TIMING RECOVERY FOR BLIND EQUALIZATION

Neil K. Jablon, C. W. Farrow, and Shao-Ning Chou

AT&T Bell Laboratories
200 Laurel Avenue
Middletown, NJ 07748

Abstract- **Timing recovery for blind equalization requires a non-equalizer-based, non-decision-directed approach, due to slow convergence of the adaptive equalizer taps and the resulting initial unreliability of the data decisions. The leading possibilities for timing recovery are either band-edge, or conventional envelope-derived using the common nonlinearities of magnitude, square, or fourth-power. A novel implementation is proposed for band-edge timing recovery. With this method, the vector representing the baud spectral line is derived by digital correlation techniques, and then this vector is filtered in such a way that the timing recovery loop can attain fast convergence while tracking worst case frequency offset between the transmitter and receiver master oscillators. It is shown experimentally that for signals chosen from the CCITT V.33 trellis-coded 128-QAM constellation having 12.5% excess bandwidth and square-root Nyquist shaping at the transmitter, band-edge timing recovery is preferred over conventional envelope-derived, as it is the only one which can recover the baud spectral line in a reasonably jitter-free manner over both moderately and severely impaired channels.**

1. INTRODUCTION

Blind equalization, namely startup of an adaptive equalizer without a predetermined training sequence, provides special challenges in designing an appropriate timing recovery system. Slow convergence of the blind equalization tap update recursions dictates a non-equalizer-based timing recovery scheme. Initial unreliability of the data decisions further dictates a non-decision-directed timing recovery scheme. Thus, tap tracking [1-2] and decision-directed [3] timing recovery schemes are both automatically eliminated. Further requirements on the timing recovery system are that it must not be excessively complicated, it must not drive the blind equalization tap update recursion and carrier recovery system unstable during joint adaptation, and it must converge rapidly (say within 1 or 2 seconds) while tracking worst case frequency offset between transmitter and receiver master oscillators. These requirements naturally lead one to consider conventional envelope-derived timing recovery (EDTR) [4] and band-edge timing recovery (BETR) [5]. The purpose of this paper will be to investigate all-digital passband implementations of the latter two approaches for trellis-coded 128-QAM signals obeying CCITT Recommendation V.33 [6], in the case of a pulse-shaping transmit filter having 12.5% excess bandwidth and square root Nyquist shaping.

If the received signal were prefiltered (i.e., the band edges filtered with narrowband bandpass filters (BPFs) having responses "optimally" chosen to minimize timing jitter over an ideal channel [4-5,7]) and EDTR then performed, the result would just be an implementation of BETR. In this paper, the term EDTR will always refer to a lack of prefiltering, and BETR will refer to an implementation of EDTR which performs prefiltering, even if it does not do so with actual BPFs. Hybrid schemes will not be considered. No other forms of receive filtering were employed for this study.

Our main results are:

- A novel implementation of BETR, specially designed for fast convergence while tracking worst case frequency offset between the transmitter and receiver master oscillators.
- Experimental determination that:
  - — EDTR using any of the common nonlinearities of magnitude, square, or fourth-power to generate the baud spectral line performed poorly, even over back-to-back channels.
  - — BETR, on the other hand, delivered consistent performance even over channels suffering from severe linear distortion, i.e., severe amplitude and envelope delay distortion.

The outline of this paper is as follows: Section 2 describes our implementation of BETR, Section 3 describes our implementation of EDTR, and Section 4 ends with some conclusions. Experimental results are interspersed with explanations of the timing recovery techniques. Joint blind equalization and carrier recovery is dealt with in [8-11].

2. BAND-EDGE TIMING RECOVERY

Figure 1 shows a real-time experimental setup to simulate one branch of a typical multipoint network [10]. Although both the control and tributary modems contain transmitters and receivers, for blind equalization we are only concerned with transmission in the *outbound* direction. Note that the receiver analog-to-digital converter (A/D) sampling clock is fixed, and we vary the interpolator taps to adjust the timing. The receiver also employs the so-called "T/4" passband adaptive equalizer [12]. The carrier frequency is 1800 Hz, the sampling rate is 9600 samples/sec., and the symbol rate is 2400 bauds.

2.1 Interpolator

The interpolator [13] is a finite impulse response filter which synthesizes a controllable fractional sample period delay $\beta(n)$ in

22ACSSC-12/88/0112 $1.00

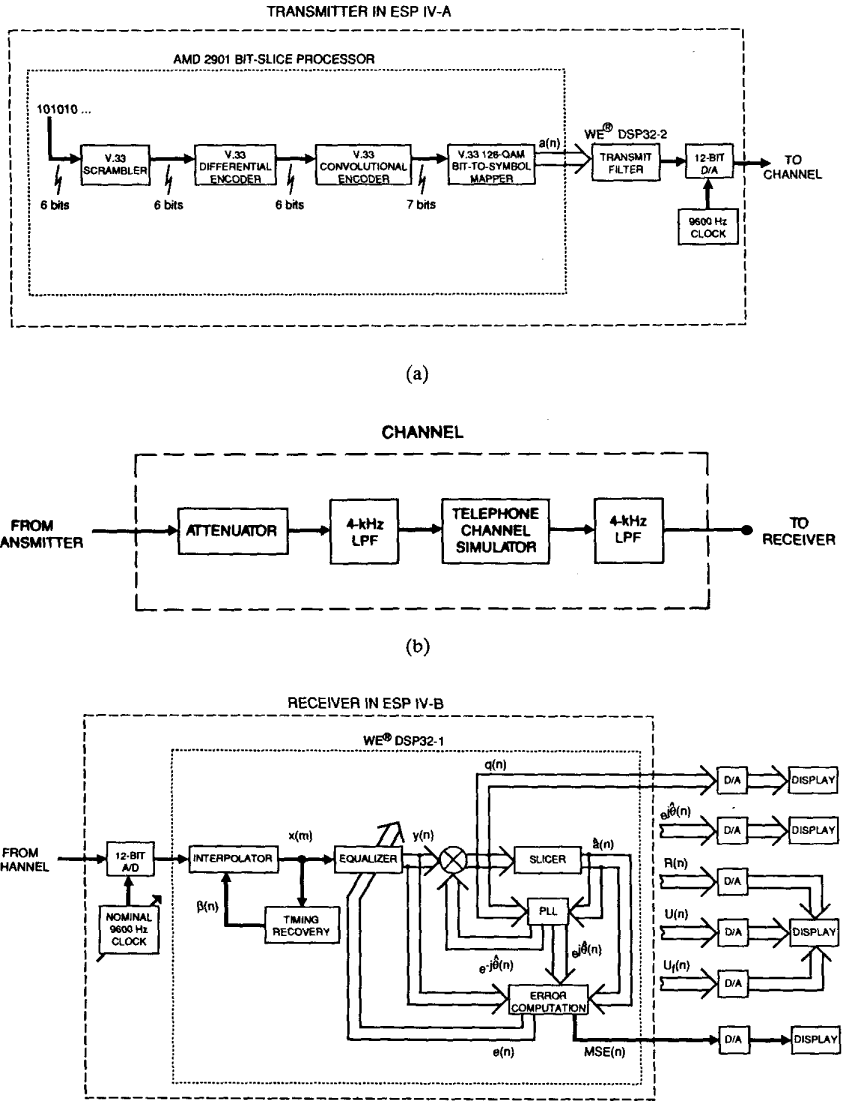


Fig. 1. Experimental setup on Exploratory Signal Processors, version IV (ESP IVs): (a) modem transmitter in ESP IV-A; (b) analog-simulated telephone channel; (c) modem receiver in ESP IV-B. Signals indicated by → are real, => are complex, and m and n are the sampling and symbol rate indices, respectively.

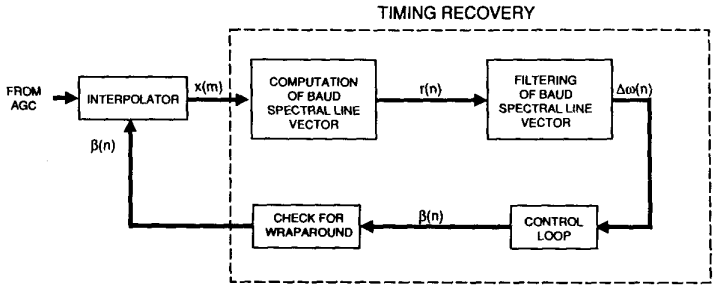


Fig. 2. Block diagram of timing recovery system.

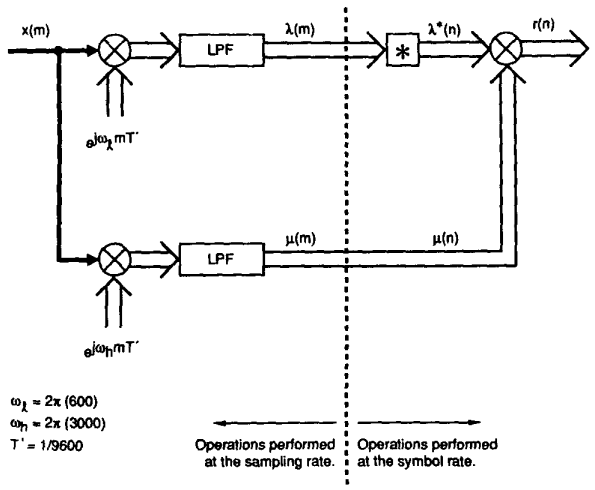


Fig. 3. Simplified baud spectral line vector computation in band-edge timing recovery system. The operation * represents complex conjugation.

the range [–0.5,0.5]. $\beta(n)$ is output from the timing recovery loop. Updating of the interpolator taps is done at the symbol rate, while computation of the interpolator output is done at the sampling rate. Our interpolator had a tap spacing of 1/9600 sec., an upper useful frequency of 3150 Hz, and a root-mean-square error of –50 dB.

2.2 Computation of Baud Spectral Line Vector

A general block diagram of the timing recovery systems to be considered in this paper is shown in Fig. 2. Two data rates are present. Signals processed at the sampling rate of 9600 samples/sec. are noted by the index m, and those processed at the symbol rate of 2400 bauds are noted by the index n. The interpolator outputs a sample $x(m)$ every 1/9600 sec. This sample is then used by the timing recovery to compute the baud spectral line vector, which is the complex representation of the recovered 2400 Hz tone. A simplified representation of this block for BETR is shown in Fig. 3. The upper arm serves to translate the negative frequency lower band edge (~–600 Hz) to DC, where it is lowpass filtered to retain only the energy near the band edge. This is also known as performing a digital correlation. The lower arm does the same, but for the negative frequency upper band edge (~–3000 Hz). The two band-edge components are then multiplied to arrive at the baud spectral line vector $r(n)$. Because of later normalization, within this block only the *phase* of $r(n)$ is of concern.

Letting $r(n) \triangleq |r(n)|\, e^{j\ \mathrm{ang}\left(r(n)\right)}$, and using an automatic gain control (AGC) to insure $|\lambda(n)|^2 = 1$:

$$\mathrm{ang}\left(r(n)\right) = \mathrm{ang}\left\{\frac{\mu(n)}{\lambda(n)}\right\} = \mathrm{ang}\left(\mu(n)\,\lambda^*(n)\right)$$

since the angle of a vector must be independent of its magnitude.

The LPFs were implemented in the form

$$v(n) = k\,u(n) + k'\,v(n-1)$$

where $u(n)$ and $v(n)$ are the LPF input and output, respectively, at the n-th iteration, the constant k is in the range $0 < k \ll 1$, and $k' \triangleq 1 - k$. With ω denoting radian frequency and T' representing the sampling rate, the LPF bandwidth $f_{3\,\mathrm{dB}}$ is

$$f_{3\,\mathrm{dB}} = \frac{1}{2\pi T'}\cos^{-1}\left\{\frac{-(k^2+2k-2)}{2\,(1-k)}\right\} \cong \frac{k}{2\pi T'} \quad \mathrm{Hz}$$

which in our case leads to a bandwidth of 24 Hz, or 1% of the symbol rate, since $k_1 = 1/64$ and $T' = 1/9600$ sec.

Comparing our implementation of the BETR to Godard's [5], it can be shown [14] that our computation of the baud spectral line vector and his are equivalent to within a constant phase shift, provided that his "baseband equivalent LPF" is identical to our LPF, and the baud spectral line vector is only utilized at the symbol rate. In most applications (including ours) the baud spectral line vector is indeed only processed at the symbol rate.

2.3 Filtering of Baud Spectral Line Vector

Although one could simply take the imaginary part of $r(n)$ to drive the control loop [5],[1] far better results are obtained by additional filtering. The block which accomplishes this is illustrated in Fig. 4. The first order of business is to smooth $r(n)$, because it is extremely noisy, varying greatly in both magnitude and phase. For the moment, we neglect the rotation of $R(n-1)$ by $e^{-j\Delta\omega(n)/2048}$, and compute the bandwidth of this second LPF as

$$f_{3\,\mathrm{dB}} \cong \frac{k}{2\pi T} = 0.37 \ \mathrm{Hz}$$

since $k_2 = 1/1024$ and $T = 1/2400$ sec. This choice of k_2 is based on tracking the worst case frequency offset between the transmitter and receiver master oscillators, without any rotation of $R(n-1)$. We assume the worst case frequency offset as 25 parts per million (ppm) per set. With worst case transmitters and receivers, the total frequency offset is 50 ppm, so the LPF must have a bandwidth of at least $2400 \times 50 \times 10^{-6} = 0.12$ Hz.

After computing $R(n)$, we normalize it. The normalized version of $R(n)$ is termed $U(n)$, for unit vector. It is generated by first computing the sign of the angle between $R(n)$ and $U(n-1)$, under the assumption that this angle is small:

$$\text{sign of the angle between } R(n) \text{ and } U(n-1)$$

$$\cong \mathrm{sgn}\left\{\mathrm{Im}\left[\frac{R(n)}{U(n-1)}\right]\right\}$$

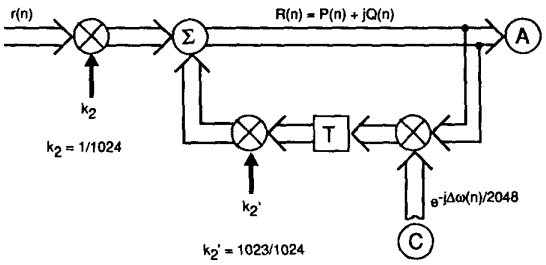
$$\cong \mathrm{sgn}\left\{\mathrm{Im}\left[R(n)\,U^*(n-1)\right]\right\}$$

where the function sgn(·) is implemented as

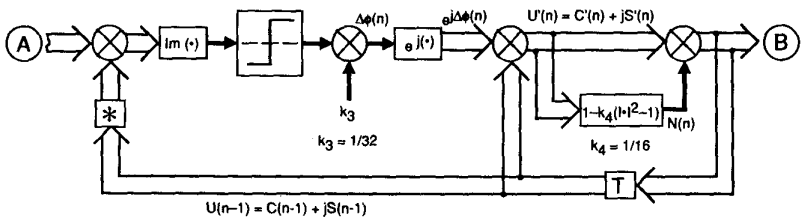
$$\mathrm{sgn}(x) \triangleq \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

and we note that the $\mathrm{sgn}\left\{\mathrm{Im}[\cdot]\right\}$ operation will be unaffected by dropping the $|U(n-1)|^2$ term in the denominator. Then the sign of the angle is scaled by $k_3 = 1/32$ to obtain $\Delta\phi(n)$, and next converted to the phasor $e^{j\Delta\phi(n)}$. The choice of k_3 is based on

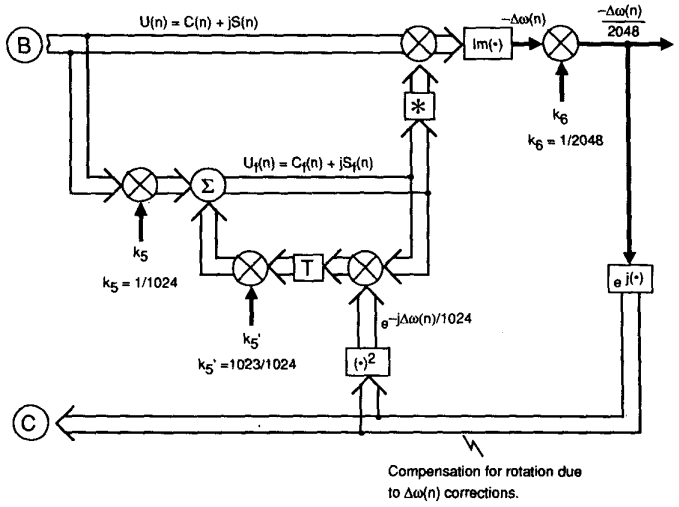
1. It will be seen that the advantage of our procedure is that we retain the valuable information about timing phase contained in the *angle* of $r(n)$.



(a)



(b)



(c)

Fig. 4. 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.

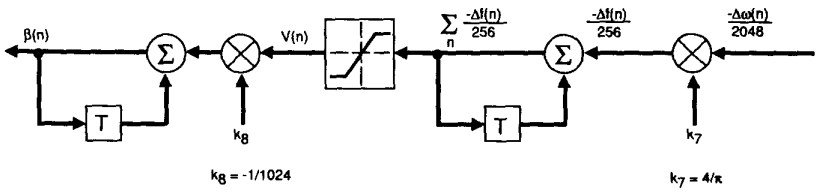


Fig. 5. Control loop in timing recovery system.

latching onto the worst case value of the angle between $R(n)$ and $U(n)$, which is $\pm\pi$ radians, within approximately 100 symbols. This value of k_3 is designed to be large enough to rapidly acquire $R(n)$, but not so large as to introduce unacceptable noise. In a practical implementation one utilizes the approximation for small angles to implement the angle to phasor conversion $e^{j\Delta\phi(n)}$.

Rotation of $U(n-1)$ by $e^{j\Delta\phi(n)}$ allows it to rapidly track $R(n)$, especially during the transient phase. However, because $|1 + j\Delta\phi(n)| \cong 1 + \Delta\phi^2(n)/2$ there would be a systematic growth in $U(n)$, so we renormalize $U(n)$ by computing

$$U(n) = \left(1 - k_4\left(|U'(n)|^2 - 1\right)\right) U'(n) = N(n)\, U'(n)$$

using $k_4 = 1/16$. This way, numerical errors cannot accumulate and drive the magnitude of $U(n)$ away from unity. The choice of k_4 was based on empirical optimization, but other values up to 1/2 may be reasonable for different applications.

From here on in, because the steady state rates of rotation are small, all vectors will be treated as unit vectors. Therefore, all angles can be well-approximated by the imaginary part of their corresponding unit vector.

Once $U(n)$ is available, it is filtered again to get $U_f(n)$, the subscript representing filtering. It is convenient to choose the bandwidth of the latter LPF the same as the bandwidth of the LPF which generated $R(n)$ from $r(n)$, because it makes the lag between $R(n)$ and $U(n)$ equal to the lag between $U(n)$ and $U_f(n)$. The lag between $U(n)$ and $U_f(n)$ is used to our advantage by computing the angle between them, assuming that this angle is small:

$$\text{angle between } U(n) \text{ and } U_f(n) \cong \text{Im}\left[\frac{U(n)}{U_f(n)}\right] \cong \text{Im}\left[U(n)\, U_f^*(n)\right]$$

where $|U(n)|^2 \cong |U_f(n)|^2 \cong 1$.

The angle between $U(n)$ and $U_f(n)$ is directly proportional to the frequency offset $\Delta\omega(n)$ between the transmitter clock and recovered receiver clock, so after exiting the second Im [·] block, we scale the result by $k_6 = 1/2048$ to obtain the term $-\Delta\omega(n)/2048$. Note that $\Delta\omega(n)$ is forced to zero in steady state. Due to scaling in the control loop (see below), it is easily verified that during the transient state, the term $V(n)$ represents the summation of 1/256 of the frequency offset in Hz. The choice of k_6 follows from aiming for removing $1/e$ of $\Delta\omega(n)$ within 256 symbols, assuming every step is in the right direction. In order to allow us to use a large step to speed up convergence of the timing recovery loop, when we adjust the control loop to compensate for $\Delta\omega(n)$, we also undo the rotation in $R(n)$ and $U_f(n)$ due to $\Delta\omega(n)$ via multiplication by the phasors $e^{-j\Delta\omega(n)/2048}$ and $e^{-j\Delta\omega(n)/1024}$, respectively. This effectively prevents the quantity $\Delta\omega(n)$ from overshooting. In a sense, the system is critically damped. The extra lag by $e^{j\Delta\omega(n)/2048}$ of $U_f(n)$ is due to the extra filtering operation required to generate it. In the actual implementation, both rotations are realized with the small angle approximation.

2.4 Control Loop

Proceeding on to the control loop shown in Fig. 5, $-\Delta\omega(n)/2048$ is scaled again by $k_7 = 4/\pi$ to convert from $\Delta\omega(n)$ to $\Delta f(n)$,[2] and then summed and limited to obtain the velocity term $V(n)$. $V(n)$ is a scaled estimate of the frequency offset between the transmitter and receiver master oscillators. The scaling factor for converting ppm timing drift to $\beta(n)$ timing drift is 4×10^{-5}. The limit V_{lim} on $V(n)$ comes about by considering the worst case frequency offset between the transmitter and receiver master oscillators. Assuming this number is 50 ppm, with reference to Fig. 5 we obtain

$$|k_8\, V(n)| \triangleq |\Delta\beta(n)| \leq 4 \times 10^{-6} \times 50 = 200 \times 10^{-6}$$

so that

$$|V(n)| \leq \frac{200 \times 10^{-6}}{|k_8|} = \frac{200 \times 10^{-6}}{1/1024} = 0.2048 \triangleq V_{\text{lim}}$$

In the actual implementation, one may wish to choose $V_{\text{lim}} = 0.2048 + \epsilon$ in order to have some "breathing room." In our implementation, it was found that $V_{\text{lim}} = 0.2050$ worked fine. The choice $k_8 = -1/1024$ was made empirically, and probably depends to a large degree on what values were used for the other constants.

$V(n)$ is termed the velocity, because it is the rate at which $\beta(n)$ must change to precisely track the constant frequency offset. In steady state, this velocity is also a constant, meaning that if $\beta(n)$ is viewed on an oscilloscope it moves smoothly.

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 the baud spectral line vector 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 $U_f(n)$]).

2.5 Check for Wraparound

When $\beta(n)$ comes out from the control loop, it must first be checked for wraparound, i.e., $\beta(n)$ outside the range $-0.5 \leq \beta(n) \leq 0.5$. Proper handling of this situation was accomplished with a sample add/delete algorithm.

2.6 Computational Complexity

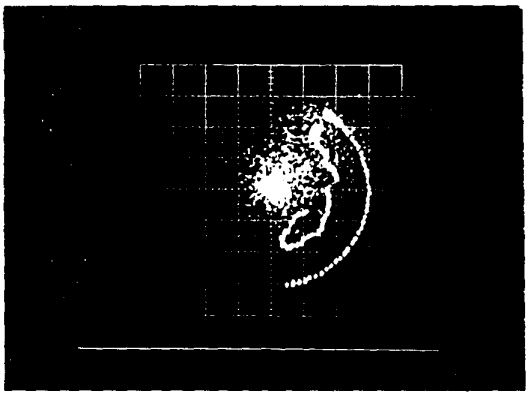
First we give figures for the latest WE® DSP32 floating point digital signal processor with an instruction cycle time of 160 nanoseconds. During an average symbol period, a total of 535 instructions (85.6 μsec.) were dedicated to the BETR. This should represent approximately 15% of the total computation. We also used a WE® DSP16 fixed point digital signal processor with an instruction cycle time of 75 nanoseconds, and adjusted the A/D sampling clock rather than employ the interpolator. During an average symbol period, a total of 613 instructions (45.975 μsec.) were dedicated to the BETR. This represented approximately 10% of the total computation for a symbol period.

2.7 Experimental Results

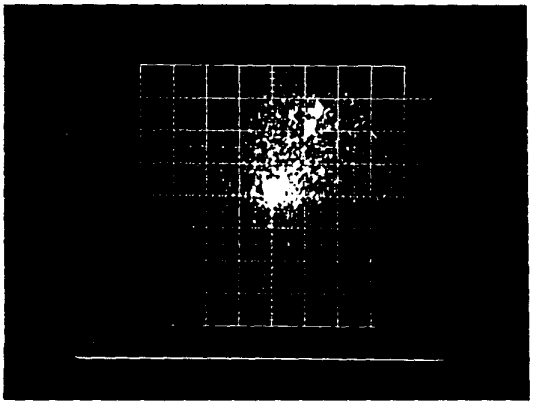
The quantities that we are interested in viewing are $r(n)$, $R(n)$, $U(n)$, and $U_f(n)$. Fig. 6(a) shows a roughly two-second time

2. This should be viewed as scaling $-\Delta\omega(n)/1024$ by $2/\pi$. The $2/\pi$ comes from converting radian-symbols to cycle-samples.

® Registered trademark of AT&T.



(a)



(b)

Fig. 6. Baud spectral line vector $r(n)$ and filtered quantities $R(n)$ and $U(n)$ with band-edge timing recovery over a back-to-back channel with 50 ppm frequency offset between the transmitter and receiver master oscillators: (a) approximately two-second exposure of transient state from startup; (b) approximately two-second exposure of steady state.

exposure of $r(n)$, $R(n)$, and $U(n)$ from the startup, over a back-to-back channel with 50 ppm frequency offset between the transmitter and receiver master oscillators. Fig. 6(b) shows the steady state situation. In both photographs, $r(n)$ appears as a "comet-like" cloud, $R(n)$ as a noisy vector of magnitude less than one, and $U(n)$ as a smooth dotted vector of unity magnitude. In the transient photograph, $R(n)$ and the unity vector $U(n)$ are both seen to initially move into quadrant IV, and ultimately settle in quadrant I at about a 60^o angle from the positive abscissa. It is seen that $U(n)$ tracks $R(n)$ quite well, always retaining its unity magnitude. In the steady state photograph, $R(n)$ and $U(n)$ simply appear as small blobs at roughly the same 60^o angle. Additional trials confirmed that solutions in other quadrants were also possible. Furthermore, experiments over 90% worst case linearly distorted channels (channels having both 90% worst case slope (amplitude) distortion and 90% worst case envelope delay distortion) demonstrated that the effect of the linear distortion was to widen the sector that $r(n)$ falls within to encompass about 90^o, as opposed to around 70^o for the back-to-back channel. The small blobs representing $R(n)$ and $U(n)$ were also slightly widened. However, no instances were ever encountered where the BETR loop failed to converge.

3. CONVENTIONAL ENVELOPE-DERIVED TIMING RECOVERY

In EDTR, we just replace the computation of the baud spectral line vector with an EDTR block rather than a BETR block. Everything else is unchanged.

A general block diagram of baud spectral line vector computation with EDTR is shown in Fig. 7. Since the necessary 2400 Hz tone will not occur naturally in the line signal spectrum, it must be generated by passing $x(m)$ through some sort of nonlinearity. Although many possible nonlinearities exist, the most popular are envelope magnitude, squared, and fourth-power. After corruption by the nonlinearity, there will be a 2400 Hz tone mixed in with all different cross products of signal and noise. In order to pick out the tone, we translate the negative frequency component ($\sim$–2400 Hz) to DC, where it is lowpass filtered to retain only the energy in its immediate vicinity.

In the laboratory, we viewed the EDTR $r(n)$, $R(n)$, and $U(n)$ for the magnitude, squared, and fourth-power nonlinearities over a back-to-back channel and a channel having 90% worst case linear distortion and a 32 dB *SNR*. In all cases the same master oscillator was used for both the transmitter and receiver, so there was neither frequency nor phase offset present between the clocks. *Under these conditions, we should expect excellent performance from our timing recovery system. However, none of the three nonlinearities delivered satisfactory performance in our system, even in the most ideal situation of a back-to-back channel.* In fact, the timing jitter was quite noticeable in the signal constellation (not shown), which was not true for BETR. We therefore conclude that with the approach taken here, EDTR performs poorly in our application, and BETR is the way to go.

4. CONCLUSIONS

Under the constraints of being prohibited from employing either equalizer-based or decision-directed timing recovery schemes, a new form of all-digital passband band-edge timing recovery (BETR) system was developed for use with blind equalization in voiceband modems. By means of experiments on a real-time digital signal processor, this BETR system was shown to be superior to the conventional envelope-derived timing recovery (EDTR) employing any of the common nonlinearities of envelope magnitude, square, or fourth-power to generate the baud spectral line. This result was obtained in the absence of any additional receive filtering, using signals chosen from the V.33 trellis-coded 128-QAM constellation and passed through a pulse shaping transmit filter with 12.5% excess bandwidth and square root Nyquist shaping.

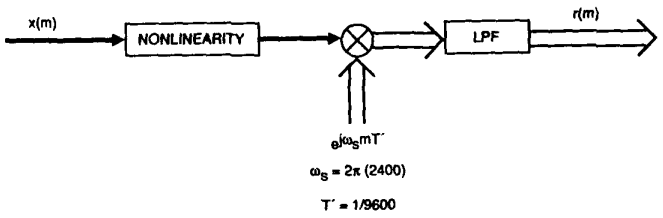


Fig. 7. Simplified baud spectral line vector computation in envelope-derived timing recovery system. Although $r(m)$ is computed at the sample rate, it is only utilized in the next block at the symbol rate.

ACKNOWLEDGEMENTS

The authors are grateful to their colleagues D. D. Harman, J. E. Mazo, M. D. Rauchwerk, B. R. Saltzberg, and J. J. Werner. Special thanks go to T. M. Dennis, G. J. Kustka, V. B. Lawrence, H. G. Mattes, and H. C. Meadors, Jr. for their helpful suggestions and support of this work.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] CCITT Recommendation V.33: "14,400 Bits Per Second Modem Standardized For Use On Point-To-Point 4-Wire Leased Telephone-Type Circuits," May 17, 1985.

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

[8] N. K. Jablon, "Joint blind equalization, carrier recovery, and timing recovery for high-order QAM signal constellations," submitted to *IEEE Trans. Commun.*

[9] N. K. Jablon, "Carrier recovery for blind equalization of high-order QAM constellations," submitted to *IEEE Trans. Commun.*

[10] 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.

[11] J. R. Treichler, C. R. Johnson, Jr., and M. G. Larimore, *Theory and Design of Adaptive Filters*, New York, NY: Wiley, 1987.

[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] C. W. Farrow, "A continuously variable digital delay element," *Proc. IEEE Int'l. Symp. Circuits Syst.*, Espoo, Finland, June 6-9, 1988, pp. 2641-2645.

[14] N. K. Jablon, C. W. Farrow, and S.-N. Chou, "Timing recovery for blind equalization of high-order QAM signal constellations," submitted to *IEEE Trans. Commun.*