Landmark Articles

Spread Spectrum Communications: Myths and Realities

Andrew J. Viterbi

Originally published in
IEEE Communications Magazine
May 1979 — Volume 17, Number 3

Author's Introduction

When I wrote this tutorial article in 1979, I did not imagine that two decades later my introductory statements would appear to have been so prophetic. Until that time, the predominant use of spread spectrum was for military anti-jam and secure communication systems and for ranging in military and space missions. Today there are well over one hundred million consumers using devices that employ spread spectrum technology to provide wireless personal communication or position-location, or both. How this came about is easily explained in retrospect. The creation of the Global Positioning Satellite (GPS) System by the U.S. Government, based solidly on spread spectrum technology, provided mobile consumers with the ability to determine their position practically anywhere on earth. But the great majority of the growth was the result of the proliferation of wireless (cellular) telephony. This provided precisely the multi-user and multi-path interference environment for which CDMA is an excellent antidote, thereby enabling highly efficient utilization of precious spectrum resources. Building on the concepts of this and other papers, a decade later I led a technical team that developed the first successful CDMA cellular telephony system. With the adoption of CDMA by the International Telecommunication Union for enhanced voice and data services, in what has imprecisely been labeled the Third Generation of Wireless Systems, it now appears likely that spread spectrum technology will enable the majority of the world's wireless communication systems, which are destined to serve billions of consumers.

Spread spectrum communication techniques date back to the early fifties. Since the earliest applications, system improvements have been more evolutionary than revolutionary. Like most improvements in electronic systems, these are due primarily to the availability of ever higher-speed integrated circuit components, which translate in this case to wider spread spectra. In three decades the achievable spreading factor has grown by about three orders of magnitude[1] to the point that we are now limited more by bandwidth allocations than by technology limitations. Before we examine the quantitative effects of spreading, let us catalog briefly the multiple purposes of spread spectrum communications.

First, we note that spreading here refers to expansion of the bandwidth well beyond what is required to transmit digital data. Thus, a system transmitting data at a rate (R) of 100 Mb/s using approximately 100 MHz of bandwidth (W) is not spread at all, while a system transmitting at 100 bits/c spread over a spectrum of about 100 MHz has a factor $W/R = 10^6$, or 60 dB of so called processing gain.

PURPOSES

The purpose and applicability of spread spectrum techniques is threefold:

- Interference suppression
- Energy density reduction
- Ranging or time delay measurement

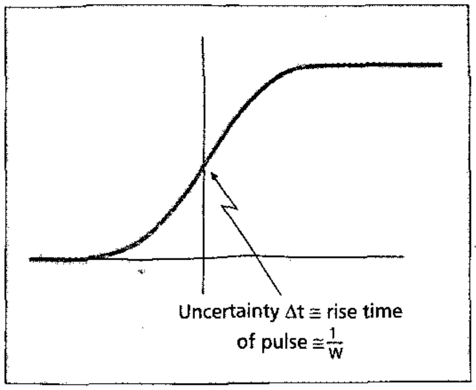
Foremost among these is the suppression of interference which may be characterized as any combination of the following:

- Other users: intentional (hostile or unintentional)
- Multiple access: spectrum sharing by "coordinated" users
- Multipath: self-jamming by delayed signal

Protection against in-band interference is usually called anti-jamming (A/J). This is the single most extensive application of spread spectrum communication. A similar application is that of multiple access by numerous users who share the same spectrum in a coordinated manner, in that each employs signaling characteristics or parameters (often referred to as codes) which are distinguishable from those of all other users. One reason for using this shared spectrum, so called code-division multiple access (CDMA), is that by distinguishing signals in this way, separation in the more common dimensions of frequency or time is not required, and hence the usual transmission tolerances need not be imposed on these parameters.

The third form of interference suppressed by spread spectrum techniques is the self interference caused by multipath in which delayed versions of the signal, arriving via alternate paths, interfere with the direct path transmission.

While the second and third forms of interference would appear more benign than that of a hostile emitter, the technique and effect are the same. What makes the intentional interference more challenging is the game aspect of the problem and the fact that the interfering source is generally granted much more power than the communicator, which is usually not the case for cooperating users and even less so for multipath interference.



■ **FIGURE 1.** *Time delay measurement.*

The second class of applications centers about the reduction of the energy density of the transmitted signal. This, too, has a threefold purpose:

- To meet international allocations regulations
- To minimize detectability
- For privacy

Downlink transmissions from satellites must meet international regulations on the spectral density of the signals received on earth. By spreading this energy over a wider bandwidth, total transmitted power can be increased, and hence performance improved. Spreading also decreases the detectability of a signal by a regulatory body which employs spectral analysis to monitor or regulate emissions. (It is not known whether bootleg radio amateurs are using spread spectrum modulation to evade FCC regulations.) Even more promising is the potential for achieving privacy in communication by spreading one's signal sufficiently to "hide" in the background noise.

The application of spread spectrum for ranging or position location is rapidly gaining in importance. In simplest terms, position location consists of measuring the delay of a pulse or pulses. Error in delay measurement is inversely proportional to the bandwidth of the signal pulse. This is most easily seen by the simple example of Fig. 1. The accuracy of the measurement Δt is obviously proportional to the rise time of the pulse, which is inversely proportional to the bandwidth of the pulse signal. Of course, a one-shot measurement on a single pulse is not very reliable. Rather, the spread spectrum signal used for ranging is a long sequence of polarity changes (binary PSK-modulated signal). Upon reception, this is correlated against a local replica and "lined up" to perform an accurate range or delay measurement.

BASIC TECHNIQUES

Having outlined the multiple uses of spectrum spreading, we must examine at least a superficial description of the concept before we can proceed to dispel myths and uncover realities about

The application of spread spectrum for ranging or position location is rapidly gaining in importance. In simplest terms, position location consists of measuring the delay of a pulse or pulses. Error in delay measurement is inversely proportional to the bandwidth of the signal pulse.

At the time of original publication Andrew Viterbi was with the Linkabit Corporation, San Diego, CA.

[1] *This parallels the evolution of data rate capabilities of digital communications.*

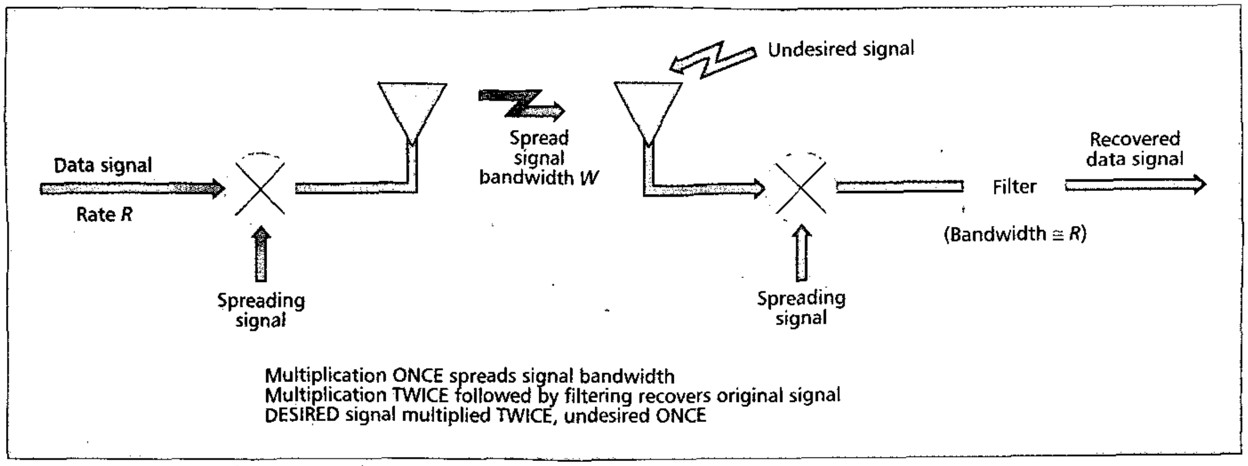


FIGURE 2. *Basic spread spectrum techniques.*

this increasingly popular technique. Fig. 2 is an all-purpose diagram to describe spread spectrum modulation. Multiplication of two unrelated signals produces a signal whose spectrum is the *convolution* of the spectra of the two component signals. Thus, if the digital data (binary) signal is relatively narrow-band compared to the spreading signal, the product signal will have nearly the spectrum of the wider (spreading) signal. So much for the modulator. At the demodulator, the received signal is multiplied by exactly the same spreading signal. Now if the spreading signal, locally generated at the receiver, is lined up (synchronized) with the received spread signal, the result is the original signal plus, possibly, some spurious higher-frequency components outside the band of the original signal, and hence easily filtered to reproduce the original data essentially undistorted. If there is any undesired signal at the receiver, on the other hand, the spreading signal will affect it just as it did the original signal at the transmitter. Thus, even if it is a narrow-band signal in the middle of the band of interest, it will be spread to the bandwidth of the spreading signal.

The result is that the undesired (jamming) signal will have a bandwidth of at least W. If its power is J watts, its average density, which is essentially uniform and can be treated as wideband noise, will be

$$N_0 = J/W \text{ watts/Hz.}$$

Let the desired component of the received signal have power S watts. Thus, if the data rate is R bits/second, the received energy per bit is

$$E_b = S/R \text{ watts} \cdot \text{second.}$$

Now it is generally recognized that digital communication system bit error rate performance is a direct function of the dimensionless ratio E_b/N_0, which for spread spectrum signals may thus be expressed as

$$\frac{E_b}{N_0} = \frac{S}{J}\frac{W}{R}$$

and hence, the jamming power-to-signal power ratio is

$$\frac{J}{S} = \frac{W/R}{E_b/N_0}. \quad (1)$$

This establishes that if E_b/N_0 is the minimum bit energy-to-noise density ratio needed to support a given bit error rate, and if W/R is the ratio of spread bandwidth to the original data bandwidth, also called the processing gain, then J/S is the *maximum tolerable jamming power-to-signal power ratio*, also known as the *jamming margin*.

We have come this far without even specifying the characteristics of the spreading signal. There are, in fact, two distinct classes of spreading techniques. The first is called *direct-sequence* or *pseudonoise* (PN) spread spectrum. Here the spreading is achieved by multiplication by a binary pseudorandom sequence whose symbol (switching) rate is many times the binary data bit rate. The spreading sequence symbol rate is sometimes called the *chip rate*.

The second class utilizes a *frequency-hopping* carrier. Here the spreading signal remains at a given frequency for each bit or even for several bits. Thus, locally it is no wider than the data signal, but when it hops to a new frequency, it may be anywhere within the "spreading" bandwidth W.

One fundamental difference between the two techniques is that direct-sequence PN spread signals can be coherently demodulated. With frequency-hopped signals, on the other hand, phase coherence is difficult to maintain when the signal frequency is hopped over a wide range; hence, this modulation is usually demodulated noncoherently.

We are now ready to explore several firmly entrenched items of common wisdom regarding the relative desirability of various features of spread spectrum systems. Often these attitudes are misguided, as we shall presently show. In all cases, the ideas hold for both classes of spread spectrum techniques, but for all but the last concept the arguments are somewhat simpler for direct-sequence spreading, which we shall therefore consider.

We are ready now to reveal the first of four myths.

First Myth

Error-correcting coding requires redundancy, which spreads bandwidth and thus reduces available processing gain for the available bandwidth.

Reality is, in fact, just the opposite. To see that coding does not reduce the *effective* processing gain, let us rewrite jamming margin (1) in terms of the *symbol*[2] *rate* R_s and the *symbol energy* E_s. These are related to the bit rate and the bit energy through the code rate r, defined as the number of data bits per transmitted symbol, or the inverse of the coding expansion factor. (For example, a rate 1/2 coded system transmits two code symbols for each data bit.) It follows that

$$R_s = R/r \quad \text{and} \quad E_s = E_b r.$$

Now if we repeat the previous dimensional argument replacing bits by symbols everywhere, we have

$$J/S = \frac{W/R_s}{E_s/N_0},$$

but substituting the preceding definitions for symbol rate and energy, we obtain for the maximum tolerable J/S ratio

$$J/S = \frac{W}{R/r}\frac{N_0}{E_b r} = \frac{W/R}{E_b/N_0}$$

which gets us back to (1). This may seem like sleight of hand, but it really is not. Moreover, although it will take some further reading to be convinced, we are really ahead of the game. For with coding, the required E_b/N_0 for a given level of performance (bit error rate) is actually *reduced*. Thus, for a given processing gain (W/R) the jamming margin is further *increased* by coding.

For those who are satisfied that spectrum spreading (especially direct-sequence PN) techniques make the noise look "white" while the signal energy, without or with coding, can be fully recovered by the receiver's "correlating" multiplier, the dispelling of the First Myth will come as no surprise. Yet it is often this sophisticated group who will fall prey to the

Second Myth

Error-correcting coding is effective only against uniform interference.

In particular, the myth continues: coding is not effective against *pulsed interference*. Yet, this is even more dramatically false than the First Myth. Let us consider what the effect of pulsed interference can be for an uncoded system. Suppose the jamming is present only a fraction $\rho < 1$ of the time, but that during this time, the noise density level is increased to a level N_0/ρ watts. This assumes spectrum spreading which turns the jamming signal into broadband noise and an *average power* rather than a peak power limitation on the jammer. (While this may be slightly pessimistic for the communicator, any other assumption is a risky bet against technological progress.) Now it is well known that with coherent demodulation an *uncoded* BPSK-modulated system produces a bit error rate P_b related to E_b/N_0 as

$$P_b = Q\left(\sqrt{2E_b/N_0}\right) \lesssim e^{-E_b/N_0} \tag{2}$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-u^2/2}du.$$

But if the noise is intermittent, and hence only with probability ρ corrupts a given transmitted bit[3] with the higher noise density N_0/ρ, the resulting bit error rate becomes

$$P_b = \rho Q\sqrt{2E_b\rho/N_0} \lesssim \rho e^{-\rho E_b/N_0}. \tag{2'}$$

Clearly, the jammer would choose the duty factor ρ which pessimizes performance—that is, maximizes bit error rate. In terms of the approximation, which is a strict upper bound, this occurs when

$$\rho = \frac{1}{E_b/N_0} \quad \text{provided } E_b/N_0 > 1$$

at which value

$$\max_{0<\rho<1} P_b \lesssim \frac{e^{-1}}{E_b/N_0}. \tag{3}$$

(Note that although we worked with the approximation for its simplicity, had we used the exact (Q-error-function) expression, the worst case ρ would be nearly the same and the maximum bit error rate would not be significantly lower.)

The result is quite dramatic. Pulse jamming—with spread spectrum but without coding—changes an exponential relation into an inverse-linear one. Numerically, if we desire bit error rate performance on the order of $P_b \simeq 10^{-5}$, stationary noise (or jamming) requires only $E_b/N_0 \simeq 10$ dB, while with pulse jamming we must have $E_b/N_0 \simeq 45$ dB, an increase in *required signal power of over three orders of magnitude*!

Amazingly, coding can almost fully restore this deplorable situation, but before we can explain why, we must briefly explore a summary of the general capabilities of coded systems.

For those who are satisfied that spectrum spreading (especially direct-sequence PN) techniques make the noise look "white" while the signal energy, without or with coding, can be fully recovered by the receiver's "correlating" multiplier, the dispelling of the First Myth will come as no surprise.

Error Correcting Coding Fundamentals

All that we need to know about coding for the present purpose is that for practically any memoryless channel, there are many good binary codes of rate r bits/symbol for which the bit error rate is upper bounded [1] by either

$$P_b < 2^{-K(\alpha-1)}, \quad \text{if it is a } \textit{block} \text{ code of block length } K \tag{4}$$

or

$$P_b < \frac{2^{-K\alpha}}{[1-2^{-(\alpha-1)}]^2}, \quad \text{if it is a } \textit{convolutional} \text{ code of constraint length } K. \tag{5}$$

In either case

2 Symbol rate refers to the code symbol of the error-correcting code, not that of the PN spreading code, which is usually called chip rate.

3 We assume for simplicity that a given interference pulse corrupts an integral number of bits. This is a reasonable assumption if the pulse width is many times the bit duration. Otherwise, the situation is actually less favorable to the jammer.

Coding Fundamentals

When a binary data stream must be transmitted over a noisy channel with a troublesome bit error rate, coding can be used to significantly reduce the error rate incurred by the message.

In **block coding** schemes, the message bit stream is partitioned into blocks of k bits, where k is the **block length**. Each such message block is replaced with an n-bit code word (n is bigger than k) which is transmitted in its place. Thus, every n bit transmitted "contains" only k message bits so that the **rate** r of the code is k/n bits per code symbol.

A common noisy channel model is the so called **Gaussian channel** where each bit, viewed as a square pulse of amplitude ±1, is independently subjected to additive noise and an error occurs when the noise alters the pulse polarity.

As a result of errors, the received n-bit block can be any of 2^n possible words. Since there are only 2^k different code words that could have been transmitted (one for each k-bit message block) and 2^k is typically much less than 2^n, the number of possible received words 2^n is much greater than the number of code words 2^k. For each received code word, the decoder decides what was the most likely code word that was transmitted, and the receiver then identifies the corresponding k-bit message block. In this way error correction can be achieved.

A notably different approach is **convolutional coding**. Here the incoming message bit stream is applied to a K-stage shift register which is shifted b bits at a time. For each K message bit stored in the register, there are n linear logic circuits which operate on the register contents to produce n code bits of the encoded output stream. For each shift of the register, b new message bits are inserted and n code bits are delivered, so that the rate is b/n information bits per code symbol. In this case, a particular code bit depends on K message bits, where K is called the **constraint length** of the code. Note also that a particular message bit remains in the register for K/b shifts, and thus influences the value of nK/b code bits.

Unlike block coding, the optimal decoding operation for convolutional codes requires a memory that stores, in effect, a function of the entire past history of the received bit stream. The performance (as measured by error rate) of a convolutional coding system improves as the complexity (i.e., memory) allowed for the decoder is increased. Several methods of decoding convolutional codes have been developed. The optimal (maximum likelihood) scheme is generally known as the Viterbi algorithm. Viterbi decoding for reasonably short constraint lengths is feasible to implement and high decoding speeds are achievable. For extremely low error probabilities, a large constraint length K is required. The computational complexity of Viterbi decoding for large K makes this approach impractical. Another approach, **sequential decoding**, then becomes more attractive. A third technique, **feedback decoding**, though inferior in performance against random errors, is particularly well suited to correcting systematic error bursts that may occur in fading channels.

Both fading and pulse jamming introduce memory in the channel and further modify the channel statistics. Yet the same coding techniques as used for the Gaussian channel are at least as effective here, provided interleaving is employed to reduce or eliminate this memory.

$$\alpha = \frac{r_0}{r} > 1 \tag{6}$$

provided the code rate $r < r_0 < 1$.

Performance then depends strongly on the value of the parameter[4] r_0. This parameter, and consequently α, is increased if the decoder is furnished with everything the receiver "knows" about the channel, that is, for binary symbols, not only the receiver's "belief" that the transmitted symbol was a "zero" or a "one," but how strongly the receiver believes this. This confidence in the decision is called a "soft decision," "quality information," or simply a "metric."

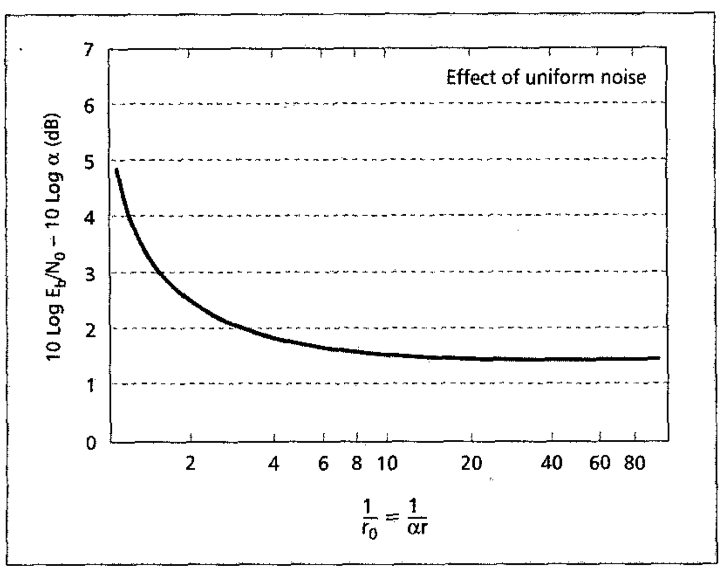


Figure 3. *E_b/N_0 requirements in additive uniform noise.*

Now for a uniform Gaussian channel, with soft decisions furnished to the decoder, the all-important parameter r_0 is a function of only the symbol energy-to-noise density. That is,

$$r_0 = 1 - \log_2[1 + e^{-E_s/N_0}] \tag{7}$$

where, as previously defined,

$$E_s = E_b r. \tag{8}$$

Combining Eqs. 6, 7, and 8, we can relate E_b/N_0 to α and r_0 as

$$\frac{E_b}{N_0} = -\alpha \frac{\ln\left(2^{1-r_0} - 1\right)}{r_0}. \tag{9}$$

This quantity expressed in decibels (10 log E_b/N_0 – 10 logα) is plotted in Fig. 3 as a function of r_0. Of course, Fig. 3 or Eq. 9 is meaningful only when taken together with Eq. 4 or Eq. 5. The interpretation is that for a given acceptable complexity of implementation, which is roughly proportional to 2^K for either class of codes, and a given code rate r, we need to select a value of α to guarantee the required P_b, according to Eq. 4 or Eq. 5. This establishes r_0 according to Eq. 6, and finally we obtain E_b/N_0 by adding 10 log α to the ordinate of Fig. 3 for the given r_0. If the resulting r_0 is greater than 1, we must choose a smaller code rate r for which $r_0 < 1$ for the

[4] *this parameter is also the so-called computational cutoff rate beyond which the sequential-decodinq mean computational load becomes unbounded.*

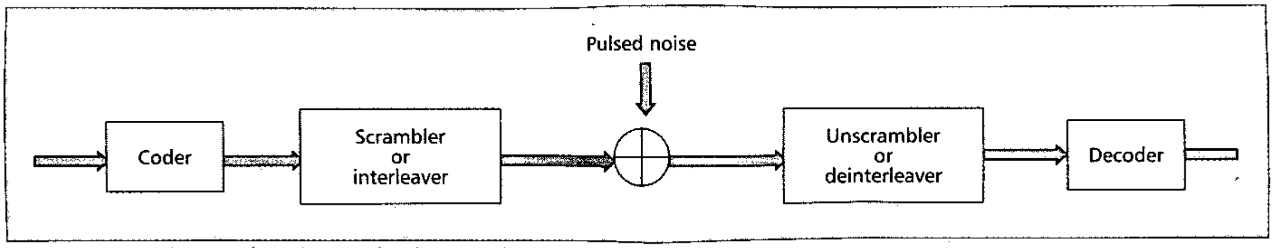


FIGURE 4. *Introduction of interleaving for disposing bursts.*

required α. Although Eqs. 4 and 5 are only upper bounds, and hence pessimistic estimates of the value of coding, they are sufficient to establish its merit, and in fact are reasonably accurate. As a practical matter, commonly used and commercially available convolutional decoders require an E_b/N_0 which is about 0.5–2 dB above the curve of Fig. 3 to achieve $P_b \approx 10^{-5}$.

For specific comparison, we have from Eq. 2 that to achieve $P_b = 10^{-5}$ on the uniform Gaussian channel without coding requires E_b/N_0 = 9.6 dB. With rate 1/2 convolutional codes, practical soft-decision decoders require between 3.5 and 4.5 dB for the same performance. Rate 3/4 codes require approximately 1 dB more. If the decoder is provided with only hard-decision inputs, it requires approximately 2 dB higher E_b/N_0.

Thus far we have considered only stationary wideband noise, whether of thermal origin, or so rendered by the direct-sequence spread spectrum technique we have investigated. We now return to the pulse jammer and show how coding remedies the deplorable situation that we left before the present digression.

Second Myth Revisited, Appropriately Armed

Suppose now that we code as before, but for the nonuniform (pulse) jammer. Spreading causes this to appear at the receiver as wideband noise of density level N_0/ρ, but for a reduced duty factor ρ. Suppose, as before, with little loss of reality, that an integral number of code symbols are affected by jamming. We cannot quite apply what we just learned about coding because the jamming pulses affect many contiguous symbols, so we can hardly call the channel memoryless, as required. But this is easily remedied. Suppose we construct a device that randomly scrambles the order of the symbols prior to transmission, but after coding, and puts them back in the right order after reception, but before decoding (Fig. 4). (Scramblers and unscramblers are more commonly called interleavers and deinterleavers.) But the unscrambler which restores the transmitted symbols to their right place in order actually scrambles the regular jamming pulses into random patterns.[5] Scrambling or interleaving thus makes our system memoryless again and we can apply our new found coding knowledge.

Without belaboring the exact details, arguing intuitively and believingly on the basis of Eqs. 2 and 2′, let us replace e^{-E_s/N_0} by $\rho e^{-\rho E_s/N_0}$ in all the formulas of the previous (uniform noise) section. Thus, Eq. 7 is replaced by

$$r_0 = 1 - \log_2\left[1 + \rho e^{-\rho E_s/N_0}\right]. \qquad (7')$$

Combining Eq. 7′ with Eq. 6 and Eq. 8, we get

$$\frac{E_b}{N_0} = -\frac{\alpha}{\rho r_0}\ln\left[\frac{2^{1-r_0}-1}{\rho}\right] \qquad (9')$$

which obviously reduces to Eq. 9 for uniform jamming (ρ = 1).

This is maximized by a jammer with duty cycle

$$\rho = (2^{1-r_0} - 1)e \qquad (10)$$

provided $r_0 > 1 - \log_2(1 + e^{-1}) = 0.548$ for which

$$\max_{0<\rho<1}\frac{E_b}{N_0} = \frac{\alpha e^{-1}}{r_0\left(2^{1-r_0}-1\right)} \quad \text{for } r_0 > 0.548. \qquad (11)$$

If $r_0 \le 0.548$, ρ = 1 maximizes E_b/N_0 and Eq. 9′ reduces to Eq. 9. This says that if r_0 is small enough, no penalty is paid to a pulse jammer. Even for $r_0 > 0.548$, the penalty is small, as seen

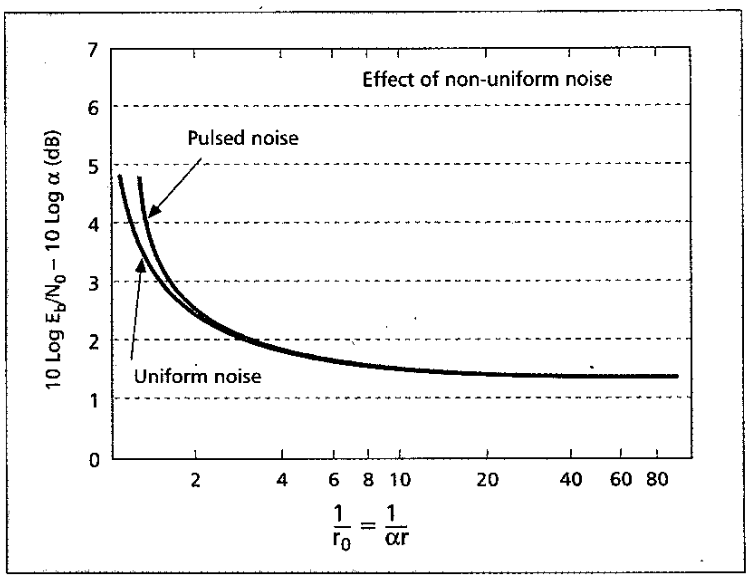


FIGURE 5. *E_b/N_0 requirement in pulsed and uniform noise.*

[5] *Note the parallel with spectrum spreading; the second multiplier unspread the signal and spread the interference; here the second device unscrambles the desired code sequence and scrambles the undesired pulsed interference.*

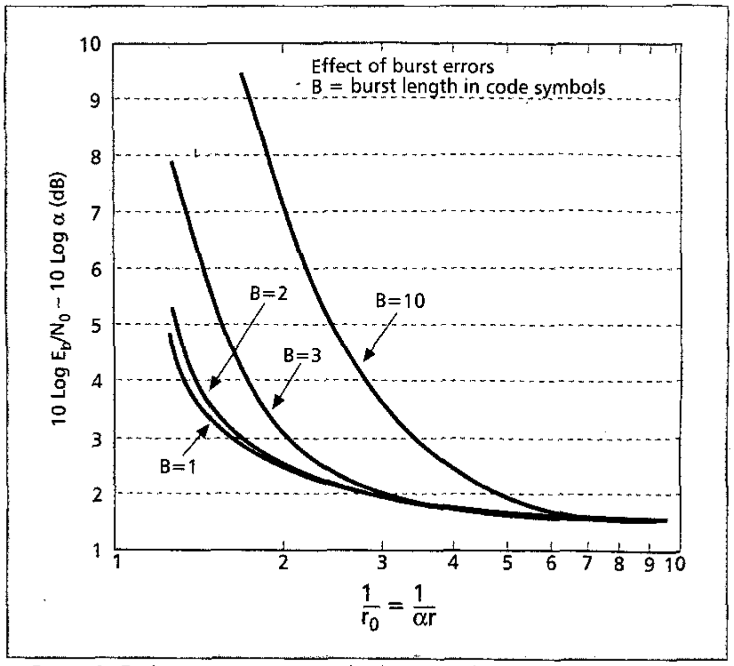


FIGURE 6. E_b/N_0 *requirement in pulsed noise without interleaving.*

in Fig. 5, which shows the new pulsed noise case and reproduces the uniform noise case from Fig. 3.

Thus, even more amazing than the original 35 dB loss to pulsed noise, *coding recovers it all and then some*. In spite of this, there still are skeptics who believe in the

Third Myth:
Interleaving destroys memory. Memory can be exploited to correct errors. Hence, interleaving is bad.

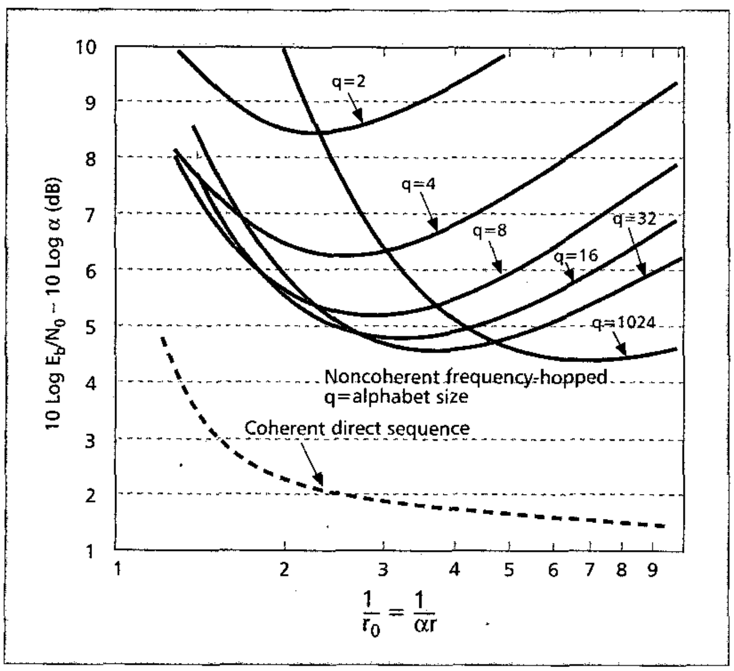


FIGURE 7. E_b/N_0 *requirement for noncoherent frequency-hopped systems.*

The discussion leading to the system design of Fig. 3 should suffice to dispel such misgivings, particularly when we recall that soft decisions contain about all the information available about the channel (granted, of course, that memory may be exploited to extract such quality information). To definitely put to rest all discussion of the matter, let us consider a burst interference phenomenon which always affects B symbols, and let there be no overlaps of bursts (overlaps can only help the communicator). In this case, the key parameter of Eq. 7′ becomes

$$r0 = 1 - \frac{1}{B}\log_2\left\{(1-\rho) + \rho\left[1 + e^{-\rho E_s/N_0}\right]^B\right\} \qquad (7'')$$

which causes Eq. 9′ to be replaced by

$$\frac{E_b}{N_0} = -\frac{\alpha}{\rho r_0}\ln\left\{\left[\frac{2^{(1-r_0)B} - (1-\rho)}{\rho}\right]^{1/B} - 1\right\} \qquad (9'')$$

which leads to the plots of Fig. 6. Clearly Eqs. 7″ and 9″ reduce to Eqs. 7′ and 9′ for $B = 1$, which is the case when interleaving is employed.

In fact, the situation is even worse than shown in Fig. 6. For as r decreases, the number of symbols per bit and, for a constant-duration interference pulse, the number of symbols per pulse (B) increases. Thus, keeping B fixed, as is done for convenience in Fig. 6, gives misleadingly favorable results.

Our final "myth" happens fortuitously to coincide with reality. We call it, therefore, a "Folk Theorem." It concerns an interesting comparison of direct-sequence (coherent) spreading with frequency-hopping (noncoherent) spreading:

Fourth Folk Theorem: (Myth = Reality)
Performance of frequency-hopped spread spectrum is 3 dB worse than that of direct-sequence (PN) spread spectrum.

The commonly invoked "mythical" argument is that noncoherent systems can utilize at best orthogonal signals (e.g., binary FSK modulation) instead of antipodal signals (binary PSK modulation) and this accounts for the 3 dB. The trouble with this argument is that it ignores the possibility of higher signaling alphabets (such as MFSK) and, worse still, the real possibility that frequency-hopped systems may be more vulnerable to nonuniform interference.

We note, in fact, that in frequency-hopped systems, the jammer need not pay the cost of a higher peak power signal, for if he jams just a fraction of the band,[6] $\rho < 1$, with power density N_0/ρ, he will appear to the receiver just as a partial-time jammer. Note also that if the hopping rate is at least as great as the symbol rate, interleaving is unnecessary.

[6] *Possibly varying this by hopping himself in order to defeat the obvious communicator strategy of determining the jammed region and staying out of it.*

For alphabets of size q, Eqs. 4 and 5 must be modified to become

$$P_b < q^{-K(\alpha-1)} \text{ block codes} \qquad (4')$$

and

$$P_b < \frac{(q-1)q^{-K\alpha}}{\left[1-q^{-(\alpha-1)}\right]^2} \text{ convolutional codes} \qquad (5')$$

which reduce to Eqs. 4 and 5 when $q = 2$. There is nothing to gain by using multiple signal alphabets for coherent, direct-sequence systems, but with frequency-hopped systems, we can show [2] that for the worst case partial-band jammer

$$\max_{0<\rho<1} \frac{E_b}{N_0} = \alpha \frac{(q-1)4e^{-1}}{(\log_2 q) r_0 \left(q^{1-r_0} - 1\right)}$$

provided

$$\frac{E_b}{N_0} \geq \frac{3}{r \log_2 q}.$$

For asymptotically large q, this approaches

$$\frac{E_b}{N_0} \simeq (4 \ln 2)\alpha$$

which is exactly a factor of 2 (3 dB) above the minimum of Fig. 3 which occurs as $r_0 \to 0$. This comparison is shown in Fig. 7, which also shows the diminishing returns of using alphabet size $q > 8$. The asymptotic minimum is virtually reached for $q > 32$.

Notice that noncoherent frequency-hopped systems exhibit a minimum, while coherent direct-sequence systems improve monotonically as $1/r_0$ increases. The explanation of this behavior is better understood by examining Fig. 8 which is a more detailed and more realistic examination of performance for an octal alphabet. Here the assumptions are more realistic. Specifically, channel quality information is limited to two bits (four levels) out of each of the q matched filters. This also allows for a practical automatic gain control (AGC) technique, which has not been mentioned up to this point.

The curve for $\rho = 1$ is, of course, for uniform noise. The increase at high rates (r and r_0 close to 1) is due to the lack of coding redundancy. The increase at low rates is due to the higher loss, characteristic of noncoherent combining of symbols in high-diversity (here low-rate) noncoherent communication systems. As ρ, the fraction of the interference bandwidth, decreases, performance gets increasingly worse at high rates since the diversity is lacking to overcome the strong jammer. But as r_0 decreases, diversity becomes sufficient to fully defeat the low ρ jammer, as shown by the family of curves of Fig. 8.

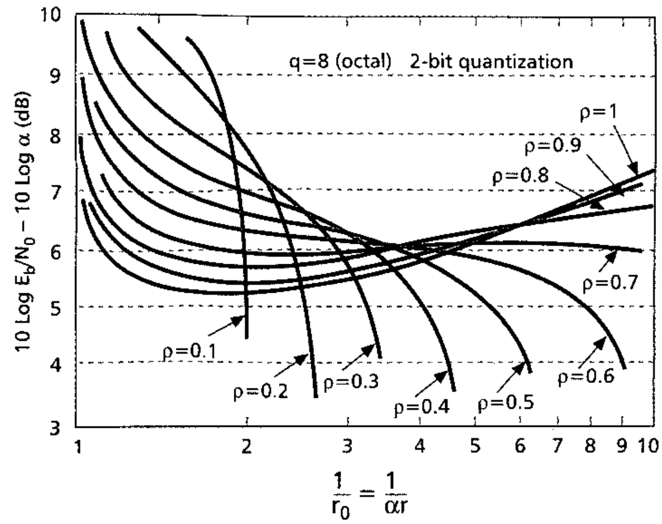


FIGURE 8. *E_b/N_0 requirement for octal noncoherent frequency-hopped system in partial band interference with receiver quantization (ρ = interference fractional bandwidth).*

SUMMARY

Beyond cataloging the many uses of spread spectrum communication, we have made no attempt to be uniform in our treatment of this extensive and many-faceted field. We have concentrated rather on its application for the suppression of interference, and have made three main points:

- Coding is always useful, and it may be critical to adequate performance of spread spectrum systems, particularly when the nature of the interference is partial-time, or in the case of frequency-hopped spreading, partial-band. Proper interfacing of the decoder to the demodulator, in utilizing quality (soft decision) information, is important to ensure maximum benefit from coding.
- Interleaving or scrambling may be equally essential in the presence of burst interference.
- Direct-sequence spread spectrum efficiency is about double that for frequency hopping. This is tantamount to doubling the processing gain W/R. However, frequency-hopping technology may have an edge in achievable band spreading of one or more orders of magnitude over direct-sequence spreading technology, which greatly overshadows the "system edge" of the latter.

REFERENCES

[1] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, New York: McGraw-Hill, 1979.

[2] A. J. Viterbi and I. M. Jacobs, "Advances in Coding and Modulation for Noncoherent Channels Affected by Fading, Partial-band and Multiple-Access Interference," *Advances in Communication Systems*, vol. 4, New York: Academic, 1975.