Signals, Samples, and Stuff: A DSP Tutorial (Part 4)

Our DSP tutorial winds to an end. As we turn toward home, we finish the Fourier Transform discussion. Along the return path, we'll explore two fascinating parts of modern smart radios: adaptive control systems and remote-control systems. We'll also take a hard look at telephone systems and their bandwidth.

By Doug Smith, KF6DX/7

In the July/August *QEX*, I ended the third part of this series with a promise to provide a method for calculation of the discrete Fourier transform (DFT) that is faster than the so-called fast Fourier transform (FFT). The algorithm and its derivation are provided in this last installment of the tutorial. A unique feature of the algorithm is its error-control mechanism. I'll show that without it, the algorithm would diverge from the correct solution.

As a follow-on to the last issue's discussion of adaptive filtering, I have some notes about other adaptive control systems. I hope readers can dream up a few of their own—try an automatic anti-VOX!

I'll also describe audio digitization at relatively low serial bit rates for use in remote-control systems. Some controversy endures over the correct SNR expression for delta (Δ) modulation. I offer my analysis without making any assumptions about the nature of the audio signal, other than its amplitude.

The Damn-Fast Fourier Transform (DFFT)

Sorry, but I have to take us through the statistical muck one more time to prove that this Damn-FFT is going to work! It will be worth it, to be sure, since the computational burden will be reduced to be proportional to 2*N*, which for large values of *N*, is quite a bit better than $N \log_2(N)$! We must deal with errors that increase arithmetically, but I'll introduce a straightforward method of limiting them and define the upper bound.

This derivation begins by looking at how DFT results change at each sample time.[1] Say we start with a DFT for an input sequence $x(n)$ at sample time *r*. Then we compute the DFT for the next sample time $r + 1$ and examine the sequences to see what's changed. For $r = 0$, each DFT sequence expands to:

$$X_0(k) = W_n^{0k}x(0) + W_n^{1k}x(1) + W_n^{2k}x(2) + \ldots + W_n^{(N-1)k}x(N-1)$$
$$X_1(k) = W_n^{0k}x(1) + W_n^{1k}x(2) + W_n^{2k}x(3) + \ldots + W_n^{(N-1)k}x(N) \quad \text{(Eq 1)}$$

and what's evident is that each input sample $x(n)$ that was multiplied by W_N^{nk} in the summation for $X_0(k)$ is now multiplied by $W_N^{(n-1)k}$ in the summation for $X_1(k)$. So the *ratio* of the two sequences is nearly:

PO Box 4074
Sedona, AZ 86340
e-mail: **kf6dx@arrl.org**

[1]Notes appear on page 29.

$$\frac{X_1(k)}{X_0(k)} \approx \frac{W_N^{(n-1)k}}{W_N^{nk}} = W_N^{-k} \quad \text{(Eq 2)}$$

We still have two terms "hanging out" of the relationship, namely the first and the last:

$$W_N^{0k}x(0) = x(0) \; and \; W_N^{(N-1)k}x(N) \quad \text{(Eq 3)}$$

which haven't been accounted for in the ratio. If we first subtract $x(0)$ from $X_0(k)$ before taking the ratio, then add the new term $W_N^{(N-1)k}x(N)$ after, we have the correct result:

$$X_1(k) = W_N^{-k}\left[X_0(k) - x(0)\right] + W_N^{(N-1)k}x(N) \quad \text{(Eq 4)}$$

Now this can be simplified a little, since:

$$W_N^{(N-1)k} = e^{\frac{-2\pi j(N-1)k}{N}} = e^{\frac{-2\pi jN}{N} \cdot \frac{2\pi jk}{N}} = W_N^{-k} \quad \text{(Eq 5)}$$

and substituting:

$$X_1(k) = W_N^{-k}\left[X_0(k) - x(0) + x(N)\right] \quad \text{(Eq 6)}$$

This is marvelous, because it means for N values of k we can compute the new DFT from the old with N complex multiplications and $2N$ complex additions, or a computational burden proportional to N! If we begin with $X(k) = 0$ and take the first N values of $x(n) = 0$, we can start the thing rolling—we save computation over the FFT by the factor:

$$\frac{\frac{N}{2}\log_2 N}{N} = \frac{\log_2 N}{2} \quad \text{(Eq 7)}$$

which for large values of N is very significant indeed! For example, if $N = 1024$, the improvement is a factor of five. Over the direct DFT, it is a factor of $N^2/N = N$ faster. But there's a catch: An error term will grow in the output because the truncation and rounding noise discussed previously is cumulative. The error will continue to grow unless we do something about it.

The safest way to handle the situation is to effectively reset the variables to zero every N samples or so, and to do this while continually generating output samples.

We'll calculate two DFFTs for all the output points k, with the first of them, $Y(k)$, taking the next N input samples $x(n) = 0$, and the second, $Z(k)$ using unmodified input samples $x(n)$. Beginning at some input sample time r:

$$For\; 0 \le n \le N-1:$$

$$Y_{r+n+1}(k) = W_N^{-k}\left[Y_{r+n}(k) + x(r+n+N)\right]$$
$$Z_{r+n+1}(k) = W_N^{-k}\left[Z_{r+n}(k) - x(r+n) + x(r+n+N)\right] \quad \text{(Eq 8)}$$

After these N iterations, both calculations have identical results except for the greater error in $Z(k)$. At this point, the result of the first calculation $Y(k)$ is transferred:

$$Z(k) = Y(k) \quad \text{(Eq 9)}$$

and then the first result is zeroed:

$$Y(k) = 0 \quad \text{(Eq 10)}$$

The calculation then continues for another N iterations, at which time the exchange and reset are again done. This places an upper bound on the cumulative error to that associated with $2N$ iterations, but it increases the computational burden by a factor of two. Now the savings over the FFT is:

$$\frac{\frac{N}{2}\log_2 N}{2N} = \frac{\log_2 N}{4} \quad \text{(Eq 11)}$$

which for $N > 16$ represents an improvement.

Behavior of the DFFT Error Term

In this system, we expect the round-off errors to grow linearly in the worst case since we're using the value of $X_{r+n}(k)$, containing an error, to compute the new value of $X_{r+n+1}(k)$. The calculation produces another error which simply adds to the old:

$$X_{r+1}(k) + \varepsilon_{r+1} + W_N^{-k}\varepsilon_r = W_N^{-k}\left\{\left[X_r(k) + \varepsilon_r\right] - x(r) + X(r+N)\right\} \quad \text{(Eq 12)}$$

and the propagation of errors depends on the value of W_N^{-k}. This coefficient might be as large as unity, so after $2N$ iterations, we expect the maximum noise power to be twice that of the direct-form DFT, or:

$$\sum_{n=0}^{N-1} E\left[\varepsilon_n^2\right] = \frac{2^{(1-2b)}N}{3} \quad \text{(Eq 13)}$$

If this were the only source of error, we could track the error term separately—using another set of registers, for example—and use it to steer the output back toward the correct result. Nonetheless, the error in the fixed-point representation of the coefficients W_N^{-k} degrades the signal-to-noise ratio (SNR) as it does in the FFT. This error cannot be represented exactly for all values of k, since the coefficients are *irrational* numbers. That is, sines and cosines. So we suspect that the exchange and reset method is the best we can do.

Fig 1 shows the result of DFFT calculation errors, as obtained from comparison with DFT results for the same input data, over many iterations for $N = 1500$ and with a random input (noise) for $x(n)$. Data and coefficients have 16-bit fixed-point representation, *convergent rounding* has been applied to the coefficients and to the result of each computation. *Block floating-point scaling* has been implemented.[2] This means that whenever the result of a calculation results in an overflow, the result is divided by two, and a scaling-factor register is updated to reflect the new scale. As indicated in Part 3 of this series,[3] the scaling required to prevent overflow increases the output noise beyond the bound of Eq 13. Clearly, the DFFT output error is twice that of the DFT, with an SNR bounded by:

$$SNR_{OUTPUT} = \frac{2^{(2b-3)}}{N} \quad \text{(Eq 14)}$$

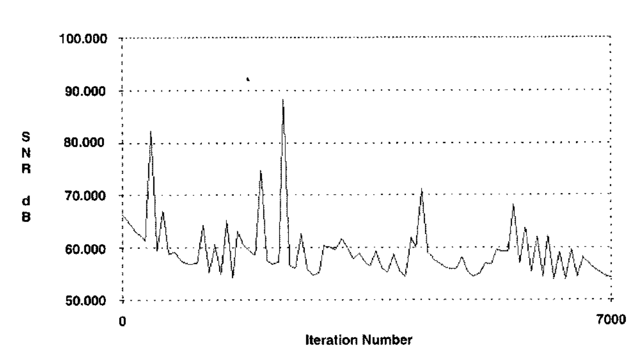

Fig 1—DFFT error noise versus sample time.

The Inverse DFFT ($DFFT^{-1}$)

There isn't an inverse DFFT ($DFFT^{-1}$)! The only direct analogy I can find for the inverse case relates to the sample-by-sample nature of the DFFT. In a "real-time" system, only the next output sample need be computed, not the next N output samples. The easiest output term to compute is $x(0)$, since in the summation, all coefficients are $W_N^0 = 1$. The output is then just:

$$x(0) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) \qquad \text{(Eq 15)}$$

and only one multiplication is involved.

The DFFT should be useful to experimenters who are looking for a quick and simple method of DFT calculation. The application of window-ing to the DFFT and further improve-ments in the algorithm are subjects of ongoing research.

Combining Adaptive Techniques with the DFFT: The Adaptive Line Enhancer

The adaptive, self-tuning filter described previously (see Note 3) can be combined with spectral-analysis techniques to create a very sensitive detector of periodic signals in the presence of noise. The basis for this method is that the Fourier transform of a filter's impulse response $h(n)$ is its frequency response $H(k)$:

$$H(k) = \sum_{n=0}^{L-1} W_L^{kn} h(n) \qquad \text{(Eq 16)}$$

That is, if we spectrally analyze an adaptive filter's time response, we get a picture of its frequency response. To exploit this combination of the two noise-reduction (NR) techniques, we can apply the DFFT to the filter coefficients $h(n)$ produced by an LMS adaptive filter.

Fig 2 shows an arrangement known as an *adaptive line enhancer*.[4] Note that the DFFT input isn't a simple, time-sampled sequence, as it was in the original derivation, because *every value* of $h(n)$ is changing at each sample time. Whereas for some filter of length L, the line enhancer output at time $r = 0$ is:

$$H_0(k) = \sum_{n=0}^{L-1} W_N^{kn} h_0(n) \qquad \text{(Eq 17)}$$

The output at the next time $r = 1$ is:

$$\begin{aligned} H_1(k) &= \sum_{n=0}^{L-1} W_N^{kn}\left[h_0(n) + 2\mu e_0 x_0(n)\right] \\ &= H_0(k) + \sum_{n=0}^{L-1} W_N^{kn} 2\mu e_0 x_0(n) \\ &= H_0(k) + 2\mu e_0 X_0(k) \end{aligned} \qquad \text{(Eq 18)}$$

This is the DFFT for the adaptive line enhancer. (See Fig 3.) It adds $2L$ complex multiplications and $2L$ real additions to the computational burden of the adaptive noise reduction system, but this sure is a lot better than applying a DFT or even FFT to the filter coefficients. We must use the simultaneous transfer and reset methods of equations 8, 9 and 10 above, because errors build up as before.

This system rivals any spectral-power-measurement algorithm around. If adaptive filtering produces some SNR improvement, then application of the DFFT to the filter co-efficients produces an additional improvement.

Further, imagine that the output of the spectral line enhancer was used as the input to a PLL! The SNR could then be whatever was achievable at the output of the oscillator—a tremendous improvement over the input SNR indeed. Of course, issues of lock time, acquisition range and so on would come into play.

The Direction of Future Research

As industry strives toward the development of hardware capable of sufficient dynamic range to eliminate more of the analog processing stages in receivers, the issues covered above will become increasingly significant. The goal remains the same: To receive a desired signal in the presence of strong undesired signals. We've seen that, in the end, only so much mathematical finesse can be applied to the problem; additional improvements come only through increased processing "horsepower." It's an unfortunate result of the way things are that when the math has exhausted all its avenues of refinement, all effort is expended on increasing the raw speed of calculation. This has the unintended consequence of allowing programmers the luxury of faster machines, of larger data and program spaces and of complacency toward continual upgrades in both of those. The emphasis drift's away from doing things intelligently, and toward brute-force approaches.

This trend is evident in the megabytes of code seen in today's application software. However, this isn't to say, as one patent office commissioner did, that: "Everything that can be in-

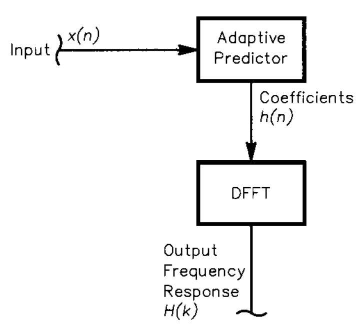

Fig 2—An adaptive line enhancer.

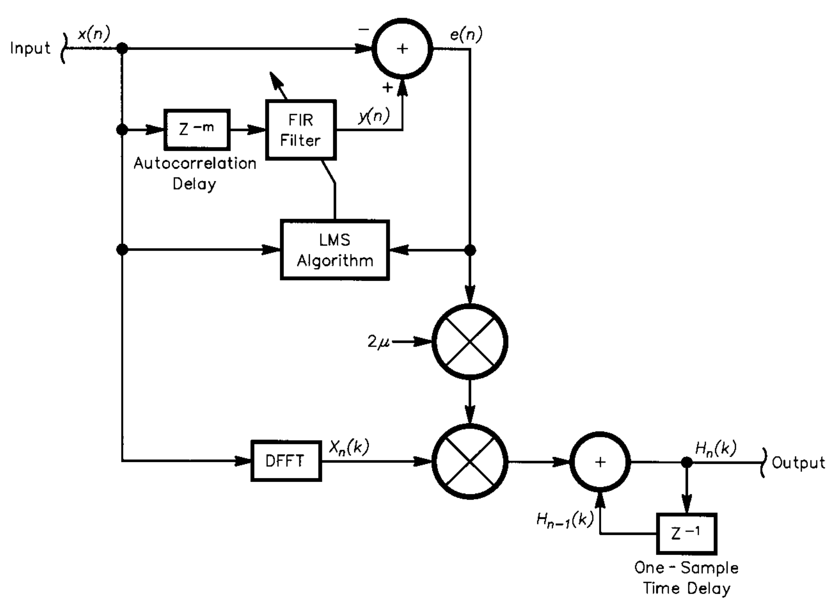

Fig 3—The adaptive line enhancer using the DFFT.

vented, has been invented." It's only a poke at those who forget the importance of questioning everything.

Adaptive Control Systems for Radio Transceivers

Microprocessor control of everything from automobiles to microwave ovens has revolutionized the way we live. It's very much like other innovations throughout history: We didn't know we needed them until they came along and made things so much easier. In fact, it's difficult for many to imagine life without automobiles, microwave ovens or microprocessors; especially, those of the newer generations cannot conceive how it was in the "dark ages."

We do well to study the past, if only to remind ourselves why we did things the way we did and refresh our memories about the mistakes that we made along the way. As is the case for so many human schemes, unintended consequences can be the most memorable part of any invention.[5] As a good friend is wont to say: "Considering all the problems with these 'new-fangled' computers, I don't think they're here to stay."

But they've sure solved a lot of problems, too. That is the subject of this section, specifically: How computers enhance the performance of radio equipment and make certain things possible that would be excruciatingly difficult without them.

Control Theory

In control theory, the system to be controlled is referred to as the *plant*.[6] The plant produces some output response to a control input. The thing providing the control input is known as the *controller*. A good example is the automobile, the engine and wheels of which respond to the input commands of the driver. He or she has their hands on the steering wheel, and presumably, their feet near the accelerator and brake pedals. The plant in this case responds to the driver's desire to steer, accelerate or decelerate the car. Most drivers understand this relationship, but there are some who clearly don't!

In a computer-controlled transceiver, the plant responds to input from the operator when it's time to change frequency, mode or alter other parameters. Central to most control systems is the idea of *feedback*, which provides the controller with information about how the plant is responding. In many cases, it's enough to know *whether* the plant has responded. A frequency display or indicator lamp on a transceiver provides visual feedback about plant response. Many responses are auditory only—the volume control is an example.

As in the case of adaptive filters, feedback provides input to the controller, which then uses some algorithm to produce new control information. The controller's algorithm may be a fixed linear or nonlinear system, or it may be time-varying so that adaptation to changing plant conditions is possible. In the following discussion, we'll address both situations, with emphasis on adaptive DSP technology.

Anatomy of a Computer-Controlled DSP Transceiver

We might be tempted to say that the digital signal processor is the heart of any DSP transceiver, but I'd like to propose that it's the brain. The synthesizer is more like the heart, the transmitter is the vocal chords and the receiver the ears! In the case of a microprocessor-controlled transceiver, almost everything is under the supervision of the processor, so it makes sense that it manages all control commands and feedback. Whether the controls are knobs attached to shaft encoders or keys on a computer keyboard matters not to this deliberation. What *does* matter is that all things impacting transceiver performance *and that can be altered* are implemented by firmware running on a DSP.

For an IF-DSP transceiver, the range of functions falling into this category is immense. In fact, it's prudent to provide as much user control as possible. The challenge is to present the options in a way that is not ambiguous and "user-friendly." Most of us have encountered "user-hostile" operating systems that make ham radio a chore rather than the pleasure it should be. Let's look at some of the basic functions of today's HF rig, and examine how DSP control improves performance.

S-Meter Calibration

Most HF transceivers have an analog gain control (AGC), another example of a feedback control system. It's desirable to meter the signal strength in the receiver, and this is usually done by using the AGC voltage to drive a visual display of some kind. Because of the nature of the physical circuits, gain control is typically proportional to the logarithm of the AGC voltage. Hence, S-meters are calibrated logarithmically—each increment of meter deflection is proportional to a fixed number of decibels.

Analog gain-controlled devices don't repeat exactly from unit to unit. To get an accurate S-meter, it's necessary to characterize each unit separately via a calibration routine. It's easy enough to measure the S-meter's performance, and to build a table of its errors versus input signal strength. We can do exactly this when the S-meter is under microprocessor control.

A correction table is stored in nonvolatile memory, and is used to adjust the S-meter values in "real time." During testing, a calibration routine is used to compare the meter reading with the input level from a known-accurate signal generator, and so to generate the table. After many units have been measured, it's found that a small number of S-meter correction curves are sufficient to account for all units; these "boiler plate" tables are used based on individual variations.

Automatic Frequency Calibration and Temperature Compensation

Another benefit of DSP feedback control systems is that it's easy to determine the frequency of a received carrier to within the accuracy of the control system's reference clock. An accurate external reference can be input to the receiver, and its frequency counted by the DSP system. The frequency error is used to generate a correction voltage, which is applied to the voltage-controlled crystal oscillator (VCXO) internal reference. The correction voltage is adjusted until the frequency error is minimized.

This VCXO-control voltage is also varied with temperature to compensate the oscillator's frequency-versus-temperature curve, making it a microprocessor-compensated crystal oscillator (MPCXO). For best accuracy, each unit can be calibrated separately. A table is downloaded and stored in nonvolatile memory representing the variation in control voltage necessary to keep the internal reference's frequency constant. The temperature sensor is located in the oscillator compartment to achieve best tracking.

As each unit is calibrated separately, the shape of the curve in the table accommodates the actual frequency-versus-temperature trait of that particular oscillator. Note that *all* frequency-determining elements in the transceiver are inside the calibration loop, and are therefore compensated to some degree using this technique.

The most convenient way to measure a carrier's input frequency is to translate it down to some IF or audio frequency, then count the number of zero-crossings per second. As the measured error decreases, the DSP system can automatically narrow the BW of the received signal to improve the SNR and, therefore, the accuracy of the result.

The Receiver as a Spectral Analyzer

As we saw above, various techniques are available to analyze signals in the frequency domain. When tuned to a fixed frequency, however, the bandwidth of the receiver is necessarily limited because of dynamic-range considerations.[7] With today's frequency-agile synthesizers, it's possible to turn our receiver into a spectrum analyzer.

The receiver is tuned rapidly across the band, and the signal strength is measured at each iteration. A very fast AGC time constant is obviously required. The resolution BW of the measurements can be altered by selecting a different BPF, with the attendant change in sweep speed. The resulting data are graphically displayed, and the operator is given a graphical user interface (GUI) with which to visually select signals, and hence manually tune the receiver.

Adaptive Path-Quality Evaluation and Automatic Link Establishment

Commercial HF operators may not be as skilled as radio amateurs in selecting operating frequencies. Hence, the need for automatic-link-establishment (ALE) systems, which select clear communication frequencies. While I don't expect ALE to catch on in ham circles, certain aspects of it have been in use by amateurs for some 15 years. It's worth examining them as forms of adaptive control.

At the lowest level of ALE architecture is a receiver's frequency-scanning capability. We can scan many frequencies in a relatively short period, look for signals of interest and stop to examine them when appropriate. After all, we're trying to find the best frequency for the communication desired; we'd better have a choice of frequencies, or the exercise would be pointless.

The next level incorporates some form of *selective calling*. We need a signaling method that discriminates between the many stations on the air. Myriad selective-calling schemes have been used over the years in commercial systems; some are more effective than others. The ham community has pioneered quite a few of the digital transmission modes used with radio, and we can easily find examples of selective calling among them. AMTOR, packet, PACTOR, CLOVER, GTOR and now PACTOR II all support selective transmissions. In fact, what transmissions other than news bulletins and CW practice are *not* selective?

At the third level, we integrate the first two levels into a path-quality evaluation (PQE) system. When not in use, all the stations in the network are scanning the assigned frequencies; we then use our selective calling tools to attempt contact with the desired station on the available frequencies, one by one. Connection with the other station results in some acknowledgement that we've been heard. At this point, scanning is suspended, and some exchange of data is made to evaluate the quality of the link. This may be as simple as a signal strength measurement, or may involve appraisal of a data-error rate, or both.

This procedure is known as *polling*. The idea is that after all frequencies have been polled between the two stations, an assessment is made regarding the best frequency to use. Either station can make this judgment, but ultimately one will call the other again on the best frequency to establish communications. In most such systems, the PQE analysis is made at the time when the connection is desired.

Another method of determining connectivity is known as *sounding*. In this procedure, stations not in use periodically broadcast messages intended for the general consumption of all the other stations in the network. Acknowledgment of sounding transmissions is not expected. Those stations that happen to be listening on that frequency evaluate the quality of reception, and make a note of it. When it's time to connect, each station has a much better idea of which frequency is the best.

Here at the fourth level, we pull it all together as an ALE system. After the thing has been operating a while, each station has data about which other stations it can contact. At connect time, therefore, the amount of delay until communications are established is at a minimum. The operator enters the selective calling code for the station with which communications are desired, and the system uses its "learned" information to make contact on the best frequency. This information is slowly changing, of course. We'll also throw in a busy-channel detector, which pre-

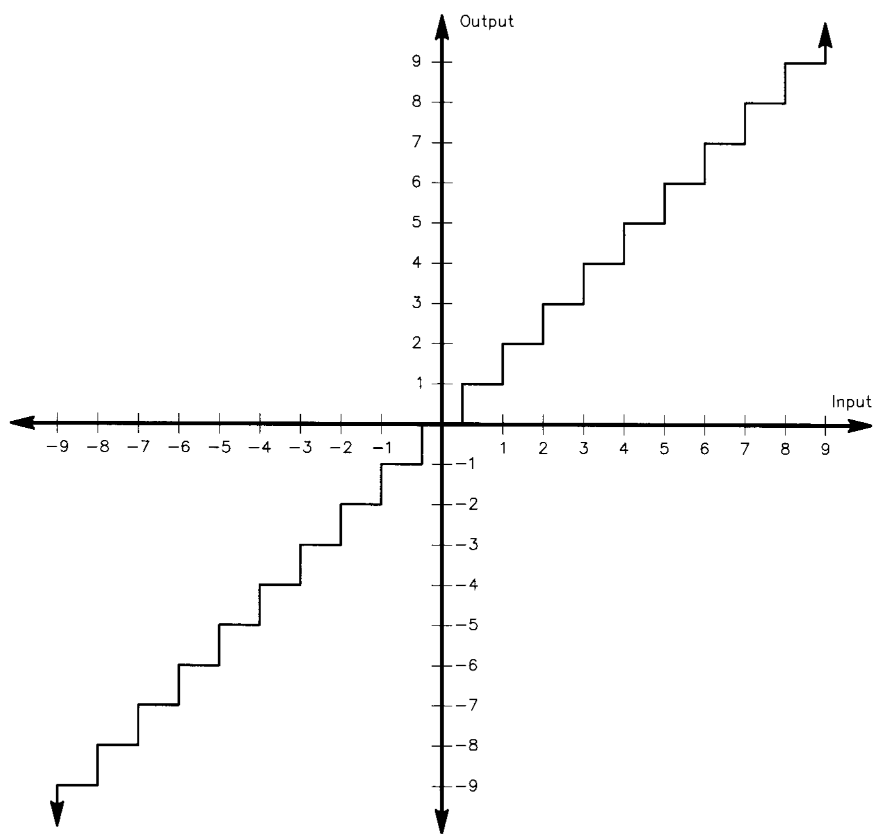

Fig 4—Uniform quantizer transfer characteristic.

vents stations from transmitting on frequencies that are already in use.

At level five, we add the ability to exchange information about network connectivity; that is, we allow the stations to periodically report to each other which others they can hear. Imagine that station A can communicate with station B, and station B can also communicate with station C, but that A cannot directly talk to C. When A attempts to contact C, the connection will fail; B then intervenes and relays the traffic from A to C.

Higher ALE levels implement the capacity to *store and forward* traffic. Stations in the network save copies of messages intended for other stations that are busy or temporarily unavailable. Later, when a connection can be established, the messages are delivered.

Standards have evolved based on these strategies, among them the AX.25 packet protocol, FED-STD-1045, 1046, 1047 etc. The advantage of standards is that they take the best of all the various techniques and meld them into a logical format. Amateur Radio has done well in fostering its packet standards. We could establish similar standards for the other digital modes and promote our ability to respond when the world needs us, such as during a major disaster or other emergency. I feel strongly that to advance the cause of Amateur Radio is not only to push the state of the art, but also to use our technology in times of need. How much more traffic could have been passed after the Mexico City earthquake with such a system?

I'll use the rest of my space here to discuss the remote control of radio equipment, as this is also important to emergency preparedness in many ways. It's also of interest to those of us in antenna-restricted areas—it's becoming so common these days!

Remote-Control Systems for Radio

This really is a hot topic because of its bearing on communication effectiveness in general. Hams who live in antenna-restricted areas (as I do) must find some means of getting on the air without upsetting their neighbors. Antennas are viewed as an eyesore by an increasing number of neighborhood associations, but it's RFI that really gets us in trouble. Remote location of the transceiver solves both problems. Commercial systems must often crowd many transmitters into a small area, so they're forced to distantly locate the receivers. Finally, the situation wherein many operators share a single transceiver occurs in both the commercial and amateur services. Repeaters on the VHF and UHF bands are examples.

No matter who is using the equipment, the requirement exists for a control link. Part 97 of the FCC Rules[8] states that there shall be a control operator present at the control site at all times. While the automatic message store-and-forward system described above stretches the rules to their limits, I think we can see that a radio without a control operator is like a sailboat without a skipper. All kinds of things go wrong with radios that require the attention of human beings! In remote control, we're just extending the control site to a point distant from the transceiver. Complete control is still possible, and the system is secure as long as it can "fail safe."

The Control Path

For a control link, two media present themselves: The public switched telephone network (PSTN) and a direct VHF, UHF or microwave radio link. It would be nice if whatever method we select were applicable to either form of remote link. It's also desirable that both control data and audio traverse the link in both directions simultaneously; ie, in full duplex. Especially in the case of a remote radio link, it's tempting to use two channels: one for the control data, and one for the audio. When considering the PSTN, however, this isn't very attractive. Telephone bills never stop coming, so it sure would be nice if we could implement the whole system on a single line.

Starting in 1996, interest in simultaneous voice and data transmission intensified. Driven by the demand for "teleconferencing," numerous companies introduced digital simultaneous voice and data (DSVD) modems. These were followed by so-called analog simultaneous voice and data (ASVD) modems, which touted improved audio quality while maintaining moderate data-transmission bandwidth. Neither of these technologies is compatible with the formats emerging for digital audio transmission over the Internet.

Meanwhile, telephone companies are grappling with a connectivity problem of their own: How can they accommodate an ever-increasing number of calls, from both voice and modem users, without increasing the number of physical lines? The tremendous increase in traffic caused by the Internet began to place a strain on the existing PSTN.

PSTN Traffic Capacity and Quality: The Bandwidth Boondoggle

This issue of traffic capacity is an enormous one for the telephone companies and for those of us wishing to pass large amounts of information through the telephone lines. As I began looking into this, several crucial questions emerged: Who or what sets the available bandwidth on the PSTN? Can the telephone companies change the bandwidth without notice?

In both wired and wireless communications, digital transmission modes are becoming more prevalent.[9] Digital formats have clear advantages over analog. The first, most obvious of these is noise immunity in detection; a detector has only to determine whether the received datum was a one or a zero. Another advantage is that error detection and correction can easily be applied to digital data. Whereas we want to pass both digital control information and audio in our remote-control application, it makes sense to use an exclusively digital mode on the remote link. It's easier and more secure to digitize the audio, and recover it error-free on the other end, than it is to devise analog control methods for a complex transceiver. Also, the data are more easily encrypted for security. The telephone companies discovered the benefits of digital transmission long ago, and while it might surprise some, virtually *all* telephone calls are digital during most of their journey.

In going digital, the folks at the telephone company (remember, it used to be just one company) found that when speech was digitized, it took up a heck of a lot more bandwidth than in analog form. Consider a speech signal occupying a bandwidth of about 3 kHz. Digitize it using 8 bits, or 256 quantization levels, at a rate just above the Nyquist limit, say 8 kHz. The bit rate is then:

$$\mathit{bit\ rate} = \left(\frac{8\ \text{bits}}{\text{sample}}\right)\left(\frac{8{,}000\ \text{samples}}{\text{second}}\right) = 64\times10^3\ \frac{\text{bits}}{\text{second}} \qquad \text{(Eq 19)}$$

Now the BW is about 10 times what's required for the actual information, but this is basically the phone company's standard. The format does have the advantages of noise immunity and the ability to *time-division multiplex* the data with those of other telephone calls. The quality of this *uniform quantization pulse-code modulation (PCM)* scheme would leave something to be desired, since its dynamic range (DR), the ratio of a full-scale signal to the smallest quantization step, using signed, fixed-point math, is only:[2]

$$DR = 2^{(b-1)} \approx 6.02\,(b-1)\text{ dB} \approx 42.2\text{ dB} \qquad \text{(Eq 20)}$$

and its maximum SNR for a sine wave input:

$$SNR_{MAX} = 2^{\left(b-\frac{3}{2}\right)}\sqrt{3} \approx 6.02\,(b-1)+1.76\text{ dB} \approx 43.9\text{ dB} \qquad \text{(Eq 21)}$$

A uniform "quantizer" has the transfer characteristic shown in Fig 4.

Compression Techniques: Non-Uniform Quantization

To improve things, we decide to take our 256 quantization levels and distribute them over the range of amplitudes such that resolution for low-level signals is increased at the expense of high-level signals. This results in a non-uniform quantization transfer characteristic, such as that shown in Fig 5. The net effect is to increase the DR for a given number of bits, and to improve the SNR for low-level signals. Alternatively, we could reduce the number of bits, and therefore our occupied bandwidth, while maintaining the same DR as before. The main drawback of non-uniform quantization is a lower limit for the maximum SNR. Eqs 20 and 21 are no longer valid.

In most non-uniform quantization systems, the transfer curve is somehow logarithmic; ie, the quantizer output is proportional to the log of the input. In North America and Japan, *μ-law* quantization is used, wherein for $|x| < 1$ and $\mu >> 1$:

$$f(x) = \text{sign}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)} \qquad \text{(Eq 22)}$$

$\mu = 255$ for North America. The DR expression must now be rewritten to reflect the new transfer function $f(x)$. Setting Eq 22 equal to the smallest-possible positive output step $1/2^{b-1}$ yields the smallest-possible positive input signal which can be resolved:

$$\frac{1}{2^{b-1}} = \frac{\ln(1+\mu x)}{\ln(1+\mu)},\ for\ x > 0 \qquad \text{(Eq 23)}$$

$$x_{MIN} = \frac{(1+\mu)^{\frac{1}{2^{b-1}}} - 1}{\mu} \qquad \text{(Eq 24)}$$

This is smaller than $1/2^{b-1}$, and for b = 8 and $\mu = 255$, the DR is now:

$$DR = x_{MIN}^{-1} = 20\log\left(\frac{\mu}{(1+\mu)^{\frac{1}{2^{b-1}}} - 1}\right)\text{dB} \approx 75\text{ dB} \qquad \text{(Eq 25)}$$

Derivation of the SNR is inherently much more difficult. Degradation of the peak SNR occurs because of the coarse quantization used for large-signal amplitudes, where finely spaced levels aren't otherwise necessary. An input that is near the boundary of two coarsely spaced quantization levels is liable to jump rapidly between those levels, producing a lower-frequency noise component at the "jitter" frequency. At low input amplitudes, quantization uses more levels, and improves the SNR there over uniform quantization. The basic idea is to hold SNR constant over the range of input amplitudes. We can analyze the SNR for μ-law quantization as a function of input level in the same manner as in the derivation of statistical truncation or rounding errors given previously. This time, however, we'll use terms for the RMS noise voltages instead of the powers.

As before, the noise voltage is the expected value of the quantization error over the "distance" between two adjacent quantization levels. Between two adjacent output quantization levels $n/2^{b-1}$ and $(n+1)/2^{b-1}$, the quantizer is liable to produce noise proportional to the difference between the corresponding *input* levels. After Eq 24:

$$x_n = \frac{(1+\mu)^{\frac{n}{2^{b-1}}} - 1}{\mu} \qquad \text{(Eq 26)}$$

and

$$x_{n+1} = \frac{(1+\mu)^{\frac{n+1}{2^{b-1}}} - 1}{\mu} \qquad \text{(Eq 27)}$$

The difference between these input levels is:

$$dx = \frac{(1+\mu)^{\frac{n+1}{2^{b-1}}} - 1}{\mu} - \frac{(1+\mu)^{\frac{n}{2^{b-1}}} - 1}{\mu} = \frac{(1+\mu)^{\frac{n+1}{2^{b-1}}} - (1+\mu)^{\frac{n}{2^{b-1}}}}{\mu} = (1+\mu)^{\frac{n}{2^{b-1}}}\left[\frac{(1+\mu)^{\frac{1}{2^{b-1}}} - 1}{\mu}\right] = (1+\mu)^{\frac{n}{2^{b-1}}}x_{MIN} \qquad \text{(Eq 28)}$$

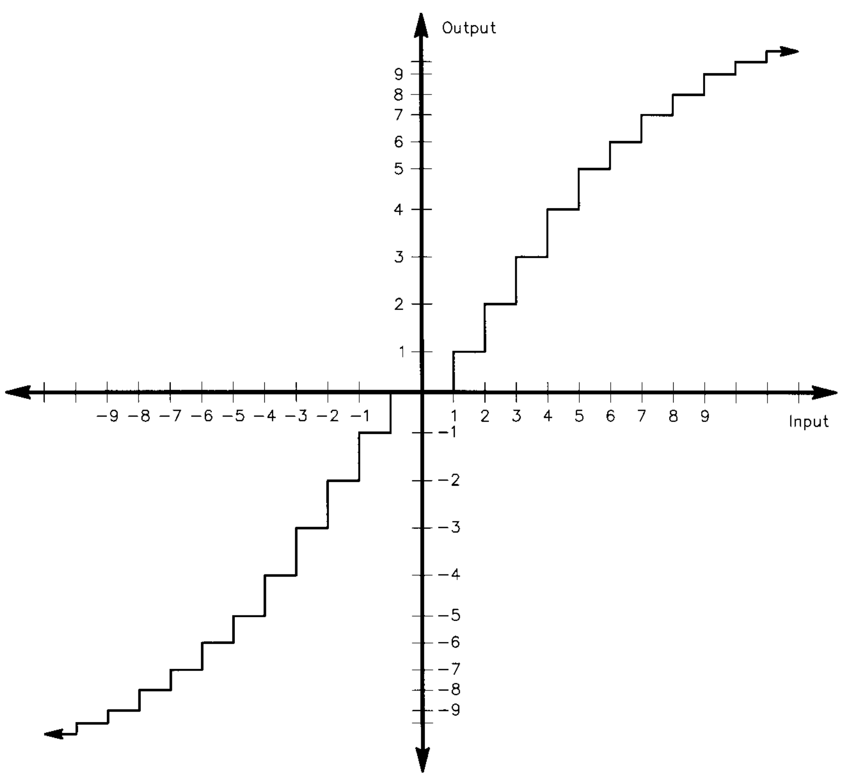

Fig 5—Nonuniform quantizer transfer characteristic.

and the expected value of the noise voltage is:

$$\sigma = \frac{(1+\mu)^{\frac{n}{2^{b-1}}} x_{MIN}}{\sqrt{12}} \qquad \text{(Eq 29)}$$

The SNR is the ratio of the input level x_n to the noise:

$$\begin{aligned} SNR &= \frac{x_n}{\sigma_n} \\ &= \frac{\left(\dfrac{(1+\mu)^{\frac{n}{2^{b-1}}} - 1}{\mu}\right)}{\left(\dfrac{(1+\mu)^{\frac{n}{2^{b-1}}} x_{MIN}}{\sqrt{12}}\right)} \\ &= \frac{\sqrt{12}\left[1-(1+\mu)^{\frac{-n}{2^{b-1}}}\right]}{\mu\, x_{MIN}} \end{aligned} \qquad \text{(Eq 30)}$$

In Fig 6, it's evident that this method does a reasonably good job of holding the SNR constant as compared with straight 8-bit PCM.

As the necessity arose to increase the total traffic-handling capacity of telephone networks, both satellite and fiber-optic technologies were developed. Each offers increased bandwidth without the radiation problems associated with a twisted-pair wire transmission line. The typical fiber-optic T1 line interface has a capacity of roughly 1.54 megabits per second (Mbps), and so can handle several simultaneous calls:

$$\begin{aligned} calls &= \frac{1.54\times10^6}{64\times10^3} \\ &\approx 24 \end{aligned} \qquad \text{(Eq 31)}$$

We infer that a data connection using a modem at 28.8 kbps occupying the same 3 kHz bandwidth is not an efficient use of resources, since were the digital interface extended to the user, 64 kbps would be possible. Therefore, we conclude that the PSTN is not optimized for data traffic. No great shock, since this isn't what it was designed for in the first place.

However, some very bright person comes along and discovers that the bandwidth occupied by digitized audio can be reduced by coding the *difference* between the samples. The difference between samples is smaller in amplitude than the actual samples, and so can be coded with fewer bits. Bandwidth is therefore reduced.

Delta Modulation

Extending this idea, we increase the sample rate by a factor of 8 (for the 8 bits per byte) and reduce the bit-resolution by the same factor. Now we're using only a single bit each sample time, and we have a one-bit quantizer. This is called linear *delta modulation* (DM).[10] The first such scheme was developed in 1946.

When the encoded bit = 1, the voltage is increasing; when the bit = 0, the voltage is decreasing. (See Fig 7.) A certain value of voltage change, dV, is associated with each bit. The encoded bit stream is integrated at the decoder to reproduce the original waveform. This design has an inherent difficulty, however: It can't reproduce waveforms that have slopes exceeding the maximum-possible integration time constant, dV/dt. This phenomenon is known as *slope overload*. The net effect is a roll-off in the high-frequency response.

Increasing the value of dV helps mitigate this problem, but it introduces *granular noise*. Large values of dV tend to mask low-amplitude signals smaller than dV. In the situation where input signals are smaller than dV, the quantizer output is an alternating sequence of ones and zeros; ie, a square wave. Since this integrates to dc, low-level signals are lost.

Continuously-Variable Slope Delta Modulation (CVSD)

We can use adaptive techniques we learned earlier to address the issues of granular noise and slope overload. Allowing the value of dV to change adaptively—based on the input slope—largely solves the problems encountered above. This technique of CVSD was first introduced by Greefkes and Riemens in 1970.[11] In it, dV is altered "on the fly" based on the values of the last three or four bits in the stream. See Fig 8.

When the last three or four bits are not identical, the system is equivalent to linear DM. When continuous strings of ones or zeros occur of at least that length, however, the integration time constant is changed to increase the slope. Thus the granularity problem is no worse than in the case of linear DM, and the slope overload problem has been greatly ameliorated. Note that the figure incorporates an adaptive predictor.

The adaptive character of CVSD results in an SNR-versus-input level performance similar to the μ-law en-

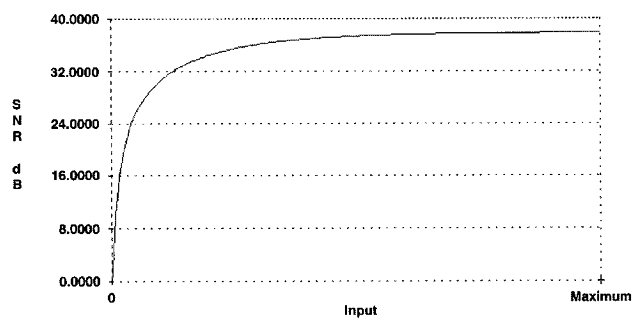

Fig 6—μ-law SNR-versus-input amplitude.

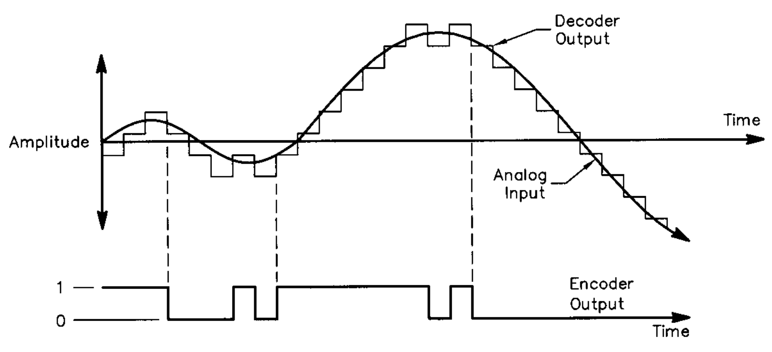

Fig 7—DM representation.

coding above. The nonlinear transfer curve in this case is due to changes in the value of dV based on past variations in the input signal. I_1 is referred to as the *primary integrator* and I_2 as the *pitch integrator*. Time constants of 1 ms and 5 ms, respectively, have been found effective.

The performance of this coding system is remarkably good. Overall, the quality of CVSD at 32 kbps is about as good as for 64 kbps PCM. Standard data modems can support 32 kbps, so the transport mechanism is readily available for use on the PSTN.

CVSD Performance

We can analyze CVSD performance by comparing it with its cousin, linear DM. When the input frequency-amplitude product doesn't exceed:

$$(Af_{in})_{MAX} = \frac{f_s dV}{2\pi} \qquad \text{(Eq 32)}$$

where f_s is the bit rate, the two systems are the same. For single integration, the quantization noise power[10] is:

$$\sigma^2 = \frac{2dV^2 f_{BW}}{3f_s} \qquad \text{(Eq 33)}$$

where f_{BW} is the system bandwidth. For a signal at the slope-overload threshold, the signal power is:

$$A_{MAX}{}^2 = \frac{1}{2}\left(\frac{f_s dV}{2\pi f_{in}}\right)^2 \qquad \text{(Eq 34)}$$

and the SNR is:

$$SNR_{MAX} = \frac{A_{MAX}{}^2}{\sigma^2} = \frac{3f_s^3}{16\pi^2 f_{BW} f_{in}^2} \approx 10\log\left(\frac{f_s^3}{f_{BW} f_{in}^2}\right) - 17.2\text{ dB} \qquad \text{(Eq 35)}$$

For our system with f_s = 32 kbps and f_{BW} = 3 kHz, Eqs 33 through 35 are used to plot SNR versus input amplitude for f_{in} = 1 kHz sine wave as Fig 9. Eq 35 predicts a maximum SNR of 23.2 dB. The curve shows that the SNR peaks at an input level near the middle of the DR. Above this level, slope overload degrades SNR because of distortion; below, quantization noise is responsible.

CVSD takes care of much of the slope-overload problems, and achieves a similar maximum SNR for voice signals. Note that in all DM systems, the maximum SNR degrades as the square of the input frequency, and with the *cube* of the bit time.

To get standard data modems to pass a more or less steady stream of bytes, we must switch off their data compression and error-correction functions. Both V.42 and MNP-5 error-correction standards are commonly used by today's V.34*bis*, 33.6 kbps modems. Both of these algorithms break the data into blocks, and the resulting intermittent output stream causes too many buffering headaches. In addition, the total end-to-end delay becomes objectionable. Without error-correction, the noise immunity of the audio-coding

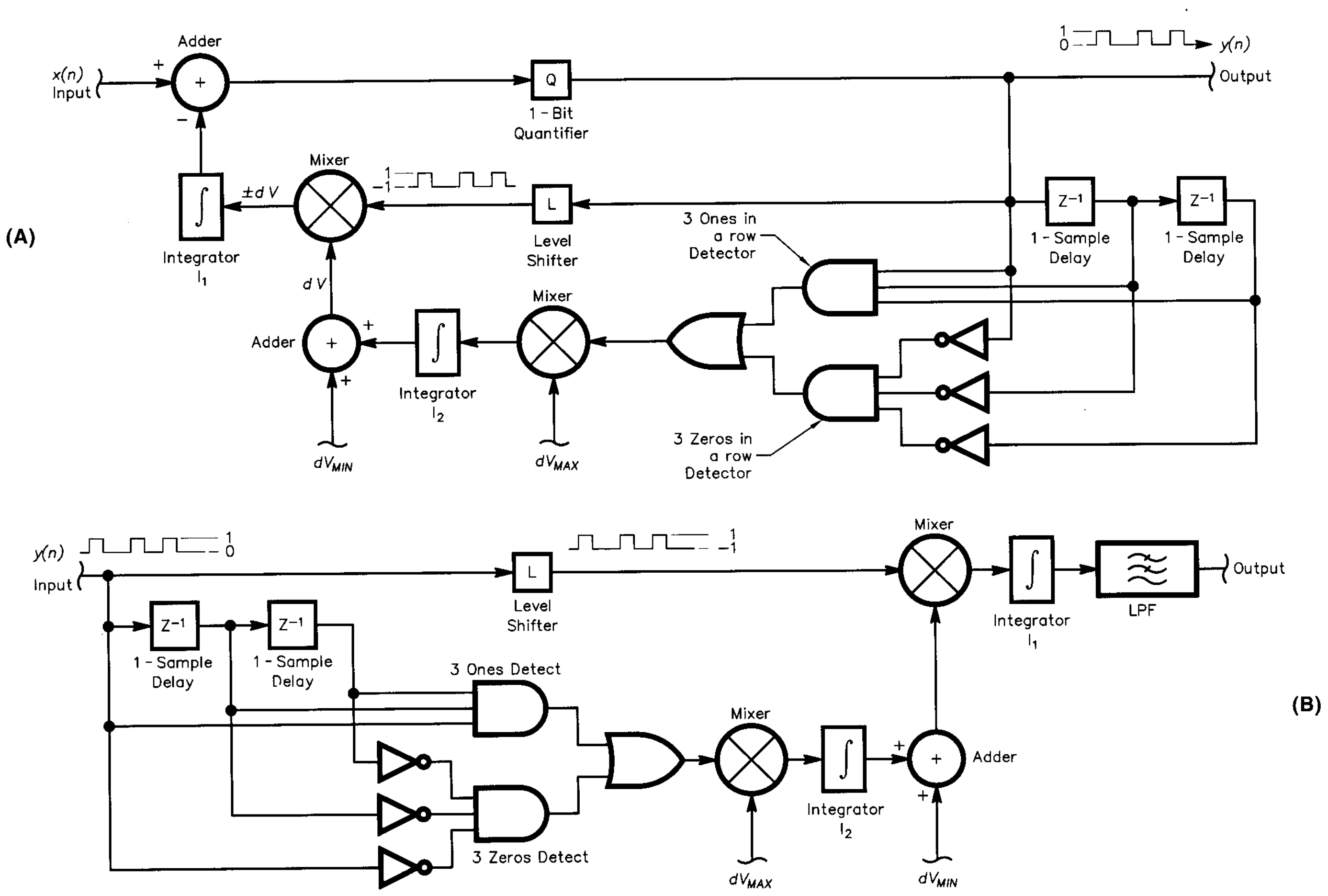

Fig 8—(A) a CVSD encoder block diagram. (B) a CVSD decoder block diagram.

scheme and that of the transceiver control protocol are of concern.

A mathematical analysis of CVSD performance in the presence of random bit errors isn't practical here. Fortunately, experiments have been performed in which subjective opinions were used to assess the perceived quality over the range of expected bit error rates (BERs).[12] On a scale from 1 to 5, listeners judged the quality of speech over BERs from 1×10^{-5} to 0.1. The scale represented the broad categories in Table 1.

Our goal of *toll quality* will be achieved with a score above four, whereas *communication quality* needs a score of at least three. This last term refers only to quality that in no way impairs intelligibility, not necessarily to good-quality audio.

In Fig 10, the results of opinion testing for the two compression methods we've discussed are plotted: μ-law PCM and CVSD. First, note that CVSD out-performs the other in the presence of errors. Second, it remains in the communications quality region until the BER rises to about 1%. This hardy performance makes CVSD an excellent choice where error-correction isn't possible.

Data Modem Performance on the PSTN

Back to the telephone company's traffic problems for a moment. Once the types of *differential quantization* such as the DM systems above were proven, "Ma Bell" decided she could double her traffic-handling capability by going to a 32 kbps system. A technique similar to CVSD called *adaptive differential pulse-code modulation* (ADPCM) was adopted as an international standard,[13] and equipment that could be retrofitted to the existing system became available for use at telephone switching centers. It's a lot less expensive to add a few boxes than to string a few thousand more miles of T1 lines! The voice quality with ADPCM remains excellent, but it ultimately limits the data rates of modems connected through it.

The new "56k" modems are capable of cross-loading information at 56 kbps, but to achieve these speeds, the bandwidth limitations of the central-office switches must be bypassed. An inherently digital line between the "uploader" and the central switch must be obtained. Because the "downloader" is still on the end of a regular telephone line, only speeds up to 33.6 kbps are possible in the other direction, and between two users without digital lines.

In fact, at my location in central Arizona, it's rare to achieve even a 33.6 kbps connection; more usually, 31.2 kbps is the best I can do. Occasionally during peak demand, the best connection possible is 26.4 kbps. This change in available connection quality occurs because the ADPCM equipment can adaptively alter its bit rate to provide greater-bandwidth connections during slack periods, and to lessen bandwidth during heavy usage to accommodate more calls. At 32 kbps, a T1 line can carry 48 calls; at 24 kbps, it'll handle 72. At 16 kbps, 96 simultaneous calls are possible, but even the voice quality starts to suffer considerably at this rate. Very occasionally—especially right after school lets out—I encounter an "all circuits busy" message! So, I've discovered that 56k modems aren't always worth the investment.

Table 1
Subjective Evaluation of Audio Quality versus BER

Score	*Quality*	*Impairment*
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

As demand continues to increase, the telephone companies will be forced to add lines and equipment. Although recent rulings have eroded their status as "monopolies," they have been in the unique situation of providing a service, the cost of which isn't proportional to usage. I can make as many local calls as I want each month, and the bill won't change. At some point, this scenario may affect the service provider's profit, and something must give.

As the PSTN actually forms the backbone of the Internet, it makes sense for the telephone companies to become Internet service providers (ISPs) as well. Then we'll see all kinds of new equipment down at the central office and bandwidth galore! The traffic capacity of fiber-optic cable (and even twisted pairs) is inherently much greater than what's currently allowed. It's a good bet that, when phone companies can justify charging for it, the bandwidth will become available—even though it could have been there all along.

In our remote-control system, the control-data throughput rate is usually much less than that required for the digitized audio. So, we invent a byte-synchronous interleaving scheme that time-multiplexes the two data streams, and we accept the very slight degradation of audio quality. Off-the-shelf DSVD modems are optimized for the opposite situation: more data and just enough digitized audio to get communication quality. Our desire is for toll quality, which implies the type of DR and SNR we derived previously.

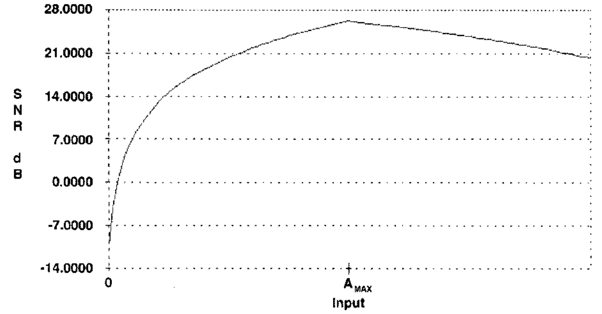

Fig 9—CVSD SNR-versus-input amplitude.

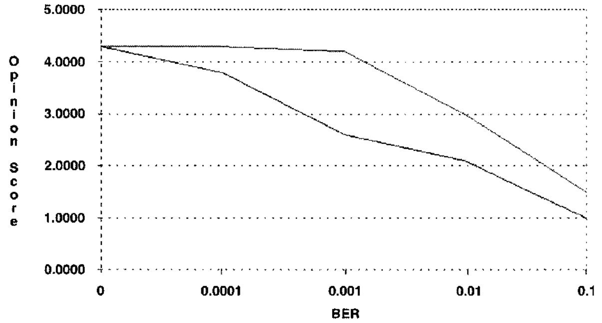

Fig 10—CVSD voice-quality score versus-BER.

Serial Remote-Control Protocols

As for the remote-control protocol, it's relatively easy to dream up byte sequences complex enough to achieve the required noise immunity. Command strings are sent from the serial port of the PC at the control point, multiplexed with any digitized audio, and transmitted to the remote site using the modem. After execution of the command, an acknowledgment is returned by the transceiver. This "handshake" is necessary to establish positive control.

Telemetry is continually passed from the transceiver to the control point using the same multiplexing method. The telemetry data indicate the received signal strength in receive mode, and the forward and reflected power in the transmit mode. Additional telemetry may be included to indicate the state of synthesizer lock, heat-sink temperature and other parameters.

Also, the byte-synchronous multiplexer can interleave control data for other devices at the remote site, such as antenna rotator controls. Over-the-air digital-mode operation of the transceiver is possible by locating the radio modem at the remote site, and interleaving the transmitted or received data into the bit stream.

Conclusion

Most of the technologies I've described in this article series are in current use by radio amateurs and others worldwide. I was motivated to write about them because the rate at which they're advancing is threatening to overcome our ability to keep up. At no point in the past have hams ever been in danger of falling behind the state of our art as much as we are today. While I believe this is true for other fields as well, it's especially true for electronics. I wish to emphasize that I *don't* believe hams will ever be very far behind the "power curve," because we're still mostly the ones pushing it forward.

Many thanks to Rudy Severns, N6LF; Bob Schetgen, KU7G, and the rest of the terrific staff of *QEX* for taking the time and having the patience to organize my ramblings into a harmonious whole. In addition, thanks to you many readers who have given me feedback. That's what it's all about!

Doug Smith, KF6DX/7, is an electrical engineer with 18 years experience designing HF transceivers, control systems and DSP hardware and software. He joined the amateur ranks in 1982 and has been involved in pioneering work for transceiver remote-control and automatic link-establishment (ALE) systems. At Kachina Communications in central Arizona, he is currently exploring the state of the art in digital transceiver design and with this issue he becomes Editor of QEX! See Empirically Speaking for more information.

Notes

[1]M. E. Frerking, *Digital Signal Processing in Communications Systems* (New York: Van Nostrand-Reinhold, 1993).

[2]A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing* (Englewood Cliffs: Prentice-Hall, 1975).

[3]D. Smith, "Signals, Samples and Stuff: A DSP Tutorial (Part 3)," *QEX*, July/August, 1998, pp 13-27.

[4]B. Widrow and S. D. Stearns, *Adaptive Signal Processing* (Englewood Cliffs: Prentice-Hall, 1985).

[5]E. Tenner, *Why Things Bite Back: Technology and the Revenge of Unintended Consequences* (New York: Alfred A. Knopf, 1996).

[6]S. M. Shinners, *Modern Control System Theory and Application* (Reading, Massachusetts: Addison-Wesley, 1973).

[7]D. Smith, "Signals, Samples and Stuff: A DSP Tutorial (Part 2)," *QEX*, May/June 1998, pp 22-37.

[8]*Part 47 CFR 97.7* and *97.109* (Washington: US Government Printing Office, 1997).

[9]N. J. Muller, *T-Carrier Compression: Adaptive Differential Pulse Code Modulation*, Strategic Information Resources, 1995; URL **www.ddx.com.**

[10]H. P. Westman, Ed., *Reference Data For Radio Engineers*, fifth edition (Indianapolis: Howard W. Sams & Co., 1968).

[11]J. A. Greefkes and K. Riemens, "Code Modulation with Digitally Controlled Companding for Speech Transmission," *Philips Technical Review*, 1970.

[12]N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video* (Englewood Cliffs: Prentice-Hall, 1984).

[13]*G.721: Adaptive Differential Pulse Code Modulation*, International Telecommunications Union, Geneva, Switzerland, 1984.