The JT65 Communications Protocol

Joe Taylor, K1JT

Abstract. JT65 is a digital protocol intended for Amateur Radio communication with extremely weak signals. It was designed to optimize Earth-Moon-Earth (EME) contacts on the VHF bands, and conforms efficiently to the established standards and procedures for such QSOs. JT65 includes error-correcting features that make it very robust, even with signals much too weak to be heard. This paper summarizes the technical specifications of JT65 and presents background information on its motivation and design philosophy. In addition, it presents some details of the implementation of JT65 within a computer program called WSJT, together with measurements of the resulting sensitivity and error rates.

1. Introduction

Spark gave way to continuous wave some eighty years ago. More or less by default, international Morse code with on-off keying has been the mode of choice for most amateur radio weak-signal work ever since. Morse is convenient, versatile, and readily encoded and decoded by humans. On-off keying is trivial to implement, and the required bandwidth is small. The choice has been an easy one.

It is easy to show, however, that neither the encoding nor the modulation of CW is optimum. When every dB of signal-to-noise ratio counts, as it does in amateur meteor-scatter and EME contacts, there are very good reasons to explore other options. Personal computers equipped with sound cards provide a golden opportunity for experimenting with the wide range of possibilities. The program WSJT[1,2,3] ("Weak Signal communications, by K1JT") is the result of my effort to introduce much more efficient coding and modulation schemes into amateur weak-signal communications. In the program's brief existence it has already become well known to nearly all weak-signal VHF/UHF operators, and is in regular use by many of them. On the VHF bands the overwhelming majority of all meteor-scatter QSOs and perhaps half of all EME QSOs are now being made with the help of WSJT.

The present paper describes JT65, one of the communications protocols supported by WSJT. JT65 is designed explicitly for communicating with extremely weak signals like those encountered on the EME path. Operational aspects of the program are described in the *WSJT User's Guide*[4]*;* here I will be concerned with a complete technical description of the protocol and a general description of the way it is implemented in WSJT.

Modern digital communication systems are based on the mathematics of information theory. This field essentially originated with two classic 1948 papers[5] in which Claude Shannon

---

[1] See the WSJT Home Page at http://pulsar.princeton.edu/~joe/K1JT.

[2] J. Taylor, K1JT, "WSJT: New Software for VHF Meteor-Scatter Communication," *QST* December 2001, pp. 36-41.

[3] J. Taylor, K1JT, "JT44: New Digital Mode for Weak Signals," *QST* June 2002, pp. 81-82

[4] The *WSJT 4.7 User's Guide* is available at http://pulsar.princeton.edu/~joe/K1JT/WSJT_User_470.pdf.

[5] Shannon, C. E., "A Mathematical Theory of Communicaton," *Bell System Tech. J.*,**27**, pp. 379–423 and 623–656, 1948.

proved that information can be conveyed over a noisy channel with arbitrarily low error rate and a throughput that depends only on channel bandwidth and signal-to-noise ratio (SNR). Achieving a low error rate at very low SNR requires the mathematical encoding of user information into a form that is compact yet includes carefully structured redundancy. Compactness is necessary in order to minimize transmitter power and maximize throughput; redundancy is needed to ensure message integrity on a noisy and variable channel.

To be transmitted by radio, an encoded message must be impressed onto a carrier wave using some form of modulation. The possibilities are almost limitless: information can be conveyed by varying the amplitude, frequency, or phase of a carrier, or any combination thereof. Commonly used digital modulation schemes include on-off keying (a limiting case of amplitude modulation), phase-shift keying, and frequency shift keying. The JT65 protocol uses 65-tone frequency shift keying with constant-amplitude waveforms and no phase discontinuities. This form of modulation is much more efficient than on-off keying, especially when combined with an optimal coding scheme. In addition, it is much more tolerant of frequency instabilities than phase-shift keying.

Section §2 of the paper begins with some background information that has helped to motivate the design philosophy of JT65, and Section §3 presents a high-level view of the overall system design. The protocol itself is defined §4–8 and in Appendix A, while Sections §9–12 describe the reception and decoding of a JT65 signal. The protocol specification completely defines the translation of any valid JT65 message into a waveform for transmission, and provides all information necessary for decoding a received JT65 signal. I include the essential details of how these tasks are actually carried out in WSJT. Different implementations of JT65, and especially the algorithms used for reception, are also feasible. I hope that this paper will motivate others to attempt this task, and that such efforts will lead to further improvements in the performance and operational convenience of this mode.

2. EME QSOs: Requirements and Procedures

Amateur Radio is a just-for-fun activity, and for many the fun has always included such goals as making contacts with all continents, all US states, and as many DXCC entities as possible. These goals are especially difficult on the EME path — and therefore, for many, all the more challenging and desirable. To make the game one that anybody can understand and play, it is necessary to agree on some basic ground rules.

When signals are reasonably strong and communication between skilled operators essentially error free, it is easy to judge whether a QSO has taken place. When a rare one shows up on the amateur HF bands, rapid-fire QSOs in the ensuing pile-up generally proceed something like the following exchange:

1.	CQ HC8N	
2.		K1JT
3.	K1JT 599	
4.		599 TU
5.	73 HC8N	

In this model contact K1JT never sends the callsign of the station he is working, because the situation has made this information implicit and moot. The signals may not be “S9” at either receiver, but no one really cares. After the exchange has taken place, both stations

confidently enter the QSO in their logs, and they may later exchange QSL cards to confirm that the contact took place.

In the VHF/UHF world, and especially when working over the EME path, signals are often very weak and communication between even the most skilled operators is far from error free. As a result, more rigorous standards need to be adopted for what constitutes a minimum legitimate QSO. Long-established rules hold that a valid contact requires each station to copy both complete callsigns, a signal report or some other piece of information, and explicit acknowledgment that all of this information has been received. These guidelines apply and work well for all types of weak-signal QSOs, whether by tropo, meteor scatter, EME, or other propagation modes, and with all types of equipment and signaling methods.

Following these guidelines closely, the minimal EME QSOs of savvy VHF operators generally proceed something like the following sequence:

1.	CQ SV1BTR ...	
2.		SV1BTR K1JT ...
3.	K1JT SV1BTR OOO ...	
4.		RO ...
5.	RRR ...	
6.		73 ...

For a scheduled QSO at prearranged time and frequency, transmission #1 is of course unnecessary. The ellipses (...) indicate repetition of messages, some form of which is nearly always used in EME contacts to help maximize chances of success. The "OOO" message component is a shorthand notation for a minimal signal report. It has an agreed-upon meaning that says, in effect, "your signals are readable at least some of the time, and I have copied both of our callsigns." Similarly, "RO" is a shorthand message conveying both signal report and acknowledgment. It means "I have copied both calls and my signal report, and your report is O". When K1JT receives the acknowledgment "RRR" sent by SV1BTR, the QSO is complete; but since SV1BTR does not yet know this, it is conventional to send "73" or some other end-of-contact information to signify "we are done."

Shorthand radio messages have been widely employed since the days of spark and land-line telegraphy; the familiar Q-signals are another universally understood type. They are simple forms of what in communication theory is called the "source encoding" of messages. The choice of "OOO…" (repeated sequences of three carrier-on intervals separated by short spaces, with a longer space after every third one) as the signal representing a positive signal report was made by wise and experienced CW operators who knew that with extremely weak signals, "dahs" are easier to copy than "dits".

3. System Design

Figure 1 presents the flow diagram of a modern digital communication system. For maximum efficiency at low signal-to-noise ratio, a user message is source encoded into a compact form having minimum redundancy. It is then augmented with mathematically defined redundancies which can enable full recovery of the message even if some parts are subsequently corrupted by noise or signal dropouts. This process is known as "forward error correction," or FEC. The encoded message, including its error-correcting information, is modulated onto a carrier. The resulting radio signal propagates over a channel that attenuates it, perhaps by 250 dB or more for an EME path, and adds noise as well as amplitude,

frequency, and phase-changing "path modulation." Upon reception the signal is demodulated and decoded, and the results presented to the user.

Except for the error-correcting enhancements, the flow diagram of Figure 1 describes traditional amateur CW communications just as well as modern digital techniques. In terms of the CW EME QSO outlined on the previous page, source encoding compresses the implied message "SV1BTR, this is K1JT, I have copied both of our calls" into the compact form "SV1BTR K1JT OOO". To provide some error-recovery capability and increase chances that the message will be copied, a CW operator repeats the compressed message many times during a timed transmission. To enhance the chances of copy even further, he may format the repetitions so as to transmit only calls for the first 75% of a transmission, followed by sending "OOO" repeatedly for the last 25%. He expects the receiving operator to know about these conventions, and to listen accordingly. All of these forms of source encoding help: the more that's known about the characteristics of a weak signal, the easier it is to copy. Under extremely marginal conditions, skilled operators listen for matches between what they hear and the types of message components they might reasonably expect. If a good match is found, message copy can be considered secure.

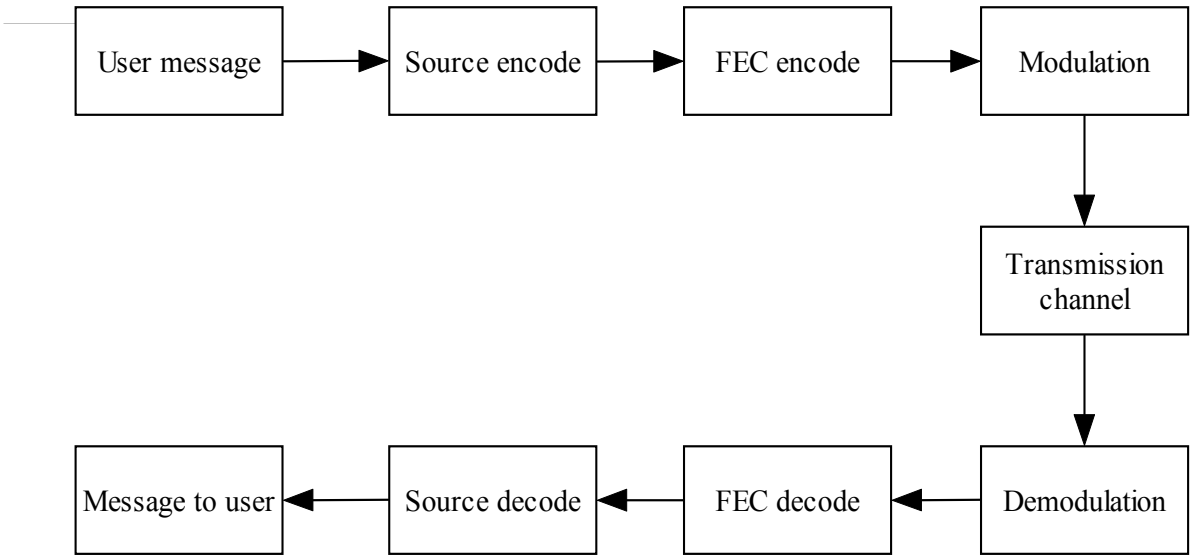

Fig. 1. – Schematic diagram of information flow in a digital communication system.

4. JT65 Source Encoding

JT65 uses exactly analogous techniques, starting out by making its transmitted messages compact and efficient. As described in the *WSJT 4.7 User's Guide*[4], the standard "Type 1" messages of JT65 consist of two callsigns, a grid locator, and an optional signal report — an enhanced form of messages 2 and 3 in the model QSO between SV1BTR and K1JT. The source encoder knows the rules by which standard amateur radio callsigns are constructed, and uses this information to minimize the required number of information bits. An amateur callsign consists of a one- or two-character prefix, at least one of which must be a letter, followed by a digit and a suffix of one to three letters. Within these rules, the number of possible callsigns is equal to 37×36×10×27×27×27, or somewhat over 262 million. (The

numbers 27 and 37 arise because in the first and last three positions a character may be absent, or a letter, or perhaps a digit.) Since 2^{28} is more than 268 million, 28 bits are enough to encode any standard callsign uniquely. Similarly, the number of 4-digit Maidenhead grid locators on earth is 180×180 = 32,400, which is less than 2^{15} = 32,768; so a grid locator requires 15 bits in a message. These important ideas for the efficient source encoding of EME messages were first suggested by Clark and Karn[6] in 1996.

Any Type 1 message can be source-encoded into 28+28+15= 71 bits, plus one more for the signal report. In comparison, sending the message "SV1BTR K1JT OOO" in Morse code requires 170 bits (where a bit is defined as the key-down dot interval), even without the grid locator. The JT65 message is much more compact than the CW message, while conveying significantly more information. In practice, the JT65 protocol encodes signal reports in another way and instead uses the 72^{nd} bit to indicate that the message contains arbitrary text instead of callsigns and a grid locator. With a 43-character alphabet, the maximum plain-text message length is 13 (the largest integer less than 71 log 2/log 43). Subject to this limiting size, JT65 can transmit and receive *anything* in a message.

As indicated above, some 6 million of the possible 28-bit values are not needed for callsigns. A few of these slots have been assigned to special message components such as "CQ" and "QRZ". CQ may be followed by three digits to indicate a desired callback frequency. (If K1JT transmits on a standard calling frequency, say 144.120, and sends "CQ 113 K1JT FN20", it means that he will listen on 144.113 and respond there to any replies.) A numerical signal report of the form "–NN" or "R–NN" can be sent in place of a grid locator. The number NN must lie between 01 and 30. If required by licensing authorities, a country prefix or portable suffix may be attached[7] to one of the callsigns, as in ZA/PA2CHR or G4ABC/P. If this feature is used, the additional information is sent in place of the grid locator. Some remaining details of message encoding can be found in Appendix A, and a list of supported "add-on" prefixes and suffixes is presented in Appendix B.

5. Forward Error Correction

After being compressed into 72 bits, a JT65 message is augmented with 306 uniquely defined error-correcting bits. The FEC coding rate is thus r = 72/378 = 0.19; equivalently one might say that each message is transmitted with a "redundancy ratio" of 378/72 = 5.25. With a good error-correcting code, however, the resulting performance and sensitivity are far superior to those obtainable with simple five-times message repetition. The high level of redundancy means that JT65 copes extremely well with QSB. Signals that are discernible to the software for as little as 10 to 15 s in a transmission can still yield perfect copy.

The source of this seemingly mysterious "coding gain" is not difficult to understand. With 72 bits the total number of possible user messages is 2^{72}, slightly more than 4.7×10^{21}. The number of possible patterns of 378 bits is a vastly larger number, 2^{378}, in excess of 6×10^{113}. With a one-to-one correspondence between 72-bit user messages and 378-bit "codewords,"

---

[6] Clark, T. W3IWI, and Karn, P., KA9Q, "EME 2000: Applying Modern Communications Technologies to Weak Signal Amateur Operations," *Proc. Central States VHF Society,* 1996.

[7] Callsign prefixes and suffixes were accommodated in a somewhat different way in WSJT versions 4.9.2 and earlier.

or unique sequences of 378 bits, it is clear that only a tiny fraction of the available sequences need to be used in the code. The sequences chosen are those that are "as different from one another as possible," in a mathematically rigorous sense.

A huge variety of efficient error correcting codes are known and understood mathematically. Among the best known are the Reed Solomon codes, used to produce the extremely low error rates characteristic of modern CD-ROMs and hard disk drives. For JT65 I chose the Reed Solomon code RS(63,12), which encodes each 72-bit user message into 63 six-bit "channel symbols" for transmission. Every codeword in this code differs from every other one in at least 52 places — which, in a nutshell, is why the code is so powerful. Even at very low SNR, distinct sequences are very unlikely to be confused with one another.

```
Message #1:  G3LTF DL9KR JO40
Packed message, 6-bit symbols:  61 37 30 28  9 27 61 58 26  3 49 16
Channel symbols, including FEC:
   14 16  9 18  4 60 41 18 22 63 43  5 30 13 15  9 25 35 50 21  0
   36 17 42 33 35 39 22 25 39 46  3 47 39 55 23 61 25 58 47 16 38
   39 17  2 36  4 56  5 16 15 55 18 41  7 26 51 17 18 49 10 13 24

Message #2:  G3LTE DL9KR JO40
Packed message, 6-bit symbols:  61 37 30 28  5 27 61 58 26  3 49 16
Channel symbols, including FEC:
   20 34 19  5 36  6 30 15 22 20  3 62 57 59 19 56 17 35  2  9 41
   10 23 24 41 35 39 60 48 33 34 49 54 53 55 23 24 59  7  9 39 51
   23 17  2 12 49  6 46  7 61 49 18 41 50 16 40  8 45 55 45  7 24

Message #3:  G3LTF DL9KR JO41
Packed message, 6-bit symbols:  61 37 30 28  9 27 61 58 26  3 49 17
Channel symbols, including FEC:
   47 27 46 50 58 26 38 24 22  3 14 54 10 58 36 23 63 35 41 56 53
   62 11 49 14 35 39 60 40 44 15 45  7 44 55 23 12 49 39 11 18 36
   26 17  2  8 60 44 37  5 48 44 18 41 32 63  4 49 55 57 37 13 25
```

Fig. 2. – Three JT65 messages shown as they appear to the user; in 72-bit packed form, displayed as 12 × 6-bit symbol values; and as FEC-enhanced sequences of 63 × 6-bit channel symbols. The channel symbols are ready to be transmitted by means of 64-tone FSK, with each symbol value corresponding to a distinct tone.

As an example, the encoded sequences for three nearly identical messages are illustrated in Figure 2. Lines labeled "packed message" show each source-encoded, 72-bit user message as a sequence of twelve 6-bit symbols. Reading from left to right, one can see that the fifth numerical symbol changes from 9 to 5 when the last letter in the first callsign changes from F to E. The final packed symbol changes from 16 to 17 when the grid locator changes from JO40 to JO41. Otherwise, the three packed messages are identical. On the other hand, the three fully encoded sequences of channel symbols appear to be almost entirely different from one another — so different that *there is virtually no chance whatsoever that, if it is decodable at all, a noise-corrupted version of one of these messages would ever be misconstrued as one of the others*. The full and exact user message has a high probability of being received, even if the key-down SNR is as low as 2 to 6 dB in 2.7 Hz bandwidth (or –28 to –24 dB in 2500

Hz, the conventional reference bandwidth used in WSJT). This statement can be quantified by explicit measurements of transmission error rates as a function of SNR, and such measurements are summarized for JT65 in Appendix C.

6. Interleaving and Gray Coding

After encoding, the order of JT65 symbols is permuted by writing them row-by-row into a 7×9 matrix, and reading them out column-by-column. I was studying FEC for the first time when JT65 was being designed, and I mistakenly believed that scrambling the symbol order would give the system greater immunity to signal dropouts. In fact, it does not; but since its effect is quite harmless, the procedure has been left intact to preserve the integrity of JT65 signals over subsequent program versions. The re-ordered symbols are converted from binary to Gray-code representation, which makes JT65 somewhat more tolerant of frequency instabilities.

7. Shorthand Messages

Like the CW methods described earlier, JT65 uses special signal formats to convey frequently used messages in a robust and efficient way. Three such messages are presently defined. They correspond exactly to the transmissions numbered 4, 5, and 6 in the model CW QSO between SV1BTR and K1JT, conveying the messages "RO", "RRR", and "73". Instead of keying a single-frequency carrier on and off according to a pattern like di-dah-dit, dah-dah-dah, …, JT65 sends "RO" by transmitting two alternating tones with specified frequencies and a specified keying rate. Such waveforms are easy to recognize and to distinguish from one another, as well as from "normal" JT65 messages. Indeed, as many users have discovered, the shorthand messages of JT65 are readily decodable by human operators using sight or sound, as well as by computer.

8. Synchronization and Modulation

JT65 uses one-minute T/R sequences and requires tight synchronization of time and frequency between transmitter and receiver. Typical amateur equipment cannot accomplish this task with sufficient accuracy in open-loop fashion, so a JT65 signal must carry its own synchronizing information. A pseudo-random "sync vector" is therefore interspersed with the encoded information bits. It allows accurate calibration of relative time and frequency errors, thereby establishing a rigorous framework within which the decoders can work. In addition, it enables the averaging of successive transmissions so that decoding is possible even when signals are too weak to accomplish it in a single transmission. The synchronizing signal is so important that (except in shorthand messages) half of every transmission is devoted to it.

A JT65 transmission is divided into 126 contiguous time intervals, each of length 0.372 s (4096 samples at 11025 samples per second). Within each interval the waveform is a constant-amplitude sinusoid at one of 65 pre-defined frequencies, and frequency changes between intervals are accomplished in a phase-continuous manner. A transmission nominally begins at $t = 1$ s after the start of a UTC minute and finishes at $t = 47.8$ s. The synchronizing tone is at frequency 1270.5 Hz and is normally sent in each interval having a "1" in the pseudo-random sequence reproduced at the top of Figure 3. The sequence has the desirable mathematical property that its normalized autocorrelation function falls from 1 to nearly 0 for all non-zero lags. As a consequence, it makes an excellent synchronizing vector.

Encoded user information is transmitted during the 63 intervals not used for the sync tone. Each channel symbol generates a tone at frequency 1270.5 + 2.6917 (N+2) m Hz, where N is the integral symbol value, $0 \le N \le 63$, and m assumes the values 1, 2, and 4 for JT65 sub-modes A, B, and C. The signal report "OOO" is conveyed by reversing sync and data positions in the pseudo-random sequence. Because normal messages depend on tight synchronization, they can be initiated only at the beginning of a UTC minute.

```
1,0,0,1,1,0,0,0,1,1,1,1,1,1,0,1,0,1,0,0,0,1,0,1,1,0,0,1,0,0,
0,1,1,1,0,0,1,1,1,1,0,1,1,0,1,1,1,1,1,0,0,0,1,1,0,1,0,1,0,1,1,
0,0,1,1,0,1,0,1,0,1,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,
0,1,0,0,1,0,1,1,0,1,0,1,0,1,0,0,1,1,0,0,1,0,0,1,0,0,0,0,1,1,
1,1,1,1,1,1
```

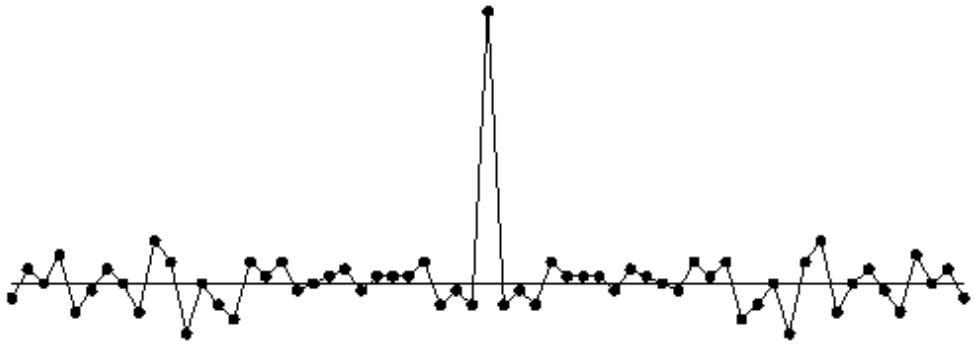

Fig. 3. – The pseudo-random sequence used in JT65 as a "synchronizing vector," and a graphical representation of its autocorrelation function. The isolated central correlation spike serves to synchronize time and frequency between transmitting and receiving stations.

Shorthand messages dispense with the sync vector and use intervals of 1.486 s (16,384 samples) for the alternating tones. The lower frequency is always 1270.5 Hz, the same as that of the sync tone. The frequency separation is 26.917 nm Hz with n = 2, 3, 4 for the messages RO, RRR, and 73. By the time shorthand messages become relevant in a QSO, the frequency offset between transmitter and receiver has already been measured with high accuracy. As a consequence, these messages can be securely identified by the operator as coming from the station whose callsign was recently decoded. Accurate time synchronization is not required for shorthand messages, so they may be started at any time during a transmission.

By now it should be clear that JT65 does not transmit messages character by character, as done in Morse code. Instead, whole messages are translated into unique strings of 72 bits, and from those into sequences of 63 six-bit symbols. These symbols are transmitted over a radio channel; some of them may arrive intact, while others are corrupted by noise. If enough of the symbols are correct (in a probabilistically defined sense), the full 72-bit compressed message can be recovered *exactly*. The decoded bits are then translated back into the human-readable message that was sent. The coding scheme and robust FEC assure that messages are never received in fragments. Message components cannot be mistaken for one another, and callsigns are never displayed with a few characters missing or incorrect. There is no chance for the letter O or R in a callsign to be confused with a signal report or an acknowledgment, or for a fragment of a callsign like N8CQ or a grid locator like EM73 to be

misinterpreted. If your sked partner does not show and another station calls in his place, you will never conclude mistakenly than the schedule was kept as intended.

9. Reception and Demodulation

Within WSJT, a received JT65 signal is converted to baseband and analyzed using a sequence of well known DSP techniques. The process begins with an audio signal in the approximate frequency range 0–3 kHz, digitized at the nominal rate 11025 samples per second. The digital signal is low-pass filtered and downsampled by a factor of two. Power spectra are computed from discrete Fourier transforms of sliding 2048-sample blocks and examined for presence of the pseudo-random sync pattern. Detection and "peaking up" on the sync pattern establishes the required frequency and time offsets, which may include Doppler shift and EME path delays as well as errors in frequency calibration and clock settings. The synchronizing accuracy is typically around 1.5 Hz in frequency and 0.03 s in time. Once "sync" has been established, the program re-measures the sync-tone frequency over small groups of tone intervals and fits a smooth curve to the results, thereby enabling the tracking and compensation of small frequency drifts. Coherent phase tracking between symbols is not required.

With accurate sync information in hand, the program computes a 64-bin spectrum for each of the 63 channel symbols. These spectra have resolution 2.7m Hz (e.g., 5.4 Hz for sub-mode JT65B, m = 2), and with very weak signals they are essentially noise-like in form. Many of the individual data tones may not be detectable above the noise. On average, however, in each tone interval the one frequency bin containing signal will have greater amplitude than the others. Using the known statistical properties of random Gaussian noise, WSJT computes the probability that a symbol was transmitted with each one of the possible values. This probabilistic information, based on measured spectra of the synchronized symbols, is the basic received information. After Gray coding and symbol interleaving have been removed, the probabilities are passed on to the decoder.

10. Reed Solomon Decoder

Even a small error-correcting code like RS(63,12) can be very difficult to "invert" or decode efficiently. The basic problem is this: given the measured spectra for each of the 63 channel symbols, is there a unique 72-bit sequence that can be confidently identified as the user's message? In principle, one might encode each of the 2^{72} possible user messages and correlate the results against the received spectra, looking for a match. Such an approach is quite impractical, however: a simple estimate reveals that with today's 3 GHz computer, unlimited memory, and a very efficient program, it would take about 200 million years to decode a single received message this way.

Reed Solomon codes are economically important because well defined mathematical algorithms exist for decoding them. The algorithms vary in complexity and in how closely they approach the ideal sensitivity of the method just described. Since program version 4.5, WSJT has used an algorithm that represents the state of the art in Reed Solomon decoding. It

is based on a research paper by Ralf Koetter and Alexander Vardy[8], and uses computer code licensed from their company, CodeVector Technologies. Furnished with soft-decision probabilistic information on received symbol values, this decoder produces a clear result for every transmission analyzed. With very high confidence, it returns either the 72 bits of the transmitted message or else a flag indicating "no result".

Error rates for the WSJT decoders have been carefully measured as a function of signal level. The results are summarized in Appendix C. Briefly stated, the K-V decoder exhibits a steep transition from "nearly always decoding" to "nearly always failing" as the signal-to-noise ratio decreases from about –23 to –25 dB (for JT65B) on the WSJT scale. The results further show that with "clean" data (additive Gaussian noise, and perhaps fading, but no interference from other signals), false decodes from the K-V decoding algorithm on RS(63,12) are so rare that you will hardly ever see one.

11. Deep-Search Decoder

What if the K-V decoder fails to produce a result? Can anything further be done? Life is too short to consider correlating all 2^{72} possible user messages in search of a match, but the number of unique messages transmitted in real EME QSOs is actually very much smaller than 2^{72}, and the ones you are most interested in are fewer still. If the more plausible and more interesting messages are tested first — more or less in the same way that one does when copying very weak CW — and if the search algorithm is instructed to "time out" if no match is found after a reasonable time, the brute-force computational approach described above can be made practical. In WSJT, a procedure I call the "deep search" algorithm attempts to do just this.

The deep search starts with a list of plausible callsigns and grid locators. Such lists have long been maintained, both mentally and in hard copy, by most EME operators. They can be of great help when trying to determine which station might be transmitting a weak CQ, answering your own CQ, or tail-ending your last QSO. In the WSJT deep search decoder, each list entry is paired with "CQ" and with the home callsign of the WSJT user, thereby creating hypothetical test messages. If N_c calls are present in the list, approximately $2N_c$ messages will be generated, fully encoded, and the channel symbols tested for good match with the observed spectra. You can define the list of likely callsigns in any way you choose. An example file is provided with WSJT, containing the calls of nearly 5000 worldwide stations known to have been active in weak-signal work on the VHF/UHF bands. Knowledgeable JT65 users maintain their own files, adding or deleting calls as they deem appropriate.

In effect, your callsign database defines a set of matched filters, custom designed for your station and tuned for optimum sensitivity to a subset of the messages you might reasonably expect to receive. The deep search is not sensitive to messages with callsigns not in the database, or arbitrary plain text, or anything besides "CQ" or your own call in the first message field. Such messages will be decoded with the already remarkable sensitivity of the K-V algorithm. However, for any message within the defined subset, the deep search decoder provides about 4 dB more sensitivity while still maintaining a low error rate. It

[8] Koetter, R., and Vardy, A., "Soft-Decision Algebraic Decoding of Reed Solomon Codes," in *Proceedings of the IEEE International Symposium on Information Theory,* p. 61, 2000.

should be obvious that those 4 dB are essentially equivalent to the widely recognized "schedule gain" that CW operators can experience when copying familiar calls or making pre-arranged contacts.

12. Decoding Shorthand Messages

In addition to seeking a synchronizing tone modulated with the expected pseudo-random pattern, WSJT searches for alternating tones having the specified modulation of a JT65 shorthand message. Frequencies are measured and compared with that of the sync tone in a previous transmission, and a test is made to be sure that the modulation follows the specified square-wave cycle. If the frequencies and modulation match, and if the amplitude exceeds a preset threshold, a shorthand message detection is declared. Because of the close frequency and timing tolerances, a low detection threshold can be set while still maintaining a very low rate of false positives. Measured sensitivity curves for shorthand messages are presented in Appendix C, along with those for the K-V algorithm and the deep search decoder.

13. Operator Responsibilities and Message Integrity

QSOs made with any of the WSJT modes, including JT65, require active user participation at all stages. In the presence of birdies, QRM, QRN, or other anomalies such as multipath signal distortions, operator involvement is necessary to avoid mistakes in interpreting program output. Most operators find that they acquire the necessary skills easily, while making their first few JT65 contacts.

In connection with the guidelines for valid QSOs outlined in Section 2, it is worth making special mention of a particular feature of JT65. Contacts made with WSJT are inherently self-documenting. When a JT65 QSO is successfully completed, both operators *know* that the requisite information has been exchanged. Moreover, if desired, they have the recorded wave files to prove it. These files provide a "bit trail," an essentially incorruptible proof of copy that anyone could examine. After especially interesting or difficult QSOs, recorded waveforms and screen images are often exchanged by email. I have accumulated a large library of JT65 wave files from my own QSOs, and by monitoring the bands, as well as many sent to me by others. These files have proven extremely valuable for refining WSJT's algorithms for optimum sensitivity and minimum error rate, under real-world conditions. Further progress will surely be made in these areas, in years to come.

14. On-the-Air Experience

The first usable version of JT65 was finished in November 2003. Early on-the-air tests with N3FZ quickly confirmed my expectation that JT65 would become a major new weapon in the arsenal of VHF/UHF weak-signal enthusiasts. The practical advantages of error-correcting codes for weak-signal amateur radio communication were very plainly evident. Little wonder, I realized, that NASA always transmits its deep-space photographs back to Earth using tight source encoding and strong FEC. In deep-space communications, every dB of improved sensitivity can save millions of dollars that would otherwise have to be spent on larger antennas or more transmitter power.

Definition of the JT65 protocol has evolved only in minor ways since the first test transmissions. Meanwhile, the decoders have been steadily improved, producing sizable advances in on-the-air performance. I have no way of knowing how many EME QSOs have

been made with JT65, but the number is surely in the many thousands. Users have not hesitated to report program bugs or suggest operational improvements, and WSJT has greatly benefited from such feedback. A sizable new group of EME enthusiasts has sprung up, attracted by the fact that JT65 QSOs can be accomplished with much more modest setups than required for traditional methods. Hundreds of JT65 EME QSOs have been made by stations running 150 W to a single yagi on the 2 m band, and QSOs with "big gun" stations have been made with as little as 5 W. Even 50 MHz EME QSOs, long considered among the most difficult of feats, have become a common occurrence.

15. Looking Ahead

I do not foresee the need for major revision or expansion of the JT65 technical specification. However, I can think of many ways in which the implementation of JT65 might be improved. To start with, received audio data should be processed as it comes in, rather than in "batch mode" after the whole reception period is complete. This would permit having a native real-time spectral display, and I can imagine an option to allow "early decoding" of signals after 20 or 30 s of received data have been acquired. I have learned that some sound cards exhibit errors as large as 0.6% in their sampling rates. The JT65 decoders presently in WSJT do not attempt to correct for such errors, and sensitivity suffers unnecessarily. A better job of detecting and suppressing interference can certainly be done. The algorithm presently used to track frequency drifts of the desired signal can be improved. Explicit tracking of Doppler-induced frequency changes is certainly desirable, especially at 432 and 1296 MHz. More accurate control of the timing of transmit/receive sequences would help, and might be possible even under Windows. Execution speed of the decoding procedures can be improved… and the list goes on and on. Perhaps others will take up the challenge to undertake some of these improvements, or will think of other enhancements that will be even more significant.

Appendix A: Details of Message Encoding

As described in Sections §4–6, JT65 message encoding takes place in several stages. A user's message is first "source encoded" into a compact form requiring just 72 bits. The bits are packed into twelve 6-bit information symbols, and a Reed Solomon encoder adds 51 parity symbols. The 63 channel symbols are interleaved, Gray coded, and transmitted using 64-tone frequency shift keying. A synchronizing vector is sent at a 65th frequency, two tone intervals below the lowest data tone.

Some arbitrary choices define further details of message packing and the ordering of channel symbols. To make it easy for others to implement the JT65 protocol, these things are best described with actual source code examples. Appended below is a Fortran program that can easily be compiled under Linux. Only the main program is listed here; the full source code, including necessary subroutines and a Linux makefile, can be downloaded from pulsar.princeton.edu/~joe/K1JT/JT65code.tgz. The compiled program accepts a JT65 message (enclosed in quotes on the command line) and responds with the packed message and channel symbols as six-bit values. Examples of program output were presented in Figure 3 and described in Section §5.

```
      program JT65code


C  Provides examples of message packing, bit and symbol ordering,
C  Reed Solomon encoding, and other necessary details of the JT65
C  protocol.

      character*22 msg0,msg,decoded,cok*3
      integer dgen(12),sent(63)

      nargs=iargc()
      if(nargs.ne.1) then
         print*,'Usage: JT65code "message"'
         go to 999
      endif

      call getarg(1,msg0)                !Get message from command line
      msg=msg0

      call chkmsg(msg,cok,nspecial,flip)  !See if it includes "OOO" report
      if(nspecial.gt.0) then              !or is a shorthand message
         write(*,1010)
 1010    format('Shorthand message.')
         go to 999
      endif

      call packmsg(msg,dgen)              !Pack message into 72 bits
      write(*,1020) msg0
 1020 format('Message:   ',a22)           !Echo input message
      if(and(dgen(10),8).ne.0) write(*,1030) !Is the plain text bit set?
 1030 format('Plain text.')
      write(*,1040) dgen
 1040 format('Packed message, 6-bit symbols: ',12i3) !Print packed symbols

      call packmsg(msg,dgen)              !Pack user message
      call rs_init                        !Initialize RS encoder
      call rs_encode(dgen,sent)           !RS encode
      call interleave63(sent,1)           !Interleave channel symbols
      call graycode(sent,63,1)            !Apply Gray code

      write(*,1050) sent
 1050 format('Channel symbols, including FEC:'/(i5,20i3))
      call unpackmsg(dgen,decoded)        !Unpack the user message
      write(*,1060) decoded,cok
 1060 format('Decoded message: ',a22,2x,a3)

 999  end
```

Appendix B: Supported Callsign Prefixes and Suffixes

Callsign prefixes and suffixes supported by JT65 are listed in the file `pfx.f` included in the source code archive at pulsar.princeton.edu/~joe/K1JT/JT65code.tgz, as described in Appendix A. Supported suffixes include /P and /0 through /9, while the full prefix list is appended below. Additional prefixes and suffixes could be added to the list in the future. Space for 450 prefixes has been reserved by not supporting any grid locators within 5° of the North Pole.

```
1A    1S    3A    3B6   3B8   3B9   3C    3C0   3D2   3D2C  3D2R  3DA   3V    3W    3X
3Y    3YB   3YP   4J    4L    4S    4U1I  4U1U  4W    4X    5A    5B    5H    5N    5R
5T    5U    5V    5W    5X    5Z    6W    6Y    7O    7P    7Q    7X    8P    8Q    8R
9A    9G    9H    9J    9K    9L    9M2   9M6   9N    9Q    9U    9V    9X    9Y    A2
A3    A4    A5    A6    A7    A9    AP    BS7   BV    BV9   BY    C2    C3    C5    C6
C9    CE    CE0X  CE0Y  CE0Z  CE9   CM    CN    CP    CT    CT3   CU    CX    CY0   CY9
D2    D4    D6    DL    DU    E3    E4    EA    EA6   EA8   EA9   EI    EK    EL    EP
ER    ES    ET    EU    EX    EY    EZ    F     FG    FH    FJ    FK    FKC   FM    FO
FOA   FOC   FOM   FP    FR    FRG   FRJ   FRT   FT5W  FT5X  FT5Z  FW    FY    M     MD
MI    MJ    MM    MU    MW    H4    H40   HA    HB    HB0   HC    HC8   HH    HI    HK
HK0A  HK0M  HL    HM    HP    HR    HS    HV    HZ    I     IS    IS0   J2    J3    J5
J6    J7    J8    JA    JDM   JDO   JT    JW    JX    JY    K     KG4   KH0   KH1   KH2
KH3   KH4   KH5   KH5K  KH6   KH7   KH8   KH9   KL    KP1   KP2   KP4   KP5   LA    LU
LX    LY    LZ    OA    OD    OE    OH    OH0   OJ0   OK    OM    ON    OX    OY    OZ
P2    P4    PA    PJ2   PJ7   PY    PY0F  PT0S  PY0T  PZ    R1F   R1M   S0    S2    S5
S7    S9    SM    SP    ST    SU    SV    SVA   SV5   SV9   T2    T30   T31   T32   T33
T5    T7    T8    T9    TA    TF    TG    TI    TI9   TJ    TK    TL    TN    TR    TT
TU    TY    TZ    UA    UA2   UA9   UK    UN    UR    V2    V3    V4    V5    V6    V7
V8    VE    VK    VK0H  VK0M  VK9C  VK9L  VK9M  VK9N  VK9W  VK9X  VP2E  VP2M  VP2V  VP5
VP6   VP6D  VP8   VP8G  VP8H  VP8O  VP8S  VP9   VQ9   VR    VU    VU4   VU7   XE    XF4
XT    XU    XW    XX9   XZ    YA    YB    YI    YJ    YK    YL    YN    YO    YS    YU
YV    YV0   Z2    Z3    ZA    ZB    ZC4   ZD7   ZD8   ZD9   ZF    ZK1N  ZK1S  ZK2   ZK3
ZL    ZL7   ZL8   ZL9   ZP    ZS    ZS8
```

Appendix C: Measured Sensitivity and Error Rates

The JT65 protocol can be defined once and for all, but on-the-air performance depends on a particular software implementation of the decoder. As outlined in §9–12, version 4.9 of WSJT does its JT65 decoding in three phases: a soft-decision Reed Solomon decoder, the deep search decoder, and the decoder for shorthand messages. Section §13 emphasizes that in circumstances involving birdies, atmospherics, or other interference, operator interaction is an essential part of the decoding process. The operator can enable a "Zap" function to excise birdies, a "Clip" function to suppress broadband noise spikes, and a "Freeze" feature to limit the frequency range searched for a sync tone. Having used these aids and the program's graphical and numerical displays appropriately, the operator is well equipped to recognize and discard any spurious output from the decoder.

Under normal conditions in which the transmission channel can be characterized by simple attenuation, the addition of white Gaussian noise, and perhaps multiplication by a "Rayleigh fading" coefficient, the sensitivities and error rates of the decoders can be accurately measured. A software simulator for doing this was written for the Linux platform as the first (and very essential) part of WSJT program development. The simulator can generate digitized waveforms for any WSJT mode and inject them into band-limited Gaussian noise with a specified signal-to-noise ratio and optional fading characteristics. The resulting audio

files can be saved in WAV format, then opened and decoded in WSJT. They can also be decoded directly within the simulator, using code identical to the WSJT decoder but compiled for Linux.

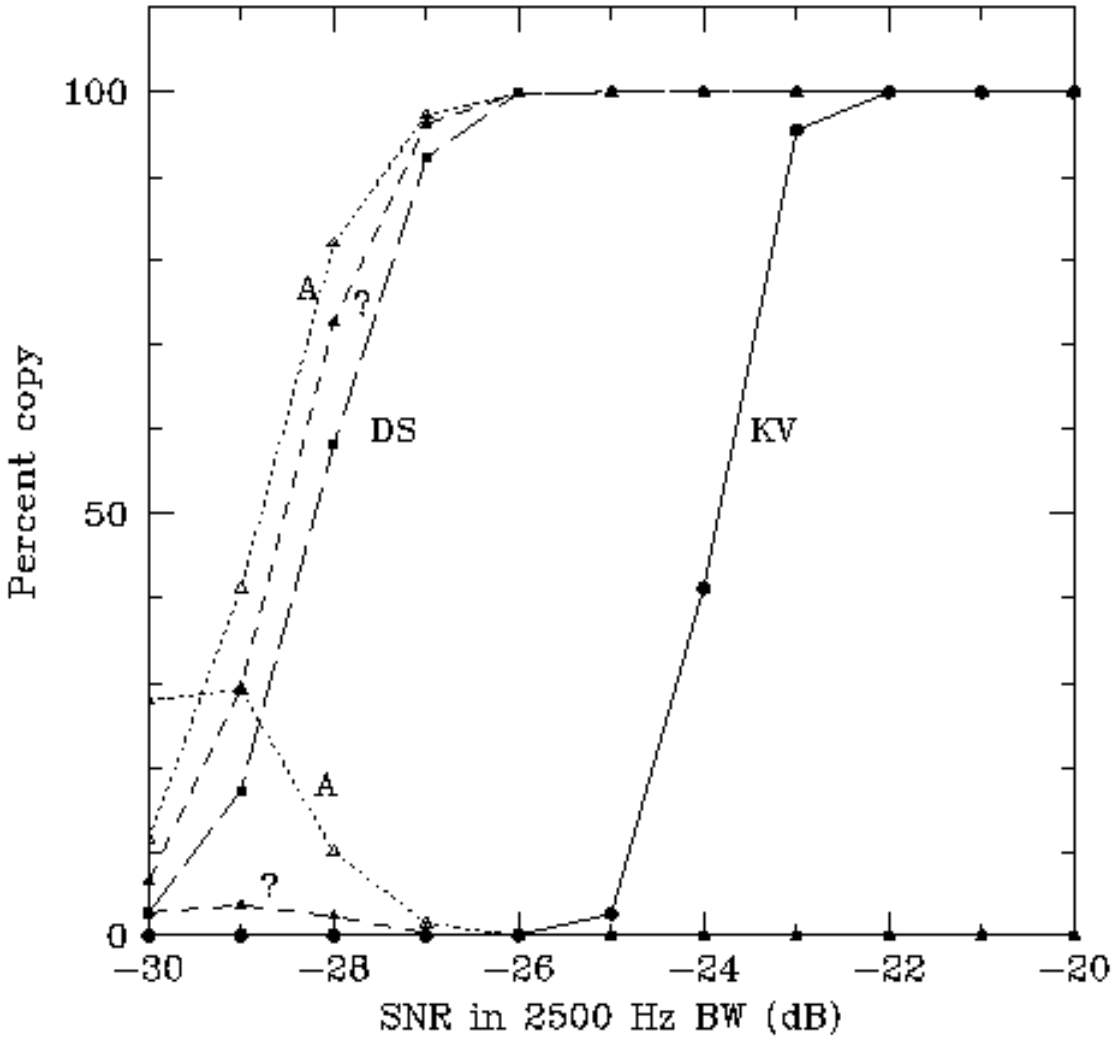

Fig. 4.– Measured rates of copy as a function of SNR for JT65B. The curve labelled KV refers to the Koetter-Vardy algorithm; DS refers to the deep search algorithm. The rate of false decodes for the KV algorithm is too small to measure; for the DS algorithm the rate of "hard errors" was about 0.03%, too small to show on this graph. Curves labelled "?" and "A" at the lower left give the deep-search soft-error rates for decoded messages marked "?" and when "Aggressive decoding" has been requested.

Several hundred thousand simulated JT65 transmissions have been tested in this way — first as a means of debugging and fine-tuning the decoders, and later as a way to measure the sensitivity and error rates of the finished program. Results of the simulations are summarized in Figures 4 and 5. To create Figure 4, 1000 simulated transmissions were generated and tested for each of the levels SNR = –30, –29, … –20 dB, using standard JT65 messages consisting of two callsigns and a grid locator. The full WSJT decoder (version 4.9.5) was run

on each of the 11,000 simulated transmissions. The filled circles and solid curve in Figure 4 illustrate results from the Koetter-Vardy decoder. The essential conclusion is that 96% of the transmissions were decoded correctly at –23 dB, 41% at –24 dB, and 3% at –25 dB. No false decodes were produced by the KV decoder in any of the tests.

For the deep search algorithm, the filled squares and long-dashed curve show that 92% of the transmissions were decoded correctly at –27 dB, 58% at –28 dB, and 17% at –29 dB. Three "hard errors" (false decodes not flagged with a question mark) were recorded in the 11,000 simulated transmissions, for an overall error rate of 2.7×10^{-4} (too small to be seen in Fig. 4). If one includes decoded messages flagged with a question mark, the numbers for correct copy increase to 96%, 73%, and 29% at signal levels –27, –28, and –29 dB (short-dash curve and filled triangles). The error rate, illustrated by the short-dash curve at the lower left of Figure 4, reaches a maximum of 3.6% at –29 dB. With WSJT's "Aggressive decoding" option selected, the percentages of correct copy increase to 97%, 82%, and 41%, at –27, –28, and –29 dB (dotted curve and open triangles). However, the rate of false decodes also increases substantially, especially at –28 dB and below, reaching a maximum of 29% at –29 dB.

Similar measurements have been made for sub-modes JT65A and JT65C. The results are qualitatively similar to those shown for JT65B in Figure 4; the curves for JT65A are shifted about 1 dB to the left (more sensitive than JT65B), while those for JT65C are shifted about 1 dB to the right.

Normal JT65 messages cannot be decoded unless the sync vector is reliably detected. In WSJT the synchronizing procedure is exactly the same for sub-modes JT65A, B, and C. For the tests illustrated in Figure 4 with SNR less than about –29 dB, failure to synchronize is the cause of many failures to decode. Synchronization is very important for another reason, as well: correct synchronization may allow the decoding of an accumulated average message, independent of whether the transmitted message is decodable with the deep search algorithm. Measured rates of synchronization are illustrated in Figure 5, again using 1000 simulated transmissions at each value of SNR over a 10 dB range. Synchronization was achieved for 93% of the test transmissions at –28 dB, 74% at –29 dB, 44% at –30%, and 19% at –31 dB. These measurements imply that message averaging should typically succeed after about 3 transmissions at –26 dB and 8 transmissions at –28 dB, but will require as many as 20 transmissions at –29 dB. These conclusions are consistent with on-the-air experience with WSJT.

The simulator was also used to measure the detection rates for JT65 shorthand messages, as illustrated in Figure 5. With 1000 trials at each SNR, shorthand messages were correctly decoded in 88% of the trials at –31 dB, 60% at –32 dB, and 26% at –33 dB. The total number of incorrectly decoded shorthand messages was five, in 11,000 trials. All five would have been recognized as spurious by an attentive operator, because the measured frequency offset was much larger than the normal tolerances used.

For any of a large number of reasons, on-the-air performance of JT65 may differ somewhat from the simulated results shown here. The measurements summarized in Figures 4 and 5 were made under idealized conditions with additive white Gaussian noise (AWGN) and no fading. (An additional set of simulations has been made with the effects of Rayleigh fading included; the results are qualitatively similar to those shown here, with the curves shifted several dB to the right.) The effects of birdies, other interference, and non-Gaussian noise are harder to quantify. Suffice it to say that I often leave WSJT running in "Monitor" mode

for days at a time, with my receiver tuned to an arbitrary frequency between 144.100 and 144.160. I live in a densely populated region where plenty of birdies as well as other signals come and go on the 2 meter band. The typical rate of false decodes when monitoring a quiet band averages no more than one or two per hour. Examination of the files producing the spurious decodes nearly always reveals tell-tale evidence that would have caused an operator to recognize and reject the invalid information. When used in the intended manner, JT65 is a highly accurate communication protocol.

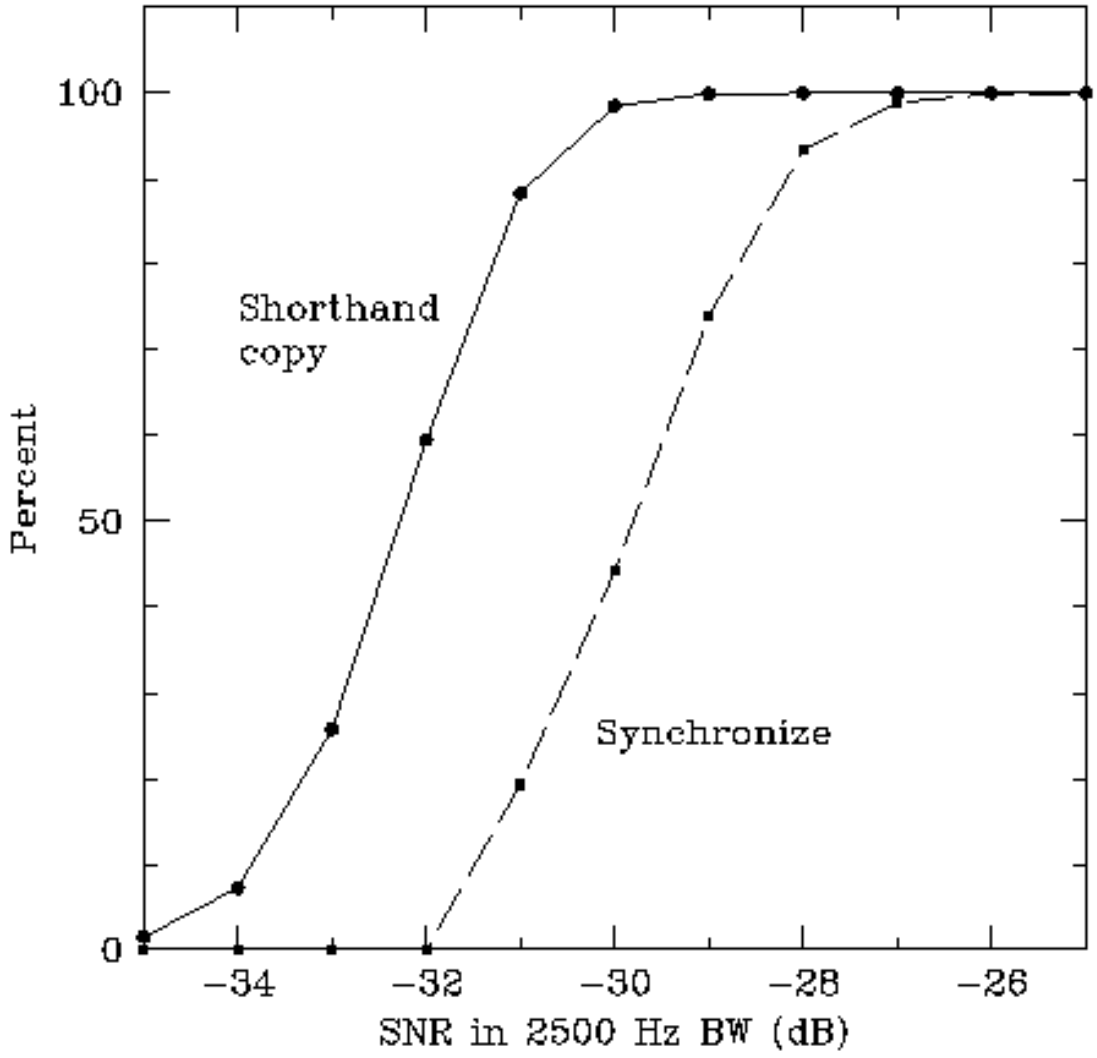

Fig. 5.– Rates of message synchronization and copy of shorthand messages as a function of SNR.

Version: March 9, 2005