated on the tallest of them by the value of the estimated minimax for the current set of good values (CE = current estimate) and a typical deviation with respect to this minimax value. If the value 0—corresponding to the case when the worst performing specifications exactly achieve their good values—is within range, it is displayed as well. Not all the specifications come with a corresponding plot, but only those for which a small change in good value would, according to our estimate, affect the minimax value to an extent greater than some given threshold. Finally, all the specifications for which a small change in good value would, according to our estimate, affect *at all* the minimax value are indicated by a star to the left of their name.

From the display of Fig. 2, the designer easily realizes that the overall gain can be much more substantial if he relaxes, say, the specification on the upper bound on the frequency

response (TopGnRatio) rather than, say, the specification on the corresponding lower bound (BotGnRatio).

CONCLUSION

We have demonstrated the crucial role of interaction in optimization-based design of engineering systems. As a complement to a previously introduced interactive design methodology, we have then shown how the task of the designer can be greatly eased if suitable sensitivity information is computed and displayed.

This supplemental tool is being implemented as part of the DELIGHT system. Preliminary experiments are promising.

ACKNOWLEDGMENTS

The authors wish to thank Dr. W.T. Nye and M.K.H. Fan for their substantial input. This research was supported by the National Science Foundation, grants No. ECS-82-04452 and OIR-85-00108, and by a grant from the Westinghouse Corporation.

REFERENCES

Bhatti, M.A., K.S. Pister, and E. Polak (1970). OPTDYN: A General Purpose Optimization Program for Problems With or Without Dynamic Constraints. Report No. UCB/ERC-79/16, Earthquake Engineering Research Center, University of California, Berkeley, California.

- Brayton, R.K. and R. Spence (1980). *Sensitivity and Optimization*. Elsevier Scientific Publishing Company, Amsterdam, The Netherlands.
- Brayton, R.K., G.D. Hachtel, and A.L. Sangiovanni-Vincentelli (1981). A Survey of Optimization Techniques for Integrated Circuit Design. *Proc. IEEE*, 69, 1334-1362.
- Davison, E.J. and I.J. Ferguson (1982). The Design of Controllers for the Multivariable Robust Servomechanism Problem using Parametric Optimization Methods. *IEEE Trans. on Automatic Control*, AC-26, 93-110.
- Deczky, A.G. (1972). Synthesis of Recursive Digital Filters Using the Minimum p-Error Criterion. *IEEE Trans. on Audio and Electroacoustics*, AU-20, 257-263.
- Fan, M.K.H., C.D. Walrath, A.L. Tits, W.T. Nye, M. Rimer, R.T. Grant, and W.S. Levine (1985a). Two Case Studies in Optimization-Based Computer-Aided Design of Control Systems. *Proc. of the IEEE Control Systems Society 2nd Symposium on Computer-Aided Control System Design (CACSD)*, Santa Barbara, California.
- Fan, M.K.H., C.D. Walrath, C. Lee, A.L. Tits, M. Rimer, R. Grant, and W.S. Levine (1985b). Design of a Flight Controller for an F14 Aircraft Using the DELIGHT.MaryLin Optimization-Based CACSD System. Technical Report TR- 85-16, Systems Research Center, University of Maryland, College Park, Maryland.
- Gill, P.E., G.H. Golub, W. Murray, and M.A. Saunders (1974). Methods for Modifying Matrix Factorizations. *Mathematics of Computation*, 28, 505-535.
- Gustafson, C.L. and C.A. Desoer (1983). Controller Design for Linear Multivariable Feedback Systems with Stable Plants, Using Optimization with Inequality Constraints. *Int. J. on Control*, 37, 881-907.
- Han, S.P. (1981). Variable Metric Methods for Minimizing a Class of Nondifferentiable Functions. *Math. Prog.*, 20, 1-13.
- Nye, W.T., E. Polak, A. Sangiovanni-Vincentelli, and A.L. Tits (1981). DELIGHT: An Optimization-Based Computer-Aided Design System. *Proceedings of the 1981 IEEE International Symposium on Circuits and Systems*, 851-855.
- Nye, W.T., A.L. Sangiovanni-Vincentelli, J.P. Spoto, and A.L. Tits (1983). DELIGHT.SPICE: An Optimization-Based System for the Design of Integrated Circuits. *Proceedings of the 1983 Custom Integrated Circuit Conference*, 233-238.
- Nye, W.T. (1983). *DELIGHT: An Interactive System for Optimization-Based Engineering Design*. Ph.D. Thesis, Department EECS, University of California, Berkeley, California.
- Nye, W.T., and A.L. Tits (1986). An Application-Oriented, Optimization-Based Methodology for Interactive Design of Engineering Systems. To appear in *International J. of Control*.