Proc. ECCTD, The Hague, August 1981

# POSTPRODUCTION TUNING EMPLOYING NETWORK SENSITIVITIES

J.W. Bandler and A.E. Salama
Group on Simulation, Optimization and Control
and Department of Electrical and Computer Engineering
McMaster University, Hamilton, Canada L8S 4L7

*Abstract*

*This paper deals with the postproduction tuning assignment problem for electrical circuits. A review of recent tuning algorithms that utilize sensitivity data is presented. We first consider functional tuning methods to which we contribute a new algorithm. Then we discuss deterministic tuning algorithms, in particular those of Alajajian et al. and Lopresti, and a modification to Lopresti's algorithm is proposed. Finally, we test three of the algorithms considered by examining their performance in tuning an active filter.*

## 1. INTRODUCTION

Postproduction tuning is often essential in the manufacturing of electrical circuits. Tolerances on the circuit components, parasitic effects and uncertainties in the circuit model cause deviations in the manufactured circuit performance, and violation of the design specifications may result. Therefore, postproduction tuning is included in the final stages of the production process to readjust the network performance in an effort to meet the specifications.

Computer-aided designers have approached the tuning problem in two ways, each emphasizing one distinct facet. Before production, at the time of designing a circuit, one can consider tuning as an integral part of the design process [1,2], the objective being to relax the tolerances on the circuit components and compensate for the uncertainties in the model parameters. The integral design problem is formulated and solved using optimization such that the essential demand of production cost reduction is optimally met. The solution of the design problem provides the manufacturer with the allowed design tolerances and the tunable parameters.

In the final production stages, the manufactured circuit is usually tested to check whether or not it meets design specifications. Tuning is usually needed and the tuning assignment problem arises. Here, it is required to find the necessary changes in the tunable parameters to adjust the manufactured circuit to satisfy the design requirements. Computer-aided designers contributed to this problem by proposing a number of algorithms [3-14]. Most of these algorithms utilize network sensitivities and first-order approximations.

This paper mainly reviews the basic tuning assignment methods with emphasis on algorithms which employ network sensitivities. We first start by considering the relevant fundamental definitions and concepts together with the points under which the methods could be categorized. In practice, one of two classes of methods for tuning is usually employed. We devote Section 3 to functional tuning algorithms, where a very recent algorithm developed by the authors [14] is also presented. In Section 4 we discuss deterministic tuning algorithms. A practical active filter example is considered in Section 5, where we put under test three of the recently developed algorithms.

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant A7239.

## 2. FUNDAMENTAL CONCEPTS

The manufactured circuit is characterized by the actual design parameter values which are given by

$$\underset{\sim}{\phi}^a \triangleq [\phi_1^a \quad \phi_2^a \quad \dots \quad \phi_n^a]^T, \qquad (1)$$

where $\phi_1, \phi_2, \dots, \phi_n$ denote the n design parameters. The parasitic effects associated with the produced circuit can be represented by the additional variables

$$\underset{\sim}{\phi}_p \triangleq [\phi_{n+1} \; \phi_{n+2} \; \dots \; \phi_p]^T. \qquad (2)$$

A subset of the design parameters is used for tuning. Let

$$\underset{\sim}{\phi}_t \triangleq [\phi_1 \quad \phi_2 \quad \dots \quad \phi_k]^T \qquad (3)$$

represent the tunable circuit parameters and let

$$\underset{\sim}{\phi}_r \triangleq [\phi_{k+1} \quad \phi_{k+2} \quad \dots \quad \phi_n]^T \qquad (4)$$

designate the nontunable parameters.

A set of performance functions of $\underset{\sim}{\phi}_t$, $\underset{\sim}{\phi}_r$ and $\underset{\sim}{\phi}_p$ given by

$$\underset{\sim}{f} \triangleq [f_1 \quad f_2 \quad \dots \quad f_m]^T, \qquad (5)$$

are usually monitored during the tuning process. These functions may or may not be related directly to the circuit design specifications.

The objective of the tuning assignment problem is to find the required changes in the tunable parameters, namely $\underset{\sim}{\phi}_t$, such that the performance functions $\underset{\sim}{f}$ satisfy the designer requirements.

Since the changes in the tunable parameters are predicted to be small, the first-order approximation is utilized in most tuning algorithms to estimate the changes in the error functions and to provide the required tuning amounts $\Delta\underset{\sim}{\phi}_t$. The first-order change in the functions $\underset{\sim}{f}$ is given by

$$\Delta \underset{\sim}{f} = \sum_{j=1}^{k} \phi_j^x \frac{\partial \underset{\sim}{f}}{\partial \phi_j} \frac{\Delta \phi_j}{\phi_j^x} = \underset{\sim}{S} \, \underset{\sim}{\chi}, \tag{6}$$

where $\underset{\sim}{S}$ defines the sensitivity matrix whose (i,j) element is given by

$$s_{ij} \triangleq \phi_j^x \frac{\partial f_i}{\partial \phi_j} \tag{7}$$

and is computed for a suitable network model ($\underset{\sim}{\phi}_t^x$, $\underset{\sim}{\phi}_r^x$, $\underset{\sim}{\phi}_p^x$). $\underset{\sim}{\chi}$ is the vector of the relative change in the tunable parameters $\Delta\phi_j/\phi_j^x$.

Several criteria have to be considered to form a basis for categorizing methods for tuning. Without being exhaustive one may list the following points. Either the network response is employed directly or auxilliary functions are considered instead. This determines a possible choice for the performance functions. In computing the amount of tuning either a point matching problem is solved or an iterative optimization procedure is carried out. The extent of the information available about the circuit to be tuned, namely, the circuit model and the circuit parameter values must be considered. One distinguishes at this time between functional and deterministic tuning. The way in which the sensitivity matrix $\underset{\sim}{S}$ is to be evaluated needs consideration. Either partial derivatives, incremental sensitivities or approximate derivatives are employed.

The categories that emerge from the foregoing points are related to each other. It has been customary, however, to distinguish the methods principally according to whether or not the circuit parameters are known. We adopt the same philosophy by considering in the next two sections functional tuning and deterministic tuning methods.

## 3. FUNCTIONAL TUNING APPROACH

Functional tuning is the traditional way of tuning electrical circuits. After manufacturing and assembling, the circuit performance specifications are checked. If tuning is necessary a sequence of tunable parameter adjustments is carried out until the specifications are met. In functional tuning methods the network elements are generally assumed unknown, for example, it may be difficult to measure or identify the actual circuit element values.

Functional tuning is obviously suitable when the sensitivity matrix $\underset{\sim}{S}$ is diagonal [9]. For every performance function there is a separate tunable parameter for its adjustment. The tuning process is noniterative, very fast and does not need a skillful operator. Normally, $\underset{\sim}{S}$ is not diagonal, but if it can be arranged to be triangular or diagonally dominant [9], then the tuning process is carried out in a certain specified sequence to reduce the iterative adjustment of the tuning elements. The reported success of this method is consequently confined to small and simple circuits, and when the circuit design is specifically chosen with appropriate tuning in mind.

Shockley et al. [10] proposed the evaluation of the tuning amounts by the direct inversion of the sensitivity matrix. The tuning amounts are directly given by

$$\underset{\sim}{\chi} = \underset{\sim}{S}^{-1} \Delta \underset{\sim}{f}. \tag{8}$$

In their implementation for tuning an active filter they used the zeros, poles and the gain factor of the transfer function as performance functions. From the measurement of the transfer function at a number of critical frequencies, equal to the number of poles and zeros plus one, the coefficients of the transfer function are evaluated. Then, a root finding routine is used to find the actual values of the zeros and the poles. The variation of these values from nominal define the vector $\Delta\underset{\sim}{f}$ and the nominal sensitivities could be used to construct the matrix $\underset{\sim}{S}$. Although the method is simple, it suffers from ill-conditioning due to the inaccuracies encountered in computing the poles and zeros. Also, as reported in [10], the method completely fails when the changes in the manufactured elements are not small enough to permit the use of the differential sensitivities.

To overcome the drawbacks of the previous method Adams et al. [11] and Müller [12] suggested the use of auxilliary functions which are almost linear in the tunable parameters. The suggestion is quite important, but it is difficult to determine these functions for a general design. Also, on the assumption that the number of tunable parameters is less than the available measurable performance functions, they have used the linear least squares optimization for estimating the tuning amounts. This provides a closed form solution which is given by

$$\underset{\sim}{\chi} = \underset{\sim}{S}^{+} \Delta \underset{\sim}{f}, \tag{9}$$

where $\underset{\sim}{S}^{+}$ is the pseudoinverse matrix defined by

$$\underset{\sim}{S}^{+} \triangleq [\underset{\sim}{S}^T \underset{\sim}{S}]^{-1} \underset{\sim}{S}^T \tag{10}$$

and the rank of $\underset{\sim}{S}$ is assumed to be equal to the cardinality of the vector $\underset{\sim}{\chi}$.

The method is usually applied iteratively [11, 12] and is reported to have very fast convergence. But, as pointed out in [13], the unconstrained solution often requires an adjustment which is infeasible either in magnitude or direction. The least squares solution should then be constrained and gradient search techniques employed to determined the best constrained solution.

Very recently, another approach which utilizes the response measurements as well as the simulated network sensitivities has been developed [14]. The method is based on formulating the tuning process as a linear minimax optimization problem. If we assume that the performance functions $\underset{\sim}{f}$ represent a set of error functions the tuning problem can be recast as

$$\text{Minimize } \chi_{k+1} \tag{11a}$$

subject to

$$f_i + \sum_{j=1}^{k} \phi_j^x \frac{\partial f_i}{\partial \phi_j} \chi_j \leq \chi_{k+1}, \; i = 1, \ldots, m, \tag{11b}$$

$$\bar{a}_j \leq \chi_j \leq \bar{b}_j, \; j = 1, \ldots, k, \tag{11c}$$

where $\chi_{k+1}$ is an additional variable. The linear constraints (11c) define the limits on the tuning amounts in size and direction. They play a very important role here since they guarantee the validity of the linear approximation and the use of the partial derivatives. The functions $f_i$ are obtained by directly measuring the response, and a new set of the measurements is needed for a subsequent iteration. The direct measurement of the functions $f_i$ compensates for the deviations resulting from the inaccurate adjustment of tunable elements and the utilization of the approximate partial derivatives. The network sensitivities are evaluated using a good approximate model of the network $\underset{\sim}{\phi}^x$, and they can be updated after each iteration using the Broyden rank one updating formula [15]. The linear program (11) is solved for the amounts of tuning required. The tunable elements are adjusted by that amount to the extent possible. The iterative tuning procedure is carried out until an optimum is reached or the amounts of tuning for all elements cannot be implemented practically. The method optimally utilizes the measurements and the simulated network sensitivities during the tuning process, and it converges in few iterations [14].

In general, functional tuning methods are not fast. Deterministic tuning methods are considerably faster although they require much more information. In the next section we examine quite closely the deterministic methods of tuning.

## 4. DETERMINISTIC TUNING APPROACH

Deterministic tuning is primarily proposed to simplify the tuning assignment problem and eliminate the iterations needed in the functional tuning approach. It is usually carried out by measuring all the parameters of the manufactured network and the possible parasitic effects. Then, a matching procedure is carried out, where it is required to match the performance functions by varying the tunable parameter values. A system of nonlinear equations usually results and for special simple problems [7] a closed form of the required tuning amounts can be obtained.

In general, a closed form expression of the required tunable amounts is usually not available and even if it does exist it needs a formidable algebraic manipulation task [7]. Most of the recently developed deterministic tuning algorithms utilize network sensitivities and first-order approximations.

Deterministic tuning suffers from the need of supporting hardware for measuring the components and the parasitic effects. Also, the network model used should quite closely simulate the actual performance of the manufactured circuit and this necessitates the consideration of all possible parasitic effects, which is impossible. So functional tuning is usually utilized after the network is deterministically tuned to improve the circuit performance [8].

As a matching problem, the tuning assignment problem is not well posed, i.e., a solution may or may not exist using the set of the tunable parameters chosen and if it exists it may not be unique. The problem of uniqueness is usually less important than the problem of existence. Lopresti [4] eliminated the problem of existence by formulating the problem using discrete optimal control theory. The algorithm was originally considered for the tuning of hybrid active filters, but it can be applicable to any filter with the appropriate choice of performance functions. In his implementation of the algorithm for the active filter case he used the transfer function coefficients as his performance functions to be matched. Recalling equation (6) the deviation of the performance functions after tuning from their nominal values is given by

$$\delta \underset{\sim}{f} = \underset{\sim}{f} - \underset{\sim}{f}^0 = (\underset{\sim}{f}^a - \underset{\sim}{f}^0) + \sum_{j=1}^{k} \phi_j^x \frac{\partial \underset{\sim}{f}}{\partial \phi_j} \frac{\Delta\phi_j}{\phi_j^x}, \quad (12)$$

where $\underset{\sim}{f}^0$ represents the nominal values of the transfer function coefficients and $\underset{\sim}{f}^a$ gives the actual values of the coefficients before tuning. The above differential function can be rewritten as

$$\underset{\sim}{x}_{j+1} = \underset{\sim}{x}_j + \underset{\sim}{s}_j \chi_j, \; j = 1, \ldots, k, \quad (13a)$$

$$\underset{\sim}{x}_1 = \underset{\sim}{f}^a - \underset{\sim}{f}^0, \quad (13b)$$

where

$$\underset{\sim}{s}_j = \phi_j^x \frac{\partial \underset{\sim}{f}}{\partial \phi_j}. \quad (13c)$$

The partial derivatives are evaluated for the nominal circuit and $\chi_j$ is defined as before.

The tuning problem can now be stated as a quadratic optimal control problem as follows. Minimize

$$\underset{\sim}{x}_{k+1}^T \underset{\sim}{Q} \underset{\sim}{x}_{k+1} + \sum_{i=1}^{k} \gamma_i \chi_i^2 \quad (14)$$

subject to (13). The second part of the objective function penalizes the excessive tuning amounts and guarantees the uniqueness of the solution.

A closed form solution is obtained using the Riccati equation [4]. If a solution to the deterministic tuning problem exists the objective function will have a zero value. Otherwise, it will approach zero as closely as possible. The process could be performed sequentially by measuring the tunable elements after adjustment and recalculating the new values of the tuning amounts accordingly. This will partially compensate for inaccurate element adjustment and an imprecise circuit model. The matrix $\underset{\sim}{Q}$ plays a very important role in this process and an intelligent choice of its elements will be important for the success of this approach. In our implementation of the process, which will be reported on in the next section, different matrices were tried before a reasonable solution was obtained.

Another quite successful deterministic tuning algorithm has been proposed in [5], which is based on the observation that the first-order approximation which utilizes differential sensitivities is not always reliable. It utilizes Tellegen's theorem to derive a large change sensitivity expression, relating large changes in the tuning elements to the desired changes in the performance functions in the manufactured circuit. The response of the filter is matched to within a multiplicative constant. Both the multiplicative constant and the tuning amounts are obtained through solving a system of linear equations. Recalling (6), this system of equations is given by

$$\underset{\sim}{S} \underset{\sim}{\chi} = c \, \underset{\sim}{f}^0 - \underset{\sim}{f}^a, \quad (15a)$$

$$[\underset{\sim}{S} - \underset{\sim}{f}^0] \begin{bmatrix} \underset{\sim}{x} \\ c \end{bmatrix} = \hat{\underset{\sim}{S}}\, \hat{\underset{\sim}{x}} = -\underset{\sim}{f}^a, \tag{15b}$$

where $\underset{\sim}{f}^0$ represents the nominal output voltages, $\underset{\sim}{f}^a$ the actual measured values defined at a set of critical frequencies and c is the unknown multiplicative constant. The matrix $\hat{\underset{\sim}{S}}$ should be nonsingular and this will restrict the number of the tunable elements and the number of independent performance functions $\underset{\sim}{f}$ needed.

The matrix $\underset{\sim}{S}$ is constructed such that its elements approximate the large change sensitivities. Since the initial state of the circuit is known from the direct measurements of the circuit parameters and the final desired state is almost known (approximately it could be considered the nominal state), an approximate expression of the incremental sensitivities can be easily derived [5]. In general, the algorithm seems to be very efficient and very simple at the same time. But it has the disadvantage of any matching technique, where there is no control over the tuning amounts, hence infeasible tuning amounts may result. In the examples considered [6], the algorithm performed quite well and the author came to the conclusion that the use of the large change sensitivity expressions together with the appropriate choice of tunable elements are the principal reasons for their success.

## 5. EXAMPLE

The two deterministic tuning algorithms we have reviewed and our newly developed functional tuning algorithm are tested by applying them to the same network example which originally appeared in [6]. The highpass notch circuit is shown in Fig. 1. The nominal circuit component values and the actual circuit values are given in Table I.

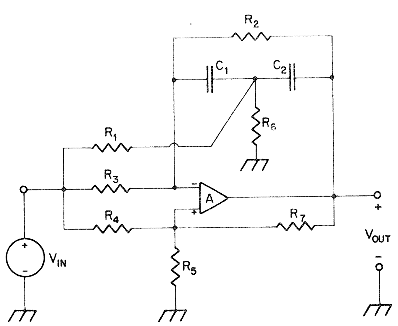


Fig. 1 The highpass notch filter circuit

We first applied the deterministic tuning algorithm proposed by Alajajian [6]. Four resistor elements are chosen as the tunable parameters, namely $R_3$, $R_5$, $R_6$ and $R_7$. To construct equation (15), the output voltage of the filter is considered at three critical frequencies 450.38, 636.94 and 676.75 Hz. Using Tellegen's theorem (15) can be rewritten as

$$\sum_{\ell=1}^{3} (V_\ell^i + \Delta V_\ell^i)\, \hat{V}_\ell^i\, \Delta G_\ell - c\, \overline{V}_{out}^i = -V_{out}^i, \tag{16}$$

where i = 1, 2, 3, correspond to the three considered critical frequencies. $V_\ell$ and $\hat{V}_\ell$ are the branch voltage of the tunable parameters in the original circuit and its adjoint before tuning, respectively. $V_{out}$ and $\overline{V}_{out}$ are the actual and the desired output voltages, respectively. $V_\ell + \Delta V_\ell$ is the tunable branch voltage after tuning which can be approximated by the nominal branch voltage. Solving (16) we get the required changes in the tunable parameters and the factor c. After three iterations the tuned response approached the nominal response very closely. The responses before and after tuning are shown in Fig. 2.

TABLE I
ELEMENT VALUES

| Element | Nominal Value | Actual Value | Percentage Deviation |
|---|---|---|---|
| $R_1$ (kΩ) | 13.260 | 13.260 | 0.0 |
| $R_2$ (kΩ) | 93.0 | 93.0 | 0.0 |
| $R_3$ (kΩ) | 214.0 | 192.6 | -10.0 |
| $R_4$ (kΩ) | 2.0 | 2.0 | 0.0 |
| $R_5$ (kΩ) | 2.0 | 1.8 | -10.0 |
| $R_6$ (kΩ) | 12.467 | 11.221 | -10.0 |
| $R_7$ (kΩ) | 10.00 | 9.00 | -10.0 |
| $C_1$ (μF) | 0.01 | 0.00973 | -2.07 |
| $C_2$ (μF) | 0.01 | 0.00965 | -3.35 |
| A | 10000.0 | 10000.0 | 0.0 |

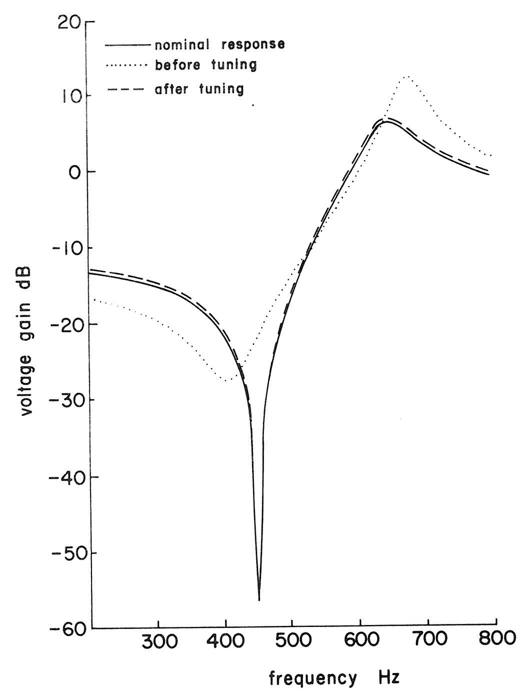


Fig. 2 The responses for the Alajajian algorithm (Method 1)

Secondly, we considered the other deterministic tuning algorithm proposed by Lopresti [4]. The transfer function of the filter considered is given by

$$T(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^2 + a_1 s + a_0}, \tag{17}$$

where $b_2 = 0.5$, $b_1 = 0.0$, $b_0 = 4 \times 10^6$, $a_1 = 500.0$ and $a_0 = 16 \times 10^6$. Due to the wide variation in the tunable parameters frequency scaling was necessary. With $\hat{s} = 1000s$ the coefficients will become $b_2 = 0.5$, $b_1 = 0.0$, $b_0 = 4$, $a_1 = 0.5$ and $a_0 = 16.0$. To construct (13) we define $\underline{f} \triangleq [a_1\ a_0\ b_2\ b_1\ b_0]^T$, and we need to find the corresponding sensitivities. We followed an approach other than the one proposed in [4]. Equation (17) can be rewritten as

$$a_1 sT(s) + a_0 T(s) - b_2 s^2 - b_1 s - b_0 = -s^2 T(s). \tag{18}$$

We use (18) to find the transfer function coefficients by evaluating T(s) at five independent frequencies. Also, by differentiating both sides of (18) w.r.t. the tunable parameter $\phi_j$ we get

$$s\,T(s)\frac{\partial a_1}{\partial \phi_j} + T(s)\frac{\partial a_0}{\partial \phi_j} - s^2\frac{\partial b_2}{\partial \phi_j} - s\frac{\partial b_1}{\partial \phi_j} - \frac{\partial b_0}{\partial \phi_j} = -(s^2 + a_1 s + a_0)\frac{\partial T(s)}{\partial \phi_j} \tag{19}$$

which is used in finding the coefficient sensitivities, or the vector $\underline{s}_j$ defined in (13c). The sensitivities are evaluated using the actual circuit parameter values. This improved the performance of the algorithm. As was pointed out before, the algorithm is sequential in nature: after adjusting every tunable element we recalculate the sensitivities and find the newly required tuning amounts using the Riccati equation [4]. The weighting factors considered in the objective function need careful choice to emphasize the effect of the different transfer function coefficients. In our implementation we took $\underline{Q} = \mathrm{diag}\{4.0, 0.04, 4.0, 10^{12}, 0.0625\}$ and $\gamma_i = 0.001$. The results of applying this algorithm to the same circuit which is considered before and using the same tunable parameters is shown in Fig. 3. The tuned response approaches the nominal response and yields acceptable results. The algorithm can be applied iteratively and an improved response may result.

Finally, we tested the functional tuning approach proposed by us [14]. To construct the optimization problem defined in (11) we defined the functions $f_i$ as the absolute deviation of the output voltage from its nominal value at 20 frequencies defined in the interval 410-505 Hz, where the notch lies. The limits given in (11c) are restricted to be $\bar{b}_j = -\bar{a}_j = 0.02$. After 11 iterations the tuned responses very closely approached the nominal response, as shown in Fig. 4. The sensitivities are updated after every iteration using the Broyden formula and the initial sensitivities are computed with the components at the nominal starting values. A tunable amount less than 0.1 percent is assumed to be infeasible and no adjustment is carried out for the corresponding tunable parameter.

The final values for the four tunable resistors for the three tuning algorithms are given in Table II.

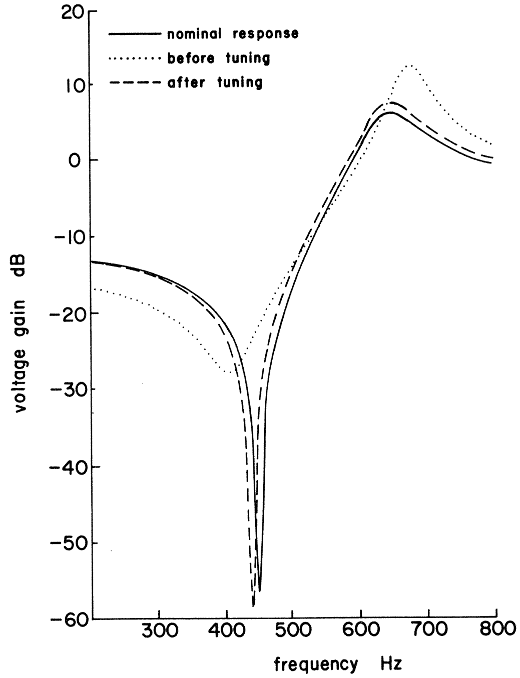


Fig. 3 The responses for the modified Lopresti algorithm (Method 2)

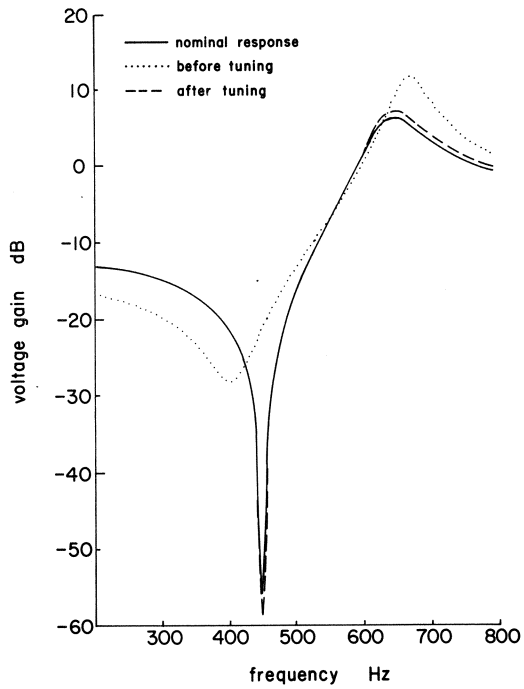


Fig. 4 The responses for our algorithm (Method 3)

Method 1 is our implementation of Alajajian et al. [5]. Method 2 is our modification of Lopresti's method [4]. Method 3 is our approach [14].

## 6. DISCUSSION

The results of the example have affirmed and clarified a number of important points. As was

TABLE II
RESULTS OF TUNING

| Tuned Element | Method 1 [5] | Method 2 [4] | Method 3 [14] |
|---|---|---|---|
| $R_3$ | 192.94 | 184.487 | 201.952 |
| $R_5$ | 2.2459 | 2.241 | 2.115 |
| $R_6$ | 13.892 | 13.747 | 13.061 |
| $R_7$ | 1.006 | 0.9993 | 0.9730 |

first pointed out by Pinel [16], network sensitivities can provide a valuable tool for network tuning. Deterministic tuning methods that utilize network sensitivities are normally superior to functional tuning methods. The more information we have about the network to be tuned the more efficient the functional tuning methods will be. We have utilized this fact in our functional tuning algorithms. We have assumed that a good starting approximate model for the circuit is available to derive the network sensitivities. Updating those sensitivities during the tuning process improves the initial assumed values, and a better approximation of the actual sensitivities is obtained after each iteration. The results indicate how the algorithm converges to the optimal solution, but the number of iterations are greater than for the other two deterministic approaches.

Lopresti's original algorithm suffers from two drawbacks which we have eliminated during our implementation of his algorithm. We used another more efficient and easily programmable algorithm for obtaining the transfer function coefficients and their sensitivities. Instead of using the nominal sensitivities, the actual sensitivities are utilized since the network element values are all known. Also, we update the sensitivities after each parameter adjustment and recalculate the required tuning amounts using the Riccati formula. The results obtained using the nominal sensitivities showed inferior performance compared with the results obtained using the modifications.

Large change sensitivities seem more promising than the differential sensitivities for deterministic tuning methods. This appears very clearly in the results obtained by Alajajian and in our implementation of his algorithm. The formulation of the tuning problem as an optimization problem provides a means for employing the different physical circuit constraints directly, especially for the functional tuning methods. The point matching scheme seems adequate for deterministic tuning methods.

## 7. REFERENCES


[1] Bandler, J.W., Liu, P.C. and Tromp, H., "A nonlinear programming approach to design centering, tolerancing and tuning", IEEE Trans. Circuits and Systems, vol. CAS-23, 1976, pp. 155-165.

[2] Polak, E. and Sangiovanni-Vincentelli, A., "Theoretical and computational aspects of the optimum design centering, tolerancing and tuning problem", IEEE Trans. Circuits and Systems, vol. CAS-26, 1979, pp. 795-813.

[3] Bandler, J.W. and Biernacki, R.M., "Postproduction parameter identification and tuning of analog circuits", Proc. European Conf. Circuit Theory and Design, Warsaw, 1980, vol. 2, pp. 205-220.

[4] Lopresti, P.V., "Optimum design of linear tuning algorithms", IEEE Trans. Circuits and Systems, vol. CAS-24, 1977, pp. 144-151.

[5] Alajajian, C.J., Trick, T.N. and El-Masry, E.I., "On the design of an efficient tuning algorithm", Proc. IEEE Int. Symp. Circuits and Systems, Houston, 1980, pp. 807-811.

[6] Alajajian, C.J., "A new algorithm for tuning of analog filters", Ph.D. Thesis, University of Illinois, Urbana-Champaign, IL, 1979.

[7] Lüder, E. and Kaiser, B., "Precision tuning of miniaturized circuits", Proc. IEEE Int. Symp. Circuits and Systems, Munich, 1976, pp. 209-211.

[8] Friedenson, R.A., Danniels, R.W., Dow, R.J. and McDonald, P.H., "RC active filters for the D3 channel bank", Bell System Tech. J., vol. 54, 1975, pp. 507-529.

[9] G.S. Moschytz, Linear Integrated Networks - Design. New York: Van Nostrand Reinhold, 1975.

[10] Shockley, T.D. and Morris, C.F., "Computerized design and tuning of active filters", IEEE Trans. Circuit Theory, vol. CT-20, 1973, pp. 438-441.

[11] Adams, R.L. and Manaktala, V.K., "An optimization algorithm suitable for computer-assisted network tuning", Proc. IEEE Int. Symp. Circuits and Systems, Newton, MA, 1975, pp. 210-212.

[12] Müller, G., "On computer-aided tuning of microwave filters", Proc. IEEE Int. Symp. Circuits and Systems, Munich, 1976, pp. 209-211.

[13] Garrison, J.L., Olsen, A., Jr. and Simmonds, T.H., Jr., "Transmission networks and magnetic components", Bell System Tech. J., vol. 53, 1974, pp. 2203-2248.

[14] Bandler, J.W., Rizk, M.R.M. and Salama, A.E., "An interactive optimal postproduction tuning technique utilizing simulated sensitivities and response measurements", IEEE Int. Microwave Symp., Los Angeles, June 1981.

[15] Broyden, C.G., "A class of methods for solving nonlinear simultaneous equations", Mathematics of Computation, vol. 19, 1965, pp. 577-593.

[16] Pinel, J.F., "Computer-aided network tuning", IEEE Trans. Circuit Theory, vol. CT-17, 1971, pp. 192-194.