HUBER OPTIMIZATION OF CIRCUITS: A ROBUST APPROACH

J.W. Bandler, S.H. Chen, R.M. Biernacki,
L. Gao, K. Madsen and H. Yu

SOS-93-6-R

March 1993

(Revised June 1993)

HUBER OPTIMIZATION OF CIRCUITS: A ROBUST APPROACH

John W. Bandler[*], Fellow, IEEE, Shao Hua Chen[*], Member, IEEE,
Radoslaw M. Biernacki[*], Senior Member, IEEE, Li Gao[†], Kaj Madsen[¤] and Huanyu Yu[‡]

Abstract

We introduce a novel approach to "robustizing" circuit optimization using Huber functions: both two-sided and one-sided. Advantages of the Huber functions for optimization in the presence of faults, large and small measurement errors, bad starting points and statistical uncertainties are described. In this context, comparisons are made with optimization using ℓ_1, ℓ_2 and minimax objective functions. The gradients and Hessians of the Huber objective functions are formulated. We contribute a dedicated, efficient algorithm for Huber optimization and show, by comparison, that generic optimization methods are not adequate for Huber optimization. A wide range of significant applications is illustrated, including FET statistical modeling, multiplexer optimization, analog fault location and data fitting. The Huber concept, with its simplicity and far-reaching applicability, will have a profound impact on analog circuit CAD.

---

[*] J.W. Bandler, S.H. Chen and R.M. Biernacki are with Optimization Systems Associates Inc., P.O. Box 8083, Dundas, Ontario, Canada L9H 5E7, and with the Simulation Optimization Systems Research Laboratory, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

[†] L. Gao is with the Department of Mathematics and Institute of Mathematics, Peking University, Beijing 100871, China.

[¤] K. Madsen is with the Institute for Numerical Analysis, The Technical University of Denmark, DK-2800 Lyngby, Denmark.

[‡] H. Yu is with the Simulation Optimization Systems Research Laboratory, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

This work was supported in part by Optimization Systems Associates Inc. and in part by the Natural Sciences and Engineering Research Council of Canada under Grants OGP0007239, OGP0042444 and STR0117819.

I. INTRODUCTION

Engineering designers are often concerned with the robustness of numerical optimization techniques, and rightly so, knowing that engineering data is, with few exceptions, contaminated by model/measurement/statistical errors.

The classical least-squares (ℓ_2) method is well known for its vulnerability to gross errors: a few wild data points can alter the least squares solution significantly. The ℓ_1 method is robust against gross errors [1,2]. We will show, however, that when the data contains many small errors (such as statistical variations), the ℓ_1 solution can be undesirably biased toward a subset of the data points. This indicates that ℓ_1 is not suitable, in general, as a statistical estimator.

Neither the ℓ_2 nor the ℓ_1 method has flexible discriminatory power to recognize and treat differently large (catastrophic) errors and small (soft) errors. We introduce the Huber function [3-5], which appears to be a hybrid of the ℓ_1 and ℓ_2 measures. Compared with ℓ_2, the Huber solution is more robust w.r.t. large errors. Compared with ℓ_1, the Huber solution can provide a smoother, less biased estimate from data which contains many small deterministic or statistical variations. We demonstrate the benefits of this novel approach in FET statistical modeling, analog fault location and data fitting.

We extend the Huber concept by introducing a "one-sided" Huber function for large-scale optimization. For large-scale problems, systematic decomposition techniques have been proposed (e.g., [6,7]) to reduce computational time and prevent potential convergence problems. In practice, the designer often attempts, by intuition, a "preliminary" optimization with a small number of dominant variables. The full-scale optimization is performed if and when a reasonably good point is obtained.

With a reduced number of variables, the optimizer may not be able to reduce all the error functions at the same time. For instance, the specification may be violated more severely at some sample points (such as frequencies) than at the others. In such situations, the minimax method is preoccupied with the worst-case errors and therefore becomes ineffective or inefficient. We demonstrate, through microwave multiplexer optimization, that the one-sided Huber function can be more effective and efficient than minimax in overcoming a bad starting point.

We present a dedicated, efficient, gradient-based algorithm for Huber optimization and show, by comparison, that generic optimization methods, such as quasi-Newton, conjugate gradient and simplex algorithms, are not adequate when directly applied to minimizing the Huber objective functions. The gradients and Hessians of the Huber objective functions are derived and their significance is discussed.

II. THEORETICAL FORMULATION OF HUBER FUNCTIONS

The Huber optimization problem is defined as [3,4]

$$\underset{x}{\mathit{minimize}} \quad F(x) \triangleq \sum_{j=1}^{m} \rho_k(f_j(x)) \tag{1}$$

where $x = [x_1 \ x_2 \ \dots \ x_n]^T$ is the set of variables and ρ_k is the Huber function defined as

$$\rho_k(f) = \begin{cases} f^2/2 & \mathit{if} \ |f| \le k \\ k|f| - k^2/2 & \mathit{if} \ |f| > k \end{cases} \tag{2}$$

where k is a positive constant and f_j, $j = 1, 2, \dots, m$, are error functions.

The Huber function ρ_k is a hybrid of the least-squares (ℓ_2) (when $|f| \le k$) and the ℓ_1 (when $|f| > k$) functions. As illustrated in Figs. 1 and 2, the definition of ρ_k ensures a smooth transition between ℓ_2 and ℓ_1 at $|f| = k$. This means that the first derivative of ρ_k w.r.t. f is continuous.

The ℓ_1 is robust against gross errors in the data [1,2]. Since the Huber function treats errors above the threshold (i.e., $|f| > k$) in the ℓ_1 sense, it is robust against those errors, i.e., the solution is not sensitive to those errors. The choice of k defines the threshold between "large" and "small" errors. By varying k, we can alter the proportion of error functions to be treated in the ℓ_1 or ℓ_2 sense. Huber gave a look-up table [3] from which k can be determined according to the percentage of gross errors in the data. If k is set to a sufficiently large value, the optimization problem (1) becomes least squares. On the other hand, as k approaches zero, ρ_k will approach the ℓ_1 function.

Gradient and Hessian

To further our insight into the properties of the Huber formulation, we derive the gradients and Hessians of the Huber objective function as follows.

The gradient vector of the Huber objective function F w.r.t. $\boldsymbol{x}$ is given by

$$\nabla F = \sum_{j=1}^{m} v_j \, f_j' \tag{3}$$

where

$$v_j \triangleq \frac{\partial \rho_k(f_j(\boldsymbol{x}))}{\partial f_j(\boldsymbol{x})} = \begin{cases} f_j(\boldsymbol{x}) & \textit{if } |f_j(\boldsymbol{x})| \le k \\ \pm k & \textit{if } |f_j(\boldsymbol{x})| > k \end{cases} \tag{4}$$

$$f_j' \triangleq \left[\frac{\partial f_j(\boldsymbol{x})}{\partial x_1} \; \frac{\partial f_j(\boldsymbol{x})}{\partial x_2} \; \cdots \; \frac{\partial f_j(\boldsymbol{x})}{\partial x_n} \right]^T \tag{5}$$

The structure of (3) is very similar to the gradient of ℓ_2 (least squares), which is

$$\nabla F_{\ell_2} = \sum_{j=1}^{m} f_j \, f_j' \tag{6}$$

By comparing (3) with (6), we can see that v_j, namely the first derivative of ρ_k w.r.t. f_j, serves as a weighting factor in the Huber gradient. For $|f_j| \le k$, v_j is defined in (4) as f_j, which is the same as in the ℓ_2 gradient given by (6). For $|f_j| > k$, v_j is held constant at the value of f_j at the threshold. In other words, the Huber gradient can be thought of as a modified ℓ_2 gradient where the gross errors are reduced to the threshold value.

The Hessian matrix of the Huber objective function F w.r.t. $\boldsymbol{x}$ can be expressed as

$$\boldsymbol{H} = \sum_{j=1}^{m} [d_j \, f_j' \, f_j'^T + v_j \, f_j''] \tag{7}$$

where

$$d_j \triangleq \frac{\partial^2 \rho_k(f_j(\boldsymbol{x}))}{\partial f_j^2(\boldsymbol{x})} = \begin{cases} 1 & \textit{if } |f_j(\boldsymbol{x})| \le k \\ 0 & \textit{if } |f_j(\boldsymbol{x})| > k \end{cases} \tag{8}$$

$$f_j'' \triangleq \frac{\partial f_j'^T}{\partial \boldsymbol{x}} \tag{9}$$

Comparing (7) to the ℓ_2 Hessian matrix given by

$$H_{\ell_2} = \sum_{j=1}^{m} [\, f_j' \, f_j'^T + f_j \, f_j'' \,] \tag{10}$$

we can see that v_j serves as a weighting factor to reduce the contribution of gross errors in the data to the Hessian matrix.

One-sided Huber Function

We present an extension of the Huber concept by introducing the "one-sided" Huber optimization defined as

$$\underset{x}{\text{minimize}} \quad F(x) \triangleq \sum_{j=1}^{m} \rho_k^+(f_j(x)) \tag{11}$$

where

$$\rho_k^+(f) = \begin{cases} 0 & \text{if } f \le 0 \\ f^2/2 & \text{if } 0 < f \le k \\ kf - k^2/2 & \text{if } f > k \end{cases} \tag{12}$$

This one-sided Huber function is tailored for design optimization with upper and/or lower specifications. f is truncated when negative because the corresponding design specification is satisfied.

The gradient vector of the one-sided Huber objective function F w.r.t. x is given by

$$\nabla F = \sum_{j=1}^{m} v_j^+ \, f_j' \tag{13}$$

where

$$v_j^+ \triangleq \frac{\partial \rho_k^+}{\partial f_j} = \begin{cases} 0 & \text{if } f_j \le 0 \\ f_j & \text{if } 0 < f_j \le k \\ k & \text{if } f_j > k \end{cases} \tag{14}$$

The Hessian matrix of the one-sided Huber objective function is given by

$$H = \sum_{j=1}^{m} [d_j^+ f_j' f_j'^T + v_j^+ f_j''] \tag{15}$$

where

$$d_j^+ = \frac{\partial^2 \rho_k^+}{\partial f_j^2} = \begin{cases} 0 & if\ f_j \le 0 \\ 1 & if\ 0 < f_j \le k \\ 0 & if\ f_j > k \end{cases} \tag{16}$$

III. A DEDICATED ALGORITHM FOR HUBER OPTIMIZATION

We present a dedicated, efficient algorithm for minimizing the Huber objective functions, both one- and two-sided. We have implemented this algorithm in the CAD system OSA90/hope™ [8] as a new standard feature and used it to generate the numerical results presented in this paper.

The numerical algorithms proposed for solving (1) are of the trust region type. We calculate a sequence of points $\{x_p\}$ intended to converge to a local minimum of F. At each iterate x_p, a linear function l_j is used to approximate the nonlinear function f_j, $j = 1, 2, ..., m$, and thus a linearized model L_p of F is constructed. This model is a good approximation to F within a specified neighbourhood N_p of the pth iterate x_p. This neighbourhood N_p is intended to reflect the domain in which the l_j approximations of the f_j are valid.

Assume a tentative step $\boldsymbol{h}$ is being searched at the pth iterate x_p. If the search is successful, we go on to the next iteration, i.e., $x_{p+1} = x_p + \boldsymbol{h}$. The problem is formulated as

$$\underset{\boldsymbol{h}}{minimize} \quad L_p(\boldsymbol{h}) \triangleq L(\boldsymbol{h}, x_p) = \sum_{j=1}^{m} \rho_k(l_j(\boldsymbol{h}, x_p)) \tag{17}$$

where

$$l_j(\boldsymbol{h}_j, x_p) \triangleq f_j(x_p) + [f_j'(x_p)]^T \boldsymbol{h} \tag{18}$$

subject to the constraint $\boldsymbol{h} \in N_p$, where

$$N_p \triangleq \left\{ x \mid \|x - x_p\| \leq \delta_p \right\} \tag{19}$$

and where $\|\cdot\|$ denotes the Euclidean (least-squares) norm.

The difference between the Hessians of the true Huber objective function (7) and this linearized model is the term

$$\sum_{j=1}^{m} v_j \, f_j''$$

This error in approximating the true Hessian (7) is smaller than in the ℓ_2 case, namely,

$$\sum_{j=1}^{m} f_j \, f_j''$$

We solve the foregoing problem (17) using an algorithm similar to that of Madsen and Nielsen for the linear Huber problem [9]. This method is based on the fact that L_p is a combination of quadratic functions which are linked together in a smooth manner. Therefore, a Newton iteration is very efficient, and can be proved to find the solution after a finite number of steps. The solution to this linear problem is denoted by h_p.

The trust region radius δ_p is updated in each iteration. We propose the usual updating scheme for trust region methods (e.g., see More [10]). This is based on the ratio

$$r_p = \frac{F(x_p) - F(x_p + h_p)}{L_p(0) - L_p(h_p)} \tag{20}$$

i.e., the ratio between the decrease in the nonlinear function and the decrease in the local approximation. If r_p is close to 1 then we can afford a larger trust region in the next iteration. On the other hand, if r_p is too small then the trust region must be decreased.

The new point $x_p + h_p$ is only accepted if the objective function F decreases. Otherwise, another tentative step is calculated from x_p using a decreased trust region.

A more precise step-by-step description of the algorithm follows.

Step 1 Given x_0 and $\delta_0 > 0$. Let $0 < s_2 < 1 < s_3$. (These constants are chosen according to our experience. The algorithm is not sensitive to small changes in these constants.) Set the iteration count $p = 1$.

Step 2 Solve the trust region linearized sub-problem to find the minimizer $\boldsymbol{h}_p$ of (17) subject to (19).

Step 3 If $F(\boldsymbol{x}_p + \boldsymbol{h}_p) < F(\boldsymbol{x}_p)$, let $\boldsymbol{x}_{p+1} = \boldsymbol{x}_p + \boldsymbol{h}_p$; otherwise let $\boldsymbol{x}_{p+1} = \boldsymbol{x}_p$.

Step 4 If $r_p \leq 0.25$, reduce the size of the trust region by letting $\delta_{p+1} = \delta_p s_2$; or if $r_p \geq 0.75$, increase the size of the trust region by letting $\delta_{p+1} = \delta_p s_3$; otherwise keep the trust region size unchanged by letting $\delta_{p+1} = \delta_p$.

Step 5 If the convergence criteria are satisfied, stop; otherwise update the iteration count by letting $p = p + 1$ and repeat from *Step 2*.

It has been proved in [4] that this algorithm obeys the usual convergence theory for trust region methods.

IV. COMPARISON OF ℓ_1, ℓ_2 AND HUBER METHODS IN DATA FITTING

To illustrate the characteristics of the ℓ_1, ℓ_2 and Huber solutions for data fitting problems in the presence of large and small errors, we consider the approximation of $\sqrt{t}$ by the rational function

$$F(\boldsymbol{x},t) = \frac{x_1 t + x_2 t^2}{1 + x_3 t + x_4 t^2} \tag{21}$$

for $0 \leq t \leq 1$ [2]. $\sqrt{t}$ is uniformly sampled at 0.02, 0.04, ..., 1. We deliberately introduced large errors at 5 of the sample points and small variations to the remaining data. The ℓ_1, ℓ_2 and Huber solutions are obtained by optimizing the coefficients x_1, x_2, x_3 and x_4 in (21) to match the sampled data using the respective objective functions. The results are shown in Fig. 3. A portion of Fig. 3 is enlarged in Fig. 4 for a clearer view of the details.

As expected, the least-squares solution suffers significantly from the presence of the 5 erroneous points. On the other hand, the ℓ_1 solution, according to the optimality condition, is dictated by a subset of residual functions which have zero values at the solution. In a sense, all the nonzero residuals are viewed as large errors. This tendency towards a biased ℓ_1 solution, as

TABLE IV
NUMBER OF FUNCTION EVALUATIONS
REQUIRED BY DIFFERENT ALGORITHMS

Algorithm	Starting Point			
	1.5	2	2.25	3
Dedicated Huber	4	4	4	4
Quasi-Newton	8	5	5	7
Conjugate-Gradient	13	13	11	14
Simplex	26	16	16	24

The optimization problem is to estimate the mean of FET parameter τ using the Huber objective function.

Figure Captions

Fig. 1. The ℓ_1 and ℓ_2 objective functions in the one-dimensional case. The ℓ_1 function is rescaled and shifted in accordance with the corresponding part in the Huber function. It has the form $F = |k|f - k^2/2$. The ℓ_2 function has the form $F = f^2/2$.

Fig. 2. The Huber, ℓ_1 and ℓ_2 objective functions in the one-dimensional case. The strikes and dots represent the discrete points on the ℓ_1 and ℓ_2 curves, respectively, in Fig 1. The continuous curve indicates the Huber objective function.

Fig. 3. ℓ_1, ℓ_2 and Huber solutions for data fitting in the presence of errors.

Fig. 4. An enlarged portion of Fig. 3.

Fig. 5. Run chart of the extracted FET time-delay τ.

Fig. 6. Percentage of "small errors" for the FET time-delay τ versus the threshold k.

Fig. 7. Percentage of "small errors" for the FET gate lead inductance L_G versus the threshold k.

Fig. 8. Percentage of "small errors" for the FET model parameter C_{10} versus the threshold k.

Fig. 9. The resistive mesh circuit.

Fig. 10. Multiplexer responses at the starting point, showing the common port return loss (———) and the individual channel insertion losses (------).

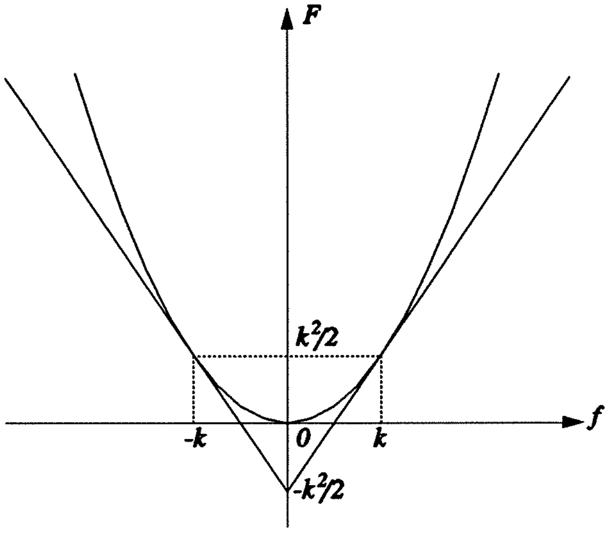
Fig. 11. Multiplexer responses after the minimax optimization with 10 variables: spacings and channel input transformer ratios; the common port return loss (———) and the individual channel insertion losses (------). This result hardly improved upon the starting point shown in Fig. 10.

Fig. 12. Multiplexer responses after the one-sided Huber optimization with 10 variables: spacings and channel input transformer ratios; the common port return loss (———) and the individual channel insertion losses (------). This result is significantly better than the minimax solution of Fig. 11.

Fig. 13. Multiplexer responses after the minimax optimization with the full set of 75 variables, showing the common port return loss (———) and the individual channel insertion losses (------).

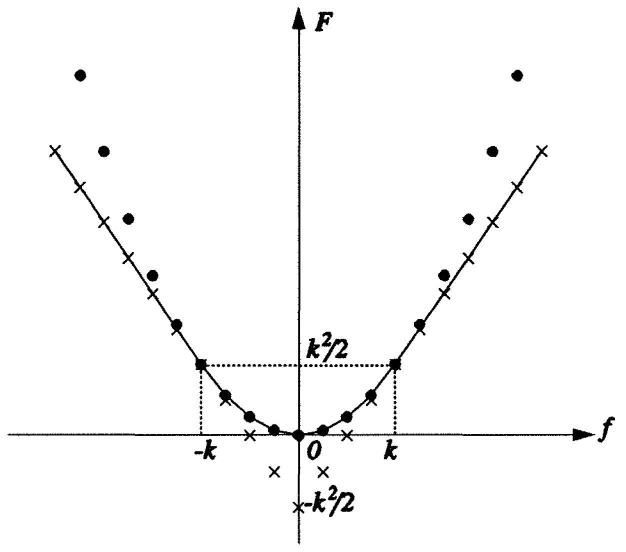
IEEE Trans. Microwave Theory Tech., vol. 41, December 1993.

Bandler *et al*. "Huber optimization ......" , Fig. 1



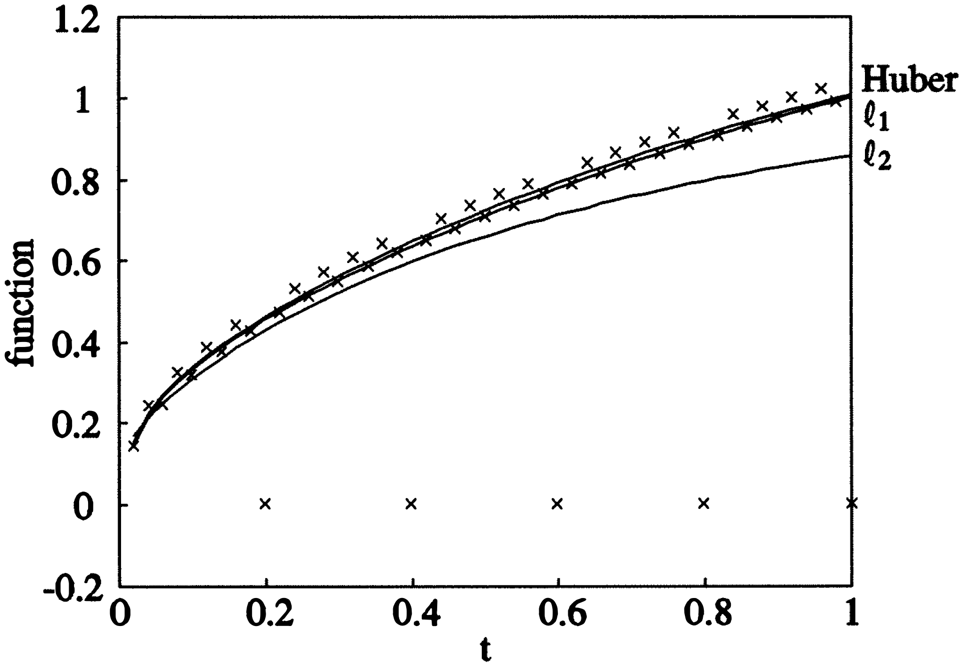
IEEE Trans. Microwave Theory Tech., vol. 41, December 1993.

Bandler *et al*. "Huber optimization ......" , Fig. 2



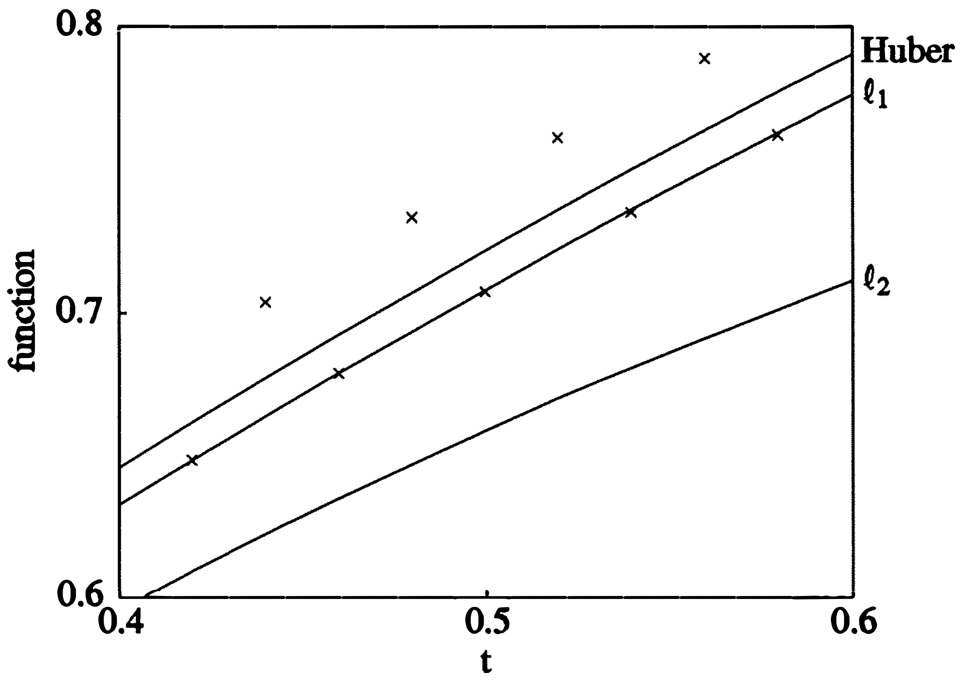
IEEE Trans. Microwave Theory Tech., vol. 41, December 1993.

Bandler *et al*. "Huber optimization ......" , Fig. 3



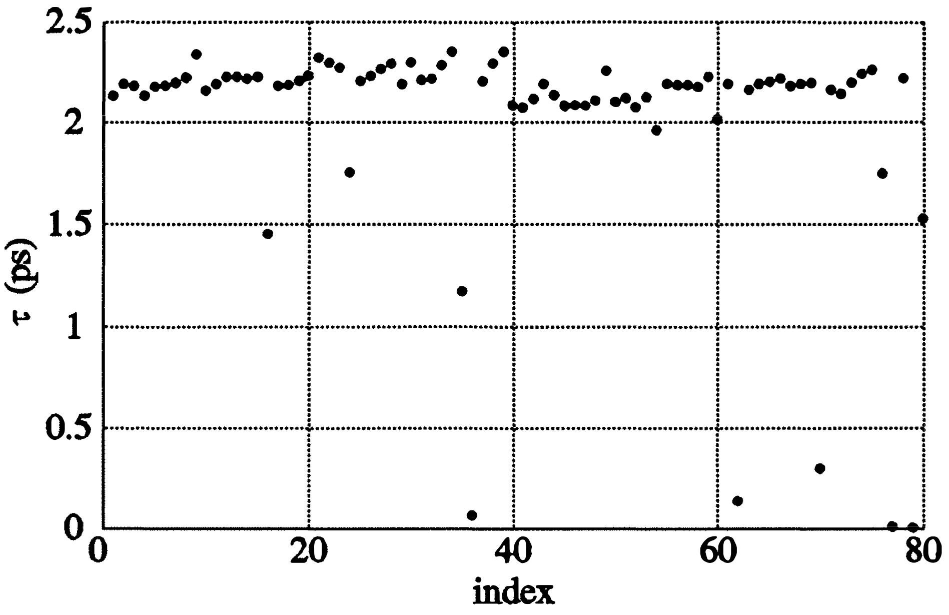
IEEE Trans. Microwave Theory Tech., vol. 41, December 1993.

Bandler *et al.* "Huber optimization ......" , Fig. 4



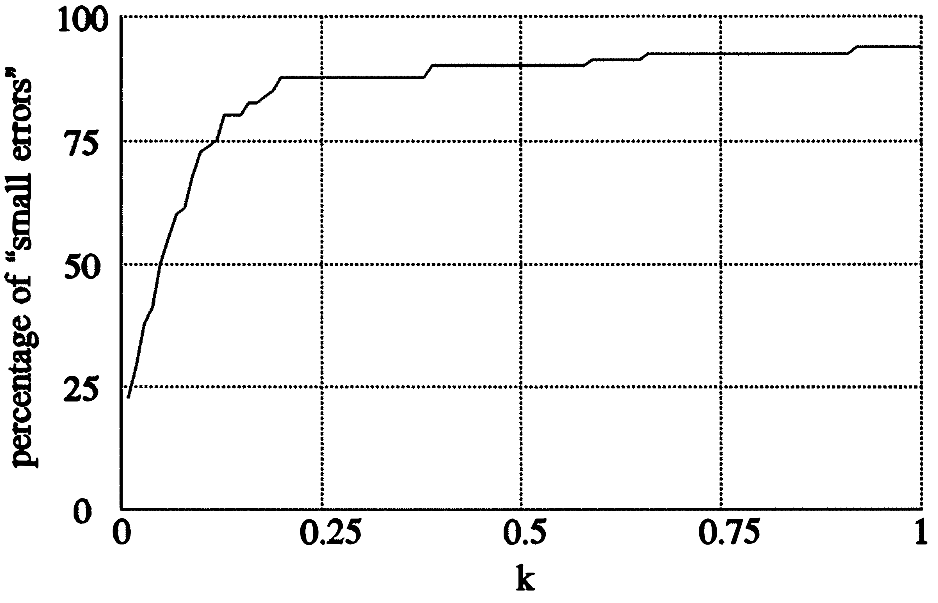
IEEE Trans. Microwave Theory Tech., vol. 41, December 1993.

Bandler *et al.* "Huber optimization ......" , Fig. 5

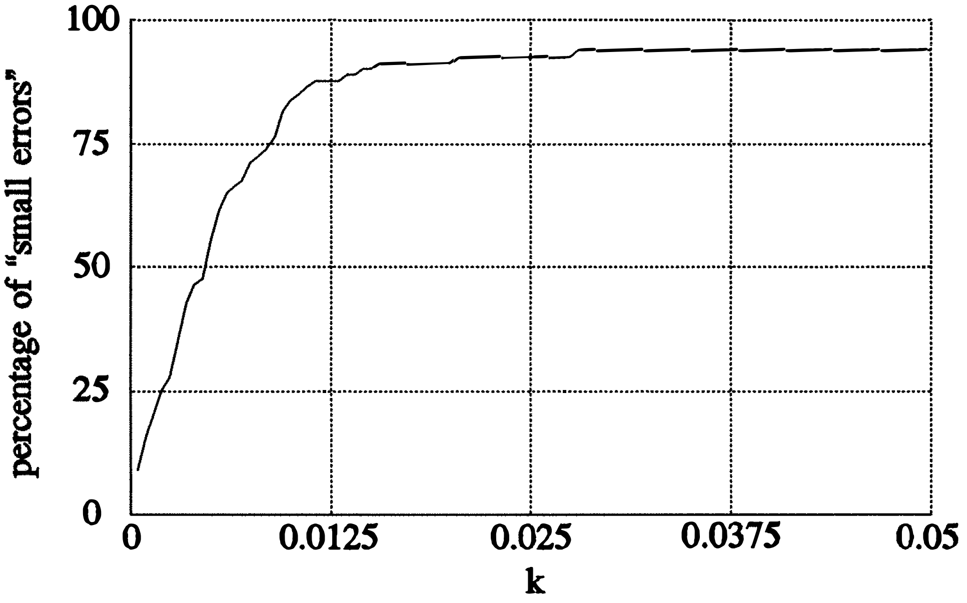


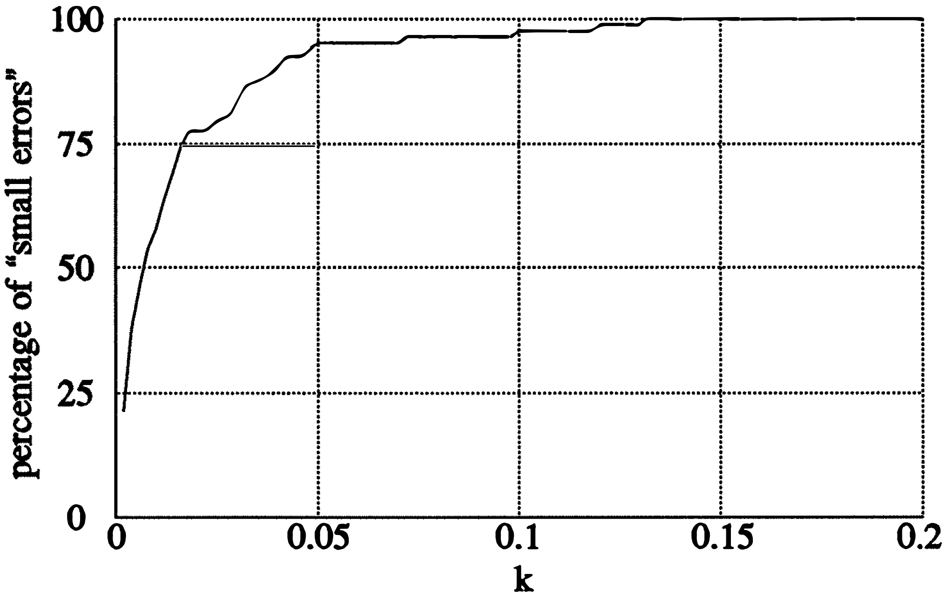
IEEE Trans. Microwave Theory Tech., vol. 41, December 1993.

Bandler *et al.* "Huber optimization ......" , Fig. 6



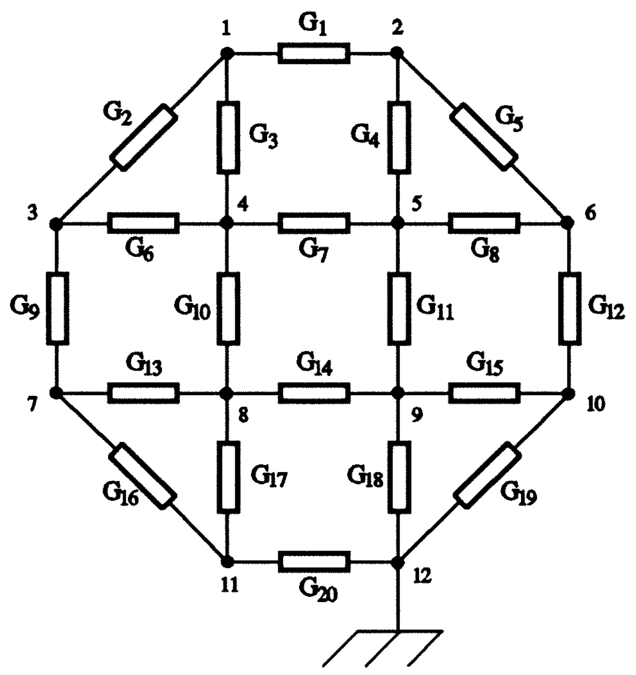
Bandler *et al.* "Huber optimization ......" , Fig. 7




100
75
50
25
0
0
0.05
0.1
0.15
0.2
k
percentage of "small errors"

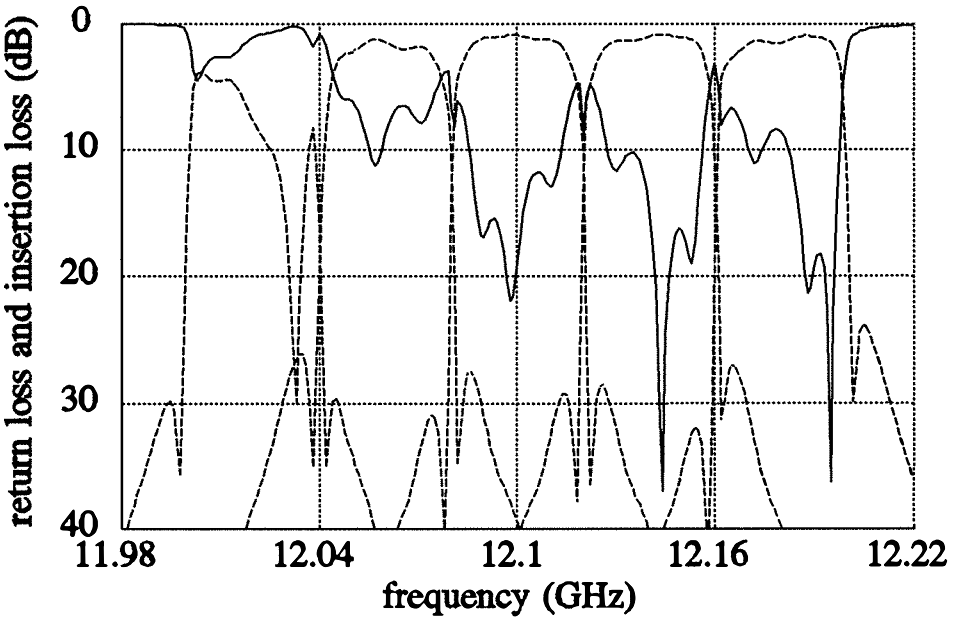
IEEE Trans. Microwave Theory Tech., vol. 41, December 1993.

Bandler *et al.* "Huber optimization ......" , Fig. 9



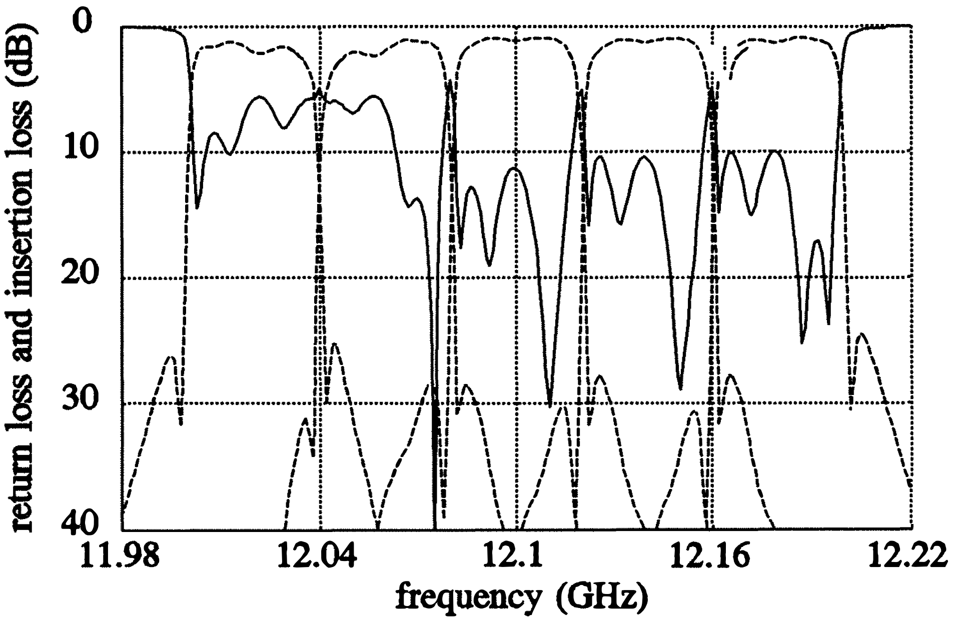
IEEE Trans. Microwave Theory Tech., vol. 41, December 1993.

Bandler *et al.* "Huber optimization ......" , Fig. 10



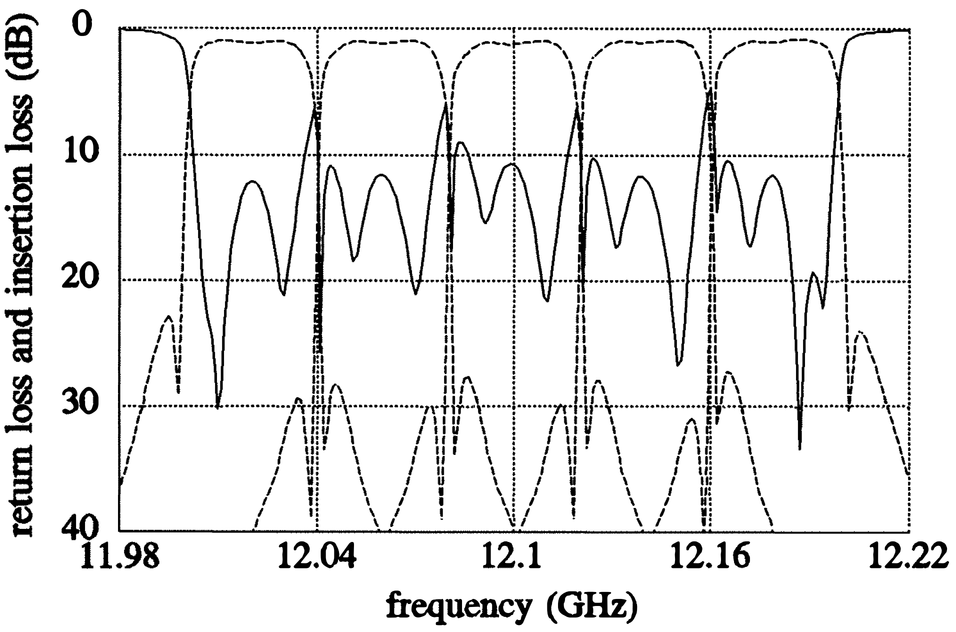
IEEE Trans. Microwave Theory Tech., vol. 41, December 1993.

Bandler *et al.* "Huber optimization ......" , Fig. 11



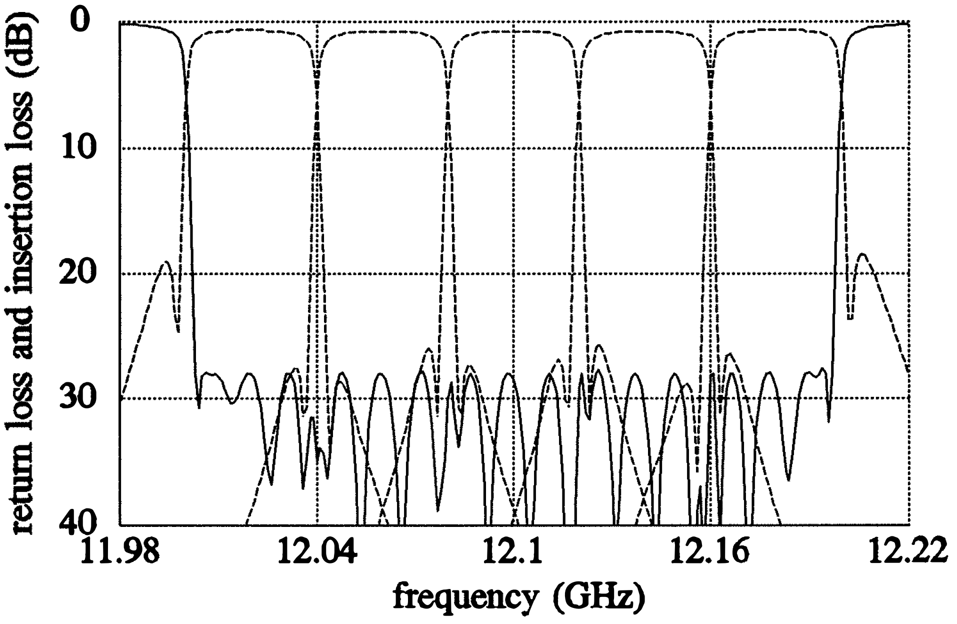
IEEE Trans. Microwave Theory Tech., vol. 41, December 1993.

Bandler *et al.* "Huber optimization ......" , Fig. 12



IEEE Trans. Microwave Theory Tech., vol. 41, December 1993.

Bandler *et al.* "Huber optimization ......" , Fig. 13


return loss and insertion loss (dB)
frequency (GHz)