Steven J. Franke, K9AN

3314 E. Anthony Dr., Urbana, IL 61802; **k9an@arrl.net**

Joseph H. Taylor, K1JT

272 Hartley Ave, Princeton, NJ 08540; **k1jt@arrl.net**

The MSK144 Protocol for Meteor-Scatter Communication

Here's a full description of the modulation, message structure, channel coding, and special operational features of the new meteor-scatter mode implemented in WSJT-X.

Meteor-scatter communication was first described[1] in the pages of *QST* in 1953 as a means for communicating on dead 15 and 20 m bands. Hams soon realized that even more impressive results could be obtained at 6 and 2 m, where background noise levels are much lower and useful low-elevation gain is obtainable with relatively modest antennas. Early meteor-scatter (MS) contacts used CW and relied on relatively rare "blue whizzer" meteor trails that last several seconds or longer. Today we can use a fast digital mode with built-in error correction to make contacts on VHF bands any day of the year, out to 1300 miles or so, using meteor-induced "pings" shorter than 0.1 s — with no dependence on weather, solar activity, position of the moon, or fickle band openings.

European hams pioneered the use of high-speed CW (HSCW) in the 1960s and 1970s, using Morse code at speeds 10 to 40 characters per second (cps; 10 cps=120 WPM) to convey short messages using pings as short as several tenths of a second. Modified cassette tape recorders saved the received audio and played it back at low speed, for decoding by ear. By the late 1990s, personal computers were used to send and receive HSCW at speeds as high as 150 cps. Shelby Ennis, W8WN, described the state of the art in HSCW circa 2000 in *QST*.[2] Soon afterward K1JT introduced computer program *WSJT*[3] with the FSK441 protocol, the first amateur digital mode designed specifically to communicate with the shortest and most frequent meteor pings. FSK441 uses 4-tone frequency-shift-keying and noncoherent demodulation. Its character transmission rate is 147 cps, and it provides reliable copy for signals a few dB above the noise in a 2500 Hz bandwidth. Since 2001 hundreds of thousands of MS contacts have been made with FSK441 on the VHF bands, and some even as high as 432 MHz.

Today's computers are considerably more powerful than those of 2001. This rapid technology advance has enabled us to develop MSK144, a practical protocol for meteor scatter that uses bandwidth-efficient modulation and cutting-edge tools for forward error correction (FEC). When designing this protocol we gave high priority to considerations of transmission speed, sensitivity, and decoding efficiency. The final design choices for MSK144 are well matched to the nature of MS signals on the amateur VHF bands and the characteristics of today's amateur transceivers. The effective transmission rate of 250 cps makes good use of very short pings and can be shown to be a practical speed limit for the typical 2500 Hz bandwidth of amateur SSB transceivers. The generated MSK144 waveform ensures that decoders can use coherent demodulation and even coherent averaging over multiple message frames. Using these techniques, we find that some MSK144 signals can be decoded with signal-to-noise ratios as low as −8 dB in the standard 2500 Hz reference bandwidth. Following its public introduction in the summer of 2016, MSK144 has rapidly become the world-wide mode of choice for amateur MS contacts.

In this paper we present technical details of the MSK144 protocol and describe its motivation and underlying design philosophy. We begin by describing the modulation, frame structure, and error-control coding, paying particular attention to the spectrum and envelope shape of generated waveforms. We include on-the-air spectral measurements and the results of simulations that establish the decoding sensitivity and false-decode rate. We then present some details of the MSK144 decoder as implemented in the popular open-source computer program *WSJT-X*,[4] followed by some special operational features of the program in MSK144 mode. These features include semi-automated tools for convenient use of a standard MS calling frequency; a contest mode to facilitate exchange of required information in North American VHF contests; a special short-message format useful at 144 MHz and higher frequencies; and a tool for measuring and compensating frequency-dependent phase shifts in the receiver passband.

Modulation and Coding

MSK stands for *minimum shift keying*, a form of continuous-phase frequency-shift keying (FSK) with shift equal to half the baud rate. MSK144 uses *message frames* of 144 bits and modulation at tone frequencies 1000 and 2000 Hz to transmit *channel symbols* at keying rate of 2000 baud. The resulting audio waveform can be viewed as a form of offset quadrature phase-shift keying (OQPSK) with individual pulses shaped like the first half-period of a sine wave. Using this OQPSK viewpoint, the continuous-time representation of an MSK144 signal can be written as

$$s(t) = I(t)\cos\omega_c t - Q(t)\sin\omega_c t$$

where the waveforms $I(t)$ and $Q(t)$ are called the in-phase and quadrature components of the signal, respectively. In *WSJT-X* the MSK144 waveform is generated with carrier frequency $f_c = \omega_c / 2\pi = 1500$ Hz, and the resulting audio signal is transmitted as upper sideband by a standard single sideband (SSB) transmitter.

Example waveforms for $I(t)$ and $Q(t)$ in Figure 1 show how the in-phase and quadrature signals are created from half-sine pulses with 1 ms duration. Note that the $Q(t)$ waveform is shifted by half a symbol relative to $I(t)$; therefore half-symbols appear at the start and end of the $Q(t)$ waveform in this plot. Pulses with positive polarity represent bits with value 1, negative pulses represent 0. With the convention that bit indices begin at zero, even-numbered bits are sent on the *Q* channel, odd-numbered bits on the *I* channel. The waveform shown in Figure 1 represents the bit sequence 01100110101101010.

Waveform Spectrum and Envelope

Figure 2 shows the average audio spectrum of an MSK144 signal as generated by *WSJT-X*. The carrier frequency $f_c = 1500$ Hz has been chosen to center the main spectral lobe in the available bandwidth of a typical SSB transceiver. The generated audio waveform has constant amplitude; however, band limiting by the transmitter's audio and RF filters will remove all spectral components above 3 kHz, so only the main lobe will be transmitted and the waveform will no longer have constant envelope. Table 1 shows some examples of the amount of envelope variation caused by different amounts of filtering. Figure 3 shows the measured spectrum of a received MSK144 signal at the receiver's audio output, as well as the spectrum of receiver background noise with no signal present.

Linear amplification is necessary to reproduce faithfully the envelope variations introduced when the MSK waveform is filtered. Such an amplifier will maintain the sidelobe-free spectrum produced by the transmitter filter. On the other hand, if the signal is hard-limited and then amplified by a nonlinear amplifier, spectral sidelobes will reappear, but at lower levels than were present before filtering.[5] In general, we expect that the MSK144 waveform will tolerate nonlinear amplification without generating excessive amounts of splatter; however anyone contemplating use of a nonlinear amplifier should conduct tests to verify that the amplifier does not cause excessive spectral broadening.

Frame Structure and Channel Code

As in all other *WSJT-X* modes with FEC, MSK144 user messages are compressed into exactly 72 bits. Transmissions consist of a sequence of identical frames that carry these bits along with synchronizing and error-correcting information. Each frame includes a 72-bit user message, an 8-bit *cyclic redundancy check* (CRC) computed from the message, and 48 bits of error-correcting redundancy. The resulting 72 + 8 + 48 = 128-bit *codeword* is combined with two 8-bit sync words to form a 144-bit message frame. The frames are constructed as {S8, D48, S8, D80}, where S8 represents an 8-bit sync word and D48, D80 represent the first 48 and last 80 bits of the 128-bit codeword. The 8-bit CRC is used by the decoder to detect and eliminate most false decodes. At 2000 baud, the frame duration is 144/2000 = 0.072 s.

The 80-bit combination of message and CRC is mapped to a 128-bit codeword using a binary (128, 80) *Low Density Parity Check* (LDPC) code designed specifically for MSK144. We chose an LDPC code because these codes provide state-of-the-art performance and can be designed with virtually any desired number of data bits and parity bits. Moreover, they can be decoded with low computational cost compared to other code types used in *WSJT-X* — specifically, the long constraint-length convolutional codes used in JT4, JT9, and WSPR and the Reed-Solomon code used in JT65. As a consequence

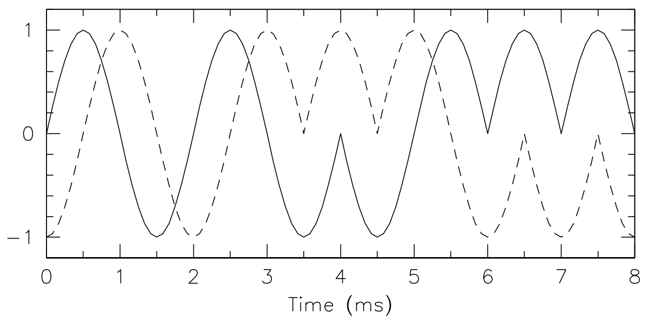

Figure 1 — Short segment of the in-phase, *I*(*t*) (*solid curve*) and quadrature, *Q*(*t*) (*dashed curve*) components of an MSK144 waveform. The in-phase waveform contains 8 contiguous half-sine pulses. The quadrature waveform is offset by half of a pulse. The offset and half-sine shaping ensures that the envelope of the MSK144 signal is constant, before band limiting by filters in the transmitter.

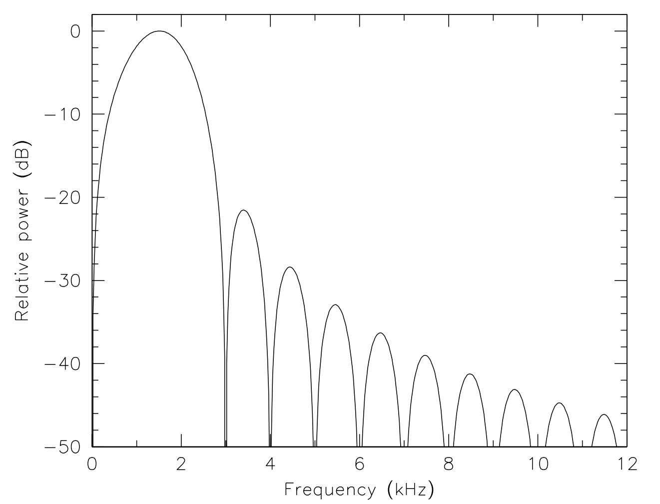

Figure 2 — Average spectrum of MSK144 audio signals generated by *WSJT-X* using 1.5 kHz audio carrier frequency. Transmitter audio and RF filters will limit the bandwidth of the transmitted signal to, typically, 300-2700 Hz, which includes most of the main lobe of this spectrum but no sidelobes.

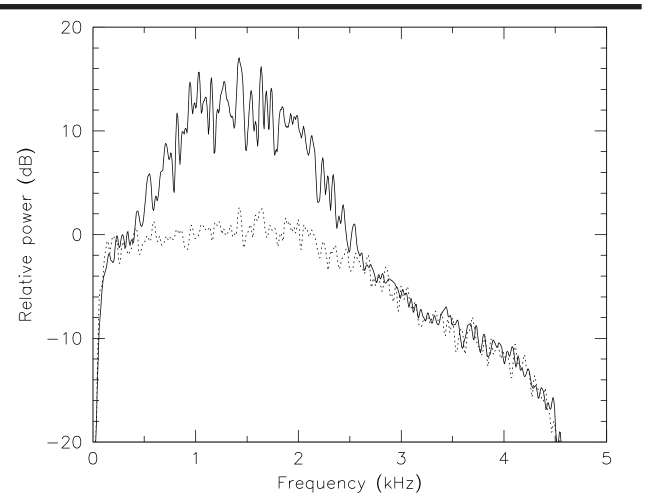

Figure 3 — *Solid curve:* received spectrum of the message "K9AN AA2UK R-05" transmitted by AA2UK and received by K9AN. *Dotted curve:* spectrum of background noise as shaped by the frequency response of K9AN's receiver.

Table 1

Representative examples of envelope variation obtained by filtering a simulated MSK144 frame. Envelope variation increases significantly as transmitter bandwidth narrows.

	Average power	*Peak envelope power*	*Peak-to-average power ratio*	*Max-to-Min power ratio*
No filtering	1.0	1.0	1.0	1.00
0-3 kHz	0.996	1.27	1.28	1.91
0.3-2.7 kHz	0.988	1.32	1.33	2.29
0.5-2.5 kHz	0.959	1.50	1.57	3.31

we can use many decoding attempts on each chunk of data, with each attempt concentrating on a different subset or weighting of the data. This procedure leads to a significantly higher probability for decoding weak or noisy signals.

The selected (128, 80) LDPC code is defined by a 48 × 128 *parity-check matrix* in which all elements are 0 or 1. The matrix has exactly three ones in each column and eight in each row. (Thus, only 3 × 128 = 8 × 48 = 384 of the 6144 entries are equal to 1; it's this low density of nonzero elements in the matrix that gives the LDPC code its name.) Each of the 48 rows of the parity check matrix defines a parity check equation. The eight 1s in each row determine which of the 128 codeword bits are included in that parity check. Each parity check is carried out by summing, modulo 2, the indicated codeword bits. All 48 of these sums will be zero for a valid, error-free codeword.

Starting with the parity-check matrix we derive a 48 × 80 *generator matrix* that will determine 48 parity bits for any 80-bit message word. The 1s in a particular row of the generator matrix determine which of the 80 message bits to sum, modulo 2, to produce a parity bit. Additional details of the way we have implemented these processes, including concise definitions of the two matrices, can be found in the open source code[6] for *WSJT-X*.

Code Performance

Figure 4 summarizes performance of the MSK144 protocol when used with the soft-decision decoder currently implemented in *WSJT-X*. As shown by the curve labeled P_c, single received frames with $SNR > 0$ dB are nearly always decoded correctly. (Here and elsewhere in this paper, signal-to-noise ratios are measured in a 2500 Hz standard reference bandwidth.) At $SNR = -1.5$ dB about 40% of received frames are correctly decoded, and about 0.1% of received frames will yield an incorrect codeword. Most of the incorrect codewords will be rejected because the falsely decoded message will not pass the CRC test. For *SNR*s near the 50% decoding threshold about 1 in 200 such incorrect codewords — about 5 per one-million decoding attempts — will accidentally have the correct CRC and will be displayed as *undetected false decodes*. The measured probability of undetected false decodes is plotted as a function of *SNR* as P_i, scaled up by a factor of 10^5 to make it visible on the plot. The false decode rate falls to negligible levels when $SNR > 0$ dB.

Figure 5 shows the fraction of noisy received codewords that will be decoded as a function of the number of hard errors in the codeword. Redundancy provided by the parity bits allows decoding of most synchronized frame-length bit sequences with fewer than 10 hard errors, and a small fraction of those with as many as 15 errors. We note that without the error-correcting code, even one hard error in the 72-bit packed message would reduce its uncompressed content to garbage.

A number of significant design choices have been built into the code/decoder combination. One such parameter is the number of iterations the decoder is allowed to try before it gives up. More iterations take more time, but (up to a point) will produce more decodes. However, the probability of false decodes increases dramatically if the maximum allowed number of iterations becomes too large. Another choice involves the detailed design of the code

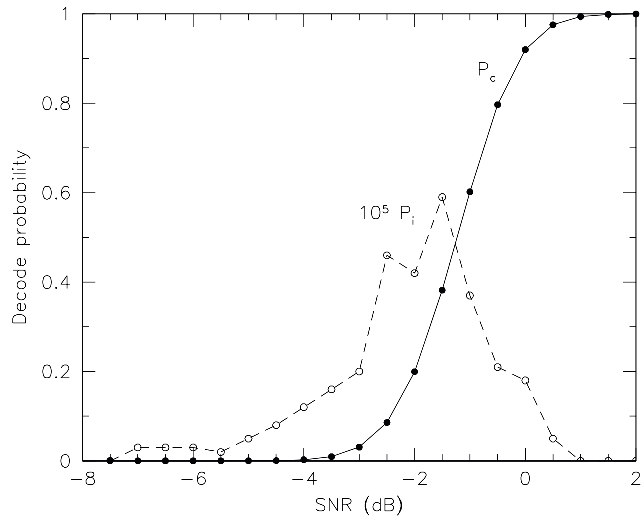

Figure 4 — Probability of a correct decode P_c and incorrect decode P_i as a function of *SNR* for standard MSK144 messages. These curves were generated by numerical simulation, based on 10^7 simulated noisy received frames. The results represent ideal performance assuming perfect frequency and time synchronization. Coherent averaging of *N* messages will slide both curves to the left by the amount $10 \log_{10} N$.

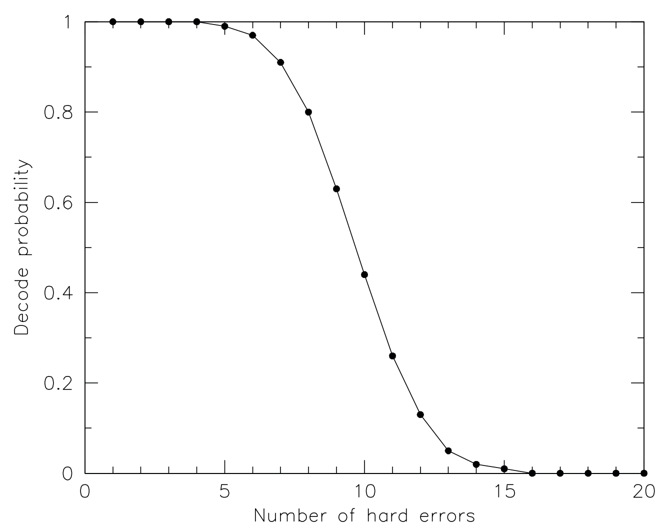

Figure 5 — Fraction of correct decodes as a function of the number of hard errors in a received codeword.

itself. Increasing the density of the parity-check matrix yields a code that requires more iterations to decode or, equivalently, produces fewer decodes for a given number of iterations. On the other hand, a more dense code that produces half as many decodes may give one fifth as many false decodes. We created a number of codes with different densities and explored the performance of each one for different iteration limits. Ultimately, we chose parameters that yield good P_c and P_i performance with the smallest number of decoder iterations. We believe the chosen code is close to optimum for the purpose at hand.

Optional Short-Message Format

On a given meteor-scatter path the duration of a ping is proportional to the inverse square of operating frequency. Pings at 144 MHz are therefore about 1/8 as long as those at 50 MHz. Most of a frame must be received in order to decode its message, so pings shorter than about 70 ms, common at 144 MHz and higher bands, are too short to convey a decodable standard MSK144 frame. To utilize even shorter pings, the protocol includes optional short messages that can be used to send signal reports and other necessary QSO information after call signs have been exchanged. Short frames are 20 ms long and consist of 40 bits: an 8-bit sync word, 4 bits to convey message information, 12 bits representing a hash of the string consisting of the DX call sign followed by the home call sign, and 16 parity bits for FEC. There are just 9 supported messages, namely R-03, R+00, R+03, R+06, R+10, R+13, R+16, RRR, and 73. The 12-bit hash serves two purposes: it gives the receiving operator high confidence that a decoded frame was indeed intended for him, and it is used to reject most false decodes.

We designed a binary (32,16) LDPC code for the MSK144 short-message frames. The 16×32 parity-check matrix has exactly three 1s in each column and 5, 6, or 7 in each row. Performance measurements for this code (including verification of the hash test) are plotted in Figure 6. The decoding threshold is almost the same as for the long code, but the peak probability of false decodes is about 30 times larger — a consequence of the short code's smaller block length.

Implementation in *WSJT-X*

MSK144 is one of seven distinct operating modes in program *WSJT-X*. An overview of the design purposes and operating characteristics of each mode has been published in the *QST* article of Note 4, and many more details can be found in the *WSJT-X User Guide*.[7] We limit the discussion here to some particular features of MSK144 as presently implemented in *WSJT-X*.

Decoder Details

With pings longer than about 100 ms we can coherently sum multiple frames to improve the signal-to-noise ratio. The current MSK144 decoder can average up to 7 frames, about half a second of data. Coherent averaging of N frames improves sensitivity by 10 log N dB, so sensitivity improves rapidly for longer pings. Coherent averages of N = 2, 3, 4, 5, and 7 frames with constant signal level yield gains of 3, 4.8, 6, 7, and 8.5 dB, respectively. These amount to very worthwhile improvements, even when the real-world signal levels are not constant over the averaging interval.

Coherent demodulation requires proper alignment of channel-symbol boundaries, precise knowledge of a received signal's carrier frequency, and an estimate of carrier phase. Averaging over multiple frames increases the necessary frequency precision in proportion to total signal duration. To ensure that carrier phase does not vary significantly over the averaging interval, 7-frame averaging requires that frequency be known to better than 1 Hz. In effect, the decoding algorithm must search all possible frequency offsets in some tolerance window using a 1 Hz step size, at the same time searching over time offsets to establish symbol synchronization. For the longest frame averages, the decoder's execution time is dominated by this synchronization requirement.

WSJT-X implements a version of the *sum-product algorithm* [8] for soft-decision decoding of the noisy received symbols. This iterative decoding algorithm accepts a numerical reliability measure for each of the received symbols. Each iteration updates the symbol reliability based on the degree to which the parity check sums are satisfied. After every iteration, hard decisions are made about the value of each symbol based on the updated symbol reliabilities; if the result is a valid codeword, the algorithm terminates. If no codeword is found before completing a predetermined maximum number of iterations, the algorithm "times out" and reports a decoding failure.

The MSK144 decoder in *WSJT-X* Version 1.7 operates in near real-time by looking at small overlapping chunks of data and completing all decoding attempts before the next chunk arrives. Each chunk contains 7 message frames, equivalent to about 0.5 s of data. For each chunk the decoder first tries to synchronize and decode the best single message frames. If this fails, it tries combinations of 2-, 3-, 4-, and 5-frame coherent averages. Finally, it tries a coherent average of all 7 frames. Older and slower computers may not be able to keep up with the demands of the MSK144 decoder when using the "Deep" decoding setting on the *WSJT-X* user interface. Selecting "Normal" decoding will eliminate the longest and most time-consuming 7-frame average to save time. The "Fast" decoding setting eliminates both 5- and 7-frame averages. Omitting the longest averages reduces sensitivity somewhat, although the penalties are modest.

Phase Equalization

Most superheterodyne SSB transceivers use narrow filters optimized for good shape

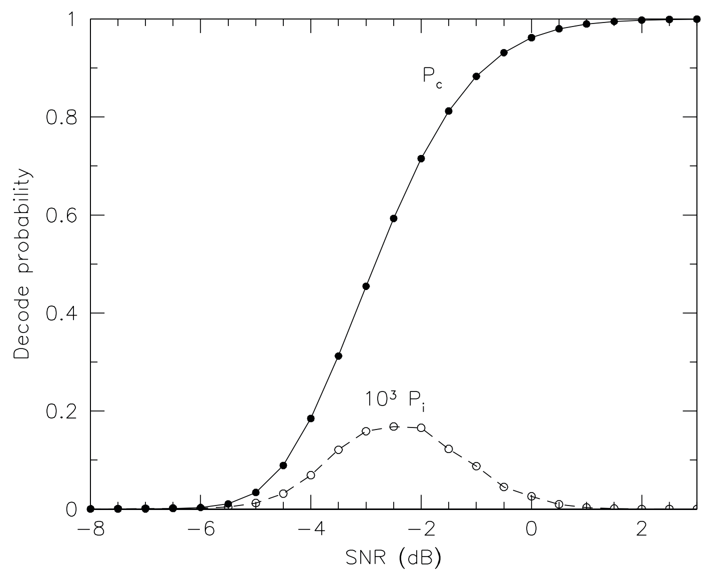

Figure 6 — Probability of a correct decode P_c and incorrect decode P_i as a function of *SNR* for the MSK144 short-message format. These curves were generated via numerical simulation, based on 10^7 simulated noisy received frames.

factor without regard to phase linearity. Group delay variation across the passband smears out MSK144 pulses, causing intersymbol interference. *WSJT-X* includes a phase equalization facility that can be used to correct any group delay variation contributed by the receiver. When a received frame has been successfully decoded, this tool generates an undistorted waveform whose Fourier transform can serve as a frequency-dependent phase reference to compare with the phase of the received frame's Fourier coefficients. Phase differences between the reference waveform and the received one will include distortions contributed by the originating station's transmit filter, the propagation channel, and filters in the receiver. If the received frame originates from a station known to transmit signals having little phase distortion (say, a station known to use a properly adjusted software-defined-transceiver) and if the received signal is relatively free from multipath distortion so that the channel phase is close to linear, the measured phase differences will be dominated by the local receiver's phase response.

The phase-response tool in *WSJT-X* fits a low-order polynomial to the measured phases. The saved polynomial coefficients can then be used to correct the phases in all received signals, effectively flattening the receiver's group-delay response. Careful use of this capability can improve the decoder sensitivity by significant amounts. As an example, Figure 7 shows the phase response of the TS-2000 receiver at K9AN, measured using a signal transmitted by the software-defined-transceiver at KØTPP. Phase errors as large as $\pi/2$ radians (90 degrees) are found near the passband edges, relative to mid-band values. The smooth curve in Figure 7 is a 4th-degree polynomial fit to the measured data. The improved decoder sensitivity can be judged from the *eye diagrams* of Figure 8, which contain overlaid plots of 72 received-symbol amplitudes constituting one frame of the signal from KØTPP. The upper part of the figure shows the diagram before phase correction; the lower part after applying the fitted phase equalization curve. Notice that the eyes are more "open" after equalization. Wider eye opening means fewer bit errors at low *SNR*, and a higher likelihood of successfully decoding a noisy received frame.

If the operator is careful to ensure that the applied phase correction accurately represents the receiver's response, then the phase equalization derived from one strong reference station will improve decoding of signals from most other stations. It should be noted that phase equalization is not likely to improve decoding performance for those who use SDRs with linear-phase receive filters.

Standard Operating Procedures

In North America the highest MSK144 activity levels are found on 6 meters. QSOs are carried out with alternating transmit/receive (T/R) sequences 15 s long; a hundred watts and a modest antenna up 20 feet is sufficient for making lots of meteor-scatter contacts. By informal convention the standard 6 m "calling frequency" is 50.280 MHz. As described in our earlier mentioned *QST* paper of Note 4, most QSOs start with someone calling CQ and proceed roughly as follows:

```
1. CQ K1JT FN20
2.     K1JT K9AN EN50
3. K9AN K1JT -01
4.     K1JT K9AN R+03
5. K9AN K1JT RRR
6.     K1JT K9AN 73
```

Each station continues sending a given message until receiving the next message from the QSO partner. The signal reports conveyed in messages number 3 and 4 are measured signal-to-noise ratios in dB. By longstanding tradition, especially for weak-signal work on VHF and higher bands, a minimal contact is considered complete and suitable for logging after both call signs, signal reports or some other previously unknown information, and acknowledgments have been exchanged. The final "73" message is a courtesy; in the above example it lets K1JT know that his final acknowledgment was received and the contact is complete. Many users like to exchange 13-character free-text "chit-chat" at this point.

As an example of on-the-air usage of MSK144, the *WSJT-X* screen shot in Figure 9 shows a sequence of messages received at K1JT after he called CQ on 50.280 MHz. In a transmission starting at 11:30:00 UTC,

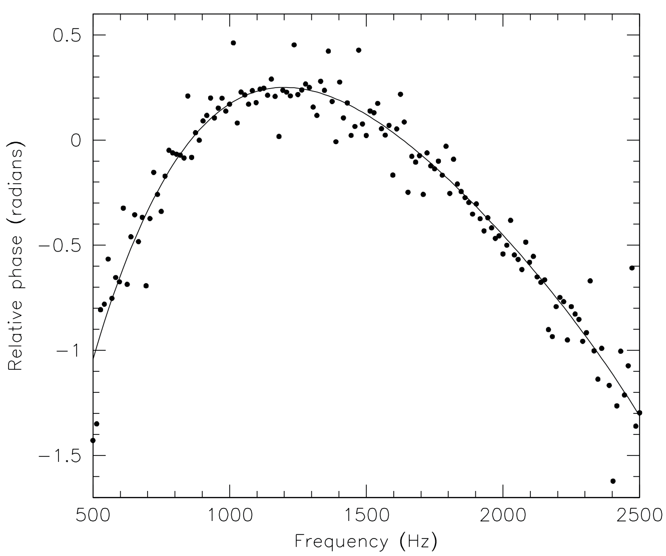

Figure 7 — *Filled circles:* measured phase differences between the complex spectra of a locally generated reference waveform and the average of several strong frames received from KØTPP. *Smooth curve:* 4th-order polynomial fit to the measured data.

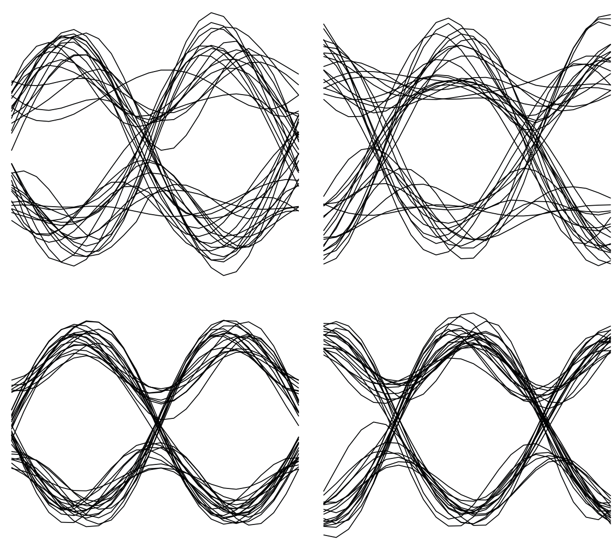

Figure 8 — Eye diagrams for the received in-phase (*left*) and quadrature (*right*) symbols for a signal from KØTPP received by K9AN. Upper curves are with no phase equalization; lower curves are after equalization using the polynomial fit shown in Figure 7.

W5ADD answered the CQ. Figure 10 is a snapshot of the *WSJT-X* horizontally scrolling spectrogram showing this ping (lower panel) and another one several sequences later that carried the message "K1JT W5ADD R-02" (upper panel). These two pings were of moderate strength, *SNR* = 5 and 4 dB, respectively, and the one in the top panel is about 70 ms wide; nevertheless these and even much weaker pings (see the "dB" column in Figure 9) are decoded without error. From Figure 9 you can see that when the QSO with W5ADD was complete N9BX and KA9CFD called as "tail-enders," and two more QSOs followed in quick succession.

UTC	dB	T	Freq		Message
113000	5	8.8	1565	&	K1JT W5ADD EM40
113230	4	11.4	1567	&	K1JT W5ADD R-02
113330	-1	1.6	1571	&	K1JT W5ADD 73
113530	5	6.3	1563	&	K1JT N9BX EM50
113600	-4	9.8	1570	&	K1JT KA9CFD EN40
113630	3	1.7	1560	&	K1JT N9BX EM50
113730	9	14.3	1558	&	K1JT N9BX R+02
113830	3	8.1	1559	&	K1JT N9BX 73
114030	2	14.0	1562	&	K1JT KA9CFD R+02
114530	14	13.4	1564	&	K1JT KA9CFD 73

Figure 9 — Messages received at K1JT in a sequence of three quick MSK144 QSOs on 6 meters.

Split Operation using "CQ nnn"

The sporadic nature of meteor pings makes it possible for many stations to share a frequency with little interference. Decoding several different stations in the same receiving sequence lets you see "who else is on," in addition to the station you may be working. However, when the frequency gets *too* busy it's a good idea to spread out. *WSJT-X* provides a mechanism for doing this while preserving the important advantage of having a known "meeting place" for initiating contacts. CQ messages may include three digits between the CQ and call sign. For example, K9AN might send "CQ 290 K9AN EN50" on 50.280 MHz to indicate that he will listen for replies on 50.290 MHz, and continue the QSO there. Note that offset CQs are not the same as typical "split" operation on HF bands. When operating split, each operator transmits on one frequency and listens on another. MSK144 QSOs using "CQ nnn" are used to move a contact off the calling frequency to a single offset frequency where both stations will transmit and receive. *WSJT-X* has facilities that allow it to recognize "CQ nnn" calls and reset the transceiver's dial frequency automatically, as required for both stations.

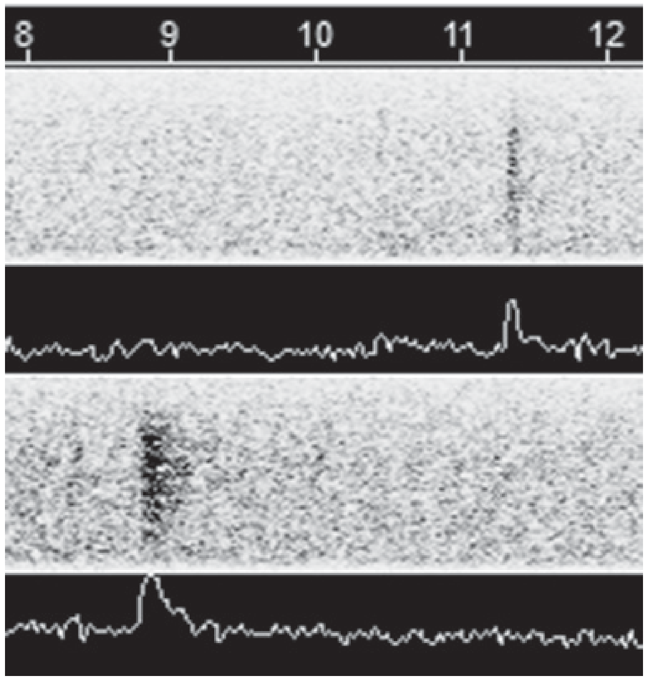

Figure 10 — Small portions of the *WSJT-X* real-time spectrogram showing pings received from W5ADD in sequences starting at UTC 11:30:00 (*lower*) and 11:32:30 (*upper*). The corresponding decoded messages are those in the first two lines of the screen shot in Figure 9.

Contest Mode

North American VHF contests use Maidenhead grid locators as multipliers and required exchange information. MSK144 offers an optional *contest mode* in which grids are exchanged and acknowledged instead of signal reports. The standard message sequence then becomes

1. *CQ K1JT FN20*
2. *K1JT K9AN EN50*
3. *K9AN K1JT R FN20*
4. *K1JT K9AN RRR*
5. *CQ K1JT FN20*

The acknowledgment "R" in message number 3 is conveyed by using the fact that propagation modes suitable for MSK144 are generally effective only out to distances of order 1300 miles. To convey the message

UTC	dB	T	Freq		Message
120300	1	9.7	1497	&	CQ K9AN EN50
120300	2	10.5	1499	&	CQ K9AN EN50
120400	-1	11.9	1497	&	K1JT K9AN +03
120430	3	0.6	1496	&	<K1JT K9AN> RRR
120500	-2	9.4	1495	&	<K1JT K9AN> 73

Figure 11 — Messages received at K1JT in a QSO with K9AN using short-format messages.

fragment "R FN20", *WSJT-X* encodes the locator as that of the diametrically opposite spot on Earth. The receiving program recognizes such a locator as special — it would imply an implausible distance approaching half the circumference of the Earth — so the software does the reverse transformation to the original grid and inserts the implied "R " in the message displayed to the user.

Short Messages

As described above, MSK144 supports short-form messages that can be used after QSO partners have exchanged both call signs. Short messages consist of 4 bits encoding R+report, RRR, or 73, together with a 12-bit hash code based on the ordered pair of "to" and "from" call signs. When short messages have been activated on the *WSJT-X* user interface, messages 4 to 6 in the standard sequence are modified as follows:

```
1. CQ K1JT FN20
2.      K1JT K9AN EN50
3. K9AN K1JT -01
4.      <K1JT K9AN> R+03
5. <K9AN K1JT> RRR
6.      <K1JT K9AN> 73
```

Individual call signs are replaced by a 12-bit hash code in the transmitted frame, and the substitution is indicated onscreen by enclosing the call signs in < > angle brackets. When the receiving program decodes the already known (and therefore expected) hash code, it displays the known call signs in angle brackets. Figure 11 is a screen shot showing messages received by K1JT in a QSO with K9AN using short messages. The short-message feature is intended for use on 144 MHz and higher bands, where very short pings make them especially beneficial.

A third party monitoring short message transmissions will not necessarily know the call signs of the stations involved in the ongoing QSO. *WSJT-X* implements an "SWL" mode that lets a listener monitor short messages exchanged by two other stations. When SWL mode is turned on, the decoder remembers recently copied call signs and compares received hashes with a list of hashes calculated from all ordered pairs of call signs currently in the list. If the received hash is found in the list, the decoder prints the decode with the "guessed" call signs in < > angle brackets. Accepting received messages with any hash found on a list increases the probability of printing a false decode in proportion to square of the number of remembered call signs. To decrease the number of spurious decodes printed in SWL mode, the software only prints decodes after the associated hash has been received at least twice in a receive sequence.

Conclusion

MSK144 is a highly efficient protocol for conducting minimal QSOs via meteor scatter. Indeed, we believe the protocol's effective character transmission rate, occupied bandwidth, and sensitivity are close to optimum for the stated purpose, while remaining consistent with the capabilities of standard amateur SSB transceivers.

We thank Bill Somerville, G4WJS, for helpful comments on an earlier version of this manuscript.

Steve Franke, K9AN, holds an Amateur Extra class license. He was first licensed in 1971 and has previously held call signs WN9IIQ and WB9IIQ. An early and abiding fascination with radio science led to his current position as Professor of Electrical and Computer Engineering at the University of Illinois in Urbana-Champaign. Steve is a member of ARRL and a Fellow of the IEEE.

Joe Taylor was first licensed as KN2ITP in 1954, and has since held call signs K2ITP, WA1LXQ, W1HFV, VK2BJX and K1JT. He was Professor of Astronomy at the University of Massachusetts from 1969 to 1981 and since then Professor of Physics at Princeton University, serving there also as Dean of the Faculty for six years. He was awarded the Nobel Prize in Physics in 1993 for discovery of the first orbiting pulsar, leading to observations that established the existence of gravitational waves. After retirement he has been busy developing and enhancing digital protocols for weak-signal communication by Amateur Radio, including JT65 and WSPR. He chases DX from 160 meters through the microwave bands.

Notes

[1]O. G. Villard, W6QYT, and A. M. Peterson, W6POH, "Meteor Scatter — A Newly Discovered Means for Extended-Range Communication in the 15- and 20-Meter Bands," *QST*, Apr. 1953, pp 11-15, 124, 126.

[2]Shelby Ennis, W8WN, "Utilizing the Constant Bombardment of Cosmic Debris for Routine Communication," *QST*, Nov. 2000, pp 28-32.

[3]Joe Taylor, K1JT, "*WSJT*: New Software for VHF Meteor-Scatter Communication," *QST*, Dec. 2001, pp 36-41.

[4]Joe Taylor, K1JT, Steve Franke, K9AN, and Bill Somerville, G4WJS, "Work the World with *WSJT-X*," *QST*, Oct., and Nov. 2017.

[5]T. Aulin and C. Sundberg, *International J. Satellite Communications*, vol 2, p 219, 1984.

[6]**sourceforge.net/projects/wsjt**.

[7]**www.physics.princeton.edu/pulsar/K1JT/wsjtx-doc/wsjtx-main-1.7.1-devel.html**.

[8]Shu Lin and Daniel J. Costello, *Error Control Coding*, Pearson Prentice Hall, 2004.