Joint Blind Equalization, Carrier Recovery, and Timing Recovery for High-Order QAM Signal Constellations

Neil K. Jablon, *Member, IEEE*

***Abstract*—Two existing blind equalization tap update recursions suitable for 64-point and greater QAM signal constellations are studied, along with existing and novel carrier and timing recovery techniques. It is determined that the superior tap update recursion is the one known as the "constant modulus algorithm." Carrier recovery requires a modified second-order decision-directed digital phase-locked loop. An all-digital implementation of band-edge timing recovery is used. With 14.4 kb/s outbound transmission using CCITT V.33 trellis-coded 128-QAM signals having 12.5% excess bandwidth, a prototype blind retrain procedure is developed to demonstrate feasibility of the new techniques for high-speed multipoint modems. A WE® DSP32-based real-time digital signal processor was employed to test the retrain over a set of severely impaired channels. For each channel in the set, the retrain succeeded at least 90% of the time.**

I. Introduction

DESIGNERS of voiceband multipoint network modems, digital microwave radio link modems, and transmission monitoring devices are all challenged by the requirement that in some instances their adaptive equalizers will be required to start up or restart without a training sequence. In multipoint networks (see Fig. 1), a "blind retrain" comes about if a channel from the master to one of the tributary stations goes down at any time following the initial training period, and it is desired to reconverge that line only. In digital microwave radio, the retrain is necessitated by loss of a return path when severe fading is encountered. In transmission monitoring [1], training sequences are not supplied by the transmitter. A common solution in all three of these applications is blind adaptive equalization as a bootstrapping approach designed to bring the receiver modem adaptive equalizer to the point where it can reliably switch to a steady state decision-directed algorithm. The purpose of this paper is to report on techniques for joint blind equalization, carrier recovery, and timing recovery when transmitting signals with high-order (64- and 128-point) quadrature amplitude modulation (QAM) and small (12.5%) excess bandwidth.

Manuscript received December 7, 1989; revised April 5, 1991. Portions of this paper were presented at Asilomar '87 and '88, Pacific Grove, CA; ICASSP '89, Glasgow, Scotland; and ICC '89, Boston, MA.

The author is with AT&T Bell Laboratories, Holmdel, NJ 07733-3030.

IEEE Log Number 9107652.

The 64-QAM constellation considered is the uncoded square one. The 128-QAM constellation is the trellis-coded "cross" one in CCITT Recommendation V.33 [2]. With a symbol rate of 2400 Bd, use of either constellation gives a data rate of $2400 \times 6 = 14.4$ kb/s.

The approach taken in this work is basically intuitive. Rather than utilizing analysis (which for these algorithms is extremely difficult under realistic channel conditions, because the blind adaptive equalizer error signal is a nonlinear function of the tap and data vectors) or computer simulations (which are unwieldy given the large amounts of CPU time that are necessary for the slow-converging blind equalization tap update recursions), we directly confirmed all of our techniques in the laboratory with a WE® DSP32-based real-time digital signal processor configured to transmit and receive over analog-simulated voiceband telephone channels. The only more rigorous verification of our algorithms would be testing over actual telephone channels. Our main results are as follows.

- A detailed comparison of existing tap update recursions for blind equalization of 128-QAM [3]. As originally discovered for 64-QAM in [4], it is shown that the "constant modulus algorithm" (CMA) [5]–[7] is preferred over the "reduced constellation algorithm" (RCA) [8]. Both CMA and RCA are two-dimensional analogs of the pioneering blind equalization algorithm reported in [9].
- The application of existing and novel carrier recovery techniques for demodulating the 64-QAM and 128-QAM constellations [10]. A modified second-order decision-directed digital phase-locked loop (PLL) is the preferred implementation.
- An implementation of band-edge timing recovery (BETR) [24], [32], [33], applied to synchronization of the 128-QAM constellation [11].
- The development of a prototype blind retrain for 128-QAM, in order to demonstrate the feasibility of our proposed techniques [3]. The retrain requires 16.0 s at a symbol rate of 2400 Bd.

The outline of this paper is as follows: Section II discusses tap update recursions, Section III discusses carrier

®Registered trademark of AT&T.

recovery, Section IV discusses timing recovery, Section V develops the proposed blind retrain, Section VI discusses experiments to verify real-time performance of the algorithms in Sections II to V, and Section VII contains the conclusions. The Appendix discusses alternatives to blind retrains.

II. Tap Update Recursions

Fig. 2 illustrates one branch in the multipoint network of Fig. 1. Although both the control and tributary modems contain transmitters and receivers, for blind equalization we are only concerned with transmission in the outbound direction, so in Fig. 2(a) we redraw Fig. 1 for a single tributary. In Fig. 2(b), we have shown a generic transmitter for the control modem, and in Fig. 2(c), a generic receiver for the tributary modem. The receiver utilized a "$T/4$" passband adaptive equalizer [12], where T represents the symbol period (1/2400 s for 2400 Bd transmission).

All adaptive equalization algorithms to be considered in this paper may be written as

$$\boldsymbol{c}(n+1) = \boldsymbol{c}(n) - \alpha e(n)\boldsymbol{x}(n) \tag{1}$$

where $\boldsymbol{c}(n)$, $\boldsymbol{x}(n)$, and $e(n)$ represent the complex tap vector, real[1] data vector, and complex passband error signal, respectively, at the nth symbol period. α is the positive real step size which controls convergence time and steady state excess mean-square error (MSE).

The two blind equalization algorithms that we will consider are CMA [5]–[7] and RCA [8]. When either of these algorithms converges we switch to the decision-directed (DD) algorithm [13]. For these three algorithms, let $y(n)$, $q(n)$, $\hat{b}(n)$, $\hat{a}(n)$, and $\hat{\theta}(n)$ denote the complex adaptive equalizer output, demodulated equalizer output, reduced constellation point to $q(n)$, full constellation point closest to $q(n)$, and carrier phase estimate supplied by the PLL, respectively, during the nth symbol period. Further define R^2 as the positive real constant in CMA, as shown in Fig. 3. We now have for $e(n)$:

$$\text{CMA: } e_{\text{CMA}}(n) = q(n)\left[|q(n)|^2 - R^2\right]e^{j\hat{\theta}(n)}$$
$$= y(n)\left[|y(n)|^2 - R^2\right]$$
$$\text{RCA: } e_{\text{RCA}}(n) = \left[q(n) - \hat{b}(n)\right]e^{j\hat{\theta}(n)}$$
$$\text{DD: } e_{\text{DD}}(n) = \left[q(n) - \hat{a}(n)\right]e^{j\hat{\theta}(n)}.$$

For our 128-QAM constellation having equiprobable transmitted points $a(n) \triangleq \text{Re}\,[a(n)] + j\,\text{Im}\,[a(n)]$, Re $[a(n)]$ and Im $[a(n)]$ are drawn from the set $\{\pm 1, \pm 3, \pm 5, \pm 7, \pm 9, \pm 11\}$. For 128-QAM this leads to $R^2 = 110.0976$ and $b(n) = \pm 7.454545 \pm j7.454545$ (based on the partitioning of Fig. 4). If R^2 and $b(n)$ were chosen to be other than these values, the blind equalization algorithms would converge to 128-QAM constellations with signal point levels linearly scaled to minimize the respective cost function. This may cause instabilities when switching to DD mode.

[1]The $T/4$ equalizer in [12] takes in real data, and multiplies it by complex taps. Most other equalizers take in complex data, and multiply it by either complex or real taps. For these other equalizers, $x(n)$ must be replaced by its complex conjugate.

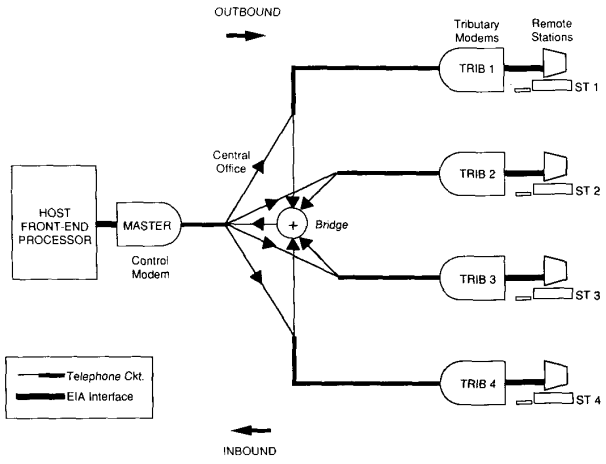


Fig. 1. Typical multipoint network.

In terms of algorithm behavior, the CMA cost function is independent of the carrier phase while the RCA cost function is not. The consequence is that, even without a PLL, CMA converges in the presence of channel frequency offset and phase fitter, but at convergence will have an arbitrary rotation. For RCA, because of possible channel frequency offset, one needs to have the PLL on even while the algorithm is converging. These properties motivate the next section.

III. Carrier Recovery

Joint blind equalization, using CMA, and conventional carrier recovery using the uncoded square 16-QAM constellation was reported in [5]. The carrier recovery system was the PLL from [13], but with frequency offset correction, shown in Fig. 5. A modified carrier recovery loop which utilized $\hat{b}(n)$ in place of $\hat{a}(n)$ as input to the phase discriminator, for the same 16-QAM constellation, was reported in [14]. "Stop-and-go" blind equalization and carrier recovery using the uncoded square 64-QAM constellation was reported in [15]. Our work consists of alternative solutions to the same problem for 64-QAM, and some new solutions for 128-QAM.

In the laboratory, we discovered that the PLL's of [5], [13], [14] were unstable when used for demodulating the 64-QAM and 128-QAM constellations. This instability is due to the large adaptation noise inherent in equalizers updated by blind equalization tap update recursions, exacerbated by the crowded signal constellations. These problems cause the slicer to make too many incorrect hard decisions on the constellation points, preventing the PLL from acquiring the correct carrier phase.

A. Carrier Recovery for 64-QAM

For 64-QAM, it was discovered that the PLL spinning problem could be solved in one of two ways. The first is

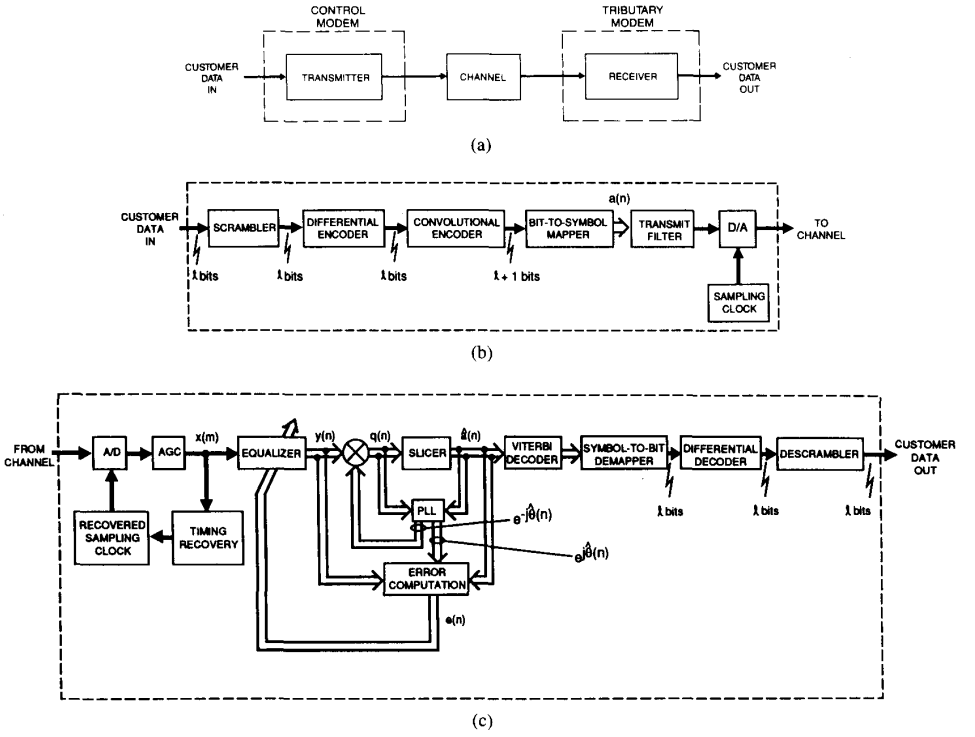


Fig. 2. Multipoint network modems: (a) modem configuration; (b) transmitter in control modem; (c) receiver in tributary modem. Signals indicated by $\rightarrow$ are real, those indicated by $=>$ are complex. Note that m and n are the sampling and symbol rate indices, respectively.

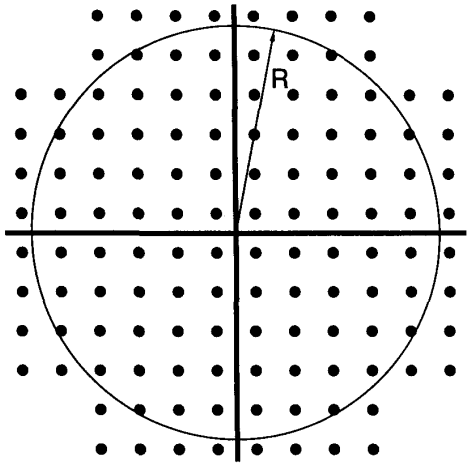


Fig. 3. Interpretation of constant R as radius in V.33 trellis-coded 128-QAM constant modulus algorithm. The signal point levels are $\{\pm1, \pm3, \pm5, \pm7, \pm9, \pm11\}$ on the real and imaginary axes, but the real and imaginary components are not independent. The uncoded 64-QAM constellation is formed from the innermost 64-point square, and thus has signal point levels of $\{\pm1, \pm3, \pm5, \pm7\}$ on both axes, with the real and imaginary components being independent.

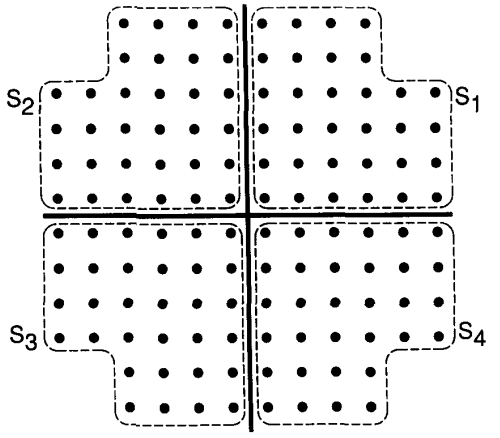


Fig. 4. Selection of reduced constellation sets in 128-QAM reduced constellation algorithm.

to reduce the large adaptation noise by reducing the equalizer step size (it is well known that adaptation noise is usually proportional to step size). We first converge the adaptive equalizer using CMA or RCA[2] with the step size which leads to the fastest convergence, while the PLL is off. At convergence of the tap update recursion, we reduce the step size, typically by a factor of 4, and wait for the adaptive equalizer to converge with the smaller step size. The PLL is still off. At convergence, we turn on the

[2]If channel frequency offset is present, one cannot use this solution with RCA.

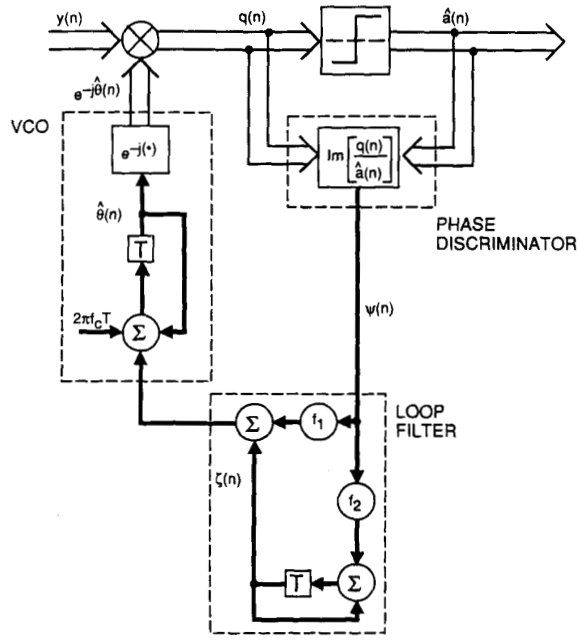


Fig. 5. Conventional phase-locked loop (PLL), with frequency offset correction.

PLL to rotate the signal constellation into place, and track any frequency offset and low frequency phase jitter. At this point, the PLL should not spin, since the adaptation noise has been greatly reduced. When the PLL is in place, we switch to DD mode.

Now a second solution will be presented for the same problem. The second solution is known to work for 16-QAM, but it was not at all obvious that it could be made to work for 64-QAM. It makes note of the fact that all constellation points are equally affected by both additive noise and adaptation noise. Based on this observation, it is safe to say that the points in the constellation that we are surest about lie at the four corners. This is because they have the longest radii, and thus their signal-to-additive-plus-adaptation-noise ratio is larger than the other sixty points. As a result, we can conceive of only performing the phase discrimination function when one of these corner points is detected. The detection is done by comparing $|q(n)|^2$ to a threshold τ. If $|q(n)|^2 \geq \tau$, we perform the phase discrimination function. If $|q(n)|^2 < \tau$, the phase discriminator output is set to zero. The loop filter is still updated normally every symbol period, because otherwise the PLL could not track frequency offset. If this scheme is implemented in real time for 64-QAM, it does indeed work in a surprisingly robust fashion. This type of PLL is termed here the "reduced constellation PLL" (RC-PLL).

The RC-PLL works well with both CMA and RCA in the presence of the common channel impairments additive noise, linear distortion (amplitude (slope) and envelope delay distortion), frequency offset, phase jitter, and nonlinear distortion (second-and third-order harmonic distortion). One disadvantage, however, is large jitter due to the reduced number of updates, which we can live with because blind equalization is only a bootstrapping procedure designed to bring us to DD mode. Once the constellation is lined up by the RC-PLL, we switch to DD mode, and once we are in DD mode we can switch our PLL to a more conventional form which updates on the full constellation, and has reduced jitter because of the tighter constellation dots.

B. Carrier Recovery for 128-QAM

With 128-QAM, not only is spinning of the PLL due to adaptation noise more serious than with 64-QAM, but two additional problems come into play. The first is a greatly increased sensitivity to phase jitter, and the second is an occasional tendency for the PLL to spin if the signal constellation has to be rotated too far when the PLL is first turned on.

Spinning of the PLL due to adaptation noise is solved by use of a combination of the strategies for 64-QAM. We first converge the adaptive equalizer using CMA or RCA[3] and the step size which leads to the fastest convergence, while the PLL is off. At convergence of the tap update recursion, we reduce the step size, typically by the same factor of 4 as for 64-QAM, and wait for the adaptive equalizer to converge with the smaller step size. The PLL is still off. At convergence, we turn on the PLL to rotate the signal constellation into place, and track any frequency offset and phase jitter. However, at this point, unlike 64-QAM, the PLL might still spin, even though the adaptation noise has been greatly reduced. Therefore, we start with the RC-PLL, and only update on the eight corner points. This scheme does not have the robustness of the 64-QAM RC-PLL, even though we are throwing away the same proportion (15/16) of the information. The main reason is that four pairs of 128-QAM corner points are only 25° apart as opposed to 90° for the 64-QAM corner points. Despite this, if we just leave the RC-PLL on long enough to rotate the large 128-QAM constellation into place (< 1 s), and then switch to the PLL which updates normally on the full constellation, followed by DD mode, the likelihood of the PLL spinning is greatly reduced. In the laboratory, it was also found helpful to add a two-sided limiter to the phase discriminator output (for the RC-PLL).

The keys to making the above procedure work are threefold. Firstly, the 128-QAM constellation dots must be tight enough when the RC-PLL is first turned on. Regardless of how the constellation quality is determined, for channels experiencing severe linear distortion, a long wait (~ 10 s) may be involved for the adaptive equalizer to converge through several stages of blind equalization tap update recursions. Second, the RC-PLL should incorporate the solutions to the other two problems of PLL

[3]If channel frequency offset is present, this will again preclude use of RCA.

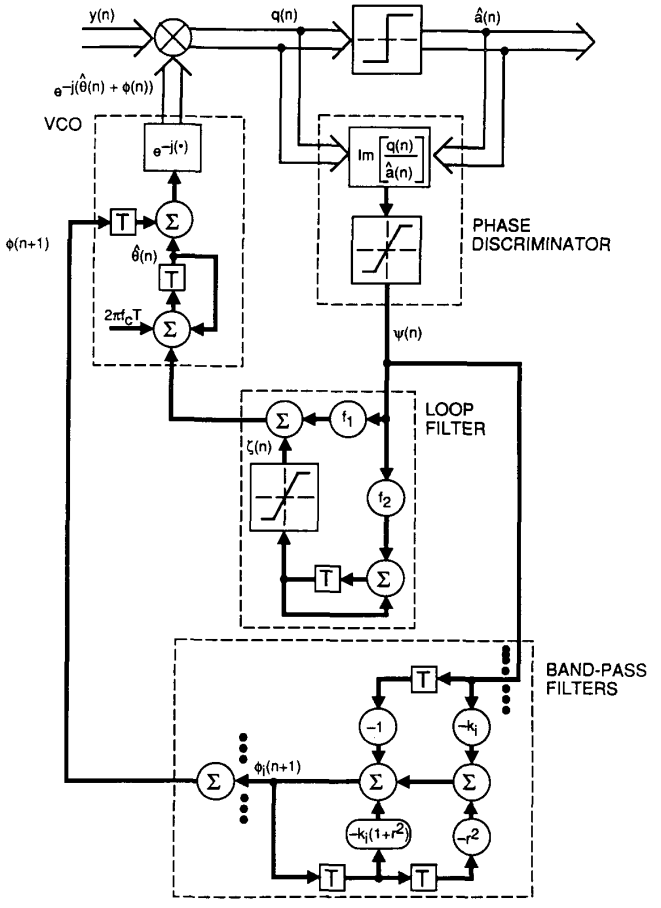


Fig. 6. Modified PLL, for joint blind equalization and carrier recovery of 128-QAM signal constellation.

spinning due to phase jitter, and large angular rotation. These solutions will now be presented.

The solution for the greatly increased sensitivity to phase jitter is quite simple. We just add bandpass filters (BPF's) at the power line harmonic frequencies in both the RC-PLL and the PLL used for updating on the full constellation. The BPF transfer function may be found in [16, eq. (50)]. Because the PLL synthesizes $e^{-j\hat{\theta}(n)}$ for use in demodulation, this effectively creates a parallel bank of notch filters at those frequencies. Fig. 6 illustrates the modified PLL. Real-time testing of this BPF with 120-Hz phase jitter confirmed that it annihilates all realistic magnitudes of phase jitter.

The solution to the large rotation problem is also straightforward. Rather than use $\hat{\theta}(0) = 0°$, we estimate whether it is closer to 0° or 45°, and "jamset" this angle into the PLL before turning it on. In order to protect ourselves against "small" frequency offset, which causes the constellation to rotate slowly, we estimate $\hat{\theta}(0)$ within a short 256-symbol window just prior to turning on the PLL. The estimation is done by counting the number of points which fit inside rotated "templates" of the constellation. The 128-QAM template is shown in Fig. 7. Assuming any frequency offset to be "small" enough so that the constellation essentially stands still during the 256-symbol window, real-time experiments have shown that this algorithm turns out to be quite successful in deciding

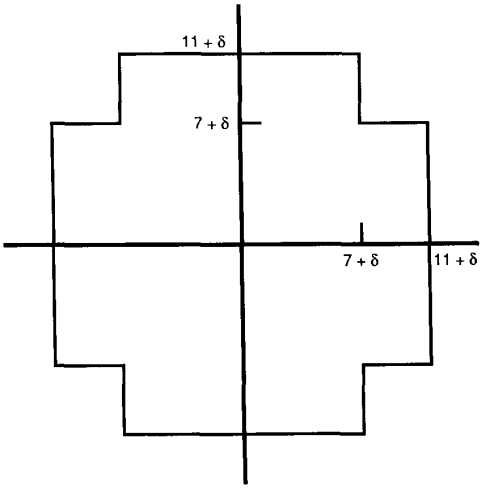


Fig. 7. 128-QAM "template" for use in estimating initial PLL angle. δ is termed the "noise immunity factor."

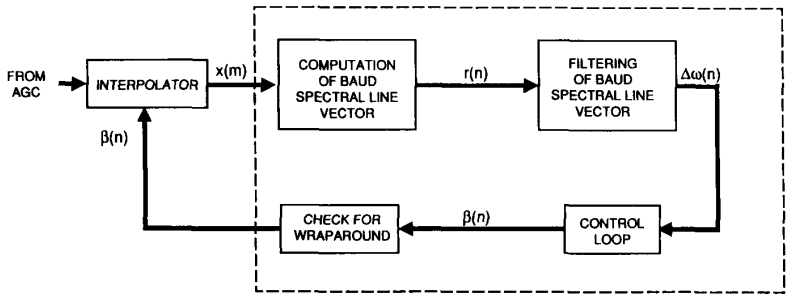


Fig. 8. Block diagram of timing recovery system.

whether the initial phase angle should be close to 45°. If the frequency offset is not "small," there is nothing lost from applying this algorithm, because all it will do is lead to a different value of $\hat{\theta}(0)$, which would be just as good, or bad, as 0°. By using more computational power, the algorithm can of course be generalized in an obvious manner to estimate $\hat{\theta}(0)$ even more accurately, say to the nearest 10° or 15°.

IV. Timing Recovery

The most important requirement for our timing recovery scheme is that it be independent of the adaptive equalizer and carrier recovery. This rules out equalizer-based [17]–[19] and decision-directed [20] approaches. However, because of the equalizer and carrier recovery dependence on the timing recovery, our scheme must acquire timing in a period that is short compared to the equalizer convergence time. Otherwise, the time required to reach DD mode would be too long. Furthermore, our timing recovery scheme must not be excessively complicated, because cost constraints require implementation in a commercial general purpose digital signal processor chip (DSP). Based on the above criteria, we decided to use a form of BETR [24], [32], [33] implemented by Farrow [11].

A general block diagram of the BETR is shown in Fig. 8. The analog-to-digital converter (A/D) sampling clock is fixed, and we vary the interpolator taps to adjust the timing. The usual method of doing timing recovery is to adjust the A/D sampling clock as in Fig. 2. However, in some applications one prefers the interpolator for other reasons, such as to minimize delay within the timing recovery system, or to shift complexity from the A/D and clock circuitry to the DSP [21]. A third possible justification for the interpolator is when one does not have the capability of adjusting the A/D sampling clock [22], as in our experimental facilities.[4] Use of the interpolator rather than adjusting the A/D sampling clock is almost transparent to the timing recovery system.

Two data rates are present in the BETR system. Signals processed at the sampling rate of 9600 samples/s are noted by the index m, and those processed at the rate of 2400 Bd are noted by the index n. The input to the interpolator [23] is a controllable fractional sample period delay $\beta(n)$, output from the timing recovery loop. The interpolator outputs a sample $x(m)$ every 1/9600 s. This sample is then used by the timing recovery to compute the baud spectral line vector $r(n)$, which is the complex representation of the recovered 2400-Hz tone. Note that without any correction from the control loop, $r(n)$ would have a frequency precisely equal to the frequency offset $\Delta\omega$ (in radians/second) between the transmitter and receiver master oscillators. *The novelty in Farrow's implementation lies in the use of a new technique for fast computation of and correction for $\Delta\omega$, within the block labeled "Filtering of Baud Spectral Line Vector."* This technique was motivated by the fact that the fractionally spaced adaptive equalizer requires a stable clock but not necessarily a given timing phase. Therefore, it is important to remove $\Delta\omega$ as soon as possible, whereas the phase "error" can be removed slowly at a later time.

A simplified representation of the block to generate $r(n)$ from $x(m)$ is shown in Fig. 9. Because of later normal-

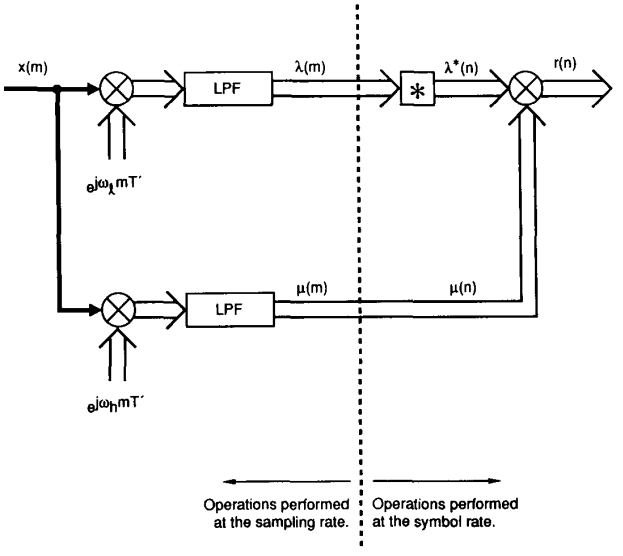


Fig. 9. Simplified baud spectral line vector computation in band-edge timing recovery system. The operation * represents complex conjugation. $\omega_l = 2\pi(600)$, $\omega_h = 2\pi(3000)$, $T' = 1/9600$.

[4]This includes the case where the incoming signal has already been digitized independently of the demodulator, e.g., the signal arrives as a digitized waveform from a time slot in a $T1$ multiplex system.

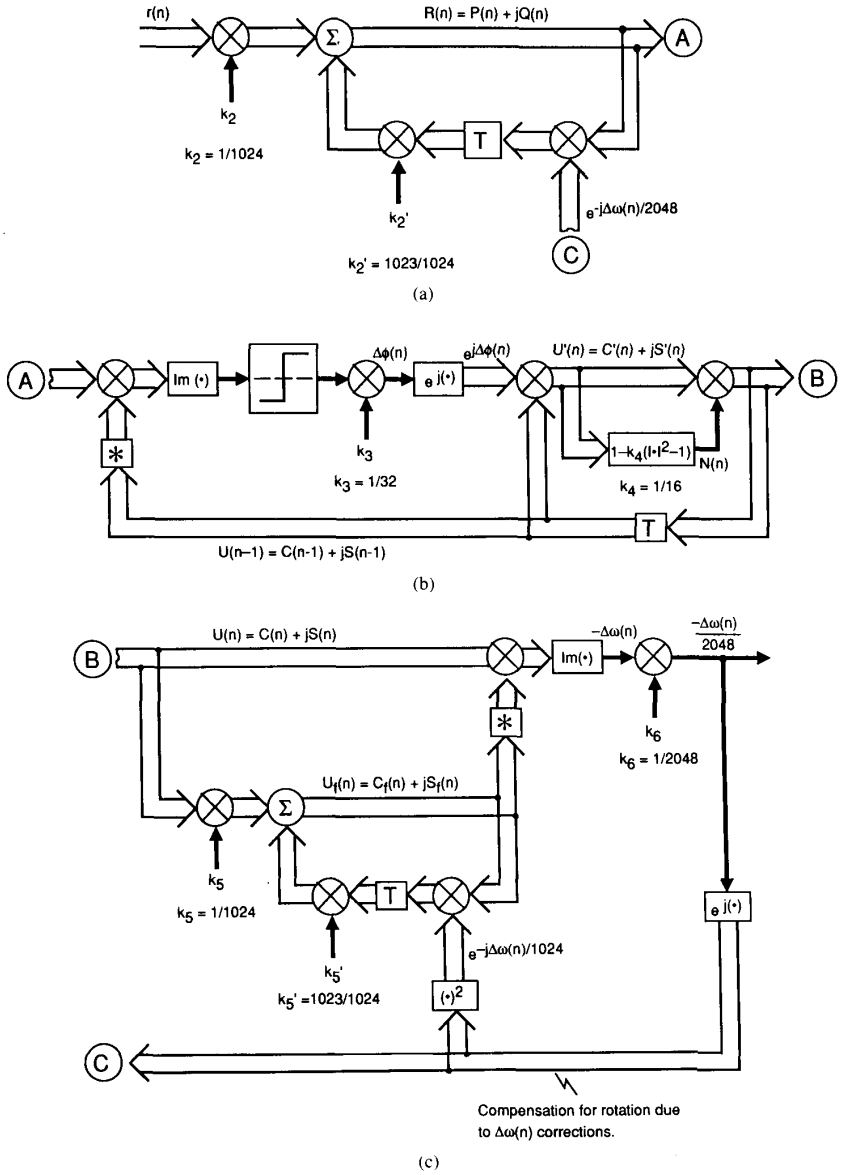


Fig. 10. Filtering of baud spectral line vector in timing recovery system. All operations are performed at the symbol rate. The operation T represents a delay of one *symbol* period. (a) Smoothing of baud spectral line vector; (b) normalization of smoothed baud spectral line vector; (c) computation of frequency offset between transmitter clock and recovered receiver clock.

ization, within this block only the *phase* $r(n)$ is of concern. Although one could simply take the imaginary part of $r(n)$ to drive the control loop [24], far better results are obtained by additional filtering. This is because we retain the valuable information about timing phase contained in the *angle* of $r(n)$. The additional filtering is illustrated schematically in Fig. 10.[5] A phasor diagram of the filtering appears in Fig. 11. The aim of the filtering and control loop (see Fig. 12) is to learn $\Delta\omega$ and cancel it out, by making $\beta(n)$ correspond to the timing drift associated with $\Delta\omega$.

The filtering begins by smoothing the noisy $r(n)$ to obtain the vector $R(n)$. We then track the vector $R(n)$ with a normalized unit vector, or phasor, $U(n)$, by rotating $U(n)$ in the direction of $R(n)$ by a small amount $\Delta\phi(n)$. This amounts to a correction

$$\Delta\phi = k_3 \operatorname{sgn}\left(\operatorname{Im}\left[\frac{R}{U}\right]\right) \cong k_3 \operatorname{sgn}\left(\operatorname{Im}\left[RU^*\right]\right)$$

$$= k_3 \operatorname{sgn}\left(R_I U_R - R_R U_I\right)$$

[5]The baud spectral line vector filtering shown in Fig. 10 illustrates the general principles behind its operation. However, in other reductions to practice, some simplification may be possible.

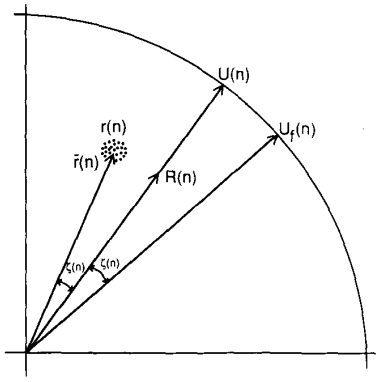


Fig. 11. Phasor diagram of filtering shown schematically in Fig. 10.

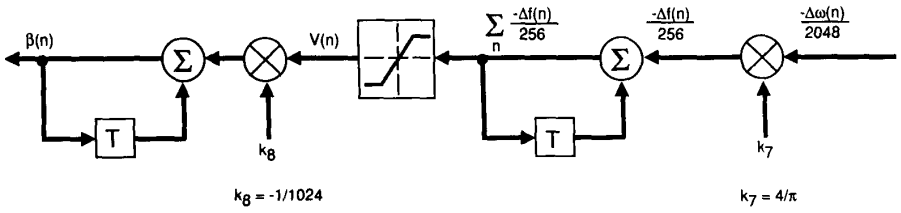


Fig. 12. Control loop in timing recovery system. All operations performed at the symbol rate.

where we have temporarily dropped symbol indices, and defined

$$R \triangleq R_R + jR_I, \qquad U \triangleq U_R + jU_I.$$

The function sgn $(\cdot)$ is implemented as

$$\text{sgn}\,(x) \triangleq \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

Letting $\Delta\omega_{\max}$ be the worst case $\Delta\omega$, the value of k_3 is chosen such that

$$\Delta\omega_{\max}\, T < k_3 < \text{angular jitter on } R.$$

The above procedure allows U to track the average value of R, while at the same time reducing the noise on U. The nonlinear nature of this operation makes it difficult to analyze. Other approaches to generating a phasor from R—for example, $R/|R|$—involve much more computation, and do not remove angular jitter.

Prior to normalization, we denote $U(n)$ by $U'(n)$, and use small angle approximations to rotate by $e^{j\Delta\phi(n)}$:

$$U'_R(n) = U_R(n-1) - \Delta\phi(n)\,U_I(n-1)$$

$$U'_I(n) = U_I(n-1) + \Delta\phi(n)\,U_R(n-1)$$

where of course

$$U'(n) \triangleq U'_R(n) + jU'_I(n).$$

$U'(n)$ is then normalized in a computationally efficient manner to obtain the phasor $U(n)$:

$$U(n) = [1 - k_4[|U'(n)|^2 - 1]]\,U'(n) = N(n)\,U'(n).$$

In order to motivate the new technique for fast computation of $\Delta\omega$, consider the following: If a "slowly rotating" phasor $F(n) = e^{j\Delta\omega nT}$ were low-pass filtered to generate a vector

$$G(n) = k'G(n-1) + kF(n)$$
$$= Ae^{j(\Delta\omega nT - \zeta)}; \qquad k \ll 1, \qquad k + k' = 1$$

it must be the case that $A \cong 1$ and $k\zeta \cong \Delta\omega T$. With reference to Fig. 10(c), we utilize this result by letting $U(n) = F(n)$, $U_f(n) = G(n)$, $A = A(n)$, $\Delta\omega = \Delta\omega(n)$, $\zeta = \zeta(n)$, and $k = k_5$:

$$\zeta(n) \cong \frac{\Delta\omega(n)\,\mathrm{T}}{k_5} \cong \text{Im}\left[\frac{U(n)}{U_f(n)}\right] \cong \text{Im}\,[U(n)\,U_f^*(n)].$$

Thus, $\Delta\omega(n)$ effectively pops right out, and can be input to a control loop[6] so that it may eventually be driven to zero. Then a phasor formed from $\zeta(n)$ is fed back to correct for rotation of the phasor diagram due to $\Delta\omega(n)$. It is straightforward to show that for slowly rotating vectors, an m times low-pass-filtered version of $r(n)$ requires a correction of $e^{jmk\zeta(n)}$, thus $R(n)$ requires a correction of $e^{jk\zeta(n)}$, and $U_f(n)$ a correction of $e^{j2k\zeta(n)}$.

Additional details concerning the design of our BETR may be found in [11].

[6]Even though we are only implementing the frequency offset correction portion of a second-order control loop, no long-term stability problems were encountered (under laboratory conditions). However, we recommend including the first-order phase correction portion of the control loop by adding a small term in the same direction as the imaginary part of $r(n)$ or any filtered version of it, i.e., Im $[r(n), R(n), U(n)$, or $U_f(n)]$ or sgn {Im $[r(n), R(n), U(n)$, or $U_f(n)]$}.

Real-time experiments [11] confirmed that the BETR delivered consistent performance, even over channels suffering from severe linear distortion. Both 32-bit floating-point and 16-b fixed-point DSP realizations were successfully tested. Furthermore, no instances were ever encountered where the BETR loop failed to converge.[7]

V. Prototype Blind Retrain

This section will develop a prototype blind retrain (PBR) procedure, in order to demonstrate the feasibility of our techniques for high-speed multipoint modems.

A. The Retrain

The first order of business is to choose a specification for our blind retrain to meet. We choose the following: The retrain must succeed at least 90% of the time over *each* channel in a set having 50% worst case background conditions and a single impairment at its 90% worst case level.[8] The 90% worst case impairment was in turn additive noise, linear distortion (amplitude (slope) and envelope delay distortion), frequency offset, phase jitter, and nonlinear distortion (second-and third-order harmonic distortion). The reasoning behind this specification is that customers will generally encounter lines that are not much more severe than the median 50% worst case line with respect to most channel impairments. However, one impairment will occasionally be present at a particularly high level, and we do not want that impairment to end up as the "Achilles' heel" of our blind retrain.

The PBR developed is enumerated below. It requires 16 s to complete when operating with a symbol rate of 2400 Bd. Recall that this is the time for a retrain, which is a relatively unusual event due to a catastrophic failure. In practice, most of the time we would only need to do a *recovery* (see the Appendix), which takes less than a second.

Prototype Blind Retrain for 128-QAM:

1) Initialize the adaptive equalizer, the carrier recovery system (PLL), and the band-edge timing recovery (BETR) system, then turn on the BETR.

2) For 3 s, update the center $8T$ span of the adaptive equalizer with the CMA tap update recursion and the step size which leads to the fastest convergence. The PLL is off.

3) For 5 s, update the full $24T$ span of the adaptive equalizer with CMA and step the size which leads to fastest convergence. The PLL is off.

4) For 3 s, update the full adaptive equalizer with CMA and the step size reduced by a factor of 4 from the previous step. The PLL is off.

5) For 1 s, update the full adaptive equalizer with CMA and the same step as in the previous step. The RC-PLL with corner point updating is turned on.

6) For 2 s, update the full adaptive equalizer with CMA and the same step size as in the previous two steps. The PLL updates on the full constellation.

7) For 2 s, update the full adaptive equalizer in the decision-directed mode and reasonably large step size. The PLL updates on the full constellation.

B. Discussion

Some explanation of the PBR will now be provided. Initialization of the adaptive equalizer is considered in Section VI. Initialization of the PLL and BETR is treated in [10], [11]. After initialization, we must converge the BETR, otherwise nothing else can happen. This takes 1 to 2 s. As soon as the BETR converges, the short equalizer will start to converge. Use of an equalizer that is $1/n$ the length of the full equalizer allows one to update with a step size approximately n times as large [25], dramatically increasing the initial convergence rate. However, the short equalizer cannot adequately compensate for linear distortion, so at the end of 3 s, we assume that the short equalizer has converged enough to lengthen the equalizer out to its full $24T$ span.[9] When lengthening the equalizer out to its full span, we keep the step size as a percentage of equalizer length roughly the same. From this point onwards, the primary concern is introducing stable carrier recovery. The adaptive equalizer step used in each stage of the PBR appear in Table I.

VI. Real-Time Experiments

This section will contain real-time experimental confirmation of both the faster convergence of CMA compared to RCA for 64-QAM and 128-QAM, and the PBR itself.

A. Experimental Setup

A simplified modem block diagram appears in Fig. 2. A schematic of the experimental setup on our homegrown real-time programmable digital signal processors, termed the Exploratory Signal Processors, version IV (ESP IV's), is shown in Fig. 13. The major differences between the two are that the receiver in the ESP IV-B was not programmed for AGC, Viterbi decoding, symbol-to-bit demapping, differential decoding, and descrambling. Additionally, the receiver in the ESP IV-B contains an interpolator. The reader will recall that we only need one outbound (master to tributary) line of the multipoint network in order to test the blind equalization algorithms.

Salient features of the setup will now be summarized.

[7]Cases where the BETR loop does fail to converge may occur when the modem signal occupies a very high percentage of the voice channel bandwidth, when the propagation characteristics are at or beyond their limit, and/or when the filtering at the band edges is very tight. A classic case is attempting to pass a V.29 or V.33 signal through a 3-kHz submarine cable voice channel.

[8]"X% worst case" for a channel impairment means that the impairment level is not exceeded on X% of all channels. "Y% worst case" for a channel means that all impairments on the channel are simultaneously at a Y% worst case level.

[9]An even greater span of $32T$ may be required when receiving over some severely slope-distorted international channels, such as submarine cables. This 33% increase in span would require additional fine-tuning of the PBR.

TABLE I
ADAPTIVE EQUALIZER STEP SIZES IN PROTOTYPE BLIND RETRAIN

Stage Number*	Algorithms	Step Size (Fraction of Signal Power**)
1	None	None
2	BETR and CMA w/out PLL	2.6×10^{-3}
3	BETR and CMA w/out PLL	6.5×10^{-4}
4	BETR and CMA w/out PLL	1.6×10^{-4}
5	BETR and CMA w/RC-PLL	1.6×10^{-4}
6	BETR and CMA w/PLL	1.6×10^{-4}
7	BETR and DD w/PLL	2.1×10^{-2}

*Corresponds to the prototype blind retrain procedure enumerated earlier.
**Measured signal power at equalizer input over back-to-back channel.
The 128-QAM signal levels are $\{\pm 1, \pm 3, \pm 5, \pm 7, \pm 9, \pm 11\}$ on both axes.
BETR—Band-edge timing recovery, CMA—Constant modulus algorithm, PLL—Phase-locked loop, RC-PLL—Reduced constellation phase-locked loop, DD—Decision-direction algorithm, w/—with, w/out—without.

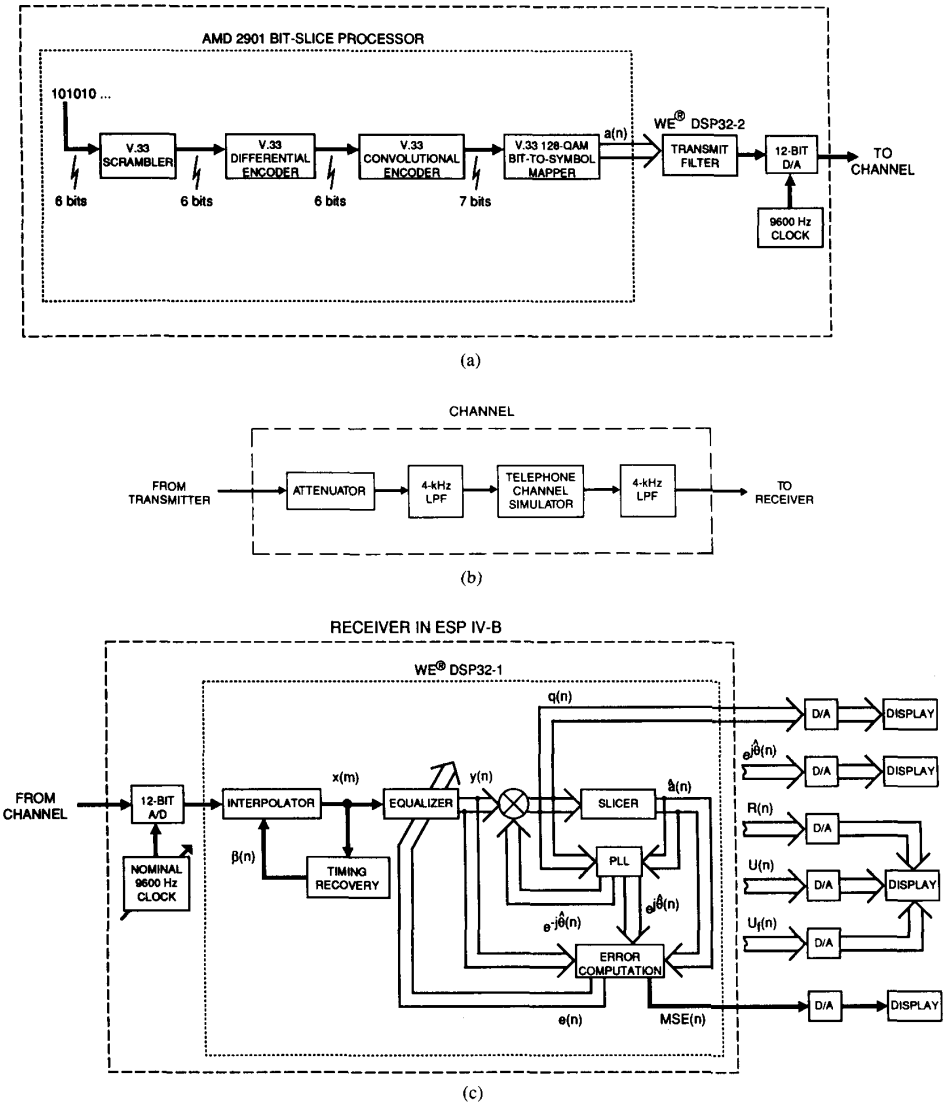


Fig. 13. Experimental setup on Exploratory Signal Processors, version IV (ESP IV's): (a) modem transmitter in ESP IV-A; (b) analog-simulated telephone channel; (c) modem receiver in ESP IV-B.

TABLE II
EXPERIMENTAL CHANNELS FOR COMPARISON OF TAP UPDATE RECURSIONS

Channel	SNR_c (dB)	EDD (μs)	Slope (dB)	Δf (Hz)	$\Delta\phi$ (deg)	$H2$ (dB)	$H3$ (dB)
A	35	1000, 950	0.0, 0.0	0	2	59	58
B	33	1170, 1350	0.4, 4.7	0	4	54	52
C	32	1850, 1350	0.4, 4.7	0	5	50	48
D	30	2300, 1850	0.8, 8.0	1*	7	42	40
E	30	2300, 1850	0.8, 8.0	0	7	42	40

*90% worst case estimate based on 1969–1970 connection survey [31].

The key to the column headings is: SNR_c—Signal to C-notched noise ratio, in decibels; EDD—Envelope delay distortion at 600 and 3000 Hz, in microseconds relative to 1700 Hz. EDD is parabolic in the entire 600–3000 Hz range; Slope—Slope (amplitude) distortion at 600 and 3000 Hz, in decibels relative to 1004 Hz. Slope is parabolic in the 600–1004 Hz range, and linear in the 1004–3000 Hz range; Δf—Frequency offset, in hertz; $\Delta\phi$—Phase jitter, in degrees peak-to-peak at 120 Hz; $H2$—Second-order harmonic distortion, in decibels; $H3$—Third-order harmonic distortion, in decibels.

The pulse shaping transmit filter interpolates a 12.5% excess bandwidth square-root Nyquist spectrum at an input symbol rate of 2400 Bds and an output sampling rate of 9600 samples/s, and modulates it up to a carrier frequency of 1800 Hz with a 3-dB bandwidth of 2400 Hz. The modulation is done via a rotator [26], [27], which for our purposes is considered to be part of the transmit filter. The "$T/4$" transmit filter coefficients are computed from [28, eq. (8b)]. With the 12-b A/D, the equalizer can perform all necessary gain adjustments, so it was decided not to include an AGC. The 12-b word from the A/D is scaled appropriately and sent to the interpolator, whose delay parameter $\beta(n)$ is output from the BETR loop [11]. The equalizer spanned 24 symbol periods. The transmit filter, interpolator, BETR, equalizer, slicer, PLL, and error computation were implemented in DSP32's, and consequently are all realized with 32-b floating point precision. No further processing was done on the slicer output, because for this study our interest is not in probability of error computation, but in simply observing the output signal constellation (i.e., demodulated equalizer output $q(n)$) and smoothed MSE.

B. Convergence Results for 128-QAM Blind Equalization Tap Update Recursions

Numerous tests of CMA versus RCA for 128-QAM were carried out over a wide variety of different channels. In order to concentrate our attention on just the tap update recursions, both the PLL and BETR were turned off, i.e., $\hat{\theta}(n) = 2\pi f_c nT$, and timing was run "under the table." The simulated channels used in the convergence results presented here included those having all of the above impairments at a 10%, 50%, 70%, and 90% worst case level on the domestic public switched network [29]. They are summarized in Table II as channels *A*, *B*, *C*, and *D*, respectively. Channel *E* is included for comparison purposes with channels *A* through *C*. The convergence results were as follows[10]:

1) The step size for CMA that led to the most rapid and reliable convergence for the set of channels was determined, and reduced by 4. With the PLL in place, the MSE at convergence was then noted.

2) With the PLL in place, for each channel tested, the step size for RCA that led to the same MSE at convergence as for CMA was determined by trial and error.

3) At least ten trials were carried out on each channel for both CMA and RCA, with convergence judged by observing by eye on an oscilloscope the time that the signal constellation fit inside the 128-QAM template from Fig. 7. The convergence time was then taken as the mean of the times recorded. The step sizes used were 4 times larger than the ones determined to lead to equal MSE with the PLL in place.

The equalizer taps were initialized by setting the center in-phase tap to 4.0, the quadrature tap advanced by one sampling period with respect to the center quadrature tap to 2.0, and the quadrature tap delayed by one sampling period with respect to the center quadrature tap to -2.0. All other taps were initialized to 0. This particular tap initialization allows the in-phase taps to very roughly approximate a square root Nyquist pulse, and the quadrature taps to very roughly approximate its Hilbert transform. The scaling used to arrive at the values 4.0, 2.0, and -2.0 was based on observing the converged equalizer tap magnitudes (recall that the equalizer performs the AGC function). Other tap initializations were tried, in particular a 96-tap truncated version of the exact square root Nyquist pulse and its Hilbert transform. However, no noticeable advantage was found over our simplified scheme.

The convergence times for the channels in Table II are summarized in Table III (channel *D* was not included because it contains frequency offset). In all cases, CMA is observed to converge faster, taking about 50% of the time required for RCA.[11] This is similar to the result obtained in [4] for 64-QAM. Additionally, RCA is seen to be

[10]In contrast to the situation for 64-QAM [4], it is quite difficult to compare the MSE of CMA and RCA using the step size that leads to the most rapid and reliable convergence of CMA. The reason is that the PLL which aligns the constellation is unstable for those step sizes.

[11]It turned out that our procedure led to converged RCA constellations that appeared to have greater MSE than the corresponding converged CMA constellations. Therefore, the result obtained, namely that CMA takes about 50% of the convergence time required for RCA, should be interpreted as an upper bound, i.e., CMA actually takes less than that for the same MSE. This further enhances the attractiveness of CMA.

TABLE III
BLIND EQUALIZATION TAP UPDATE RECURSION CONVERGENCE TIMES

Channel	Convergence Time* (s.) CMA	RCA
A	3.	7.**
B	5.	9.***
C	6.	14.****
E	8.	16.*****

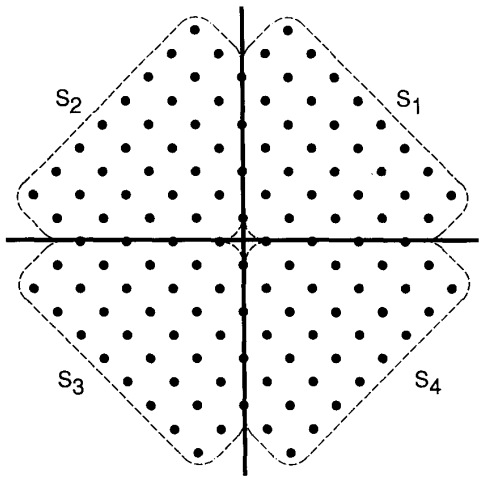
*Convergence times do not include undesired solutions (see below).
**Out of fifteen trials, one cross solution and four 45° solutions observed.
***Out of thirteen trials, one cross solution, one 45° solution, and one 144-QAM solution observed.
****Out of twelve trials, two cross solutions observed.
*****Out of eleven trials, one cross solution observed.
The experimental error is ±20%.
Carrier and timing recovery are both off.
CMA—Constant modulus algorithm. RCA—Reduced constellation algorithm.

plagued by convergence to undesired solutions. One of these solutions is the desired 128-QAM constellation, rotated by 45°. This solution, termed the "45° solution," corresponds to choosing the four reduced constellation sets as shown in Fig. 14(a). A photograph is shown in Fig. 14(b). Over time, the 45° solution can become the "cross solution," shown in two forms in Fig. 15. The third undesired solution is termed the "144-QAM solution," because it generates a received 144-QAM signal constellation, i.e., the 128-QAM constellation with the corners filled in. This solution may be related to the other two. *In any case, due to both its faster convergence and absence of undesired solutions, it is obvious that CMA is more suitable for our blind retrain.*

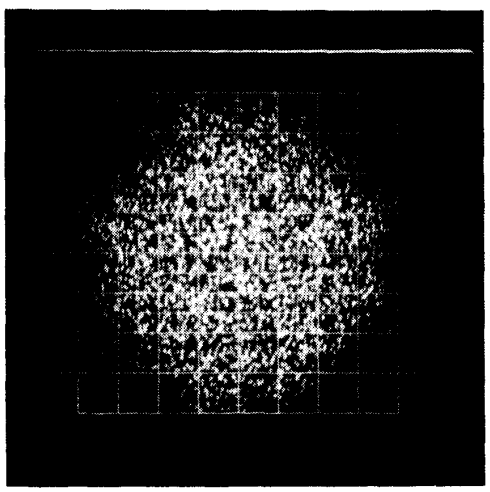
C. Experiments with the Prototype Blind Retrain

Experimental results for the PBR are presented in Table IV. Each trial was run over a channel having 50% worst case background conditions (channel *B* in Table II) and a single impairment at a higher than 50% worst case level. The tap initialization was done in the same manner as in the tests of CMA versus RCA for 128-QAM. As can be seen, if we do not count unsuccessful retrains due to the "rotated 144-QAM solution,"[12] the PBR converges at least 90% of the time on every channel tested, in accordance with our original requirements. It should also be noted that the retrain is insensitive to master oscillator frequency offset up to at least 25 ppm per set.

[12]In principle, the rotated 144-QAM solution can be avoided through redesign of the tap update recursion, because we determined that it is due to the equalizer choosing one of the solutions Im $[q(n)] = -\text{Re}\,[q(n-1)]$ or Im $[q(n)] = \text{Re}\,[q(n+1)]$. This discovery is consistent with the known properties of the $T/4$ equalizer [30]. Redesign of the tap update recursion is a subject for future study.



(a)



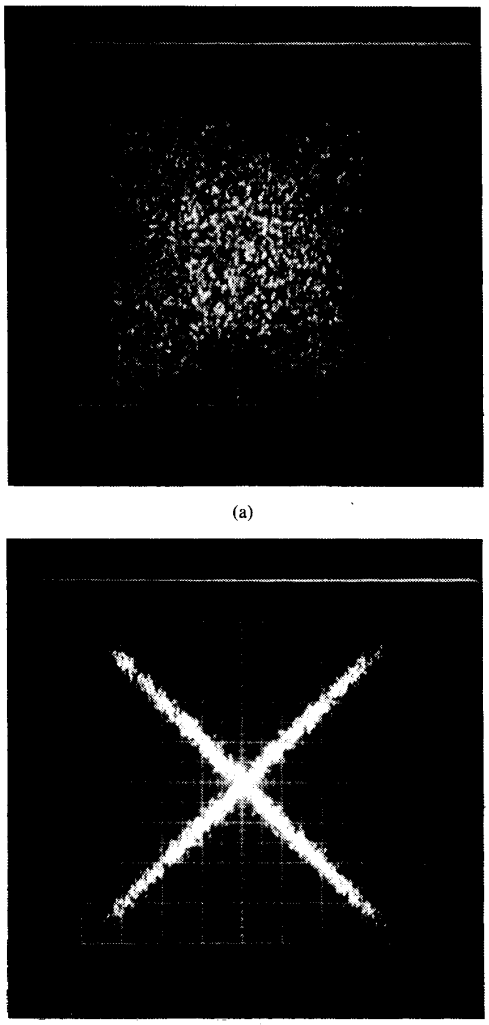
(b)

Fig. 14. Undesired "45° solution" when using reduced constellation algorithm for blind equalization of 128-QAM: (a) selection of reduced constellation sets corresponding to this solution; (b) photograph from real-time experimental setup.

Some photographs of the PBR over a 50% worst case channel are presented in Fig. 16. Fig.'s 16(a)-(g) show the signal constellation (i.e., $q(n)$) at the end of stages 1-7, respectively.

VIII. CONCLUSIONS

For voiceband modem multipoint networks, blind equalization would be used in the tributary modem receiver. Two of the most suitable blind equalization algorithms for this application are the "constant modulus algorithm" (CMA) and the "reduced constellation algorithm." For CCITT V.33 trellis-coded 128-QAM signals with 12.5% excess bandwidth, the convergence properties of these two algorithms were compared over a variety of different analog-simulated channels using a



(a) (b)

Fig. 15. Undesired "cross solutions" when using reduced constellation algorithm for blind equalization of 128-QAM: (a) form I, which follows from "45° solution" of Fig. 14; (b) form II, which follows from form I.

TABLE IV
PROTOTYPE BLIND RETRAIN OVER CHANNELS WITH 50% WORST CASE BACKGROUND CONDITIONS

Added Impairment	No. Trials	No. 144-QAM Solns.	% Successful Blind Retrains*
None	41	1	97.5
Oscillator Δf @ − 50 ppm	40	0	97.5
Oscillator Δf @ − 25 ppm	40	0	97.5
Oscillator Δf @ + 25 ppm	43	3	100.0
Oscillator Δf @ + 50 ppm	42	2	95.0
SNR_c @ 90% worst case**	40	0	95.0
LD @ 90% worst case**	40	0	95.0
Ch. Δf @ 90% worst case**	40	0	97.5
$\Delta\phi$ @ 90% worst case**	44	4	97.5
NLD @ 90% worst case**	40	0	90.0

*Unsuccessful blind retrains do not include failures due to rotated 144-QAM solution.

**90% worst case values correspond to those from channel *D* in Table II.

The experimental error is ±1 retrain.

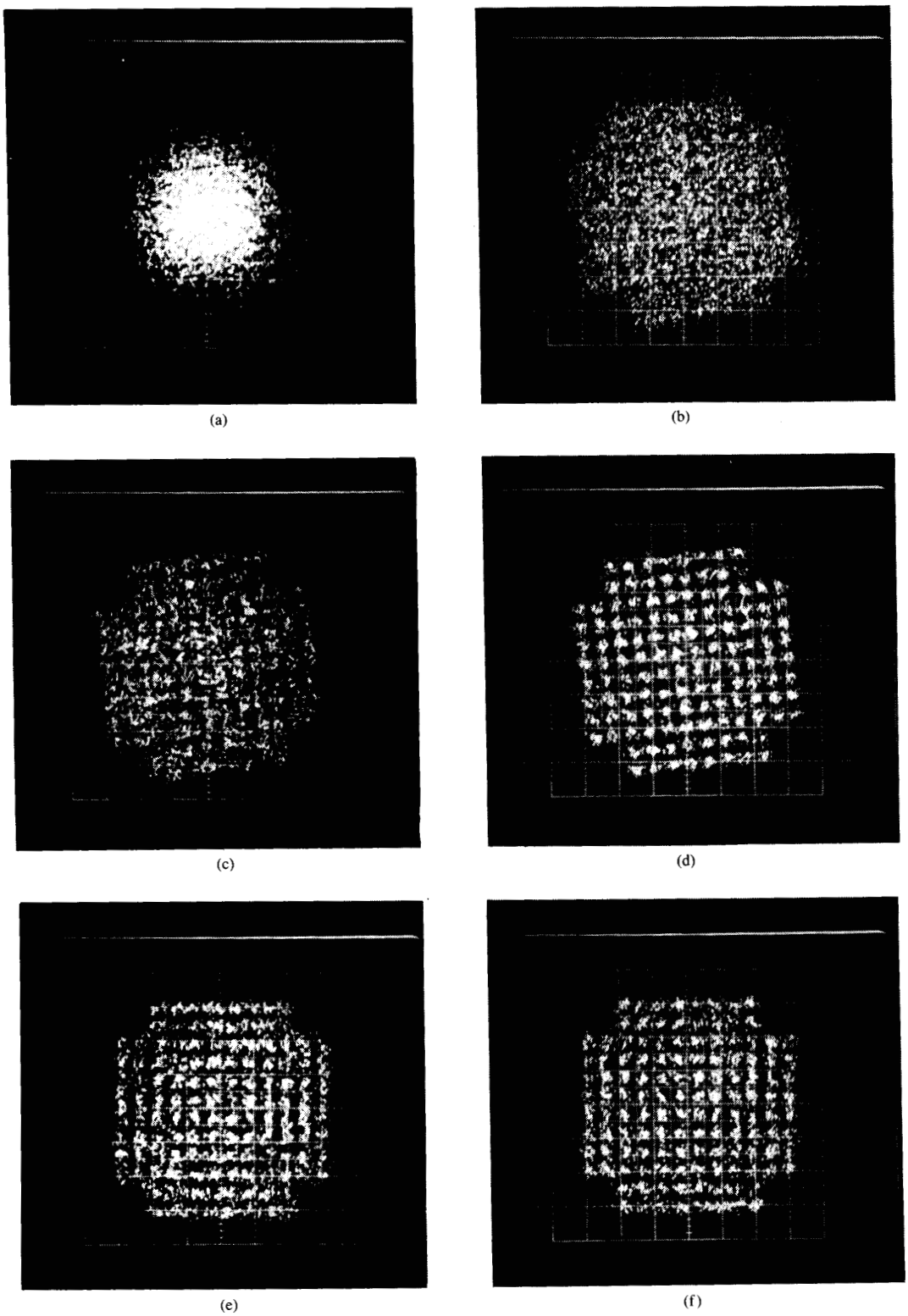
Δf—Frequency offset (refers to both master oscillators and channel); ppm—Parts per million (>0 indicates faster receiver clock), SNR_c—Signal to *C*-notched noise ratio; LD—Linear distortion (amplitude (slope) and envelope delay distortion); Ch.—Channel; $\Delta\phi$—Phase jitter at 120 Hz; NLD—Nonlinear distortion (second- and third-order harmonic distortion).

WE® DSP32-based real-time digital signal processor. CMA emerged as the undisputed victor. Additionally, several novel carrier and timing recovery implementations were developed for use with the 128-QAM and square uncoded 64-QAM constellations. The feasibility of the joint blind equalization, carrier recovery, and timing recovery techniques was demonstrated at an outbound data rate of 14.4 kb/s via a prototype blind retrain procedure. This retrain was successfully tested in real time over a variety of severely impaired channels.

APPENDIX
ALTERNATIVES TO BLIND RETRAINS

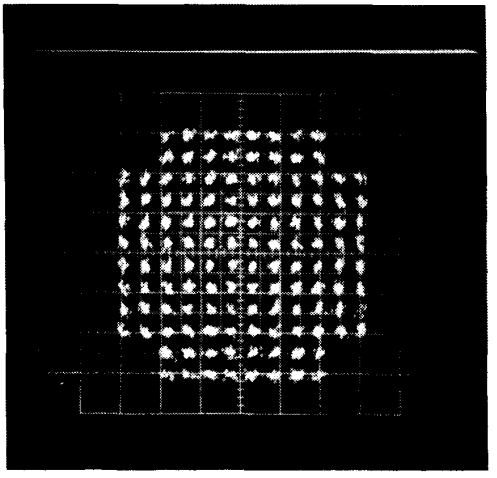
An obvious alternative to blind retrains is to periodically store the current equalizer taps. When the received data is sensed as being rapidly degraded, rather than do a blind retrain, one just substitutes the stored taps into the equalizer, and then reconverges the AGC, carrier, and timing recovery loops. Like the blind retrain, this method only involves the one line in the multipoint network that went down. The major benefit of this approach is quite rapid (< 1 s) equalizer convergence once the line comes up. This "tap substitution" approach will be termed a *recovery*, as distinguished from the blind *retrain*. Here the recovery is not part of the retrain. The recovery would cover all situations where the line transfer function was unchanged during downtime, such as a dropout, defined as a 12 dB drop in signal level lasting one second or longer. It would not cover catastrophic situations requiring a retrain, like someone at the central office accidently cutting off power to a twisted-pair connection for the line. This first alternative of recovery is thus recommended whenever possible, and it should be understood that the blind retrain strategies outlined in this paper are only for use if it cannot be done, or if it is tried without success.

A second alternative to blind equalization is for the tributary to request retransmission of the training sequence by the master, which involves reconverging the entire network. The main advantages of this approach are the speed and reliability with which the tributary modem receiver can be brought up. The primary disadvantage is disruption caused to the network. In the inbound (tributary to master) direction the disruption is due to the tributary's request interfering with inbound messages from other tributaries. For instance, one form of request consists of a tone within the primary channel. In the outbound (master to tributary) direction the disruption is due to the fact that no useful data can be sent during the time that the master is broadcasting the training sequence to everyone. This second alternative of request for retransmission of training sequence is used in practice, but only if any



(a) (b) (c) (d) (e) (f)

Fig. 16. Photographs of "prototype blind retrain" signal constellations over a 50% worst case channel: (a) end of stage 1; (b) end of stage 2; (c) end of stage 3; (d) end of stage 4; (e) end of stage 5; (f) end of stage 6. (*Continued on next page.*)



(g)

Fig. 16. (*Continued.*) (g) End of Stage 7.

recoveries or blind retrains are unavailable, or have repeatedly failed.

ACKNOWLEDGMENT

The author is grateful to his colleagues in the Data Communications Research Department at AT&T Bell Laboratories. C. W. Farrow's suggestions led to significant improvements in the section on timing recovery, which describes a scheme that he implemented. The reviewers' comments also considerably enhanced the manuscript. Special thanks go to R. D. Gitlin, V. B. Lawrence, H. G. Mattes, and H. C. Meadors, Jr., for their helpful suggestions and support of this work.

REFERENCES

[1] N. Carder *et al.*, "In-service transmission impairment testing of voice-frequency data circuits," *H-P J.*, pp. 4–15, Oct. 1987.

[2] CCITT Recommendation V.33: "14 400 bits per second modem standardized for use on point-to-point 4-wire leased telephone-type circuits," *CCITT Blue Book*, Vol. VIII-Fascicle VIII.1, pp. 252–266, Nov. 1988.

[3] N. K. Jablon, "Joint blind equalization, carrier recovery, and timing recovery for 64-QAM and 128-QAM signal constellations," in *IEEE Int. Conf. Commun. Rec.* (Boston, MA), June 11–14, 1989, pp. 1043–1049.

[4] ——, "A real-time comparison of two blind equalization algorithms," in *21st Asilomar Conf. Signals, Syst., Comput. Rec.* (Pacific Grove, CA), Nov. 2–4, 1987, pp. 452–457.

[5] D. N. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Trans. Commun.*, vol. COM-28, no. 11, pp. 1867–1875, Nov. 1980.

[6] D. N. Godard, "Method and device for training an adaptive equalizer by means of an unknown data signal in a transmission system using double sideband-quadrature carrier modulation," U.S. Patent 4 309 770, Jan. 5, 1982.

[7] J. R. Treichler and B. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, no. 2, pp. 459–472, Apr. 1983.

[8] D. N. Godard and P. E. Thirion, "Method and device for training an adaptive equalizer by means of an unknown data signal in a QAM transmission system," U.S. Patent 4 227 152, Oct. 7, 1980.

[9] Y. Sato, "A method of self-recovering equalization for multilevel amplitude-modulation systems," *IEEE Trans. Commun.*, vol. COM-23, pp. 679–682, June 1975.

[10] N. K. Jablon, "Carrier recovery for blind equalization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Glasgow, Scotland), May 23–26, 1989, pp. 1211–1214.

[11] N. K. Jablon, C. W. Farrow, and S.-N. Chou, "Timing recovery for blind equalization," in *22nd Asilomar Conf. Signals, Syst., Comput. Rec.* (Pacific Grove, CA), Oct. 31–Nov. 2, 1988, pp. 112–118.

[12] K. H. Mueller and J. J. Werner, "A hardware efficient passband equalizer structure for data transmission," *IEEE Trans. Commun.*, vol. COM-30, no. 3, pp. 538–541, Mar. 1982.

[13] D. D. Falconer, "Jointly adaptive equalization and carrier recovery in two-dimensional digital communication systems," *Bell Syst. Tech. J.*, vol. 55, no. 3, pp. 317–334, Mar. 1976.

[14] A. Benveniste and M. Goursat, "Blind equalizers," *IEEE Trans. Commun.*, vol. COM-32, no. 8, pp. 871–883, Aug. 1984.

[15] G. Picchi and G. Prati, "Blind equalization and carrier recovery using a 'stop-and-go' decision-directed algorithm," *IEEE Trans. Commun.*, vol. COM-35, no. 9, pp. 877–887, Sept. 1987.

[16] R. L. Cupo and R. D. Gitlin, "Adaptive carrier recovery systems for digital data communications receivers," *IEEE J. Select. Areas Commun.*, Dec. 1989.

[17] G. Ungerboeck, "Fractional tap-spacing equalizer and consequences for clock recovery in data modems," *IEEE Trans. Commun.*, vol. COM-24, no. 8, pp. 856–864, Aug. 1976.

[18] R. D. Gitlin and H. C. Meadors, Jr., "Center-tap tracking algorithms for timing recovery," *AT&T Tech. J.*, vol. 66, no. 6, pp. 63–78, Nov./Dec. 1987.

[19] R. L. Cupo and C. W. Farrow, "Equalizer-based timing recovery," U.S. Patent 4 815 103, Mar. 21, 1989.

[20] K. H. Mueller and M. Muller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. Commun.*, vol. COM-24, no. 5, pp. 516–531, May 1976.

[21] D. D. Harman, private communication.

[22] A. Haoui, H.-H. Lu, and D. Hedberg, "An all-digital timing recovery scheme for voiceband data modems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Dallas, TX), Apr. 1987, pp. 1911–1914.

[23] C. W. Farrow, "A continuous variable digital delay element," in *Proc. IEEE Int. Symp. Circuits Syst.* (Espoo, Finland), June 6–9, 1988, pp. 2641–2645.

[24] D. N. Godard, "Passband timing recovery in an all-digital modem receiver," *IEEE Trans. Commun.*, vol. COM-26, no. 5, pp. 517–523, May 1978.

[25] G. Ungerboeck, "Theory on the speed of convergence in adaptive equalizers for digital communication," *IBM J. Res. Develop.*, pp. 546–555, Nov. 1972.

[26] J. J. Werner, "Modulated passband signal generator," U.S. Patent 4 015 222, Mar. 29, 1977.

[27] S. U. H. Qureshi, "Digital modem transmitter," U.S. Patent 4 358 853, Nov. 9, 1982.

[28] P. R. Chevillat and G. Ungerboeck, "Optimum FIR transmitter and receiver filters for data transmission over band-limited channels," *IEEE Trans. Commun.*, vol. COM-30, no. 8, pp. 1909–1915, Aug. 1982.

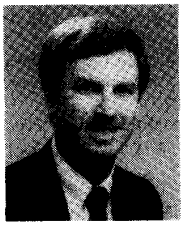
[29] M. B. Carey *et al.*, "1982/83 end office connection study: Analog voice and voiceband data transmission performance characterization of the public switched network," *AT&T Tech. J.*, vol. 63, no. 9, pp. 2059–2119, Nov. 1984.

[30] J. D. Wang and J. J. Werner, "On the transfer function of adaptive $T/4$ equalizers," in *22nd Asilomar Conf. Signals, Syst., Comput. Rec.* (Pacific Grove, CA), Oct. 31–Nov. 2, 1988, pp. 260–264.

[31] F. P. Duffy and T. W. Thatcher, Jr., "1969–70 connection survey: Analog transmission performance on the switched telecommunications network," *Bell Syst. Tech. J.*, vol. 50, no. 4, pp. 1311–1347, Apr. 1971.

[32] L. E. Franks and J. P. Bubrouski, "Statistical properties of timing jitter in a PAM timing recovery system," *IEEE Trans. Commun.*, vol. COM-22, no. 7, pp. 913–920, July 1974.

[33] R. D. Gitlin and J. F. Hayes, "Timing recovery and scramblers in data transmission," *Bell. Syst. Tech. J.*, vol. 54, no. 3, pp. 569–593, Mar. 1975.



Neil K. Jablon (S'78-M'86) was born in Albany, NY, on June 7, 1959. He received the B.E. degree from the State University of New York at Stony Brook in 1981, and the M.S. and Ph.D. degrees from Stanford University in 1982 and 1985, respectively, all in electrical engineering. While at Stanford, he also studied the spoken and written Chinese language (Mandarin dialect) for three years. During Fall 1987, he studied classical Chinese at Rutgers University, NJ.

Since 1986 he has been with AT&T Bell Laboratories, where he currently performs systems engineering for consumer electronics products. From 1989 to 1992, he performed systems engineering, systems testing, and design of quality processes for original equipment manufacturer (OEM) products purchased by AT&T from several East Asian companies. Previously at AT&T (1986-1989), he conducted exploratory studies in digital signal processing for data modem and facsimile applications. From 1983 to 1986, he was with the Information Systems Laboratory, Stanford University, Stanford, CA, where he researched adaptive antenna beamforming algorithms. He has several publications dealing with adaptive filtering for antenna and data modem applications. Currently, his major interests are data modems, facsimile, quality control, the East Asian electronics business, and nearly all aspects of sinology.

Dr. Jablon is a member of Eta Kappa Nu, Sigma Xi, and Tau Beta Pi. He received the Ward Melville Valedictory Award (SUNY, 1981), the National Space Club Robert H. Goddard Scholarship (1981), a Tau Beta Pi Fellowship (1981-1982), and a Fannie and John Hertz Foundation Graduate Fellowship (1981-1985). He has contributed to organizing the Asilomar Conference on Signals, Systems, and Computers each year since 1987. In 1991, he served as Technical Program Chairman of that conference. During 1992, he is serving as General Chairman of the same conference.