Copyright © 1976 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 55, No. 4, April 1976
Printed in U.S.A.

Digital Single-Tone Generator-Detectors

By R. P. KURSHAN and B. GOPINATH

(Manuscript received December 5, 1975)

A class of digital, linear generator-detectors, based upon cyclotomic polynomials, which have simple implementation and operate without roundoff errors, is proposed. It is shown how these filters are optimal among all linear generator-detectors which have no roundoff required in the feedback loop. The complexity of various cyclotomic filters are compared. These filters in general require far fewer binary adds/s than conventional second-order filters used for the same purpose.

I. INTRODUCTION

Devices for pure tone generation and detection have widespread applications. The most notable examples are *Touch-Tone*® signaling, frequency shift keying (FSK), and multifrequency (MF) signaling. Associated with such devices are problems of stability and predictability, which in practice are dealt with on an individual basis, using techniques peculiar to the particular application. When these devices are realized digitally, the above problems are manifest from errors due to operational roundoff.

Generally, tones for signaling are analog signals of the form $A \sin \omega t$ (A is the amplitude, $2\pi/\omega$ is the period, and $\omega/2\pi$ is the frequency). Devices that generate these tones are usually oscillators of various kinds. Because of the requirement of structural stability, in practice these devices are limit cycle oscillators. These are simulations and realizations in hardware of nonlinear differential equations that have limit cycles. Because of the complexity of these equations, the amplitude and frequency are not easily predicted from given values of resistors and capacitors in the network.

For detection of these tones, linear analog filters are frequently used. These are also used as generators, when the duration of the signal is not too long compared to the period. However, passive linear analog oscillators require inductors which are bulky, and the frequency and amplitude of these oscillators can vary with changes in value of the inductors and capacitors due to environmental conditions. Active linear oscillators using RC elements are used in many applications. However, they also generally need some form of limiting and end up being nonlinear devices, thus usually preventing them from being used as receivers.

Digital oscillators, on the other hand, are almost insensitive to changing parameter values and produce stable repeatable waveforms. However, in the mechanization of these oscillators (which are usually based upon second-order linear equations), roundoff in multiplication and addition produce errors in the feedback that lead to limit cycles and can significantly impair the signal quality. Also, when such linear digital devices are used as receivers, the precision required for satisfactory performance goes up quite rapidly with increasing Q. Although the effects of this can be satisfactorily controlled in certain specific applications (see, for example, Ref. 1), the difficulties, in general, cannot be ameliorated except by increasing the accuracy of computations.[2]

In this paper, we present a class of digital filters that operate without arithmetic roundoff. These filters are linear, and can be used both as oscillators for signal generation and also as receivers for signal detection. The feedback loop of each filter is constructed in such a way as to eliminate the possibility of roundoff or truncation errors, thus insuring perfect arithmetic. This entirely eliminates the problem of limit cycles. The filters presented, when used as generators, produce quantized values of $A \sin \omega t$ of arbitrary accuracy. Implementation of these filters as receivers involves first sampling an analog input signal to produce a digital input into the filter. The filter is designed to resonate for a particular input frequency, thus enabling detection.

The means by which arithmetic errors are eliminated in the feedback loop involves constraining all feedback coefficients to be integers (a constraint which turns out to be necessary to guarantee perfect arithmetic in any digital filter). Thus multiplication by these coefficients can be performed as additions, simplifying implementation.

The behavior of the feedback loop of this filter is modeled by a linear recursion whose characteristic polynomial is a cyclotomic polynomial. In recognition of this, we call the filter consisting of the feedback loop alone a "cyclotomic filter." It will be demonstrated that the only way to ensure perfect arithmetic with no limit on the period of operation (and thus avoid limit cycles) in a filter modeled by a linear recursion (i.e., a linear digital filter) is to constrain the feedback coefficients to be integers. Furthermore, it will be shown that, with this constraint on the feedback coefficients and also subject to minimizing memory and eliminating as many resonant harmonics as possible, the cyclotomic filter is uniquely optimal among all digital linear filters, both for the purpose of tone generation and the purpose of tone detection.

In subsequent sections of this paper, it is demonstrated how a weighting function can be applied externally to the cyclotomic filter to drastically reduce the impact of those higher-order resonances that remain. This is applied also to determine those impulse responses which have a small number of integer levels and lack higher-order harmonics. All the cyclotomic filters of practical significance, along with their associated weighting functions and impulse responses, are examined.

In Ref. 3, a specific proposal is described for the *Touch-Tone* receiver (and tone generator), utilizing eight cyclotomic filters.

II. CYCLOTOMIC FILTERS

The purpose of this filter, as discussed in Section I, is to generate or detect a single pure tone $u(t) = A \sin (2\pi ft + \varphi)$ of frequency f. Digital implementation involves realizing a discrete time filter with k stages of memory (see Fig. 1), which is described recursively in terms of an input sequence u_n as

$$x_n = \sum_{i=1}^{k} a_i x_{n-i} + u_n. \tag{1}$$

The numbers $a_i (i = 1, \cdots, k)$ are the feedback coefficients of the filter. The filter is driven by a clock with the time interval τ between pulses. In tone generation, the filter must satisfy

$$x_n = u(n\tau), \tag{2}$$

at least for some initial conditions $x_0, \cdots, x_{k-1}$. When used as a receiver, the analog input $u(t)$ is sampled, producing a discrete input $u_n = A \sin (2\pi fn\tau + \varphi)$; the filter (1) must distinguish between the desired frequency f_0 and all other frequencies in a band containing f_0.

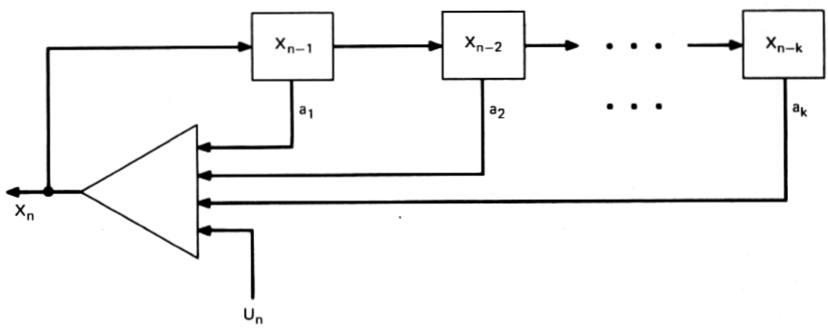

Fig. 1—Recursive filter in k stages of memory.

Specifically, it must satisfy the resonance property

$$\limsup |x_n| = \infty, \tag{3}$$

when $f = f_0$, and in a sufficiently large band B including f_0 there must be no other such resonances. Then $|x_n|$ will be uniformly bounded in B in the complement of any small interval δ about f_0, say, $|x_n| \leqq m(\delta)$ for all $f \in B$, $f \notin \delta$, for all n. A threshold detector can thus detect in a finite amount of time $N\tau$, the presence (or absence) within B of an input frequency f_0 (with error $\pm\frac{1}{2}|\delta|$). It does this by comparing the gain $\sup_{n \leqq N} |x_n|$ with the bound $m(\delta)$; if $\sup |x_n| > m(\delta)$, then $f \in \delta$; otherwise it is not. Of course, the smaller the allowable error δ, the larger N must be.

To know precisely when an input u_n will resonate with respect to this filter, we first observe that the general solution to (1) is

$$x_n = \sum_{j=0}^{n} \sum_{i=1}^{k} b_i \rho_i^{n-j} u_j, \tag{4}$$

where $\rho_1, \cdots, \rho_k$ are the roots (assumed to be distinct) of the characteristic equation

$$\lambda^k - \sum_{i=1}^{k} a_i \lambda^{k-i} = 0 \tag{5}$$

and $b_1, \cdots, b_k$ are complex functions of the roots. [This is derived in (17) below.] If the magnitude of a root of (5) is greater than 1, the filter will be unstable. However, if all roots are inside the unit circle, then (1) will not have any resonance as defined in (3). Hence, in general we will assume that all roots of (5) lie inside or on the unit circle.

Hence, the resonance (3) will occur if and only if the frequency f is such that with $\theta(i) = \arg \rho_i$ either

$$2\pi f\tau \equiv \theta(i) \text{ (modulo } 2\pi) \quad \text{or} \quad 2\pi f\tau \equiv -\theta(i) \text{ (modulo } 2\pi) \tag{6}$$

for some $i = 1, \cdots, k$ with the property that $|\rho_i| = 1$. That is, the detector (see Fig. 2) will give a "yes" response iff (6) is satisfied. As we are trying to detect the presence of the frequency $f = f_0$, let us suppose by way of example that $\theta(1) = 2\pi f_0\tau$ ($|\rho_1| = 1$). Then an input $A \sin (2\pi f_0 t + \varphi)$ would elicit a "yes" response from our receiver. (Any phase shift of $A \sin 2\pi f_0 t$ will not affect the resonance of this signal, as $A \sin (2\pi f_0 t + \varphi) = (A \cos \varphi) \sin 2\pi f_0 t + (A \sin \varphi) \cos 2\pi f_0 t$, and $\cos \varphi$ and $\sin \varphi$ never simultaneously vanish.) However, let us now suppose that also $\theta(2) = 2\pi f_1\tau$ ($|\rho_2| = 1$). Then the receiver would also detect an input frequency f_1 (and would not differentiate between f_0 and f_1). Hence, one would know only whether or not *either* f_0 or f_1 is among the inputs. To positively identify the presence of f_0, one

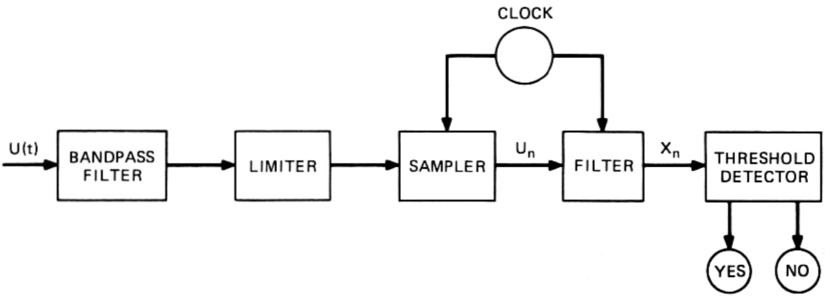

Fig. 2—Structure of a tone detector.

must either insure that f_1 is out of band or use some other means to differentiate between f_0 and f_1.

Similarly, because of (6), the filter cannot distinguish between the frequency f and the frequency $\tau^{-1} - f$, since $2\pi(\tau^{-1} - f)\tau = 2\pi - 2\pi f\tau \equiv -2\pi f\tau$ (modulo 2π). In fact, $2\pi f\tau$ and $-2\pi f\tau$ are the respective arguments of complex conjugates, and thus we see from (6) that no new resonances can occur if the characteristic polynomial (5) is altered to include among its roots any complex conjugates of $\rho_1, \cdots, \rho_k$. We shall use this fact in our determination of a good structure for the recursion (4). When the filter is such that an input of frequency f will resonate, we shall say that the filter resonates (or has a resonance) at f.

Recapitulating, because of (6), whenever the filter has a resonance at a frequency f, it will also necessarily and unavoidably resonate at the frequency $\tau^{-1} - f$. To counter the effect of this in practice, τ must be made sufficiently small so that $\tau^{-1} - f$ is out of band. In keeping with (6), we refer to resonance at the frequency f as "resonance at the root $e^{i2\pi f\tau}$," and resonance at $\tau^{-1} - f$ as "resonance at the conjugate root $e^{-i2\pi f\tau}$" [the roots in question being, of course, roots of (5)].

The remaining resonances described by (6) are those due to aliasing. These also are intrinsic to the system—a consequence of using discrete (rather than continuous) input samples u_n. Indeed, if resonance occurs at a frequency f (or, equivalently, at the root $e^{i2\pi f\tau}$), it will also occur at all the frequencies $f + m\tau^{-1}$ for any integer m, as $2\pi f\tau \equiv 2\pi(f + m\tau^{-1})\tau$ (modulo 2π) or, equivalently,

$$e^{i2\pi f\tau} = \exp[i2\pi(f + m\tau^{-1})\tau].$$

In practice, if conjugate resonances are out of band, resonances due to aliasing will also necessarily be out of band.

Hence, if $\rho_1, \cdots, \rho_m$ are those roots of (5) of modulus 1, the filter will have resonances within the band $[0, \tau^{-1}]$ at the frequencies $\theta(1)/2\pi\tau$, $[2\pi - \theta(1)]/2\pi\tau$, $\theta(2)/2\pi\tau$, $\cdots$, $[2\pi - \theta(m)]/2\pi\tau$. The number of distinct resonances in the interval $[0, (2\tau)^{-1}]$ is m, less the number of roots among $\rho_1, \cdots, \rho_m$ which appear along with their conjugates. This picture is repeated in each successive interval $[n\tau^{-1}, (n+1)\tau^{-1}]$ $(n = \pm 1, \pm 2, \cdots)$.

It should be clear that, in choosing the recursion (1), one desires to have the number of resonances as small as possible—for the purpose of generation, to minimize the number of harmonics that can be produced by perturbations of the initial conditions, and for the purpose of detection, to maximize the band in which the filter can detect a unique signal. Also, of course, one desires to have the memory k (a measure of the complexity of implementation) as small as possible.

Ideally, one would like to have only one resonance, namely at the frequency one is trying to detect or generate. This is possible within the band $[0, \tau^{-1}]$, by using the recursion $x_n = -x_{n-1} + u_n$. However, this resonates at a frequency equal to half the clock frequency τ^{-1} and thus also resonates at the third harmonic $(2\tau)^{-1} + \tau^{-1}$ due to aliasing. As the third harmonic is frequently in band, this recursion is generally not satisfactory.

On the other hand, for some complex number ρ of unit modulus, one could use the recursion $x_n = \rho x_{n-1} + u_n$ which also has memory one. By adjusting τ, one could make the argument of $\rho = \exp(i2\pi f_0\tau)$ small, thus avoiding any resonance up to as high a frequency as desired. However, there are problems with this recursion. First of all, the memory (in implementation) is not really one but two, as the real and imaginary parts of ρ must be handled separately. In fact, as seen before, no new resonances would be introduced by including the complex conjugate $\bar{\rho}$ of ρ to form a recursion of order two. Hence, one does just as well by replacing the characteristic equation $\lambda - \rho = 0$ with $0 = (\lambda - \rho)(\lambda - \bar{\rho}) = \lambda^2 - a\lambda + 1$ (where the real number $a = \rho + \bar{\rho}$). The corresponding recursion replacing $x_n = \rho x_{n-1} + u_n$, also (but now explicitly) of memory two, is $x_n = a x_{n-1} - x_{n-2} + u_n$. This is the recursion after which digital linear filters are customarily modeled. However, as a (ρ) is, in general, not a rational number (gaussian rational*), it must in general be truncated, leading to slight frequency shifts, and multiplication round-off error in the feedback loop of these filters (Fig. 1); this could lead to unwanted limit cycles.[2] To avoid this, a (ρ) is restricted to be rational (gaussian rational). Even for rational numbers, however, truncation error would occur if

* Has rational real and imaginary parts.

the number of bits necessary to represent the number x_n exceeded the word length allowed. In Section V we show that this can be controlled only if a is an integer.

Hence, in the case of the real recursion, we restrict a to be an integer, and the only possibilities are $a = 0, \pm 1, \pm 2$. We have already ruled out $a = -2$ (this gives the square of the characteristic equation of $x_{n+1} = -x_n + u_n$). If $a = 2$, this gives the square of the characteristic equation of $x_{n+1} = x_n + u_n$, which is even worse, as it produces resonance at the second harmonic. The remaining three possibilities for a correspond to cyclotomic polynomials of orders 3, 4, and 6 (as defined subsequently in this section). It will be shown that, by taking a cyclotomic polynomial for the characteristic equation (5), one always obtains the best possible recursion (1) for the given amount of memory.

In general, to have perfect arithmetic (the only means by which to uniformly avoid unwanted limit cycles), it is necessary to constrain the feedback coefficients a_i, $i = 1, \cdots, k$ [see (1)] to be gaussian integers (see Section V). In fact, it will be shown that one can take each $a_i = 0, \pm 1$ so that each tap in the feedback loop involves at most changing the sign. Hence, from here on we restrict ourselves to three cases: the a_i's are gaussian integers, are integers, or are 0, ± 1. In what follows, we will show that the three are, for practical purposes, equivalent.

For the first case, in the recursion corresponding to $\lambda - \rho = 0$ (no complex conjugate), the restriction to integer real and imaginary parts requires $\rho = \pm 1, \pm i$ resulting in less generality than possible, as this corresponds to the recursions of the previous example with $a = \pm 2$ only. In fact, we can generalize this, and say it is always better to include among the roots of (5) all the complex conjugates, and thus to have a recursion (1), all of whose coefficients are real (and hence integers). We will make this explicit in a moment, but let us first indicate the reasoning. First of all, by including the conjugates, no new resonances are introduced (as has already been demonstrated). Second, if among the roots of (5) even one conjugate were missing, the coefficients of (1) would not all be real. In this case, the real and imaginary parts of x_n would have to be considered separately, and one would thus need an effective memory of $2k$. On the other hand, if one multiplies (5) by factors of the form $(\lambda - \bar{\rho})$, one for each root ρ of (5) whose complex conjugate is not also a root of (5), then the resulting polynomial and the corresponding recursion will have real coefficients. The respective degree and memory will thus be raised to no more than $2k$ (the effective memory of the complex recursion). Furthermore, as will be shown in Theorem 1 below, the new polynomial (and recursion) obtained from multiplication by the factors $(\lambda - \bar{\rho})$ will also be guaranteed to have integer coefficients. Thus, we will do at least as well (and, as we have seen above, even better) by restricting all the recursions (1) to have real (and hence integer) coefficients.

Let us now make this explicit. Suppose one has the recursion

$$x_n = \sum_{j=1}^{k} \alpha_j x_{n-j} + u_n,$$

where $\alpha_j (j = 1, \cdots, k)$ are gaussian integers: $\alpha_j = a_j + b_j i$ (a_j and b_j integers, $i = \sqrt{-1}$). Let y_n and z_n be, respectively, the real and imaginary parts of x_n. Then

$$y_n = \sum_{j=1}^{k} (a_j y_{n-j} - b_j z_{n-j}) + u_n,$$

$$z_n = \sum_{j=1}^{k} (b_j y_{n-j} + a_j z_{n-j}).$$

The only feature possibly mitigating in favor of the complex recursion is this: We are constrained to have a_j and b_j be integers. If the new recursion with added roots did not have integer coefficients, then in spite of the other considerations above, one would choose the complex recursion. However, in the following theorem we show this is not possible.

Theorem 1: *Suppose* $F(\lambda)$ *is a polynomial with gaussian integer coefficients, and suppose* $\rho_1, \cdots, \rho_m$ *are those roots of* $F(\lambda)$ *whose complex conjugates are not also roots of* F. *Then* $F(\lambda) \prod_{i=1}^{m} (\lambda - \bar{\rho}_i)$ *has integer coefficients. Furthermore, if* $F(\lambda)$ *has no polynomial with integer coefficients as a factor, then deg* $F = m$.

Proof: Write $F(\lambda) = g(\lambda)h(\lambda)$, where $h(\lambda) = \prod (\lambda - \rho_i)$. Then g has real coefficients. Let $p(\lambda)$ be any irreducible factor of $F(\lambda)$ (considered as a polynomial over the gaussian integers). Suppose p has the root r in common with g and the root s in common with h. Then $\bar{p}$ (the polynomial in λ whose coefficients are the complex conjugates of the coefficients of p) has $\bar{r}$ as a root, and hence $\bar{p}$ must also be a factor of F. But $\bar{s}$ is also a root of $\bar{p}$, whereas $\bar{s}$ is expressly not a root of F. Hence, any irreducible factor of F must be a factor of either g or h. It follows that g has integer coefficients, and h (and thus $\bar{h}$) have gaussian integer coefficients. As $h(\lambda)\bar{h}(\lambda)$ has real, and hence integer, coefficients the theorem follows.

Thus, it is best to take the coefficients of the recursion (1) to be integers. The theorem which follows completely characterizes those recursions.

First, however, a short description of cyclotomic polynomials must be given. The Euler φ-function is a function on the positive integers, defined as follows: $\varphi(m)$ is the number of positive integers less than or equal to m and having no integer factor in common with m, other than 1 (such integers are said to be *relatively prime* to m). For example, $\varphi(1) = \varphi(2) = 1$, $\varphi(3) = \varphi(4) = 2$, $\varphi(9) = 6$. The cyclotomic ("circle-dividing") polynomial of order m, denoted $F_m(\lambda)$, is that monic polynomial (coefficient of the term of highest degree is 1) with integer coefficients all of whose roots are primitive mth roots of unity (that is, $r^m = 1$, and $r^n \neq 1$ for $0 < n < m$). Over the integers, $F_m(\lambda)$ is irreducible (not a nontrivial product of polynomials with integer coefficients).[4] From the definition, one can explicitly determine that $F^m(\lambda) = \prod_d (\lambda - \exp[2\pi i(d/m)])$, where the product is taken over all d, $1 \leqq d < m$ such that d and m are relatively prime. Thus, the degree of F_m is $\varphi(m)$.

The next theorem shows that, whatever constraints there are on available memory and acceptable resonant harmonics, the characteristic polynomial of the optimal recursion will be a cyclotomic polynomial.

Theorem 2: Let $F(\lambda) = \lambda^k - \sum_{i=1}^{k} a_i\lambda^{k-i}$, *where* $a_k \neq 0$, a_i $(i = 1, \cdots, k)$ *are integers. Suppose every root* ρ *of* $F(\lambda) = 0$ *satisfies* $|\rho| \leqq 1$. *Then* F *is a product of cyclotomic polynomials.*

This is proved in Section V. Recall from our prior discussion that all the roots of F must be chosen to satisfy $|\rho| \leqq 1$ to have stable detection. As it is, of course, better to have fewer resonances, one would hence choose for (4) a single cyclotomic polynomial. The cyclotomic polynomials make very desirable characteristic polynomials because of their extremely simple structure. For example, for $m < 105$ or for m a product of two primes, the coefficients of F_m are all 0, ± 1! For m a power of a single prime, the coefficients are all 0, 1 and for $m < 385$, the coefficients do not exceed 2 in absolute value. If m is a product of three distinct odd primes, all the coefficients are less than the smallest of those primes. These assertions are cited in Ref. 5.

This means that implementation of the recursion (1) in the filter shown in Fig. 1 is very simple indeed. For all cases of practical interest, the feedback coefficients a_i will be 0, ± 1. Of course, when $a_i = 0$, one simply does not put a tap on the ith stage. Because of the relation

$$F_{p_1^{\alpha} \cdots p_n^{\alpha_n}}(\lambda) = F_{p_1 \cdots p_n}(\lambda^{p_1^{\alpha_1 - 1} \cdots p_n^{\alpha_n - 1}})$$

(p_i distinct primes—see Ref. 4), most of the coefficients of F_m will usually be zero, and hence the taps-to-memory ratio is generally low (see Table I).

In the preceding discussion, the principal emphasis has been on the use of the filter as a receiver. However, considerations relating to its use as a generator lead to the same conclusion: that the characteristic polynomial (5) of the recursion (1) should be a cyclotomic polynomial. Indeed, for a generator, the problem of unwanted limit cycles is more critical. There is again the requirement that all the roots ρ_i of (5) satisfy $|\rho_i| \leqq 1$, as small perturbations in the initial conditions $x_0, \cdots, x_{k-1}$ from the (ideal) values 0, $\sin 2\pi f_0 \tau, \cdots, \sin 2\pi f_0(k-1)\tau$ (to generate $\sin 2\pi f_0 n\tau$) are inevitable; if such a perturbation occurs along an eigenvector corresponding to a root ρ_i, where $|\rho_i| > 1$, it produces a nonzero coefficient b_i for that root in the general solution $x_n = \sum_{i=1}^{k} b_i \rho_i^n$ (where $b_1, \cdots, b_k$ are functions of the initial conditions $x_0, \cdots, x_{k-1}$; see Section III). This component would attain an arbitrarily large amplitude (with time) and overwhelm the desired tone.

Hence, one again requires a filter that can perform perfect arithmetic and whose characteristic equation has all its roots on the unit disc. From Theorem 2 we thus deduce that (5) should be a product of cyclotomic polynomials for the generator as well. As tone generation is impeded by the presence of harmonic resonances at other roots (due, again, to perturbation of initial conditions), one takes for (5) a single cyclotomic polynomial.

Thus we have shown that, for both generating and receiving, the best linear recursion is one whose characteristic polynomial is cyclotomic. As the roots in this case are all of the form $\exp[2\pi i(d/m)]$, the resonant frequencies can be expressed as

$$2\pi f\tau \equiv 2\pi \frac{d}{m} \text{ (modulo } 2\pi) \tag{7}$$

for all positive integers $d < m$ such that d is relatively prime to m. Resonance at the fundamental is described by $2\pi f\tau = 2\pi(1/m)$, that is, the fundamental of the filter is $f = \tau^{-1}/m$. Hence, if one requires a fundamental frequency of f_0 (i.e., if f_0 is the frequency of the tone to be generated or detected) and one intends to use a filter with memory $k = \varphi(m)$, the clock rate τ^{-1} is set at $\tau^{-1} = f_0 m$. All other resonances occur at various harmonics (multiples of f_0) as follows: the resonant harmonics in the band $0 \leqq f \leqq \tau^{-1}$ occur when $f\tau = d/m$, that is, at $f = df_0$ for all those integers d as above. For example, if $m = 30$ then $k = 8$ and d assumes the values 1, 7, 11, 13, 17, 19, 23, 29. Hence, this filter has no resonances between the fundamental f_0 and the seventh harmonic. It resonates at the seventh harmonic $7f_0$, and thereafter at $11f_0$, $13f_0$, and so on. The resonances are at all the prime harmonics greater than 5, since in general those integers less than and relatively prime to the product m of the first p primes, are those primes lying between the pth prime and m. Furthermore, note that $30 = 1 + 29 = 7 + 23 = 11 + 19 = 13 + 17$. The first resonance due to aliasing will always be at $f = f_0 + \tau^{-1} = f_0 + f_0 m = (m + 1) f_0$. In the case of the previous example, this is the thirty-first harmonic.

Factors pertinent to the choice of which cyclotomic polynomial to use are relegated to Section VI. Suffice it to say at this point that the more memory available, the farther away from the fundamental can the first resonance be made due to aliasing. However, except for the cases $m = 1$ and $m = 2$, the first resonance after the fundamental will be below the clock frequency τ^{-1}. In these cases, for a given amount of memory k, if the interest is to have the first higher-order resonance as far from the fundamental as possible, one would find the largest integer r such that the product m of the first r primes satisfies $\varphi(m) \leqq k$. Then the first higher-order resonance would occur at the qth harmonic, where q is the $(r + 1)$st prime.

III. ELIMINATING IN-BAND HIGHER-ORDER RESONANCES

The preceding analysis has indicated that, within the constraints established, various higher-order resonances are unavoidable. This could lead to difficulties. In practice, many higher-order harmonics are introduced in the process of limiting the input signal. The limiter (see Fig. 2) limits the amplitude of the input signal $u(t)$. For example, a common limiter is a "hard-clipper." This has output ± 1, depending upon whether $u(t) \geqq 0$ or $u(t) < 0$. The effect of hard-clipping on an input is to produce all the odd harmonics: $\sin 2\pi ft \rightarrow 2/\pi \sin 2\pi ft + 2/3\pi \sin 6\pi ft + 2/5\pi \sin 10\pi ft + \cdots$. Hence, a filter with more resonances frequently must be run for a longer period of time to attain a threshold sufficiently high to reject spurious signals. Also, when used as a generator, perturbations of the initial conditions of the filter could lead to unwanted harmonics at all the resonances of the filter. As such perturbations are inevitable, it is usually necessary to make allowance for eliminating these harmonics.

While resonances due to aliasing are inherent to the discrete-time nature of the system and are hence unavoidable, resonances below the clock frequency τ^{-1} can be handled outside the feedback loop. In particular, it is possible (in theory) to eliminate (in practice, to reduce the Fourier coefficients of) any or all resonances at a frequency f, $0 < f < (2\tau)^{-1}$, along with the conjugate resonance at $\tau^{-1} - f$. This is effected through operations *outside* the feedback loop. Specifically, this is accomplished either through alteration of the input before it enters the filter: $u_n \rightarrow v_n = \sum_{i=1}^{d} c_i u_{n-i}$, or equivalently through alteration of the filter output before it enters the threshold detector: $x_n \rightarrow y_n = \sum_{i=1}^{d} c_i x_{n-i}$ (see Figs. 3 and 8). Although these two options are mathematically equivalent, considerations with respect to minimizing the word length necessary for perfect arithmetic would mitigate in favor of one or the other. This will be discussed in Section V. Here, we will describe the latter option only.

Let $X(\lambda)$ be the generating function for the sequence

$$x_n = \sum_{i=1}^{k} a_i x_{n-i} + u_n,$$

and let $U(\lambda)$ be the generating function for the input u_n. That is,

$$X(\lambda) = \sum_{n=0}^{\infty} x_n \lambda^n, \qquad U(\lambda) = \sum_{n=0}^{\infty} u_n \lambda^n. \tag{8}$$

Then

$$X(\lambda) = \sum_{i=1}^{k} a_i \lambda^i X(\lambda) + U(\lambda), \qquad \text{or} \qquad X(\lambda) = \frac{1}{1 - \sum_{i=1}^{k} a_i \lambda^i} U(\lambda).$$

Notice that defining $F(\lambda) \equiv \lambda^k - \sum a_i \lambda^{k-i}$, the characteristic polynomial of the filter, we obtain

$$X(\lambda) = \frac{1}{\lambda^k F(\lambda^{-1})} U(\lambda). \tag{9}$$

Since $F(\lambda)$ is assumed to be a cyclotomic polynomial, it is real and all its roots are of unit modulus. Hence ρ is a root if and only if $\bar{\rho} = \rho^{-1}$ is a root. It follows that $\lambda^k F(\lambda^{-1}) = F(\lambda)$. Thus (9) may be rewritten as

$$X(\lambda) = \frac{1}{F(\lambda)} U(\lambda). \tag{10}$$

We define a weighting function $W(\lambda)$ with the property that the resulting output function

$$Y(\lambda) \equiv W(\lambda) X(\lambda) \tag{11}$$

has poles only at those roots of $F(\lambda)$ corresponding to those resonances actually desired. Specifically, $W(\lambda)$ will be a real polynomial of degree $k - 2r$, where r is the number of resonances desired in the band $[0, (2\tau)^{-1}]$; the roots of W shall be those roots of F corresponding to the unwanted resonances. Typically, one desires to eliminate all resonances but the fundamental, in which case $r = 1$ and $W(\lambda)/F(\lambda) = 1/(\lambda^2 - a\lambda + 1)$ for an appropriate real number a. Then, from (10) and (11), one obtains $Y(\lambda) = W(\lambda)X(\lambda) = (1/(\lambda^2 - a\lambda + 1))U(\lambda)$ so $Y(\lambda) = -\lambda^2 Y(\lambda) + a\lambda Y(\lambda) + U(\lambda)$, and

$$y_n = a y_{n-1} - y_{n-2} + u_n. \tag{12}$$

This corresponds to a second-order filter with only one resonance in the band $[0, (2\tau)^{-1}]$ as shown in Fig. 3. Although there will be trunca-

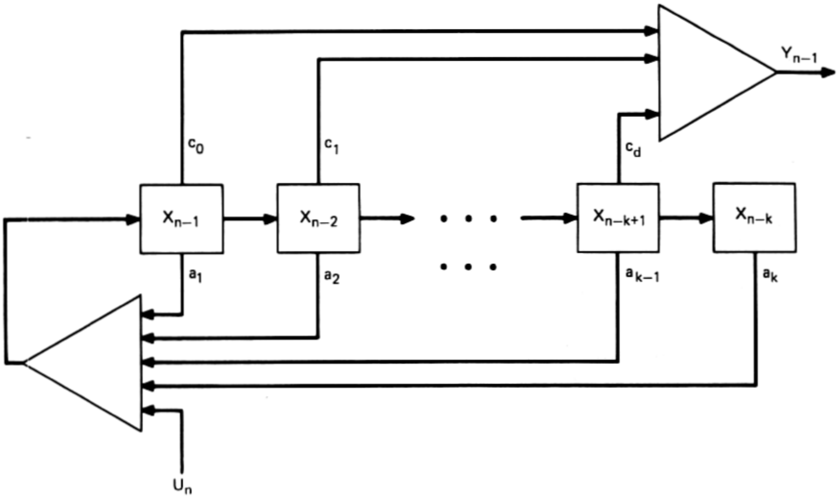

Fig. 3—Implementation of the weighting function.

tion error in (12), this will not lead to limit cycles, as there is no feedback from this to the filter [although (12) represents the performance of the filter in terms of resonances, the filter, of course, is not realized in this way]. Specifically, the weighting function is implemented as in Fig. 3. This is derived from definition (11): if $W(\lambda) = \sum_{i=0}^{d} c_i\lambda^i$, then equating terms in (11) yields

$$y_n = \sum_{i=0}^{d} c_i x_{n-i}, \tag{13}$$

where, typically, $d = k - 2$.

As mentioned earlier, the arithmetic of the weighting function is only approximate; since there is truncation error in the computation of the coefficients c_i, the roots of W will not precisely cancel out the roots of F. Rather, the roots of W will be slightly perturbed from the corresponding roots of F. The effect of this, as will be shown, is that all the resonances due to the roots of F (i.e., all the resonant harmonics of the original feedback loop) will be present in the output y_n—however, they will have reduced energy (but for the fundamental). That is, the less the error in the implementation of W, the smaller the Fourier coefficients of the higher resonant harmonics of the filter. This is demonstrated below.

Suppose F is the cyclotomic polynomial of order m (or any polynomial whose roots $\rho_i, \cdots, \rho_k$ are distinct mth roots of unity, so that each $\rho_j = e^{i2\pi q/m}$ for some integer q, $0 \leqq q < m$). A continuous-time extension $x(t)$ of the discrete-time function x_n, satisfying $x(n\tau) = x_n$ can be defined as

$$x(t) \equiv \sum_{n=0}^{m-1} x_n v(t - n\tau), \tag{14}$$

where $v(t)$ describes a continuous-time extension of x_n. Specifically, $v(t)$ is a periodic input pulse satisfying $v(t + m\tau) = v(t)$ for all t [typically, $v(t) = 1$ for $0 \leqq t < \tau$]. In (4), set $u_n = v(n\tau)$ and normalize $v(0) = 1$. Then $x_n = \sum_{j=1}^{k} b_j \rho_j^n$ for $n < m$. Let $\hat{x}(q)$ [$\hat{v}(q)$] denote the qth Fourier coefficient of $x(t)$ [$v(t)$]. It follows that

$$\begin{aligned}
\hat{x}(q) &\equiv \frac{1}{m\tau} \int_0^{m\tau} x(t) \exp\left(-i2\pi \frac{q}{m} t\right) dt \\
&= \hat{v}(q) \sum_{n=0}^{m-1} x_n \exp\left(-i2\pi \frac{q}{m} n\right) \\
&= \hat{v}(q) \sum_{j=1}^{k} b_j \sum_{n=0}^{m-1} \rho_j^n \exp\left(-i2\pi \frac{q}{m} n\right) \\
&= \hat{v}(q) b_j,
\end{aligned} \tag{15}$$

where j is that index such that $\rho_j = \exp[i2\pi(q/m)]$; if no such index exists, then $\hat{x}(q) = 0$. To simplify matters, we will use the expression "the Fourier coefficient at (the root) ρ_j" to indicate what in the case of (15) is the qth Fourier coefficient $\hat{x}(q)$.

These Fourier coefficients can be computed explicitly from (9). Indeed, factoring $\lambda^k F(\lambda^{-1}) = \prod_{j=1}^{k} (1 - \rho_j \lambda)$ obtains

$$\begin{aligned}
X(\lambda) &= \prod_{j=1}^{k} \frac{1}{1 - \rho_j \lambda} U(\lambda) \\
&= \sum_{j=1}^{k} B_j \frac{1}{1 - \rho_j \lambda} U(\lambda),
\end{aligned} \tag{16}$$

where the B_j's are the coefficients of the partial fraction decomposition, derived explicitly in Lemma 3 below (it is assumed that all the roots ρ_j are distinct; in the case of multiple roots, however, similar results obtain). From (16) one obtains

$$\begin{aligned}
X(\lambda) &= \sum_{j=1}^{k} B_j \sum_{n=0}^{\infty} (\rho_j \lambda)^n \sum_{i=0}^{\infty} u_i \lambda^i \\
&= \sum_j B_j \sum_{i,n} \rho_j^{n-i} u_i \lambda^n,
\end{aligned} \tag{17}$$

so $x_n = \sum_{j=1}^{k} B_j \sum_{i=0}^{n} \rho_j^{n-i} u_i$ [which is (4) above]. Hence, $B_j = b_j$ $(j = 1, \cdots, k)$ and their explicit form is given in the following lemma.

Lemma 3: *Suppose* $\rho_1, \cdots, \rho_k$ *are distinct numbers. Then*

$$\prod_{i=1}^{k} \frac{1}{1-\rho_i\lambda} = \sum_{i=1}^{k} \left(\prod_{\substack{j=1 \\ j\neq i}}^{k} \frac{\rho_i}{(\rho_i - \rho_j)} \right) \frac{1}{1-\rho_i\lambda}.$$

Proof: The residue of the left-hand side at the ith pole is the coefficient of that term in the sum above. The decomposition follows from the Cauchy residue theorem.

Notice that, as the roots of F occur in conjugate pairs, a direct consequence of (17) is that, if ρ_i and ρ_j are conjugate roots, then the corresponding Fourier coefficients are also conjugate: $b_i = \bar{b}_j$.

The Fourier coefficients for the sequence y_n can be determined as in (15). For $x_n = \sum b_j\rho_j^n$ as before, we obtain from (13)

$$\begin{aligned} y_n &= \sum_{i=0}^{d} c_i \sum_{j=1}^{k} b_j\rho_j^{n-i} \\ &= \sum_{j=1}^{k} W(\bar{\rho}_j) b_j \rho_j^n. \end{aligned} \tag{18}$$

Thus, the Fourier coefficient of the sequence y_n at the root ρ_j is $W(\bar{\rho}_j)b_j$ (as could be expected, since Fourier transformations are multiplicative). Again, the conjugate coefficient $W(\rho_j) = \overline{W(\bar{\rho}_j)}$. Observe that, if ρ_j is a root of W, then the Fourier coefficients of y_n vanish at the roots ρ_j and $\bar{\rho}_j$ (W was chosen to be real). If W' is the result of perturbing the coefficients of W to correspond to truncation error, then $W'(\bar{\rho}_j)$ is (by continuity) close to zero. Hence, as errors in the weighting functions are reduced, so is the power at each of the resonant harmonics above the fundamental (running the system for finite time, of course). Surprisingly, W is very stable; if the coefficients of W' are simply those of W rounded to the nearest integer (!), the results are frequently virtually as good as if W itself were used. This is exhibited in Table I and illustrated in Figs. 4, 5, and 6. These figures correspond to a filter using the cyclotomic polynomial F_{30}. The input is a hard-clipped sine wave for each given frequency up to 15 times the fundamental. The input frequencies are normalized to units of the fundamental frequency for each filter. For each input frequency, the filter is run for an amount of time equal to seven cycles of the fundamental. If this time corresponds to N steps of the filter, the output is $\max_{n \leqq N} |x_n|$, as measured at each input frequency (1500 samples). Using a W' with integer coefficients (or any W' with uniformly truncated coefficients) enables one to perform all the multiplications as additions, simplifying implementation and eliminating any further errors. As one expects, upon

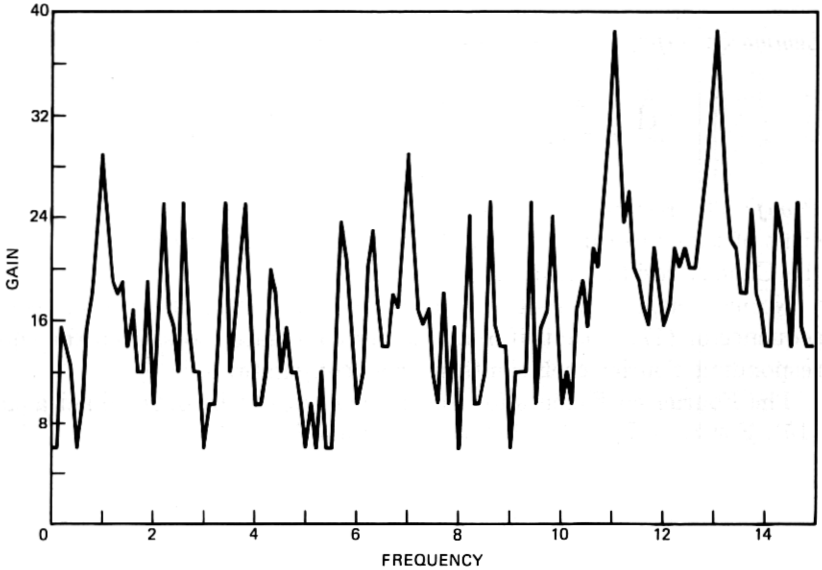

Fig. 4—Hard-clipped/no weighting.

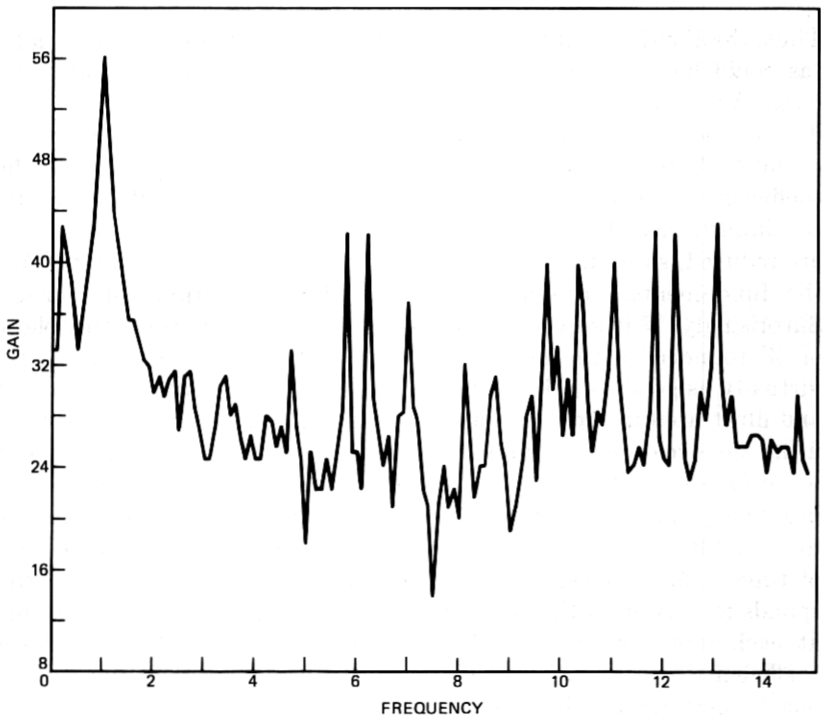

Fig. 5—Hard-clipped/rounded weighting.

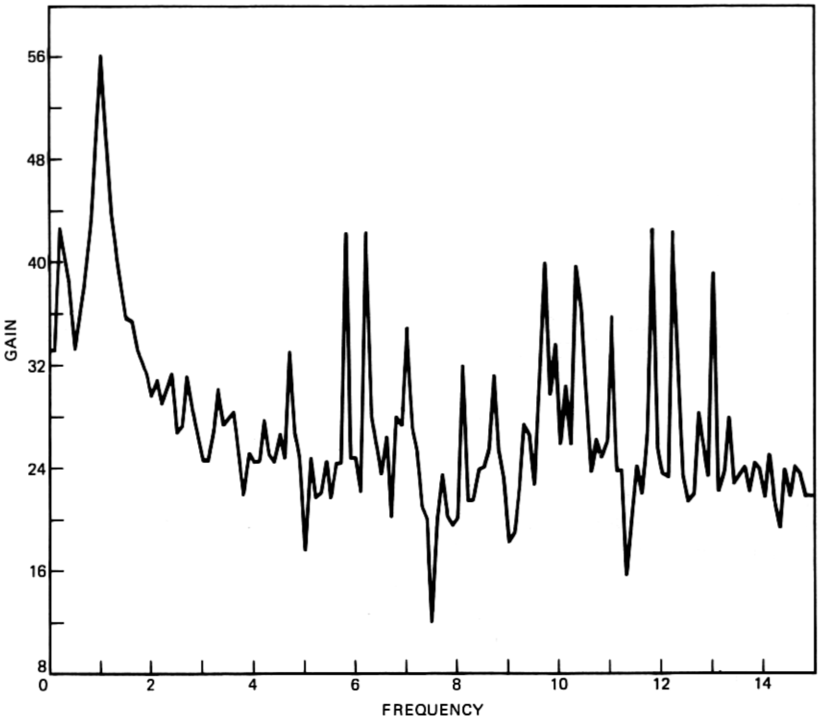

Fig. 6—Hard-clipped/exact weighting.

setting $\rho_2 = \bar{\rho}_1$, the Fourier coefficient of y_n at the fundamental

$$W(\bar{\rho}_1) \prod_{j=2}^{k} \frac{\rho_1}{\rho_1 - \rho_j} = \frac{\rho_1}{\rho_1 - \rho_2} \prod_{j=3}^{k} \frac{\rho_1(1 - \rho_j \bar{\rho}_1)}{\rho_1 - \rho_j} = \frac{\rho_1}{\rho_1 - \rho_2},$$

is the Fourier coefficient of (12) at the fundamental.

IV. IMPULSE RESPONSE

The impulse response is the output resulting from an input of a single pulse: $u_0 = 1$, $u_{n>0} = 0$. Since this output can also be produced by appropriately setting initial conditions, we will refer to it as a pulse train. From (4) we see that if the input u_n is a single pulse, then the output x_n reduces to

$$x_n = \sum_{i=1}^{k} b_i \rho_i^n. \tag{19}$$

In the context of the previous sections, it is assumed that the characteristic polynomial of the sequence x_n is cyclotomic. Since each ρ_i is then an mth root of unity, the sequence x_n is periodic: $x_{n+m} = x_n$ for all n. As before, the resonant harmonics present in the pulse trains x_n correspond to the mth roots of unity which are roots ρ_i $(i = 1, \cdots, k)$

of $F = F_m$; the Fourier coefficient of the pulse train at the root ρ_j is b_j (see Section III).

In particular, using the notations of Section III, $U(\lambda) = 1$ and thus (10) reduces to

$$X(\lambda) = \frac{1}{F(\lambda)}. \tag{20}$$

But $X(\lambda) = \sum_{n=0}^{\infty} x_n\lambda^n = (\sum_{n=0}^{m-1} x_n\lambda^n)(\sum_{n=0}^{\infty} \lambda^{mn})$ since $x_{n+m} = x_n$. Defining $f(\lambda) = \sum_{n=0}^{m-1} x_n\lambda^n$, one obtains

$$f(\lambda) = \frac{1-\lambda^m}{F(\lambda)} \tag{21}$$

from (20). Notice that f has integer coefficients (the input u_n is integer, as are the coefficients a_i). Indeed, $1 - \lambda^m$ is a product of cyclotomic polynomials, one of which is $F(\lambda)$. Specifically,

$$1 - \lambda^m = \pm \prod_{n|m} F_n(\lambda)$$

[the product is taken over all n which divide m; hence, for example,

$$\begin{aligned} 1 - \lambda^6 &= -F_1(\lambda)F_2(\lambda)F_3(\lambda)F_6(\lambda) \\ &= (\lambda - 1)(\lambda + 1)(\lambda^2 + \lambda + 1)(\lambda^2 - \lambda + 1)]; \end{aligned}$$

and, from (21),

$$f(\lambda) = \pm \prod_{\substack{n \mid m \\ n \neq m}} F_n(\lambda)$$

obtains. Consequently, $f(\rho) = 0$ for all mth roots of unity ρ, *except* for the primitive roots of unity [the roots of $F_m(\lambda)$]. This was anticipated by E. N. Gilbert in Ref. 6, where he showed that a pulse train x_n of period m has resonances at those harmonics corresponding to the mth roots of unity which are not roots of $\sum_{n=0}^{m-1} x_n\lambda^n = 0$. Equation (21) covers the general situation where $f(\lambda)$ [and consequently $F(\lambda)$] are arbitrary products of cyclotomic factors of $1 - \lambda^m$.

In the same paper, Gilbert was concerned about the problem of increasing the power of the pulse train at the fundamental (relative to the power at the other resonances). This could be done by shaping the input u_n for one period, but it is usually undesirable to do this. As explained in Section III, however, the same effect is obtained by utilizing a weighting function W. If utilized directly, this will introduce noninteger levels into the pulse train. Nonetheless, it is possible to avoid this by replacing W with W^I where the latter is obtained through rounding off to the nearest integer the coefficients of the former. The pulse train resulting from W^I will have integer levels, but the truncation error will again introduce higher-order resonances. However, Table I shows that these are very small indeed, leaving typically about 98 percent of the power at the fundamental. This compares with 25 percent or less (for F_{16}, F_{24}, F_{30}) without W^I. Note, for example, F_8. The pulse train 1, 0, 0, 0, -1, 0, 0, 0 has resonances at the third, fifth, and seventh harmonics. However, by simply altering this to 1, 1, 1, 0, -1, -1, -1, 0, the first appreciable resonance does not come until the seventh harmonic. In this case, use of W^I does not introduce any new levels in the pulse train.

The worst case in Table I is F_9 where 92 percent of the power is at the fundamental. E. N. Gilbert has pointed out that if one wished to increase the proportion of the power at the fundamental of this train (or any other), one could multiply the output $Y(\lambda)$ by some constant $c > 1$, chosen so that the roundoff error of $cW \to (cW)^I$ is smaller than that for W^I alone [recall (11)]. This, however, would introduce more levels into the pulse train (although no more than c times as many).

Table I gives an indication of the possibilities for various filters. Included are the filters with memory less than 12 which provide the greatest separation between the fundamental and the first resonant harmonic, either with or without the weighting function. The asterisks and daggers indicate those which, for the amount of memory, have the largest possible separation without or with the weighting function. For utilization with a "hard-clipper" (which has all odd harmonics), F_3, F_9, and F_{15} are included. Although these resonate at all even harmonics, they have the same response to a hard-clipped input at the fundamental as the respective cyclotomic filters of twice the sampling rate. To have the first resonant harmonic higher than the seventh (without W) would require a memory of 48 (and F to have a coefficient of 2). The next interesting entry with respect to W is F_{36} with memory 12. The columns to the right of the double line all deal with the integer-rounded transfer function W^I. Columns A and B give $|b_1|^2/\sum_{i=1}^{k/2} |b_i|^2$ and $|b_1 W^I(\bar{\rho}_1)|^2/\sum_{i=1}^{k/2} |b_i W^I(\bar{\rho}_i)|^2$ as a percent, respectively, where b_i is the Fourier coefficient of the sequence x_n at the root ρ_i [see (6) and Section III)]. Column C gives $(\max_{2 \leqq i \leqq k/2} |b_i W^I(\bar{\rho}_i)|^2)/|b_1 W^I(\bar{\rho}_1)|^2$ as a percent. The roots ρ_i ($i = 1, \cdots, k/2$) are assumed to be in order of ascending argument $<\pi$ (so ρ_1 is the fundamental). Columns D and E give the moduli of the Fourier coefficients b_i and $b_i W^I(\bar{\rho}_i)$ of the sequences x_n and y_n, respectively. Columns F and G give the pulse trains of x_n and y_n, respectively, with initial pulse $u_0 = 1$, $u_{n>0} = 0$. The exponent denotes repeated digit; the arrow indicates that the preceding train is followed by another identical train, but that each digit is the negative of what it was.

Memory	Characteristic Polynomial F	Resonant Harmonics in the Band $[0, \tau^{-1}]$, Aside From Fundamental: Without Weighting Function	Resonant Harmonics in the Band $[0, \tau^{-1}]$, Aside From Fundamental: With Weighting Function	First Resonant Harmonic Due to Aliasing	Number of Taps on Filter (Without Weighting Function)	Integer-Rounded Weighting Function W^I
*1	$F_2 = \lambda + 1$	none	—	3	1	—
2	$F_3 = \lambda^2 + \lambda + 1$	2	—	4	2	—
2	$F_4 = \lambda^2 + 1$	3	—	5	1	—
*2	$F_6 = \lambda^2 - \lambda + 1$	5	—	7	2	—
4	$F_8 = \lambda^4 + 1$	3, 5, 7	7	9	1	$1 + \lambda + \lambda^2$
†4	$F_{12} = \lambda^4 - \lambda^2 + 1$	5, 7, 11	11	13	2	$1 + 2\lambda + \lambda^2$
6	$F_9 = \lambda^6 + \lambda^3 + 1$	2, 4, 5, 7, 8	8	10	2	$1 + 2\lambda + \lambda^2 + 2\lambda^3 + \lambda^4$
†6	$F_{18} = \lambda^6 - \lambda^3 + 1$	5, 7, 11, 13, 17	17	19	2	$1 + 2\lambda + 3\lambda^2 + 2\lambda^3 + \lambda^4$
8	$F_{15} = \lambda^8 - \lambda^7 + \lambda^5 - \lambda^4 + \lambda^3 - \lambda + 1$	2, 4, 7, 8, 11, 13, 14	14	16	6	$1 + \lambda + \lambda^2 + \lambda^3 + \lambda^4 + \lambda^5 + \lambda^6$
8	$F_{16} = \lambda^8 + 1$	3, 5, 7, 9, 11, 13, 15	15	17	1	$1 + 2\lambda + 2\lambda^2 + 3\lambda^3 + 2\lambda^4 + 2\lambda^5 + \lambda^6$
8	$F_{24} = \lambda^8 - \lambda^4 + 1$	5, 7, 11, 13, 17, 19, 23	23	25	2	$1 + 2\lambda + 3\lambda^2 + 3\lambda^3 + 3\lambda^4 + 2\lambda^5 + \lambda^6$
†*8	$F_{30} = \lambda^8 + \lambda^7 - \lambda^5 - \lambda^4 - \lambda^3 + \lambda + 1$	7, 11, 13, 17, 19, 23, 29	29	31	6	$1 + 3\lambda + 5\lambda^2 + 5\lambda^3 + 5\lambda^5 + 3\lambda^5 + \lambda^6$

*† See text for explanation

V. CONDITIONS FOR PERFECT ARITHMETIC

Here we indicate why cyclotomic polynomials yield optimal recursions for generating sinuosidal signals. When we use (1) to generate tones, the u_n is set to zero and some initial condition $x_0, x_1, \cdots, x_{k-1}$ is chosen to generate the required samples x_n:

$$x_n = \sum_{i=1}^{k} a_i x_{n-i}. \tag{22}$$

If we use the usual second-order recursion, then (22) is of the form

$$x_n = a x_{n-1} - x_{n-2}, \tag{23}$$

where $|a| < 2$, so we have complex roots. In this case, we show below that the number of distinct values that x_n, $n = 0, 1, \cdots, N$ can take grows at least as fast as $N/2$, with N. So, to simulate (22) with perfect arithmetic, the number of "words" needed grows at least as fast as N, the number of samples needed.

Proposition 4: Suppose $|a| < 2$, and rational but not an integer. Then for any initial conditions x_0, x_1 (not both zero) and any positive integer N, the number of distinct values among $x_0, \cdots, x_N$, where $x_n = a x_{n-1} - x_{n-2}$, for $2 \leqq n \leqq N$, is at least $N/2$.

some cyclotomic filters

A	B		D	E	F	G
% of Total Power in $[0, (2\tau)^{-1}]$ at Fundamental		Highest Power at a Rejected Resonance (% of Fundamental)	Modulus of Fourier Coefficients (in Order of Arguments $< \pi$)		Pulse Trains	
Without W^I	With W^I		x_n	y_n	x_n	y_n
100	—	—	1	—	1, −1	—
100	—	—	0.58	—	1, −1, 0	—
100	—	—	0.5	—	1, 0, −1, 0	—
100	—	—	1.73	—	$1^2 0 - 1^2 0$	—
50	97.1	2.9	All 0.25	0.60, 0.10	$10^3 - 10^3$	$1^3 0 - 1^3 0$
50	99.5	0.5	All 0.29	1.08, 0.08	$1010^3 - 10 - 10^3$	$12^3 10 - 1 - 2^3 - 10$
33.3	92.7	7.8	All 0.19	0.85, 0.04, 0.24	$10^2 - 10^5$	$121^2 - 1^2 - 2 - 10$
33.3	99.6	0.3	All 0.19	7.6, 0.04, 0.06	$10^2 10^5 \rightarrow$	$123^4 210 \rightarrow$
53.3	98.9	0.8	0.33, 0.27, 0.09, 0.11	4.78, 0.51, 0.55, 0.75	$1^3 0^2 - 1^3 0^7$	$123^3 21 - 1 - 2 - 3^3 - 2 - 10$
25	98.0	1.8	All 0.13	1.29, 0.02, 0.06, 0.17	$10^7 - 10^7$	$12^2 32^2 10 \rightarrow$
25	99.5	0.3	All 0.14	1.97, 0.05, 0.09, 0.11	$10^3 10^7 - 10^3 - 10^7$	$123^2 4^3 3^2 210 \rightarrow$
6	98.4	1.0	0.11, 0.09, 0.27, 0.33	2.41, 0.04, 0.19, 0.24	$1 - 110^2 1 - 110^7 \rightarrow$	$123^2 545^2 453^2 210 \rightarrow$

Proof: We can write $x_n = b_1\rho_1^n + b_2\rho_2^n$, where ρ_1, ρ_2 are the distinct roots of $\lambda^2 - a\lambda + 1$, as in (19). Since the roots are not real, let $\rho = \rho_1(= \bar{\rho}_2)$, $b = b_1(= \bar{b}_2)$. Then $x_n = x_m$ implies $\text{Re}\,(b\rho^n) = \text{Re}\,(b\rho^m)$. In this case, letting $\theta = \arg \rho$, $\varphi = \arg b$, we obtain $\cos(\varphi + n\theta) = \cos(\varphi + m\theta)$ so $\varphi + n\theta \equiv \pm(\varphi + m\theta) \pmod{2\pi}$. Since ρ is not a root of unity, the numbers $n\theta$ $(n = 0, 1, 2, \cdots)$ are all distinct and hence for fixed m either $n = m$ or $n\theta \equiv -2\varphi - m\theta \pmod{2\pi}$. As this last congruence can be satisfied by at most one n, it follows that, for each m, there is at most one $n \neq m$ such that $x_n = x_m$.

The following result shows that, if one wishes to generate $\sin \pi n\theta$ with perfect accuracy using a linear recursion, $e^{i\pi\theta}$ must be a root of the corresponding polynomial (5).

Proposition 5: If $s_n = \sin \pi n\theta$ is a solution of $x_n = \sum a_j x_{n-j}$ and θ is not an integer, then $e^{i\pi\theta}$ is a root of the polynomial $\lambda^k - \sum a\lambda^{k-j}$.

Proof: From $\sin \pi(n + 1)\theta = \sum a_j \sin \pi(n + 1 - j)\theta$, we expand both sides using a familiar trigonometric identity and get

$$\begin{aligned} \sin \pi n\theta \cos \pi\theta + \cos \pi n\theta \sin \pi\theta &= \sin \pi(n + 1)\theta \\ &= \sum a_j \sin \pi(n + 1 - j)\theta = \sum a_j \sin \pi(n - j)\theta \cos \pi\theta \\ &\quad + \sum a_j \cos \pi(n - j)\theta \sin \pi\theta = \sin \pi n\theta \cos \pi\theta \\ &\qquad + \sum a_j \cos \pi(n - j)\theta \sin \pi\theta. \end{aligned}$$

Since θ is not an integer, $\sin \pi\theta \neq 0$, and thus from the equality of the first and last expressions, we obtain $\cos \pi n\theta = \sum a_j \cos \pi(n - j)\theta$. Hence, $\cos \pi n\theta$ is also a solution to the recursion, and it follows that $e^{i\pi n\theta} = \cos \pi n\theta + i \sin \pi n\theta$ is a solution too. Consequently, $e^{i\pi k\theta} - \sum a_j e^{i\pi(k-j)\theta} = 0$.

The next theorem shows that every recursion which satisfies the stability criterion $|\rho| \leqq 1$ for all its roots, and for which perfect arithmetic is possible, is cyclotomic.

Theorem 6: Suppose every root ρ *of the polynomial* $F(\lambda) = \lambda^k - \sum_{i=1}^{k} a_i\lambda^{k-i}$ *satisfies* $|\rho| \leqq 1$.

(i) If $a_1, \cdots, a_k$ *are integers and* $a_k \neq 0$, *then* $F(\lambda)$ *is a product of cyclotomic polynomials.*

(ii) If $a_1, \cdots, a_k$ *are rational numbers and* $x_n = \sum a_i x_{n-i}$ *is periodic* ($x_{n+p} = x_n$ *for some* p, *all* n) *for some nonzero initial conditions* $x_0, \cdots, x_{k-1}$, *then* $F(\lambda)$ *has as a factor a cyclotomic polynomial.*

Proof: For case (*i*), each irreducible factor (over the integers) of $F(\lambda)$ has the same form as $F(\lambda)$ itself by' 'Gauss' Lemma".[6] Thus, it suffices to assume that $F(\lambda)$ is irreducible, in which case all its roots are distinct. In this case, we can write $x_n = \sum b_i\rho_i^n$ where the ρ_i's are the roots of $F(\lambda)$ and x_n is as in case (*ii*). But then $|x_n| \leqq \sum |b_i|$, and as for any integer initial conditions $x_0, \cdots, x_{k-1}, x_n$ will be an integer for all n, x_n can in such a case assume only a finite number m of distinct values ($m = [\sum|b_i|]$). Hence for all n, the $k$tuple $(x_{n+1}, \cdots, x_{n+k})$ can assume at most m^k distinct values, and as x_n is recursively generated with memory k, x_n must be periodic, of period $p \leqq m^k$. This brings us to case (*ii*).

For case (*ii*), let L be the rational canonical form associated with the recursion x_n (see Ref. 7, Section 5.2.1), and J be the Jordan canonical form of L. Then for some initial state vector $\mathbf{x}$, $J^p\mathbf{x} = \mathbf{x}$, and it follows that some diagonal element of J, that is, some root of $F(\lambda)$, must be a pth root of unity. Hence, the irreducible factor of $F(\lambda)$ having that root must be cyclotomic.

Hence, from the above the θ of Proposition 5 must be rational when perfect accuracy is required.

In all the preceding, the basic assumption has been that all the coefficients of the recursion (22) are real. We can infer from Theorem 1 that this is no loss of generality as, if the recursion had complex coefficients (with rational real and imaginary parts) and was irreducible over the field $Q(i)$ (the field of gaussian rationals), then the roots of the characteristic polynomial would be distinct, no pair being conjugate. Indeed, Theorem 1 remains true if the word "integer" is every-where replaced by "rational number." The arguments of Section II show that we may as well assume all the coefficients are real.

VI. COMPUTING WORD LENGTH AND ADDITIONS PER CYCLE

To realize the cyclotomic filters in hardware with perfect arithmetic, the necessary amount of memory and adder complexity must be provided. We describe here how to estimate the word length and the rate of additions required to implement a cyclotomic filter with a weighting function. It shall be assumed that all operations are performed in binary form. The number of binary bits required to store each x_n is called the word length ω of the system. For generators that produce a signal approximating a sinusoid, the word length required will depend on the accuracy of approximation needed. When the filter is used as a tone detector, the word length required will depend on the duration of operation, since the signal level tends to build up, especially at frequencies close to any resonant frequency (Fig. 7). The signal level, of course, does not uniquely specify the minimum word length. Even though for storing x_n we may need only ω bits, it is conceivable that during the computations numbers greater in magnitude than x_n, which need more bits for storage, could arise. To perform operations in a serial-multiplexed fashion, it is desirable to have uniform word length for all operations in the feedback loop of the filter. Hence, the word length will have to be increased to accommodate any number encountered during the computations. However, for the filters considered

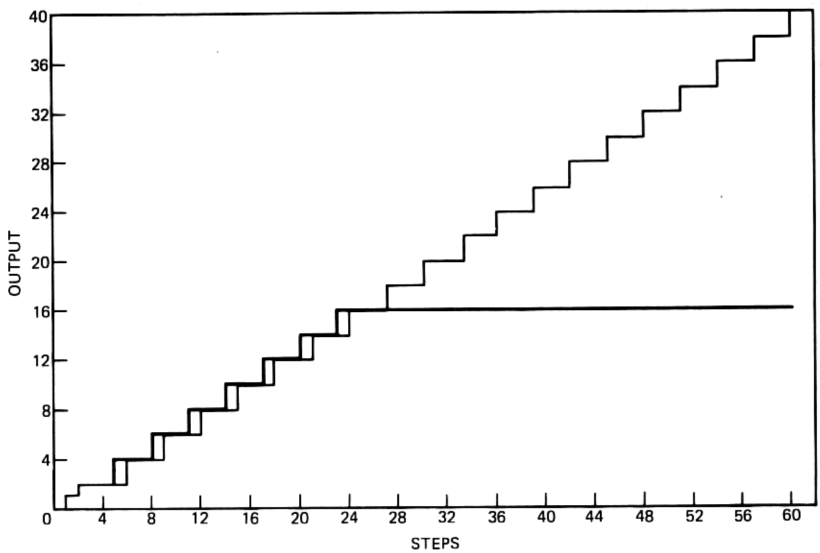

Fig. 7—Growth of output.

in Table II, it is possible to arrange the computations in such a way that the word length is determined by the maximum magnitude of x_n. In general, there are a finite number of ways in which the additions involved in the filter can be arranged. By simulation of the different arrangements, the word length required can then be determined.

There are two possible ways of implementing the cyclotomic filters as generators. The first is to generate the impulse response (19); this is generally sufficient (see Table I). In this case, the weighting function (13) shapes the effect of this impulse to simulate the initial conditions $x_0, \cdots, x_{k-1}$ of the tone being generated. As the input is zero after the initial pulse $u_0 = 1$, the weighting function need only be used during the first $d + 1$ steps of the filter. Let m be the largest number in the pulse train y_n of Table I, and let $[[x]]$ be the smallest integer larger than x. The word length necessary for perfect arithmetic is at least $\omega = [[\log_2 m]] + 1$ and, for the filters considered here, ω is also sufficient. (We add 1 for a sign bit.) This word length is shown in column B of Table II.

However, rounding off in the weighting function introduces errors in the effective initial values of the signal. If this approximation is not sufficiently good, then the initial conditions of the filter $x_0, \cdots, x_{k-1}$ can be set as accurately as needed, and then the filter is operated with the feedback loop alone. In particular, one can set the initial conditions of the filter such that $|x_n - \sin 2\pi n/p| < 2^{-m}$ $(n = 0, \cdots, k - 1)$ where $\sin 2\pi n/p$ is the desired signal. One can then compute the minimum word length required by simulating the filter for one period. In all cases of interest here, the word length including sign is $(m + 1)$ for $m \leqq 12$. Hence, as an example, the cyclotomic filter of order 30 can generate a sequence (x_n) such that $|x_n - \sin 2\pi n/p| < 2^{-10}$ if the initial conditions are set such that $|x_n - \sin 2\pi n/p| < 2^{-10}$ $(n = 0, \cdots, 7)$, using a word length of 11.

To determine the number of binary additions per period of the filter (i.e., per cycle of the fundamental), one counts the number of bit additions per step. If m denotes the number of additions per step, then $pm\omega$ is the number of binary additions per cycle, where p is the period of (22) and ω the word length used in the feedback loop (see above). When the generator is implemented in the first way (using an initial pulse and the weighting function), the number of additions is shown in column C of Table II (not including those necessary in the initial $d + 1$ steps for the weighting function). When the generator is implemented in the second way (setting the initial conditions), the number of additions can be computed by multiplying the value in column C by ω/ω', where ω is the word length chosen and ω' is the corresponding word length from column B.

When the filter is used as a detector, we assume that the input to the filter is a sequence which only assumes the values $+1$ and -1. This is true, for example, when the analog signal to be detected is either hard-clipped or delta-modulated. In these cases, it is advantageous to apply the weight function to the input sequence u_n rather than to the sequence x_n; since, in general, x_n can assume many values other than $+1$ and -1, computations involving the weighting function are simplified if they are performed on the input (see Section III). In fact, applying the weighting function to the input is so simple arithmetically that it can be implemented with read-only memory. On the other hand, if read-only memory is not used and one wishes to save on computations by checking the threshold (max $\{x_n\}$) only in the last cycle of the filter (with respect to its duration of operation for detection), then the weighting function is best implemented as in Section III, on the output of the feedback loop. Then the filter can be run during all but the last cycle, without computing the weighting function.

When the weighting function is applied to the input, the filter is described by

$$v_n = \sum_{i=0}^{d} c_i u_{n-i} \tag{24}$$

$$x_n = \sum_{i=1}^{k} a_i x_{n-i} + v_n, \tag{25}$$

where u_n is the input into the filter and v_n is the result of the weighting function. Figure 8 describes this filter.

For the filters in Table I, the effect of rounding c_i to the nearest integer is slight. Hence, it is *a fortiori* suitable to round off $v_n = \sum c_i u_{n-i}$ to the nearest integer. Therefore, since the only values

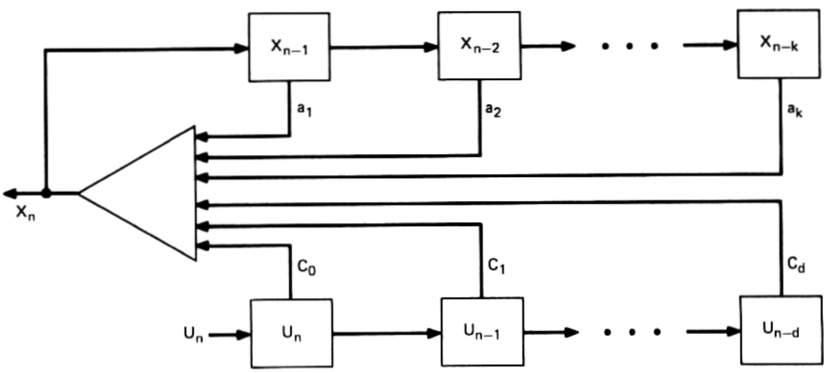

Fig. 8—Implementation of the weighting function at the input.

assumed by u_i are ± 1, it suffices to have for v_n a word length of $\omega = [[\log_2 \{\sum |c_i|\}]] + 1$ (where $\{x\}$ is the integer closest to x and $[[x]]$ is the smallest integer larger than x; 1 is added for a sign bit). The sequence v_n can then assume any value between $-\{\sum |c_i|\}$ and $\{\sum |c_i|\}$. With d as in (24) and ω as above, implementations of the weighting function with read-only memory then requires $2^{d+1}\,\omega$ memory bits. The respective values for this are shown in column D of Table II. When a bank of such tuned filters is used in one receiver (for example, in a *Touch-Tone®* system such as described in Ref. 3), all the filters could use one read-only memory for the weighting functions. Also, by increasing ω, we can make the round-off error as small as we wish.

To determine the word length for use in the feedback loop of the detector, the maximum signal level can be determined by using an input u_n of the same frequency as the resonant frequency. Since the impulse response [see (19)] of these filters is periodic and of the same period as the resonant frequency, the latter produces the maximum signal level $\sup_{n \leq N} x_n$, for duration of operation $N\tau$. Let this maximum be M. The word length required should then be at least $[[\log_2 M]] + 1$. For all the filters considered here, $[[\log_2 M]] + 1$ is also sufficient. The number of M, of course, is determined by N. If the cyclotomic filter is of period p (i.e., Theorem 1 is F_p), then the filter runs through N/p periods, corresponding to N/p cycles of the fundamental. Calculations have been made for two values of N/p: 7 (the number of cycles computed in Ref. 3 to be necessary for *Touch-Tone* interchannel rejection), and 10 (a more uniform point of reference).

In Table II, column E shows the word length required in the feedback loop for the indicated durations, when the weighting function is computed on the input as in (24), implemented equivalently with or without read-only memory, producing the filter response (25).

When there is no weighting function on the input, the word length required is shown in column F (of course, a weighting function may be applied to the output as in Section III).

The number of binary additions per cycle for the detector is determined in the same way as for the generator; the number is $pm\omega$ as defined above. These numbers are shown in columns G, H, and K of Table II. Column G shows the number of binary additions per cycle in the feedback loop when read-only memory is used to implement the weighting function, applied to the input as in (24). If read-only memory is not used, then the weighting function has to be computed. Since the numbers involved in the computation of the weighting function [when implemented as in (24)] are generally smaller than those in the feedback loop, the word length required for their computations are smaller. Hence, one can use two different adders, one for the weighting function

Table II — Complexity of some cyclotomic generator-detectors

	Generator Using Impulse Response			Detector: 7 Cycles					Detector: 10 Cycles				
				Word Length for (x_n)		Adds/Cycle for (x_n)			Word Length for (x_n)		Adds/Cycle for (x_n)		
Period	Word Length	Adds/Cycle	R.O.M.	Weighted Input	Unweighted Input	Weighted Input	Unweighted Input	For (v_n)	Weighted Input	Unweighted Input	Weighted Input	Unweighted Input	For (v_n)
A	B	C	D	E	F	G	H	K	E	F	G	H	K
6	2	24	—	6	6	72	72	—	7	7	84	84	—
8	2	16	24	7	5	56	40	112	8	6	64	48	128
9	3	54	128	8	5	144	90	360	8	6	144	108	360
12	3	72	24	8	6	192	144	288	9	7	216	168	324
15	3	270	512	9	7	810	630	810	10	7	900	630	900
16	3	48	640	9	5	144	80	1276	10	6	160	96	1440
18	3	108	128	9	6	324	216	1134	10	7	360	252	1260
24	4	192	640	10	6	480	288	2160	11	7	528	336	2376
30	4	720	768	11	8	1980	1440	3630	11	8	1980	1440	3630

and one for the feedback loop. Using this arrangement, the number of additions per cycle for calculating the weighting function is shown in column K. The number of binary additions per cycle when no weighting function is used is shown in column H. This, of course, applies when the weighting function is applied to the output as in Section III (but does not include the number of additions necessary for the weighting function). To calculate the number of additions when the weighting function is applied to the input, but read-only memory is not used, add columns H and K.

Column A indicates the respective cyclotomic filters described by their periods.

One important consideration that affects the choice of the order of cyclotomic filter is the noise level at the input to limiter (together with the noise in the limiter). This affects the output of the limiter when the signal level is low. One could divide the period of the signal to be detected into regions where errors could affect the decision about the sign of the signal, and regions where no errors will occur. Those sampling instances where errors could occur lie in regions where the absolute value of the signal is small. Suppose these regions are intervals of length ϵ around the zero crossings of the signal. The worst case corresponds to a phase shift of the signal with respect to the sampling interval which maximizes the number of samples in the error regions. For $\epsilon = 1/63$ (corresponding to approximately 20 dB s/n), there are at most two samples per period that are subject to errors for all the filters we have considered here. Hence the ratio of error-susceptible samples to error-free ones decreases in this case as the period p increases (for $p \leqq 30$). This ratio indicates the perturbation of the threshold one has to make in order to compensate for errors in the limiter.

VII. APPLICATIONS

Possible uses for the systems described in this paper have been mentioned in Section I. In particular, a scheme is proposed in Ref. 3 for utilizing eight cyclotomic filters as channel detectors in a *Touch-Tone* receiver.

Another application of cyclotomic filters may be FSK. As described earlier, by selecting the initial conditions of a cyclotomic filter of period p, one can approximate uniformly sampled values of a sinusoid of period p, i.e., $\sin 2\pi n/p$. By changing the clock rate of the filter, one can shift the frequency of the sinusoid to any preassigned value. Hence, when using the filter as a generator, one can shift the clock rate to shift the frequency. This method of shifting frequencies does not introduce any "discontinuities" in the signal. If, instead of changing clock rate, one were to change the coefficient of a filter, then the filter has to be reinitialized to have constant amplitude, thus producing a discontinuity in the signal. In a similar manner, when using the filter as a detector, one can shift the resonant frequency by shifting clock rate. Hence, with the same filter, one can generate and detect both tones used in a typical FSK arrangement. Furthermore, cyclotomic filters have infinite Q, allowing for the possibility of increasing signaling rate above the presently used systems with finite Q.

REFERENCES

1. L. B. Jackson, "An Analysis of Limit Cycles due to Multiplication Roundoff in Recursive Digital Filters," Proceedings of the Seventh Allerton Conference on Circuits and Systems Theory, 1969, pp. 69–78.
2. H. Breece, private communication.
3. B. Gopinath and R. P. Kurshan, "A *Touch-Tone*® Receiver-Generator with Digital Channel Filters," B.S.T.J., this issue, pp. 455–467.
4. S. Lang, *Algebra*, New York: Addison-Wesley, 1965.
5. E. Lehmer, "On the Magnitude of the Coefficients of the Cyclotomic Polynomial," Bull. Am. Math. Soc., *42* (June 1936), p. 389.
6. E. N. Gilbert, "Pulse Trains Which Lack Prescribed Harmonics," unpublished document.
7. B. Gopinath and R. P. Kurshan, "Recursively Generated Periodic Sequences," Canadian Journal of Math., *XXVI*, No. 6, 1974, pp. 1356–1371.