Spread-Spectrum Applications in Amateur Radio

Through the properties of their coded modulation, spread-spectrum systems can provide multiple-access, low-interference communications to radio amateurs.

By William E. Sabin,* W0IYH

Traditionally, the emphasis in Amateur Radio has been to make a transmitted signal as narrow in bandwidth as possible. Also, receivers are made as narrow and as interference-immune as possible. In this way, many signals can occupy a ham band successfully. This approach has been successful, to a point. But if a group of stations is on one frequency (the pileup!), the system does tend to break down, with disastrous results.

A new approach is being advanced that amateurs should take a look at. Military and commercial organizations are developing spread-sprectrum systems. Such systems deliberately occupy a wide band of frequencies, as part of a strategy to make communications more reliable and more secure, or private. (The word "privacy," as used here, has a special meaning in Amateur Radio, which will be considered later.)

To be more exact, a transmitter sends its message in such a way that a wide spectrum is used, according to a very carefully designed plan. The receiver has the ability to use this same plan in reverse, to convert the signal back to narrow-band form. By performing these actions in the right way, privacy and interference immunity are improved. Fig. 1A shows a conventional transmitter output of, say, 1 kW. With spread-spectrum operation, this same power is spread out as shown in Fig. 1B. There is no strong carrier at any one frequency. Within a 3-kHz band, the amount of signal is greatly reduced. In fact, it may be less than the noise level. But after "despreading," the signal once again looks like the signal at A.

This scheme is different from wide-band fm in that the message itself does not produce the spread spectrum. Instead, another agent is employed to spread the signal. Also, there is no carrier, as in an fm system.

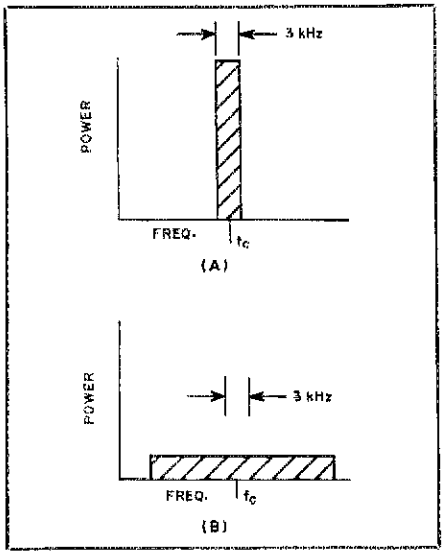

Fig. 1 — The power distribution of a conventional communications signal (A) versus a spread-spectrum signal (B). The same total power is contained in both signals.

Why Amateur Spread Spectrum?

Consider the network shown in Fig. 2. Using spread spectrum, stations A and B can communicate privately, while C and D do the same. Or, A can address all stations. Also, a member of this net can address an entirely different net or a single member of the other net. Or, B could address all nets simultaneously. In a particularly large region, these nets all use the same frequency band, and no equipment retuning is needed for any of the above operations.

In any of these operations, a degree of "privacy" is achieved, in the sense that communications are programmed according to the requirements of the moment. This "selective calling" is achieved by using the microcomputer-based "protocol," or message-routing procedures. The use of spread spectrum is an enhancement of the "packet radio" techniques that advanced amateurs are now experimenting with. By adding to this packet system a carefully managed spread-spectrum protocol, it should be possible to greatly reduce "collisions," avoid interference and add significantly to the repertoire of the packet system.

This extra element of spread-spectrum management, in addition to time management, makes it possible to reduce the guesswork with respect to frequency selection, which is a major problem in Amateur Radio. The difficulty is that a clear frequency at my station may not be clear at

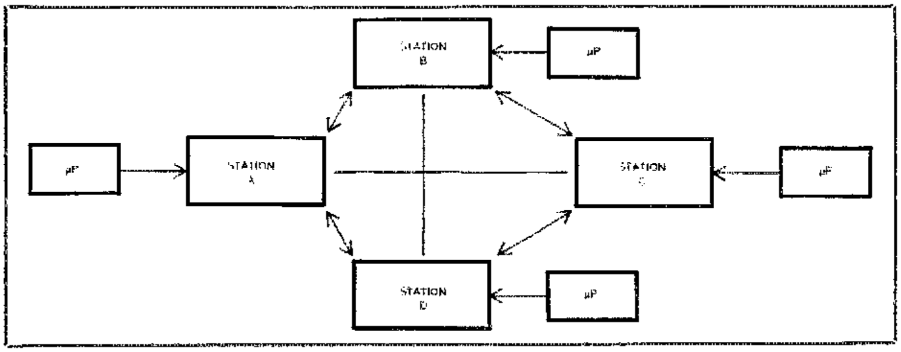

Fig. 2 — A network of stations using μP or microprocessor-based protocols. Network and internetwork communication may be obtained without equipment retuning.

---

*Rockwell-Collins, Telecommunication Products Div., Cedar Rapids, IA 52498

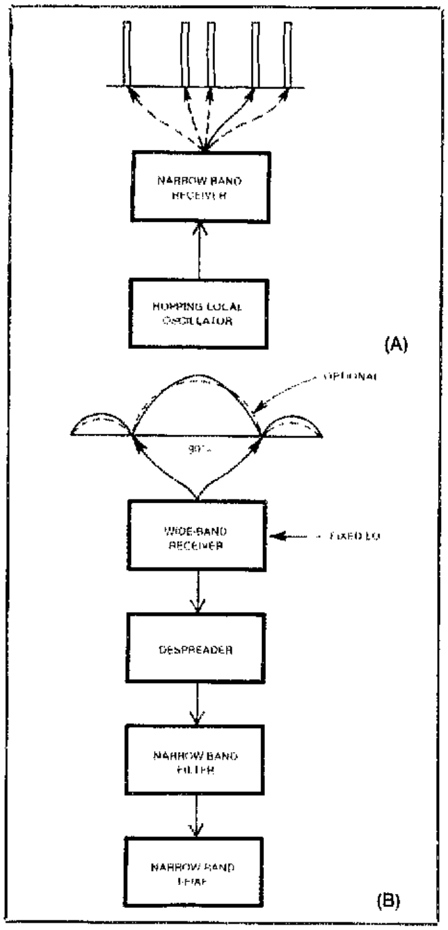

Fig. 3 — The two major types of spread-spectrum operation. At A, the frequency-hopping mode is illustrated; at B is the direct-sequence mode. The spreading is caused by applying a coded modulating signal at the transmitter that is independent of the intelligence modulation. The same predetermined code is applied in reverse at the receiver to despread the signal, and the intelligence is then demodulated in conventional fashion. If different spreading codes are appropriately chosen, interference-free operation may be obtained with different stations sharing the same frequency spectrum.

your station. We need a way to improve this situation. The communication is not restricted to data. Digitized voice messages can be sent, stored at the receiving station and converted back to speech at the completion of the message.

The information below summarizes the possible advantages of combining spread spectrum with packet-network protocols. There is no particular argument in favor of using spread spectrum by itself.

- Voice or data
- Simultaneous net combinations
- Network privacy
- Reduced collisions between
  A) spread and nonspread systems
  B) other spread systems
- Combines well with packet protocols
- Enhances packet repertoire

The important extra element is the voluntary "discipline" to which the various players can subscribe. The discipline is handled by the various personal computers involved, leaving the operator with a great deal of freedom.

Fig. 3 illustrates the two major types of spread-spectrum reception that amateurs might consider: frequency hop and direct sequence. At A, a narrow-band receiver is rapidly tuned through a predetermined set of frequencies. The desired signal is available at each of these frequencies when the receiver tunes there, according to a predetermined plan. Other signals are programmed so that they are seldom on the frequency to which the receiver happens to be tuned at one particular moment, although they may share this frequency at different times.

At B, a wide-band receiver, tuned to a fixed frequency, listens to a signal that has been carefully spread out in the manner shown, according to a predetermined plan. The receiver possesses the key by which to despread this signal and put it through a narrow filter. All other signals using the same frequency band are essentially ignored. A different spread-spectrum signal fails to despread. A conventional signal is likewise unable to penetrate the narrow filter, because it is converted to a wide-band signal by the receiver.

On the hf bands, frequency hopping would be better because of its narrow-band nature. At uhf, direct sequence offers advantages. By carefully selecting the kind of modulation in direct sequence, the spectrum is improved as shown in the dashed line in Fig. 3B. This could be msk (minimum-shift keying) modulation.

Direct-Sequence Spreading

Fig. 4A shows a conventional phase-shift-keyed data signal and the transmitted spectrum it produces. The width of the main lobe is twice the data rate, and in an amateur RTTY system would be less than 1.5 kHz wide. In Fig. 4B, each data bit is modulated by a PN code, where PN means pseudo-noise. There may be thousands of PN code bits for each data bit. The data bit has the effect of inverting, or not inverting, a group of PN bits, depending on a data 1 or 0. After combining these code streams, a mixer with a fixed-frequency local oscillator produces the spread-spectrum rf signal. The width of the main lobe is twice the PN code rate.

The design of the PN code is critical to the performance of the spread-spectrum system, and will be covered in the next section. The spectrum in Fig. 4B shows that, when listened to by a conventional receiver, a weak, hissing, noise-like signal is heard. It is not truly random noise, because of the nonflat nature of the spectrum, and because the PN code does repeat itself after some long time interval. The expression "pseudo" noise is therefore appropriate.

Fig. 5A shows a PN code generator. A shift register with N stages, initially loaded with all 1s, generates an output that is

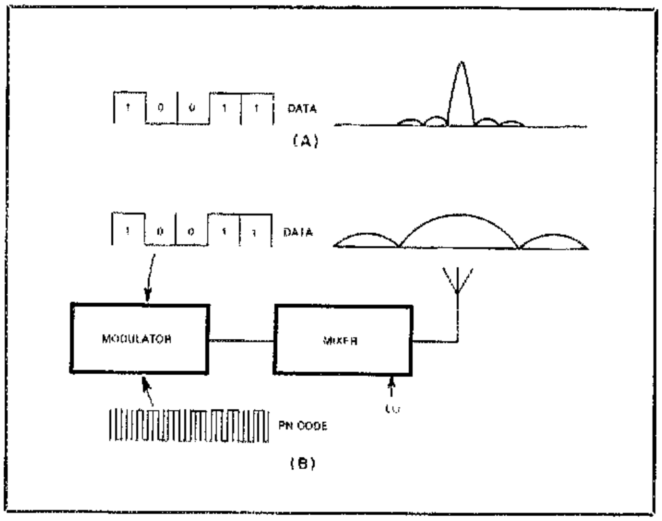

Fig. 4 — At A, a conventional phase-shift-keyed (psk) data signal and the transmitter spectrum that it produces. At B, the same data signal and a pseudo-noise (PN) code are combined to produce a direct-sequence psk signal with its much broader spectrum.

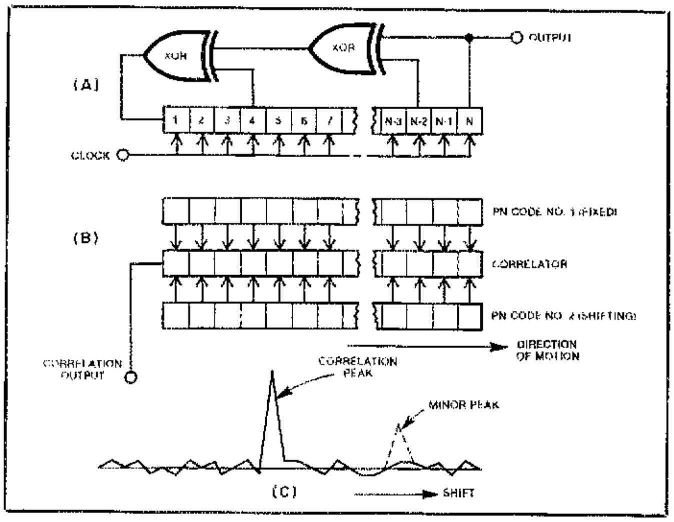

Fig. 5 — A PN code generator, A, and a correlator, B. The strings of blocks represent shift registers. At C is shown the correlation output with shifts of data in the lower shift register. This output peaks when synchronization is obtained.

influenced by the XOR feedback gates. After as many as $2^N - 1$ pulses, the code repeats itself. Code length can be from, say, 10 bits, to 10 million or more. If my transmitter and your receiver use the same PN generator circuit, we can communicate.

Fig. 5B shows a correlator. One input is a fixed (or static) PN code that has been stored. The other input is the received signal, which ripples through the bottom section. The output of the correlator, Fig. 5C, is very small (±1 or 0) except when the two codes exactly coincide, at which time the output equals N. When this peak occurs, the two codes are "synchronized," or correlated.

A well-designed code has only one of these sharp peaks. Other codes will have other minor peaks that can produce "false" synchronization. Also, a short segment of a good code may not be so good in this respect. Therefore, the register length should be as long as the code, or at least as long as possible. A good possibility for amateur use is to combine two short Gold codes.[1] The resultant codes have good correlation properties and are easy to generate.

Fig. 6 shows two types of direct-sequence receivers. In Fig. 6A, the despreading is done at i-f. A narrow-band crystal filter lets only the despread signal get into the data or voice detector. In B, a double conversion takes place, and the correlator, operating at baseband, performs the despreading. A low-pass filter then passes the desired signal only, which can be derived from the correlator output. Some receivers use both methods, that at A for data detection, and that at B for synchronization.

Direct-Sequence Synchronization

Fig. 7 shows, in a very general way, how synchronization is achieved and maintained. One way is to slow down or speed up the PN code generator in such a way that it can search backward and forward in time to "acquire" the incoming code. Once the code has been acquired, a tracking operation takes place, so that the PN code stays closely aligned with the desired signal. Acquisition is greatly aided by the use of very stable clocks and by prealignment of the codes, so that only a small amount of searching is needed. For example, using WWV, all code generators could be initialized each hour.

Another method uses a special, short preamble that the receiver quickly recognizes. This recognition starts the tracking operation. In an amateur application, this would be easier and cheaper to implement. In this case, the prealignment

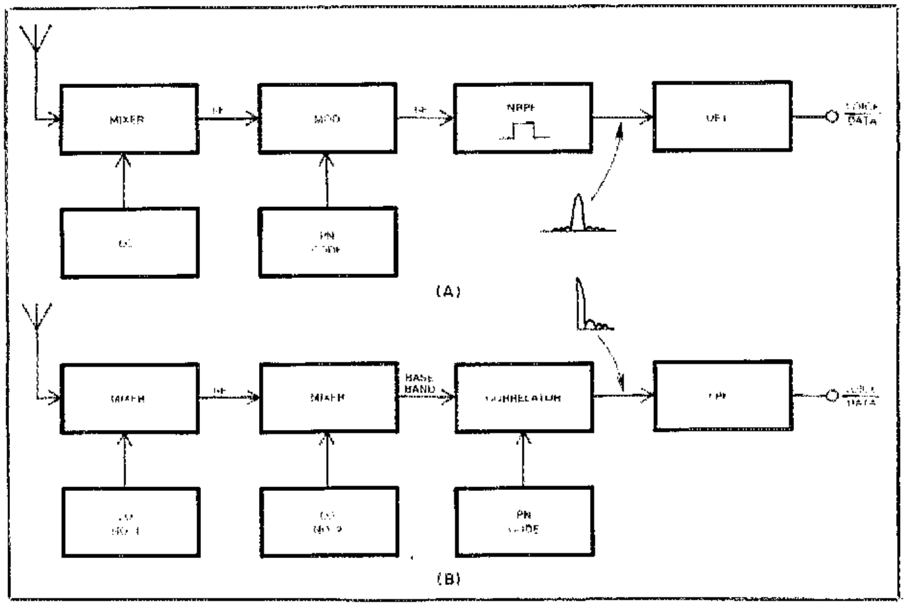

Fig. 6 — Block diagrams of two types of direct-sequence receivers. See text.

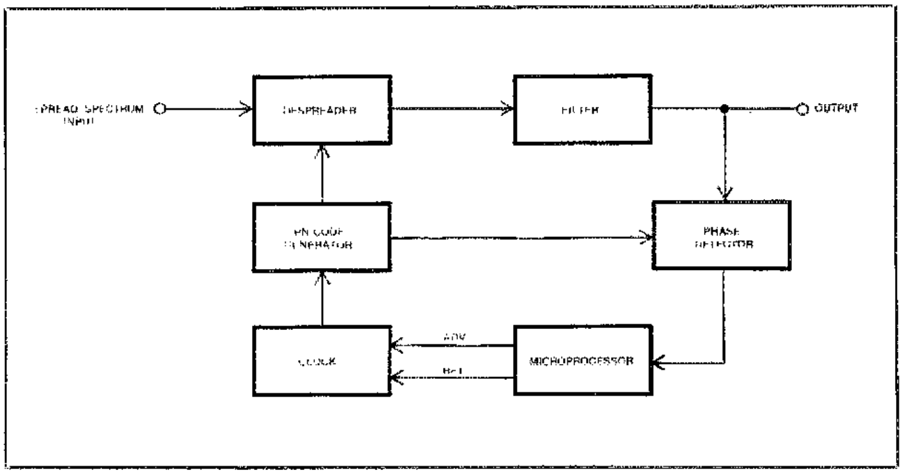

Fig. 7 — A system for synchronizing and tracking with direct sequence.

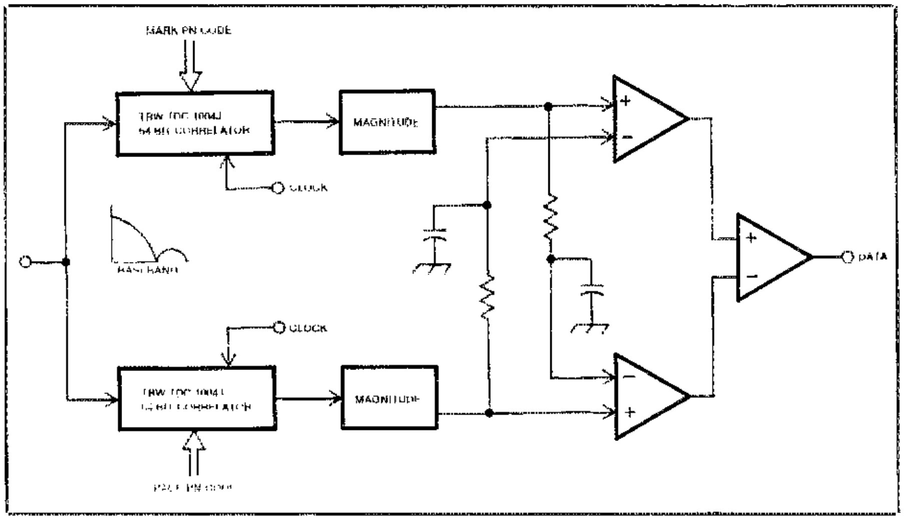

Fig. 8 — A correlator that operates at baseband with direct sequence.

---

[1] [Editor's Note: Gold codes are a family of codes named after the developer, R. Gold. The significance of the family is that a large number of codes may be obtained with relatively short shift registers in the code generator.]

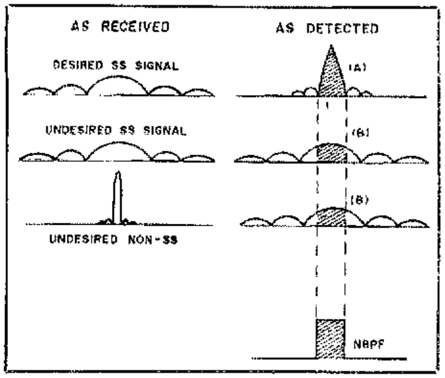

Fig. 9 — Illustrating interference rejection in direct sequence. Processing gain is obtained with despreading, and in decibels is equal to 20 log (A)/(B), where (A) and (B) are the amplitudes depicted in the shaded areas at the right.

of codes and time-of-day clocks would not be needed. When the preamble is sent, it is immediately followed by the start position of the long PN code. The receiver performs the identical operation. Sometimes, the preamble is sent as a frequency-hopping signal rather than direct sequence.

The circuit of Fig. 8 uses two TRW 64-bit correlators. One recognizes a PN sequence that identifies a mark, and the other recognizes an unrelated sequence that signifies a space, using bi-phase modulation. The correlator outputs can be positive or negative, so a full-wave rectifier (using op amps and diodes) is needed.

Each comparator looks at the instantaneous output of its correlator and the average output of the opposite correlator. This circuit, with very little modification, can detect marks and spaces in a frequency-hop radio, in which up to 64 bits of information would signify the message bit.

Fig. 9 shows how a desired spread spectrum and an undesired signal, either spread or narrow-band, are interpreted by a spread-spectrum receiver. The narrow-band signal is "smeared" by the receiver so that little energy is passed by the narrow band-pass filters. If the undesired local signals are strong enough, however, they can still override the desired distant signal, even after it has been despread, as shown. This "near-far" problem in direct sequence is an important limitation. Frequency hop seems to be more immune.

Frequency-Hop Spreading

Fig. 10 is a block diagram of a frequency-hop transmitter. A fast-frequency-change local oscillator is needed. The time that it takes to settle on the next frequency should be less than 10% of the dwell time on that frequency. Hop rates of 10 to several thousand per second are feasible with today's technology, and amateurs should experiment with speeds over the entire range. For example, a real opportunity exists for innovation in a low-cost, fast-hopping synthesizer design.

Shortly before changing frequency, the signal is smoothly attenuated, as shown. After the hop, the signal is brought up again, smoothly. This is necessary to reduce the transmitted spectrum (key clicks) as much as possible and to allow the LO time to make its frequency change. The hopping pattern is all under microprocessor control and determines which station or network will be addressed.

The analog voice or fsk data is filtered by the narrow band-pass filter. In other words, frequency hop is basically a narrow-band mode at any one frequency. Following the mixer, wide-band amplifiers and a wide-band antenna are needed so that equal power output occurs at each frequency.

For voice, a-m (with carrier) is preferred to single sideband because ssb causes phase jumps between hops, which produce excessive noise and distortion. Analog voice is not a preferred mode, in general, in frequency hop. At very low hop speeds, however, ssb should be considered as a possibility.

Fig. 11A shows the transmitted signal, smoothly attenuated between hops. The dashed lines show possible amplitude variations with a-m operation. Fig. 11B shows the spectrum. The tapered Christmas-tree shape at each frequency is caused by the turning on and off (Fig. 11A), and also by the mark/space information or a-m sidebands. To minimize interference, the drop-off in the spectrum should be as fast as possible, consistent with good communication. We can also visualize that the receiver bandwidth should be only wide enough to receive, say, 90% of the total signal energy at each frequency.

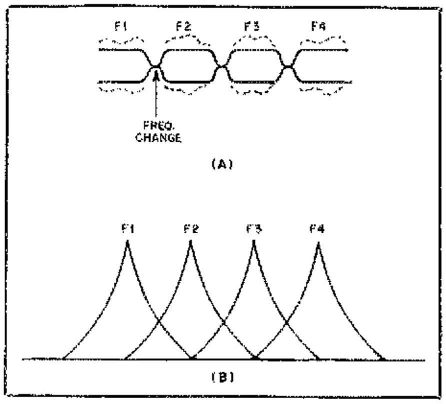

Fig. 11 — At A, an amplitude-versus-time representation of a frequency-hopping signal. The broken lines indicate possible variations arising from amplitude modulation of the signal. At B, an amplitude-versus-frequency representation of the same signal; any predetermined arrangement of frequency hopping may be used.

A frequency-hop receiver is shown in Fig. 12. An antenna input switch controls the turn-on at each frequency in a way that reduces intermodulation with strong, undesired signals on nearby frequencies. After mixing, a narrow filter leads to the signal detectors. The outputs of these detectors provide signal information to the microprocessor to control the synchronization algorithm and to determine that sync has occurred. When synchronization is achieved, a tracking operation is started in which the hop clock rate is adjusted momentarily. Later paragraphs cover this topic more thoroughly.

Instead of a single filter plus discriminator, consider two narrow filters, one for mark and one for space. The outputs are rectified and compared to determine the

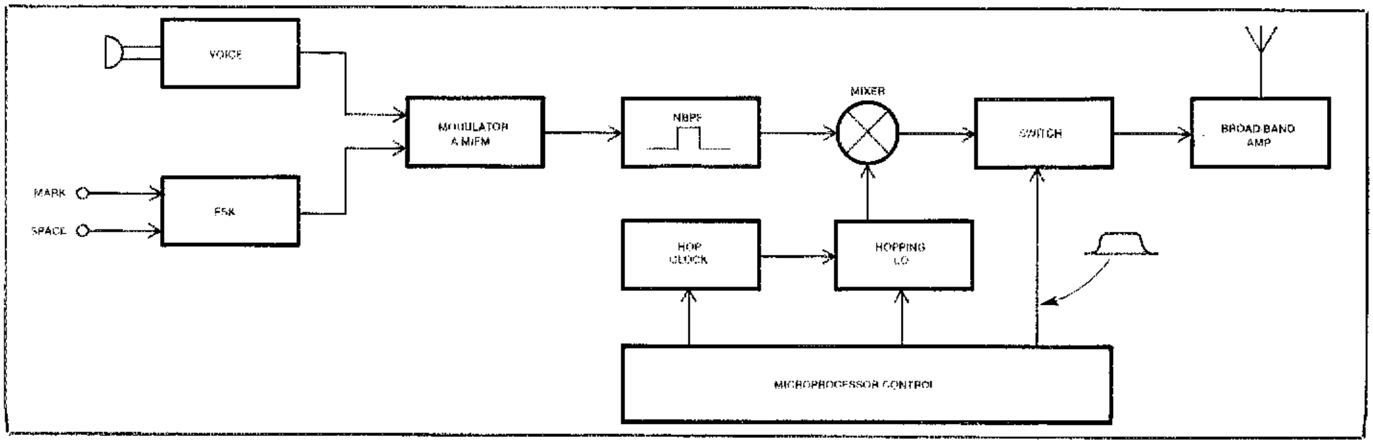

Fig. 10 — A frequency-hop transmitter. The hopping LO must be capable of making fast frequency changes with short settling times.

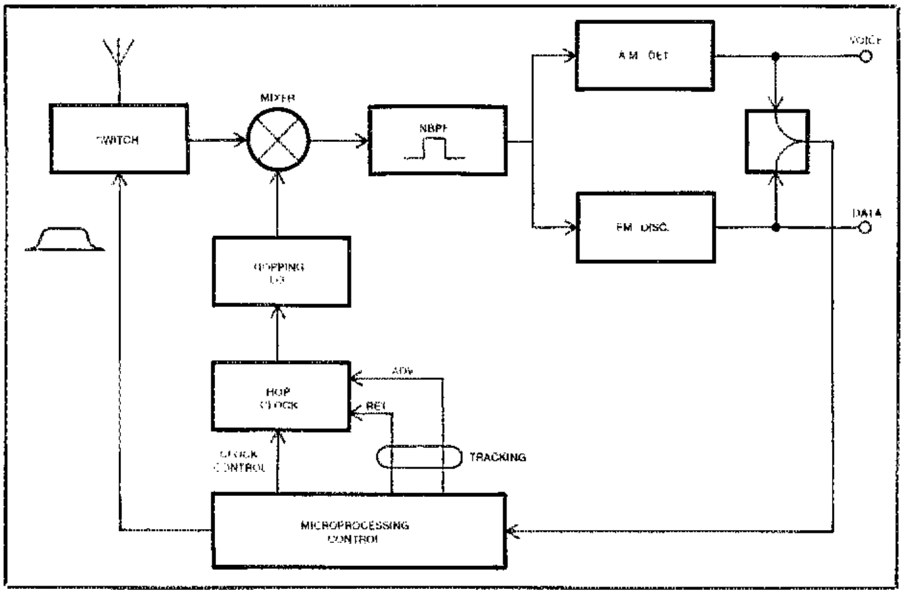

Fig. 12 — A frequency-hop receiver. The antenna switching reduces intermodulation from strong adjacent-frequency signals.

mark or space condition. For data reception, each filter has a 3-dB bandwidth that in hertz is about 1.25 times the number of hops per second. Linear-phase filters, called "matched" filters, are needed.

Frequency-Hop Synchronization

One way to synchronize is to use very stable hop-clock oscillators. The hopping code patterns are then all initialized at some time, say each hour, using WWV as a time reference. Then, only a slight amount of searching back and forth in time is needed to align the receiver with the signal.

At the start of each reception time, a small sync adjustment is made. A block of data would be preceded by a special segment that sets up the receiver to copy data. A latecomer in a net would also need opportunities to get fully synchronized.

Sync searching is illustrated in Fig. 13. We see that the incoming signal, at A, does not completely coincide in time with the receiver tuning, at B. The signal switches to frequency f2 before the receiver is ready to switch to f2. The result is that the receiver output has a signal only during the interval shown at Fig. 13C. The receiver uses this information to advance or retard the hop clock slightly until the overlap has improved sufficiently. In the example shown, the receiver hop clock would be speeded up.

In another synchronization method that is somewhat more complicated, the receiver slow hops until the computer recognizes that sync has occurred. Then, the receiver fast hops in sync with the signal. The advantages of this method are that very stable hop-clock oscillators are not needed, and no prealignment of hopping codes is needed.

An additional method is to reserve certain frequencies as "start" or "stop" frequencies. When the start frequency is detected, hopping automatically begins according to the hop pattern plan. A stop frequency advises that the net control is available.

Interference Rejection

In Fig. 14, two kinds of interference are shown. In one case, a steady signal appears on one of the frequencies. In the second case, another hopping signal occasionally occupies the same frequency at the same time. Also shown in Fig. 14 is a common occurrence, signal fading from time to time on various channels.

Ways to combat these conditions are (1) make the receiver bandwidth narrow, to reduce interference; (2) design the hop codes to minimize "collisions" between nets and between net members; (3) use a fast-responding agc; (4) use a lot of message redundancy, i.e., repeat everything on several frequencies; (5) use error-correction codes; and (6) delete occupied frequencies and insert clear frequencies. Possibly certain frequencies would be reserved for backup use only.

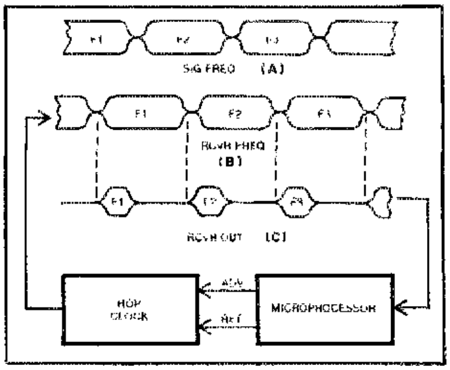

Fig. 13 — Illustrating frequency-hop sync searching or tracking.

Network Protocol

How could frequency hop be used to enhance an amateur packet network? Consider the format in Fig. 15.

Block 1: The net control has a hop code that addresses the net. The stations in the net are using this hop code to monitor the net control station and synchronize to it.

Blocks 2 and 3: The net control also sends data that contains the information in blocks 2 and 3.

Block 4: Having acquired frequency-hop sync, the net members acknowledge.

Block 5: The net control listens for my message using my frequency-hop code. Other stations in the net communicate with each other at the same time, using predetermined hop patterns that do not interfere with other members.

Block 6: The stations use two kinds of code simultaneously: frequency-hop pattern and mark/space code. This makes it possible, for example, to address a particular station and to identify the caller at the same time, or to inform the recipient how to respond to the call. Once the net members become synchronized, it should be possible to maintain sync for a long period of time, with tracking adjustments as required. If a member requires sync, he can send a sync-request message to which the recipient responds.

These ideas are offered for illustrative purposes and do not represent any known system in use. The important thing is that

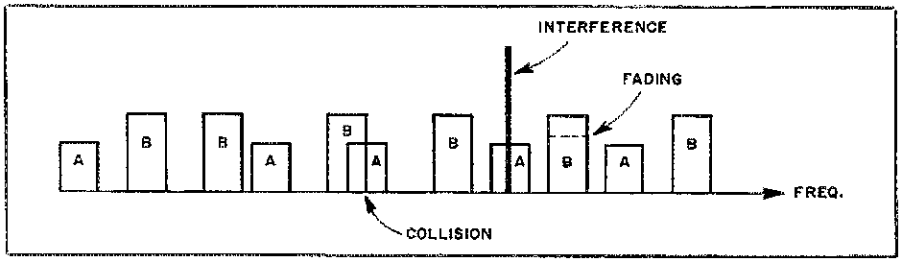

Fig. 14 — This drawing depicts interfering signals and fading in a frequency-hop system. The letters A and B identify emissions from two different transmitters sharing the spectrum, but with different hopping codes. The two frequency sets are almost orthogonal. The coding system is designed to reject interference, fading and collision. The same hop frequencies may be used by both stations, A and B, but at different times.

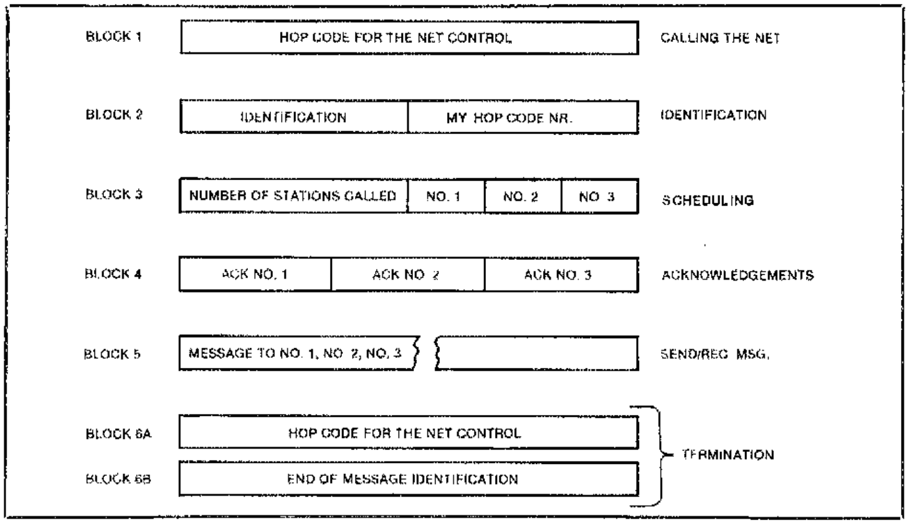

Fig. 15 — Possibilities for network protocol.

amateurs are free to devise schemes that are right for them.

The key to success in frequency hop is to repeat the message often and to provide error correction. This means the message rate must be reduced to improve reliability. In frequency hop, the possibilities of interference are strong. This means that any part of the message must be repeated on several different frequencies. In a slow-hop system, a block of data is sent on each frequency and then the frequency is changed. A typical dwell time on each frequency might be 0.1 second.

It is important to have some way of detecting "bad" frequencies and moving away from them. The network protocol should include an avoidance strategy. The key is to find out if excessive errors occur on certain frequencies. This frequency management is under computer control. My computer tells your computer what action is needed to improve the message reliability.

In a fast-hop system, the dwell time on each frequency might be 1 ms. The message rate is the same as with slow hop, but now each data bit is repeated on several different frequencies. After each bit is received, a vote is taken and the majority decides whether the bit is a 1 or 0. The addition of error correction adds to the reliability. The fast-hop method should be a better scheme for amateur use because, with good design of hopping code patterns, collisions and loss of data should be less.

The Amateur Radio Research and Development Corporation (AMRAD) is a group of amateurs who are dedicated to advanced technology in Amateur Radio. Within this group are subgroups interested in spread spectrum and packet networks. The *AMRAD Newletter* is helpful to anyone wanting to learn more, or to get in touch with others having similar interests. (Another interesting newsletter, *QEX*, an experimenter's exchange, is published by the ARRL.) AMRAD is also in touch with the FCC and has obtained special permits to do various kinds of experimental work. The address is AMRAD, P.O. Drawer 6148, McLean, VA 22106. Terry Fox, WB4JFI, is president of AMRAD. Hal Feinstein, WB3KDU, heads the spread-spectrum subgroup, and Dave Borden, K8MMO, heads the packet-protocol group.

New Books

MICROPROCESSORS AND MICROCOMPUTERS

by Ronald J. Tocci and Lester P. Laskowski. Published by Prentice-Hall, Inc., Englewood Cliffs, NJ 07632. Second edition, 1982. Hardbound, 7-1/4 × 9-1/2 inches, 404 pages including index. $20.95.

This comprehensive book does exactly what the authors intend it to do. It addresses a broad spectrum of readers, providing a practical introduction to the world of microprocessors and microcomputers. Concentrating on the fundamentals of microprocessor-based systems, it leans specifically toward the 6502.

The authors' writing style is about the nearest you can come to having a personal tutor without actually employing one. If you can't find the time (or money) to attend classes on this subject, you can bring the class into your home for a small cash outlay. And should you attend a formal class, you'll still find this book to be a valuable reference.

From number systems and codes used in the world of computers, you're stepped through digital circuits, an introduction to computers and then into microcomputer hardware. After voyaging through the inner workings of the microprocessor itself, you're led to the ports of I and O — input/output interfacing. These are important ports of call, as this interfacing allows you to communicate with the computer, and the computer to respond to you. The many facets of microcomputer software are covered in the last section, and the appendix contains the 6502 instruction set and op codes. Budding assembly-language programmers will find some helpful examples.

Each chapter has a number of questions, plus problems for the reader to solve. This helps you to discover gaps in your understanding of the material. More than likely you're not going to remember *everything* you've read the first time around, but it's a simple matter to turn back a few pages and refresh your memory — sort of a dynamic RAM operation!

Weak points? My only gripe is that there is no answer section. It would be nice to have absolute verification of the answers you've provided for the example questions and problems.

If by now you haven't gathered that I recommend the book, let me state emphatically that I do! I wish all my schoolbook texts had been written this clearly. This is one volume that shouldn't collect dust on your bookshelf. — *Paul K. Pagel, N1FB*