University of New Mexico
UNM Digital Repository

4-1-1975

Two Design Techniques for Digital Phase Networks

Francis Brophy

Andres C. Salazar

Follow this and additional works at: http://digitalrepository.unm.edu/ece_fsp

Two Design Techniques for Digital Phase Networks

By F. J. BROPHY and A. C. SALAZAR

Two computer-aided algorithms for the design of all-pass digital filters are presented. The first technique is based on a linear programming approach to solving the approximation problem posed by the minimax design of an all-pass digital filter. A new iterative algorithm with stability constraints is offered for direct form design. The second technique implements a gradient search for those quadratic factors of an all-pass transfer function that lead to a locally optimal approximation (in the least-squares sense) of a desired phase function. New initial guess procedures and the parameterization of linear-phase offset enhance the least-squares design procedure. Examples illustrating the result of both procedures are included.

I. INTRODUCTION

The increasing availability of digital signal processors such as those described in Refs. 1 and 2 has generated much interest in the algorithmic design of digital filters. One particular class of recursive digital filters commonly referred to as all-pass digital networks has an important and interesting design problem associated with it. That is, the design objective for this type of filter involves the following transfer function

$$H(z^{-1}) = \frac{\sum_{k=0}^{N} b_{N-k} z^{-k}}{\sum_{k=0}^{N} b_k z^{-k}}.$$

Because of the relationship between numerator and denominator polynomials, the number of degrees of freedom in filter design has been reduced to N from the usual $2N$. Since the magnitude function of $H(z^{-1})$ is precisely 1.0 on the unit circle, the design problem is focused directly on the phase variation of $H(z^{-1})$. The importance of this design problem does not arise from an academic viewpoint.

There are signal processing applications in which an influential factor in signal fidelity is the amount of phase distortion present in
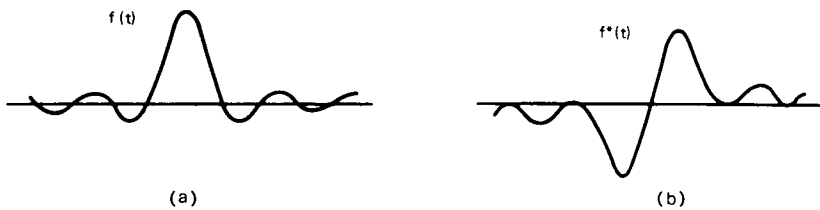
Fig. 1—(a) Original pulse. (b) Phase-distorted pulse.

the medium. The effects of phase distortion in communication systems are illustrated in Refs. 3 and 4. Apart from nonlinear phase equalization applications, all-pass networks can be used to provide a constant phase shift over a specified frequency band or bands. The Hilbert transformer commonly found in bandpass modulation systems is just one example of this application. In constructing phased arrays in radar and seismic research, constant phase shifters are also found to be useful.[5,6] Figure 1 illustrates how a constant phase offset can shape (or distort) the impulse response of a system where $f(t)$ and $f^*(t)$ differ by a constant phase offset of $\pi/2$. A constant phase shift of any amount besides an odd multiple of $\pi/2$ will produce a pulse with a single large lobe. Equalization of this type of distortion is again possible by all-pass networks.

Previous work[7] has addressed the envelope delay design problem. In many cases, this is sufficient but, as seen above, there are applications where the phase function must be treated directly.

Our design techniques are for all-pass structures where the design criteria stem from the phase function directly. The first technique, described in Section II, is a new method for designing all-pass networks using linear programming. This approach allows for fast (at least quadratic), always convergent design of phase networks. For the first time, stability can be treated directly in the design procedure. The second algorithm is based on a gradient search procedure on a least-squares criterion. The basic approach is analogous to those described in Refs. 7 to 9. The all-pass structure reduces the number of variables and simplifies the gradient calculations. In addition to developing the algorithm, we provide initial guess procedures and linear-phase offset parameters that enhance the algorithm. These initial guess procedures are new noniterative filter designs that can serve as excellent all-pass approximations in their own right.

II. A LINEAR PROGRAMMING APPROACH

A need for fast, reliable design of all-pass digital filters has been shown in the previous section. Linear programming techniques have

been found to be useful in rational function approximations[10,11] and have been applied to the magnitude-only design of digital filters.[12,13] Here we show how the all-pass structure in digital filters can be transformed into a problem that can be handled by linear programming techniques also. As we shall see, the rational function differs from the magnitude-only case. Most importantly, this technique allows the question of stability to be handled directly in the design procedure. Other techniques that consider the phase or envelope delay variation of the digital filter (see Refs. 7 and 9 and Section III of this paper) deal with stability with a more heuristic approach.

To develop the linear programming design method, we first recall that the all-pass transfer function is

$$H(z^{-1}) = \frac{P(z^{-1})}{Q(z^{-1})} = \frac{b_N + b_{N-1}z^{-1} + b_{N-2}z^{-2} + \cdots + b_0 z^{-N}}{b_0 + b_1 z^{-1} + \cdots + b_N z^{-N}} \quad (1a)$$

$$\frac{P(z^{-1})}{Q(z^{-1})} = \frac{z^{-N}(b_N z^N + b_{N-1}z^{N-1} + \cdots + b_0)}{(b_N z^{-N} + b_{N-1}z^{-N+1} + \cdots + b_0)}. \quad (1b)$$

Hence, the phase function of (1) on the unit circle is

$$\phi\left(z^N \frac{P(z^{-1})}{Q(z^{-1})}\right)\bigg|_{|z|=1} = -2\phi[Q(z^{-1})]\big|_{|z|=1}. \quad (2)$$

From (2) we note that the phase variation of $H(z^{-1})$ is equivalent (modulo a constant multiplier and an N sample delay term) to the phase of $Q(z^{-1})$. Henceforth, we address the problem of synthesizing $Q(z^{-1})$. The phase variation of $Q(z^{-1})$ on the unit circle is

$$\phi[Q(z^{-1})]\big|_{|z|=1} = \tan^{-1} \frac{-\sum_{k=1}^{N} b_k \sin 2\pi kf}{\sum_{k=0}^{N} b_k \cos 2\pi kf}$$

$$\tan \phi[Q(z^{-1})]\big|_{|z|=1} = \mathrm{Imag}\,[Q(e^{-j2\pi f})]/\mathrm{Real}\,[Q(e^{-j2\pi f})]. \quad (3)$$

Further,

$$\tan \phi[Q(e^{-j2\pi f})] = \frac{-\sum_{k=1}^{N} b_k \sin 2\pi kf}{\sum_{k=0}^{N} b_k \cos 2\pi kf} \triangleq \frac{R(f)}{S(f)}. \quad (4)$$

Our design criterion is chosen to be

$$\min_{\{b_k\}} \max_{n} \left| D(f_n) - \frac{R(f_n)}{S(f_n)} \right| \qquad n = 0, 1, 2, \cdots, M,$$

where $D(f)$ is the tangent desired phase function and M is a number

of frequency points* ($\gg N$) chosen to ensure adequate approximation over a subinterval of $|f| \leqq \frac{1}{2}$, namely, $0 < f_0 < f_1 \cdots < f_M < \frac{1}{2}$. We recall that the desired phase function has been scaled down by $-\frac{1}{2}$ because of the factor of -2 appearing in (2) and will have a delay of N samples inherent in its design by the z^{-N} factor of (1b). It is important to note here that the norm is applied to the tangent of the desired phase function instead of the desired phase function itself.[†]

If we prevent $S(f)$ from assuming the value zero, we seek the minimum value of Δ,

$$|D(f_n)S(f_n) - R(f_n)| \leqq \Delta S(f_n). \tag{5}$$

Using the differential correction idea of Ref. 10, we expand the right-hand side of (5) in an iterative form:

$$\Delta S(f_n) \approx \Delta_k S_k(f_n) + (\Delta - \Delta_k) S_k(f_n) + [S(f_n) - S_k(f_n)]\Delta_k. \tag{6}$$

The intention is to iterate toward those values of $\{b_j\}$ that minimize Δ. The subscript k indicates the kth iteration. We then have, from (5) and (6),

$$|D(f_n)S(f_n) - R(f_n)| - \Delta_k S(f_n) - (\Delta - \Delta_k)S_k(f_n) \leqq 0,$$

which translates into a familiar pair of equations[10]

$$[D(f_n) + \Delta_k]S(f_n) - R(f_n) + (\Delta - \Delta_k)S_k(f_n) \geqq 0 \tag{7}$$

$$[-D(f_n) + \Delta_k]S(f_n) + R(f_n) + (\Delta - \Delta_k)S_k(f_n) \geqq 0. \tag{8}$$

Substituting the series forms for $R(f_n)$ and $S(f_n)$, we have

$$\sum_{j=1}^{N} \{[D(f_n) + \Delta_k] \cos 2\pi j f_n + \sin 2\pi j f_n\}b_j + (\Delta - \Delta_k)S_k(f_n)$$
$$\geqq -D(f_n) - \Delta_k \tag{9}$$

$$\sum_{j=1}^{N} \{[-D(f_n) + \Delta_k] \cos 2\pi j f_n - \sin 2\pi j f_n\}b_j + (\Delta - \Delta_k)S_k(f_n)$$
$$\geqq D(f_n) - \Delta_k, \tag{10}$$

where $b_0 \equiv 1$ is the normalization made. We have in (9) and (10) an over-determined set of $2M$ equations in $N + 1$ variables. The objective is to minimize Δ, one of the variables. It would seem that the condition $S(f_n) > 0$ would be necessary to solve (9) and (10). But the phe-

---

* An extension into a weighted criterion can be handled, but is suppressed in this presentation. M was chosen to be in the range $4N$ (N large) $\leqq M \leqq 10N$ (N small) in our implementation of the algorithm.
† Therefore, the nonlinear nature of the tangent transformation may inhibit designs in the neighborhood of π.

nomenon experienced in Ref. 10 occurs here also. That is, if $S_k(f_n) > 0$, $0 \leqq n \leqq M$, then $S_{k+1}(f_n) > 0$, $0 \leqq n \leqq M$, also.

Standard linear programming techniques can now be used on (9) and (10) to iterate toward a minimum Δ. However, no restriction has been made on the locations of the zeros of $Q(z^{-1})$. Now there exist sufficient conditions for stability that can be written as linear constraints. We have looked at two of these, e.g., a restriction that $b_1, b_2, \cdots, b_N$ of (1) form a monotonic sequence[14] or the restriction that the sum $\sum_{k=0}^{N} b_k \cos 2\pi kf > 0$, $\forall f \in [0, \frac{1}{2}]$.[15] (The formulation of the linear programming problem gives us this condition on the subset of $[0, \frac{1}{2}]$ over which we are approximating.) For an example of a filter designed using this technique and the latter constraint to assure stability, refer to Fig. 2. Curve B is the sixth-order approximation to Curve A (only approximated over $[0.075, 0.425]$).

However, the filter designer may decide that these types of constraints are too restrictive for his particular applications. Nonlinear stability constraints, such as those found in Ref. 14, Chapter 3, can be included via the cutting planes algorithm,[16] but this may require excessive computation times. Another suggestion involves interrupting the standard simplex method for solving the linear programming problem after each iteration. We may then further constrain the **b** vector used in the next basic feasible solution to a choice (i.e., some
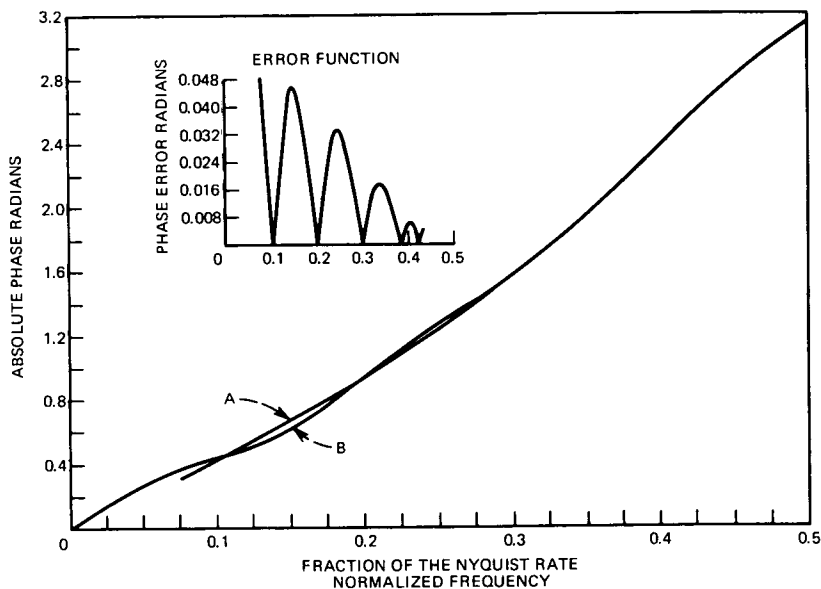

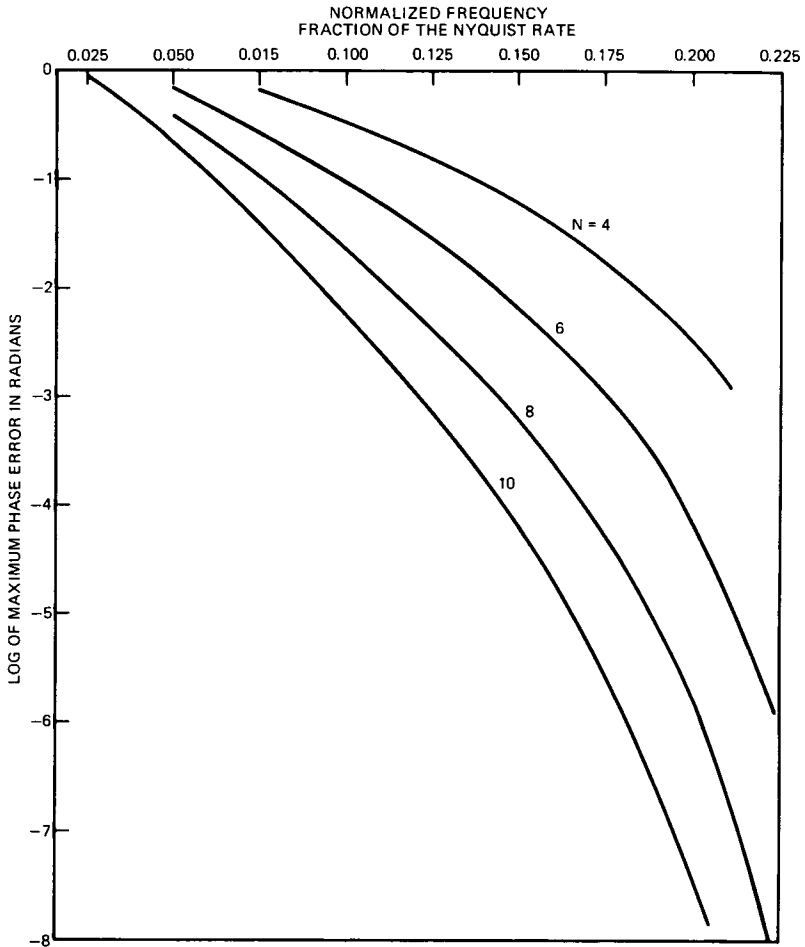Fig. 2—Sixth-order approximation using linear programming method.

Fig. 3—Phase error vs bandwidth for various orders of Hilbert transformers.

"maximum") from among those vectors that would result in a stable filter in addition to the normal improvement of an object function.

Using the standard formulation of the problem with no additional constraints or techniques necessary to assure stability, we were able to design many Hilbert transformer filters.[*] Figure 3 shows the relationship between the maximum error (recall that the tangent of the desired function is approximated) and a bandwidth (the filters were

---

[*] FIR designs of Hilbert transformers are well documented (see Ref. 17). There, 90-degree phase is guaranteed, and the magnitude of 1.0 is approximated.
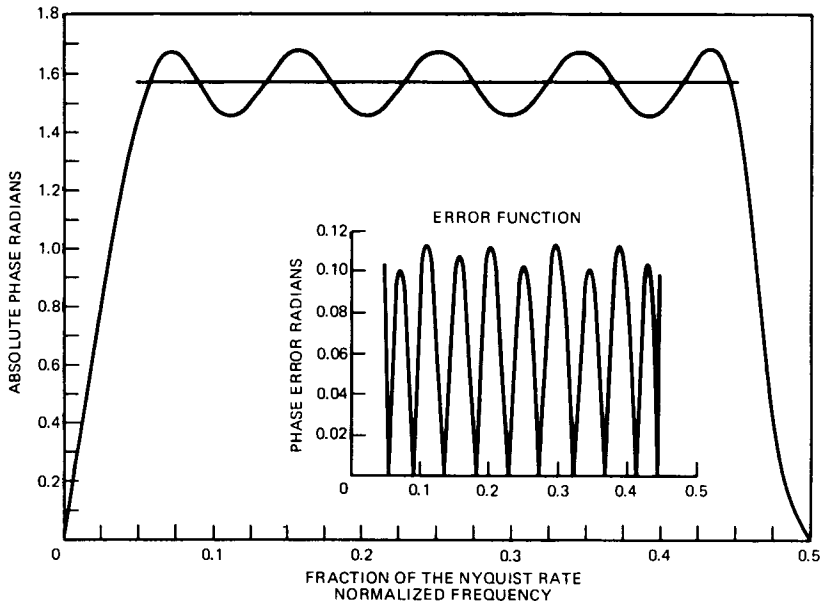
Fig. 4—Tenth-order Hilbert transformer.

designed* over $[f, 0.5 - f]$, $f = 0.075$, 0.05, $\cdots$, 0.225) for various orders of filters $N = 4, 6, 8, 10$. The log of the maximum error is given in the figure.

The minimax approximation formulated here is performed on the tangent of the desired phase and not on the desired phase itself. For very good approximations, however, no penalties seem apparent. We have briefly looked at methods to design minimax phase approximations based on the algorithm we have presented here. Our conclusions are that a two-stage design algorithm is required to iteratively locate a proper weight function that will "prewarp" the "tangent" design so that the weighted "tangent" design is minimax and the phase approximation is itself equiripple.

Figure 4 illustrates the effect of the tangent transformation in this design procedure. In this figure, we see the phase of the resultant design (and its error function). This is a 10th order approximation to a 90-degree phase shift over $[0.05, 0.45]$. While the design guarantees a minimax solution (equiripple) to the tangent, we can see that the resultant phase approximation is not exactly equiripple.† We have not

---

* Each design only required a few (e.g., 5) iterations.
† We can see from Fig. 2 that the effect that the tangent transformation has on the error curve also depends on the values of the desired function.

implemented an algorithm to find the minimax solution to the phase, since, for our needs, the improvement in the phase approximation from the method outlined here did not seem to justify the use of a modified algorithm.

III. A GRADIENT SEARCH TECHNIQUE FOR LEAST-SQUARES DESIGN

The next design algorithm we describe involves the computation of the gradient vector relative to the set of coefficients in a product of quadratic factors. The transfer function of an all-pass digital filter, expressed as a product of second-order sections, is:

$$H(z^{-1}) = \prod_{i=1}^{M} \left(\frac{\beta_i + \alpha_i z^{-1} + z^{-2}}{1 + \alpha_i z^{-1} + \beta_i z^{-2}} \right).$$ (11)

The least-squares form

$$E = \sum_{k=1}^{L} [D(f_k) - \text{Ang } H(e^{-j2\pi f_k})]^2 w(f_k)$$ (12)

will be used as a measure of the approximation error from the desired function $D(f)$ on the set of frequency samples $\{f_k\}_1^L$. Here, $w(f)$ denotes a nonnegative weighting function. A. G. Deczky has also considered gradient techniques applied to the least-square design of all-pass digital filters.[7] In that paper, the emphasis was on envelope delay design. However, as shown in Section I, there are applications where envelope delay approximations are not adequate. Specifically, there are cases where phase distortion (e.g., constant phase offset) must be eliminated with an all-pass structure.

Our design algorithm stems from familiarity with Ref. 8, which considers magnitude-only designs. With the least-squares criterion, the cascade second-order section form can be used. The advantage is that coefficient accuracy problems are minimized. As an alternative to the linear programming approach considered in Section II, this least-squares approach also enables one to more easily control the linear-phase offset permitted in the design. However, a disadvantage of the least-squares approach is that stability of the designed filter cannot be handled directly. Stability is obtained by confining the gradient movement to within the unit circle. This constraint may increase the likelihood of reaching an unsatisfactory local optimum. As we see later, there are initial guess procedures that provide excellent approximations to the desired phase function which, through the design algorithm, increase the likelihood of reaching a satisfactory local optimum.

3.1 Gradient calculations

We find the entries of the gradient vector are

$$\frac{\partial E}{\partial \alpha_i} = -2 \sum_{k=1}^{L} [D(f_k) - \text{Ang } H(e^{-j2\pi f_k})] w(f_k) \frac{\partial \text{ Ang } H(e^{-j2\pi f_k})}{\partial \alpha_i} \quad (13)$$

$$\frac{\partial E}{\partial \beta_i} = -2 \sum_{k=1}^{L} [D(f_k) - \text{Ang } H(e^{j2\pi f_k})] w(f_k) \frac{\partial \text{ Ang } H(e^{-j2\pi f_k})}{\partial \beta_i}. \quad (14)$$

Here we define $\phi(f) = \text{Ang } H(e^{-j2\pi f}) = \tan^{-1} I(f)/R(f)$, where $I(f) = \text{Imag } H(e^{-j2\pi f})$ and $R(f) = \text{Real } H(e^{-j2\pi f})$. We seek

$$\frac{\partial \phi(f)}{\partial \alpha} = R(f)I_\alpha'(f) - I(f)R_\alpha'(f)$$

$$\frac{\partial \phi(f)}{\partial \beta} = R(f)I_\beta'(f) - I(f)R_\beta'(f),$$

where prime (′) denotes the partial derivative relative to the subscript. After some algebra, we find

$$\frac{\partial \phi}{\partial \alpha_i} = 2(1 - \beta_i)F_i(f) \sin 2\pi f \quad i \leqq i \leqq M \quad (15)$$

$$\frac{\partial \phi}{\partial \beta_i} = 2F_i(f)(\sin 4\pi f + \alpha_i \sin 2\pi f) \quad 1 \leqq i \leqq M, \quad (16)$$

where $F_i(f) = |1 + \alpha_i e^{-j2\pi f} + \beta_i e^{-j4\pi f}|^{-2}$. Finally, (13) and (14) can be simplified for $1 \leqq i \leqq M$ to

$$\frac{\partial E}{\partial \alpha_i} = -4(1 - \beta_i) \sum_k [D(f_k) - \phi(f_k)]F_i(f_k)w(f_k) \sin 2\pi f_k \quad (17)$$

$$\frac{\partial E}{\partial \beta_i} = -4 \sum_k [D(f_k) - \phi(f_k)]F_i(f_k)w(f_k)(\sin 4\pi f_k + \alpha_i \sin 2\pi f_k). \quad (18)$$

The minimization of E in (12) then proceeds with an iterative algorithm that is based on the formula

$$\mathbf{c}^{(n)} = \mathbf{c}^{(n-1)} - \epsilon_{n-1} A_n (\nabla E)_{n-1}, \quad (19)$$

where $\mathbf{c}^{(n)}$ is the coefficient vector $(\alpha_1, \beta_1, \alpha_2, \beta_2, \cdots, \alpha_M, \beta_M)$ at the nth iteration, ϵ_n is the nth step size in the coefficient adjustment, A_n is a positive definite matrix at the nth step ($\equiv I$ in the case of the steepest descent algorithm) and $(\nabla E)_n$ is the gradient vector whose entries are given by (17), (18) at the nth iteration (we use the Fletcher-

Powell algorithm). An initial guess procedure is required to start an iterative algorithm such as that of (19).

3.2 Initial guess procedures for all-pass networks

Convergence to a local minimum at which the approximation to a desired phase function is satisfactory can be made easier if a good initial guess is provided to $c^{(0)}$ of (19). A desired feature of an initial guess procedure is that it be simple in nature. After all, excessive computation and effort should not be expected in simply starting a complex algorithm. In this section, we consider two procedures in which only a linear set of equations need be solved to obtain initial values for $\{b_k\}_{k=0}^{N}$ of (1). The value of having several initial guess procedures is that the designer may want to exercise his algorithm from multiple starting points to choose the best from a set of local optima. The following initial guess procedures operate on the direct form of $H(z^{-1})$ (1) which can be factored to the cascade form (11).

3.2.1 Tangent approximation by Gauss' method

From (4) in Section II we know that a desired phase function can be approximated by considering a monotonic function of the phase, namely the tangent. Hence,

$$\tan \phi(f) = - \frac{\sum\limits_{k=1}^{N} b_k \sin 2\pi kf}{\sum\limits_{k=0}^{N} b_k \cos 2\pi kf} \tag{20}$$

is the approximating function of the tangent of half the desired phase. If we require the estimates of the desired phase tangent $[\tan \phi_d(f)]$ to be "good" at a number of frequencies, we then have the following equations:

$$\tan \phi_d(f_0) \sum_{k=0}^{N} b_k \cos 2\pi f_0 - \sum_{k=1}^{N} b_k \sin 2\pi kf_0 = r_0$$

$$\tan \phi_d(f_1) \sum_{k=0}^{N} b_k \cos 2\pi kf_1 - \sum_{k=1}^{N} b_k \sin 2\pi kf_1 = r_1$$

$$\vdots$$

$$\tan \phi_d(f_M) \sum_{k=0}^{N} b_k \cos 2\pi kf_M - \sum_{k=1}^{N} b_k \sin 2\pi kf_M = r_M. \tag{21}$$

If $\{r_n\}_0^M$ were all zero, then the approximation would be exact. The objective then is to minimize $\sum_{n=0}^{M} r_n^2$, where $M > N$. This problem is a least-squares minimization problem for which the solution is

derived from solving a set of normal equations:

$$
\begin{bmatrix}
(\mathbf{a}_1, \mathbf{a}_1)(\mathbf{a}_1, \mathbf{a}_2)\cdots(\mathbf{a}_1, \mathbf{a}_N) \\
(\mathbf{a}_2, \mathbf{a}_1)(\mathbf{a}_2, \mathbf{a}_2)\cdots(\mathbf{a}_2, \mathbf{a}_N) \\
\vdots \qquad\qquad \vdots \\
(\mathbf{a}_N, \mathbf{a}_1) \qquad\qquad (\mathbf{a}_N, \mathbf{a}_N)
\end{bmatrix}
\begin{bmatrix}
b_1 \\ b_2 \\ \vdots \\ b_N
\end{bmatrix}
=
\begin{bmatrix}
(\mathbf{a}_1, \mathbf{d}) \\
(\mathbf{a}_2, \mathbf{d}) \\
\vdots \\
(\mathbf{a}_N, \mathbf{d})
\end{bmatrix}
\tag{22}
$$

or

$$A\mathbf{b} = \mathbf{e},$$

where

$$
\begin{aligned}
\mathbf{a}_n = (&\tan \phi_d(f_0) \cos 2\pi n f_0 - \sin 2\pi n f_0, \\
&\cdot \tan \phi_d(f_1) \cos 2\pi n f_1 - \sin 2\pi n f_1, \cdots, \\
&\cdot \tan \phi_d(f_M) \cos 2\pi n f_M - \sin 2\pi n f_M) \quad n = 1, 2, \cdots, N
\end{aligned}
$$

and

$$
\begin{aligned}
\mathbf{d} &= [- \tan \phi_d(f_0), - \tan \phi_d(f_1), \cdots, - \tan \phi_d(f_M)] \\
\mathbf{b} &= (b_1, b_2, \cdots, b_N) \quad \text{and} \quad \mathbf{e} = [(\mathbf{a}_1, \mathbf{d}), \cdots, (\mathbf{a}_N, \mathbf{d})].
\end{aligned}
$$

Let

$$
\rho_{\max} = \max_{0 \leq n \leq M} \{|r_n^*|\} \quad \text{and} \quad \bar{\rho} = \sqrt{\frac{(\mathbf{r}^*, \mathbf{r}^*)}{M}},
$$

where $\mathbf{r}^* = (r_0^*, r_1^*, \cdots, r_M^*)$, the residual values after the least-squares approximation. If $\rho_{\max} - \bar{\rho}$ is large (it is always positive), then a Chebyshev approximation may be desirable.[18]

3.2.2 Tangent approximation in Chebyshev sense

It is well known that the minimax solution to (21) requires solving an appropriate subsystem of $N + 1$ equations. Further, the minimax solution of $N + 1$ inconsistent equations can be effected by examining the least-squares solution to the same set of equations and proceeding to solve a set of N linear equations.[19]

For our purposes here, an effective method of obtaining an initial guess for the iterative procedure implied by (19) is that of choosing $M = N$ and obtaining the minimax solution to (21). This can be done by solving (22) for $\mathbf{b} = (b_1, b_2, \cdots, b_N)$ and then evaluating (21) for the residuals $r_0^*, r_1^*, \cdots, r_N^*$. The minimax solution to (21) is then given by the linear set of equations

$$B\mathbf{b} = \boldsymbol{\sigma}, \tag{23}$$

where $B = (b_{jk})$, $N + 1$ by N matrix with $b_{jk} = \tan \phi_d(f_j) \cos 2\pi k f_j - \sin 2\pi k f_j$, $\boldsymbol{\sigma} = \epsilon[\text{sign}\,(r_0), \text{sign}\,(r_1), \cdots, \text{sign}\,(r_N)]$, and

$$\epsilon = \sum_{k=0}^{N} r_k^{*2} \Big/ \sum_{k=0}^{N} |r_k^*|.$$

It may be noted that only N of $N + 1$ equations are used in the solution of (23).

3.2.3 Discussion

It should be noted that no constraint has been made on the initial guess procedures of A or B to ensure that the resulting digital filter is stable. In fact, if $\sum_{k=0}^{N} b_k \cos k2\pi f$ should ever change sign in $|f| \leqq \frac{1}{2}$ or at least in the subinterval of approximation $[f_0, f_M]$, then the transition from (20) to (21) is not really valid since a division by zero is implied. Should $\sum_{k=0}^{N} b_k \cos k2\pi f$ be strictly positive over $|f| \leqq \frac{1}{2}$, then stability results.[5] (The interesting point is that stability can result even if the cosine series does change sign in $|f| \leqq \frac{1}{2}$). However, the point to remember is that the resulting initial guess may be unstable. In our experience, we have not encountered any serious problems using these initial guess procedures.

We must further remark that the inherent N sample delay present in these approximations [see (2)] could present a problem when designing filters with $M \neq N$ sample delays. However, we feel, intuitively, that since some delay is unavoidable, a delay of the order of the filter will not, for most applications, be overly restrictive.

The last point to consider is that the initial guess procedure of Sections 3.2.1 and 3.2.2 obtains a direct form estimate of the digital filter coefficients. What is really required for $c^{(0)}$ of (19) are quadratic factors. We remark that we make the transition from the direct form estimate of (20) to quadratic factors by using a Bairstow quadratic factorization routine.

3.3 Some considerations for least-squares design

Often the engineering systems requirement of a digital filter can tolerate a linear-phase offset. While the systems engineer cannot always adapt to an arbitrary delay, there will usually be a range of delays permissible to him. How then can a designer incorporate these relaxations into the design mechanism? One technique for doing this is to add an acceptable delay to the desired function to create a new desired function and proceed from there. By designing filters for each of the permissible delays, one can choose from among the delays and their associated errors to decide which filter to implement.

In Fig. 5 we can see the error function of a sixth-order filter[*] (B)

---

[*] We have not tested the limit of the order of filters that can be designed by this method, but we have obtained a twentieth-order approximation (20-sample delay) to the desired function in this example. Quality initial guess procedures help us do this without excessive computation times.
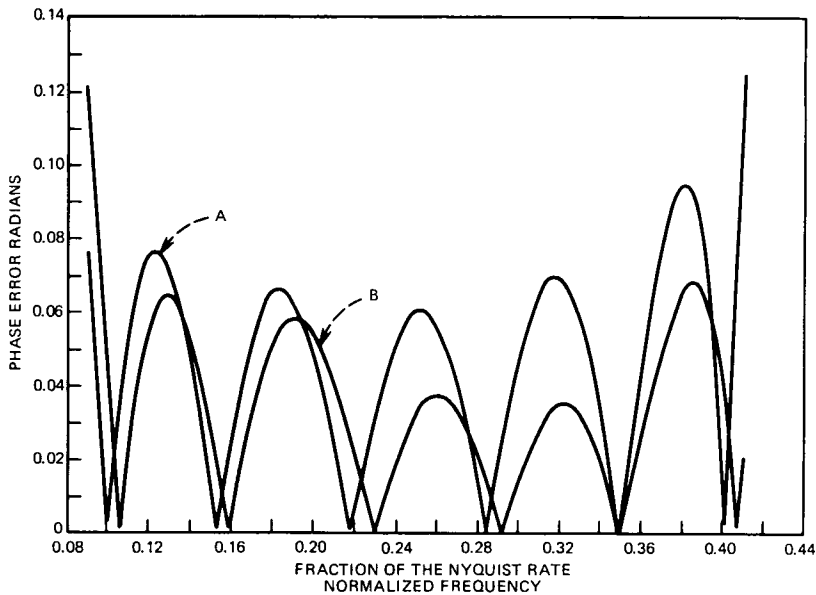
Fig. 5—Error curves for initial and final sixth-order Hilbert transformer designs.

designed for a delay of six samples. The desired function is a 90-degree phase shifting filter with the approximation having weight 1 on [0.08, 0.41] and 0 otherwise. Note the quality of the initial approximation (A) using the first initial guess technique outlined in Section 3.2. Of course, the disadvantage of presetting the delay is obvious; the choice of the optimal delay from those that are acceptable is not automatic but requires a separate design for each delay. However, eq. (12) can be expanded to include delay as parameter A

$$E = \sum_k [D(f_k) - \text{Ang } H(e^{-j2\pi f_k}) - A 2\pi f_k]^2 w(f_k).$$

An optimal A can be found analytically at each step in the gradient search and at convergence A will represent the amount of delay which, in conjunction with the filter, produces the best design.[*] Of course, we cannot expect that this delay will represent an integral number of samples or even a delay that the designer can tolerate. Figure 6 shows a desired function (A) (this curve is only shown where the weight of the approximation is nonzero) and its fourth-order approximation (B),

---

[*] It is possible to include a constant phase angle as parameter B similar to the A used here. In such a case, our procedure becomes an envelope delay design technique.
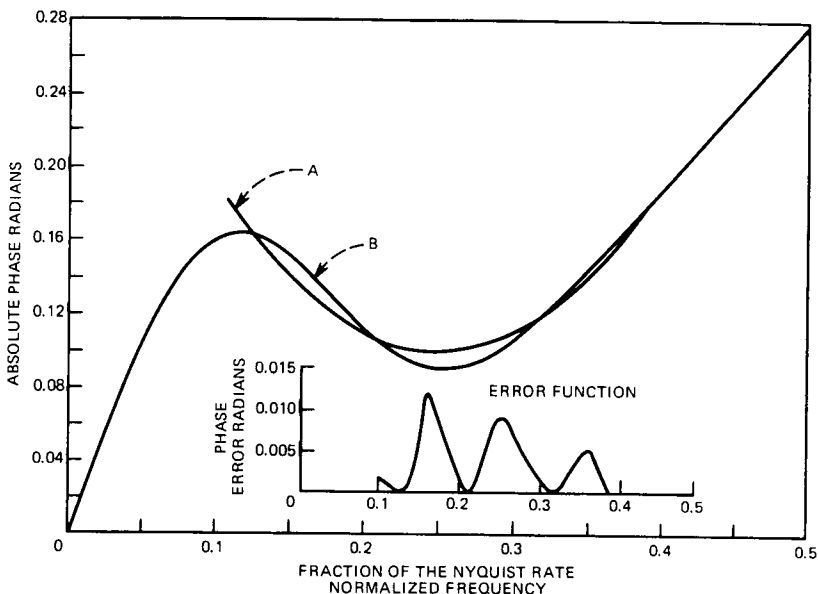
Fig. 6—Fourth-order approximation to desired shape (A).

which is by solving for optimal A. Allowing for arbitrary delay, the algorithm obtained this optimal design with a 4.1-sample delay.

We can offer an heuristic solution to guarantee integer delays in an automatic fashion; namely, at each step (that is, at each calculation of A), the nearest acceptable delay* is used to replace A in the algorithm. This, of course, places a serious strain on optimality, although it does permit an automatic design procedure.

As a footnote to this algorithm, we remark that there is a tendency, when working with procedures for designing filters in the cascade form, to use a previous optimal design of order n as the initial starting point in the design of filters of order $n + 2$. In the case of magnitude-only design, this is easily implemented since the appended second-order section can be initialized with magnitude 1. However, in the all-pass presentation there does not exist any second-order section that can be added which does not distort the overall phase when using a previous optimal design of order n to provide the initial guess for a design of order $n + 2$. And so the user of this algorithm must consider the effect of the appended second-order section if he does not want to obviate the value of a previous design toward providing an initial guess.

---

* Nearest in the sense of greatest reduction of (12); "acceptable" here means "integer."

REFERENCES

1. G. D. Hornbuckle and E. I. Ancona, "The LX-1 Microprocessor and Its Applications to Real Time Signal Processing," IEEE Transactions on Computers, C-19, August 1970.
2. B. Gold, I. L. Lebow, P. G. McHugh, and C. M. Rader, "The FDP, a Fast Programmable Signal Processor," IEEE Transactions on Computers, C-20, January 1971.
3. R. W. Lucky, J. Salz, and E. J. Weldon, Jr., Principles of Data Communication, New York: McGraw-Hill, 1968, pp. 190–192.
4. F. D. Sunde, Communication Systems Engineering Theory, New York: John Wiley, 1969, Chaps. 6–7, notably, pp. 244–246, pp. 229–232, and Figs. 6.4 and 6.6.
5. E. A. Robinson, Statistical Communication and Detection, New York: Hafner, 1967, pp. 325–326.
6. L. V. Blake, Antennas, New York: John Wiley, 1965, pp. 250–252.
7. A. G. Deczky, "Synthesis of Recursive Digital Filters Using the Minimum p-Error Criterion," IEEE Trans. on Audio and Elect., October 1972.
8. K. Steiglitz, "Computer-Aided Design of Recursive Digital Filters," IEEE Transactions on Audio and Electroacoustics, AU-18, June 1970.
9. R. E. King and G. W. Condon, "Frequency Domain Synthesis of a Class of Optimum Recursive Digital Filters," Int. J. on Control, 1973, No. 3.
10. I. Barrodale, M. J. D. Powell, and F. D. K. Roberts, "The Differential Correction Algorithm for Rational L_∞-Approximation," SIAM J. of Num. Math., 9, September 1972.
11. H. L. Loeb, "Algorithms for Chebyshev Approximations Using the Ratios of Linear Forms," J. Soc. of Ind. Appl. Math., September 1960.
12. F. Brophy and A. Salazar, "Synthesis of Spectrum Shaping Digital Filters of Recursive Design," IEEE Trans. on Circuits and Systems, March 1975.
13. L. Rabiner, N. Y. Graham, and H. D. Helms, "Linear Programming Design of IIR Digital Filters with Arbitrary Magnitude Function," IEEE Trans. on Acoustics, Speech, Signal Proc., April 1974.
14. E. I. Jury, Theory and Applications of the z-transform Method, New York: John Wiley, 1964, p. 116.
15. Ref. 5, p. 188.
16. B. S. Gottfried and J. Weisman, Introduction to Optimization Theory, Englewood Cliffs, N. J.: Prentice-Hall, 1973, p. 255.
17. L. R. Rabiner and R. W. Schafer, "On the Behavior of Minimax FIR Digital Hilbert Transformers," B.S.T.J., 53, No. 2 (February 1974), pp. 363–390.
18. E. L. Stiefel, An Introduction to Numerical Mathematics, New York: Academic Press, 1963, pp. 54–55.
19. E. W. Cheney, Introduction to Approximation Theory, New York: McGraw-Hill, 1966, pp. 36–41.