

The JT65 Communications Protocol

Joe Taylor, K1JT

日本語訳 Toshihisa Takahashi, JH1OQW

要約 JT65 はアマチュア無線の極めて微弱な信号処理を対象としているデジタルプロトコルである。これは VHF に於ける EME 通信が最適化出来るように設計されている。そしてこのような QSO の手順と標準化を効率良く行うことを目的としている。JT65 は非常に弱い信号に対しても対処できる強力なエラー訂正プログラムを組み込んでいる。本論文は JT65 の技術仕様とその意図する動機及び設計フィロソフィの背景情報を扱っている。さらに WSJT と呼ばれるコンピュータプログラム内の JT65 遂行の詳細とその感度及びエラーレートの関連について解説している。

1 Introduction

火花通信は 80 数年前に連続信号に取って代られている。多かれ少なかれ通常の場合、on off キーイングによるモールスコードを用いた国際通信はその時から、また殆どのアマチュア無線でも微弱信号通信モードとして採用されて来ている。モールスは便利であり、応用自在であり、人手によって符号化と復調が出来る。On-off キーイングは簡単な装置であり、その占有バンド幅は小さい。その選択は誰でも簡単に出来ている。しかしながら CW の符号化も CW の変調も、最適の条件ではないということを簡単に示すことが出来る。信号とノイズの比を夫々 dB カウントする時に、アマチュア無線に於ける流星通信や EME では、他の選択肢を探した方が大変良い理由がある。サウンドカードを備えたパソコンは幅広い可能性を持った実験をする時に黄金のような機会を提供する。

そのプログラムである WSJT^{1,2,3} (" Weak Signal communications, by K1JT ") は非常に効率的なコーディング(符号化)と変調の技術をアマチュア無線の Weak-Signal 通信へ導入した私の努力の結果である。プログラムの要点は殆ど全ての VHF/UHF の Weak-Signal 運用者の知るところとなり、彼等の多くは通常使用を行っている。VHF に於ける流星通信では圧倒的の大多数が、また EME 通信の半数が現在 WSJT の支援を利用している。

本論文での JT65 は、WSJT によってサポートされている通信プロトコルの一つである。JT65 は EME 伝播路を通るような極めて微弱な電波を用いた通信用にハッキリした意図を持って設計している。このプログラムの操作手順については *WSJT User 's Guide* ⁴ に述べられている。ここにはこのプロトコルの完全な技術説明と WSJT に実装している方式の一般解説を述べている。最新式のデジタル通信システムは情報理論の数学をベースにしている。この分野は 1948 の 2 つの論文 ⁵ Claude Shannon が証明した。情報は任意の低いエラーレートを持ったノイズチャンネル上を運ばれ、その情報処理量はチャンネルのバンド幅と信号と雑音比 (SNR; signal - to - noise ratio) のみに依存する。

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 Communication," *Bell System Tech. J.*, **27**, pp. 379-423 and 623-656, 1948.

非常に低い SNR に於ける低エラーレートを達成するためには数学的にユーザ情報を符号化してコンパクト（簡潔）にする必要があり、注意深く作られた冗長性を持たせている。この簡潔さは送信パワーを最小にしたり、情報処理能力を最大にするために必要である。冗長性は雑音と変化するチャンネル上を伝わるメッセージエラーの無いことを保証するために必要である。

無線伝送のためには、符号化されたメッセージはある種の変調を使用したキャリアに乗せる必要がある。その可能性の制限はほとんど無い；情報は増幅度、周波数、キャリアの位相、またはそれらの組み合わせによって伝送することが出来る。一般に使用できるデジタル変調手法は on-off キーイング（増幅変調を一定にする） phase-shift キーイング（PSK）、frequency-shift キーイング（FSK）を含んでいる。JT65 プロトコルは 65-tone(65 個のトーン)の FSK を使い、これは一定の増幅度と位相が連続した波形を用いている。この変調方式は on-off キーイングよりも非常に効率が高く、最良のコーディング技術と結びつくと効果が発揮できる。さらに phase-shift キーイングよりも周波数変動に対する許容度ははるかに高い。

セクション §2 は JT65 の設計思想を誘導するための背景情報で始まる。セクション §3 はシステムデザイン全体を高いレベルで紹介する。プロトコル自体は §4 - 8 と Appendix A である。§9 - 12 は JT65 信号の受信と解読についてである。プロトコルの仕様については JT65 の有効なメッセージの符号化を送信波形にする定義をしている。JT65 信号を解読するため必要な全ての情報を用意している。WSJT に於いて、これらのタスクがいかに実施されるかの本質的な情報を含んでいる。各種の JT65 の遂行、特に受信に使用されるアルゴリズムについて、そしてその可能性についての説明をしている。この論文は他人がこの WSJT を使用するための動機付けになることを希望し、このような努力がこのモードの処理能力と利便性の更なる改良に繋がることである。

2 EME QSO's : Requirement and Procedure

（EME QSO に於ける要求事項とその手順）

アマチュア無線はまさに趣味の活動である。多くの人々にとって全ての国々、全てのアメリカ州、そして可能な限り DXCC のエンティティを増やすためのコンタクトが目標ゴールに上げられる。これらのゴールは EME を用いると特に困難が伴う。その理由から多くのハムは大きな欲望を持ち挑戦的になるのである。ゲームに参加するために、参加者はいくつかの基礎的なグラウンド・ルールに同意する必要がある。

信号が適度に強くそして熟練を積んだハム同士がエラーフリーでコンタクトが出来た場合には QSO が完結したかどうかの判定は容易である。HF バンドで珍局が現れた時、パイルアップのなかでの速攻の QSO は一般に次のようなコンタクトのやりとりが行われる。

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

この通信モデルでは K1JT は通信相手のコールサインは決して送信しない。この状況下では相手のコールサインが明々白々であるからである。この時の信号強度は S9 ではないかもしれないが、これについては意に介さない。コンタクトの交換が行われた後に、両局は自信を持って、その QSO をログブックに記録する。そして、その後このコンタクトを確認するために QSL カードを

交換するかもしれない。

VHF/UHF の世界で、特に EME 通信では信号はしばしば特に弱く、大いに熟練を積んだ局同士間でもエラー無しでのコンタクトは少ない。結果として、どうしたら必要最小限の QSO を構成できるか、信頼できる標準が必要がある。長い間確立されてきたルールは両局の完全なコールサインのコピー、シグナルレポートまたはその他の情報、そしてこれらの情報の全てが受信されたことを明白に確認できたこと。これらのガイドラインは全てのタイプの Weak-signal QSO にも当てはめることが出来る。これらのタイプとは Tropo, Meteor scatter, EME, その他の伝播モード、すべての信号方式と装置などである。

これらのガイドラインに綿密に従うと、常識ある VHF 局による必要最小限の EME QSO は一般に次のような手順によって進行する：

1. CQ SV1BTR...
2. SV1BTR K1JT...
3. K1JT SV1BTR 000...
4. RO...
5. RRR...
6. 73

スケジュールを決めた QSO に対しては、時間と周波数は前もってアレンジしているので、# 1 は勿論必要は無い。省略の (...) はメッセージの繰返しを含み EME コンタクトでは殆ど日常的に使用され成功のチャンスを最大にする。000 のメッセージ部は最も短いシグナルレポート短縮表現である。この表現は、あなたの信号は少なくともある時間解読が可能であり私は両方のコールサインをコピーしている。“RO” は短縮メッセージ (shorthand message) でシグナルレポートおよび承認の両方を示している。この意味は“私は両方のコールサインと私のシグナルレポートをコピーした。そして貴方のレポートは O である。” K1JT が SV1BTR から送られた承認である“RRR”を受信する時にその QSO は完全であるが、SV1BTR はまだこれを知らないので、73 またはコンタクトの終了を示す“我々は完了している”などを送信するのが習慣的である。

短縮メッセージは無線と地上電信が始まって以来広く使用されており、慣れ親しんでいる Q 符合も広く世界的に理解されている。これらは“情報の符号化”とよばれる通信理論の最も単純な形式である。“000...” (連続波の間隔が短いスペースによって区切られてこれが 3 個続くの繰返し。長いスペースの後はこの 3 分の 1 のスペースが来る) 短点よりも長点の方がコピーしやすく、ハッキリしたシグナルレポートを表す信号として、賢く経験を積んだ CW マンが極めて弱い信号を識別するために編み出したものである。

3. System Design

Figure 1 は現代のデジタル通信システムのフローダイアグラムを表している。低い S/N に於いて最大の効果を上げるために、ユーザーメッセージは最小限の冗長性を持たせて、情報の符号化をしたコンパクトな形にする。後から議論されるが、数学的に冗長性を持たせているので、もしノイズまたは信号の欠落によりある部分が破壊されてもそのメッセージを完全に回復させることが出来る。このプロセスは“前方エラー訂正”または FEC として知られている。符号化情報はエラー訂正情報を含み、キャリア上に配置している。この無線信号レベルは EME では伝播ロスが 250dB 又はそれ以上になる。そして増幅度とノイズ、周波数、“path modulation”と呼ばれる位相変化などが加わる。受信では、その信号が検波され解読されて、その結果をコ

ーザーに知らせる。

エラー訂正を高めることを除いて、Figure 1のダイアグラムの流れは、伝統的なアマチュア無線のCW通信も現代のデジタル技術も同じ流れであることを示している。前のページでCW EME QSOアウトラインの語句の中で、“SV1BTR こちらはK1JT、私は両方のコールサインをコピーした”を”SV1BTR K1JT 000“と圧縮フォームにして情報の符号化ができる。一定量のエラー回復能力を高めるために、そしてそのメッセージがコピー出来るチャンスを増加するために、CWオペレータは送信時間内に圧縮メッセージを多数回繰返して送信する。更にコピーのチャンスを高めるために、彼は1回の送信でただコールサインの繰り返し送信に75%の時間を割り、000の繰り返し送信に残りの25%の時間を費やす。彼は受信オペレータがこれらの習慣について知っており、それに従って受信することを期待する。これらの情報符号化の全ては、weak-signalの特長を知れば知るほど、そのコピーが更に容易になることである。極めて限界ギリギリの条件下では、熟練したオペレータは彼が前もって期待するメッセージの部分と受信した内容とを照合しながら受信をする。もし良好な照合が見つければメッセージのコピーは保証されたと考えることが出来る。

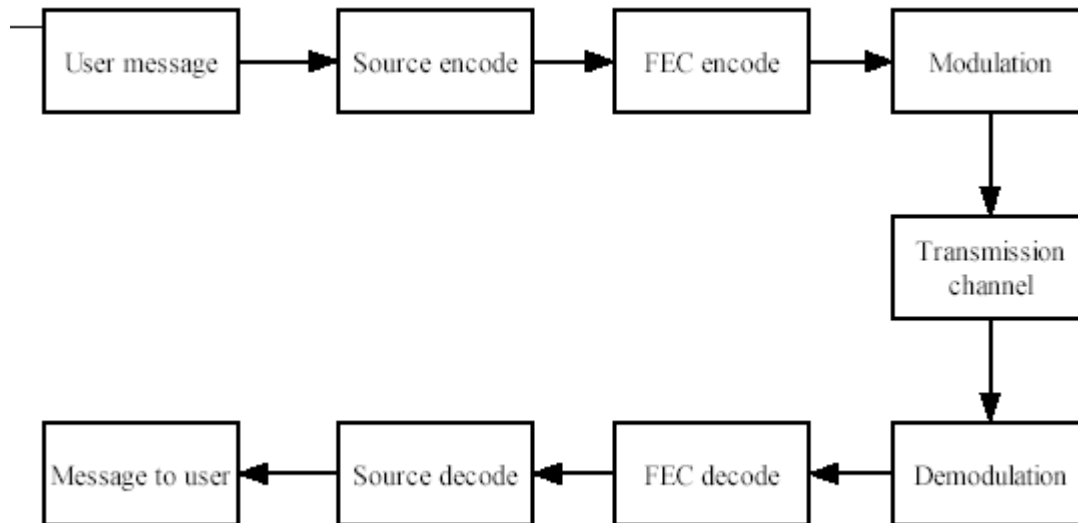


Fig 1 デジタル通信システム情報の流れの概要を示す。

4.JT65 Source Encoding (JT65における情報の符号化)

JT65はまさに同類の技術を使用して送信されるメッセージはコンパクトで効率的であることから始まる。WSJT 4.7 User's Guide⁴に記載しているように、JT65の標準メッセージ“Type 1”は2つのコールサイン、1つのグリッドロケータと視覚のシグナルレポートから構成されている。先のモデルQSOであるSV1BTRとK1JTのようにQSOが進みメッセージ2、3が続く。情報の符号化は標準のアマチュア局のコールサインが作られるルールを用いている。このため必要とする情報ビット数を最小化するためにこの情報を使用する。アマチュア局のコールサインは1つ又は2つの接頭辞から構成されており、その内少なくとも1つは文字であらねばならない。続いて1つの数字と1つから3つの文字が続く。このルールを内で、コールサインの可能な局数は $37 \times 36 \times 10 \times 27 \times 27 \times 27$ に等しいか幾分2.67百万局を超える。(27と37は最初なるが故に発生して、続く3つの位置は1文字が欠落かまたは1文字または多分1つの数字となると思われる)。だから²⁸2²⁸は268百万より大きく、28ビットあれば如何なる標準コールサインも重複す

ること無く符号化に十分である。同様に地球を蔽う4数字グリッドロケータは $180 \times 180 = 32400$ 、
で $2^{15} = 32768$ よりも少なく、グリッドロケータを表すには15ビットあれば十分である。EMEメ
ッセージを効率良く符号化するこれらの重要なアイディアは1996年にClarkとKarn⁶によって初
めて提案された。如何なるType 1のメッセージも $28+28+15 = 71$ ビットへ情報の符号化が出来
るので、さらにもう1つシグナルリポートが追加できる。比較のために、メッセージ“SV1BTR
K1JT000”はモールスコードでは、グリッドロケータを除いても、170ビット(ここで1ビッ
トはキーダウンのドットのインターバルとして定義する)が必要になる。JT65のメッセージは
CWのメッセージよりも非常にコンパクトになる。だからこの間更に多くの情報を伝えることが
出来る。実際に、JT65のプロトコルは他の方法でシグナルリポートを符号化する。グリッドロ
ケータとコールサインに代わる任意の文字を含むメッセージを 72 ndビットを使用して表す。43キ
ャラクタとアルファベットを用いて、最大平文の長さは13文字(最も大きい整数は $71 \log_2 / \log_2 43 = 13.084$ よりも小さい)である。この制限の大きさに従うと、JT65は1つのメッセージの中で
何でも送信と受信が出来る。

上記で示したように、可能な28ビット値の中の約600万はコールサインに必要ではない。これら
のスロットの一部は“CQ”、“QRZ”など特別なメッセージ類に割り当てられる。CQは望まれる応
答周波数を示すために3数字を従える場合がある。(もしK1JTが標準呼び出し周波数144.120で“CQ
113 K1JT FN20”を送信するならば、この意味は144.113を受信し、いかなる応答にも答える
ことを示している)。“-NN”または“R-NN”形式の数値シグナルリポートはグリッドロケータの
代わりに送信することが出来る。NN数は01から30の間であることが必要である。もしライセン
スを発行する当局が国の識別接頭辞または移動接尾辞をコールサインに付加する要求がある場
合には、例えばZA/PA2CHRまたはG4ABC/Pなどようになる。もしこの形が使用されるならば、
この付加情報はグリッドロケータの場所で送られる。メッセージの符号化についての詳細の一
部はAppendix Aのなかにある。そして“add-on”の接頭辞と接尾辞はAppendix Bに掲載され
ている。

5. Forward Error Correction

72ビットの中に圧縮した後に、JT65のメッセージは定義された重複の無い1306のエラー訂正ビッ
トへ拡大される。このFECコーディングレート $r = 72/378 = 0.19$ となる。また各々のメッセージ
は冗長比 $378/72 = 5.25$ で送信されるといえよう。良いエラー訂正コードを持つことはただ単に
同じメッセージを5回繰返すよりも、結果として実行能力と感度が遥かに優れたものになる。高
い冗長性を持つことはJT65がQSBを受けても極めて高い受信能力を持つことを意味している。
認識できる信号は送信してから10ないし15秒の間はソフトウェアに対して完全な解読を続ける
ことが出来る。このように見える不思議な現象の源は“コーディングゲイン”と呼ばれるもの
でその理由は難しくは無い。可能なユーザーメッセージの合計は72ビットであり 2^{72} である。こ
れは 4.7×10^{21} よりも僅かに大きい。378ビットの可能なパターン数は極めて大きな数 2^{378} であり、
 6×10^{113} を越えている。72ビットのユーザーメッセージと378ビットの“コードワード”または
重複の無い378ビットシーケンス間の1対1の対応は、このコードで使われる必要なシーケンス
の割合が非常に小さなものであることが明白である。選択されたこれらのシーケンスは、数学
的に厳密なセンスで、お互いに異なっている可能性がある。

⁶ Clark, T. W3IWI, and Kam, 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.

非常に多様性に富んだ数多くの効率の良いエラー訂正コードが知られておりそして数学的にも理解されている。その中でベストなものがReed Solomon コードであり、現代のCD-ROMsとハードディスクに使用されており極めて低いエラーレートを作り出すために使用されている。JT65のために、私はこのReed Solomon コードRS(63,12)を選択しており、72ビットのユーザーメッセージを63個の6ビット“チャンネルのシンボル; Channel Symbols”を送信のために符号化する。このコードの中で各々のコードワードは少なくとも52個所(1つのナットシェル)内でお互いに異なって格納されている。これがパワフルの理由である。非常に低いSNRの時でさえハッキリしたシーケンスはお互いに全く混乱無しに行われる。

```
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 3つのJT65メッセージは72ビットをパックした形でユーザーに見えるようにした。これは12×6ビットシンボル値で示している;そしてこれはFEC-エンハンスしたシーケンス63×6ビットのチャンネルシンボルである。このチャンネルシンボルは64トーンのFSKを用いて送信するようになっており、各々のシンボル値は夫々1つのトーンに分けて送信される。

1つの例として符号化したシーケンスを3つのメッセージ例を用いてFig.2に解説している。“パックされたメッセージ: packed message”のラベルの付いたラインは各々の符号化した情報を示し、72ビットのユーザーメッセージは12-6ビットシンボルのシーケンスとして表示されている。左から右へ読んでゆき5番目の9から5へ数字が変わった時に時に、最初のコールサインの最終文字がFからEに変わる。最終のパックされたシンボルが16から17に変わった時にはグリッドロケータがJ040からJ041に変わる。他の点ではこのパックされた3メッセージは完全に一致している。一方この完全に符号化されたチャンネルシンボルのシーケンスは殆ど全くお互いに異なって現れる。(このように異なっていることはどんなチャンスも生まれない。もしそれが解読可能であっても、これらのメッセージの1つがノイズ破壊された場合は別のものの1つとして解釈を誤る。)もしSNRがバンド幅2.7Hzでないし6dBよりも低くても(WSJTで使用している

通常のバンド幅での換算では2500Hz、-28から - 24dBになる)完全で正確なユーザーメッセージを高信頼度で受信ができる。この計算書はSNRの関数として伝達エラーレートを明快な計測によって定量化することが出来る。そのような計測はAppendix CのなかのJT65に対して要約されている。

6. Interleaving and Gray Coding

(配列を変えて高速化を計り更にグレーコード化する)

符号化ができれば、JT65シンボルは横から横への流れを7×9のマトリックの中へ入れて順序を変える。それから読み出す場合には縦から縦への順序で行う。私は最初にJT65をデザインした時にFECを勉強した。シンボルの順序を急激に変えることはそのシステムの大きな免疫性を脱落させると誤って信じていた。事実は、そうではなく、その効果は無害であり、その順序はその後続いて起こるプログラムバージョンでもJT65の信号の一貫性を完全に残している。並び替えられたシンボルはバイナリ - からグレーコードに変換される。そしてこれは周波数の不安定性に対する許容度をJT65に対して大きくする。

7. Shorthand Message

先に話題にしたが、CWモードのように、JT65は強力で効率的な方法として、度々使用されるメッセージに対して特別なシグナルフォーマットを使用する。現在3メッセージが定められている。送信No4,5、と6のSV1BTRとK1JT間のCW QSOのモデルに使用されており、このメッセージは“R0” “RRR” と“73”である。単一周波数のキャリアのオンオフではト ツー ト ツー ツー ツーがJT65では“R0”となわり、これは2つの特別に設定された周波数とキーイングレートをを用いて交互に送信する。このように波形は容易に識別され易くまた他の信号との区別が付きやすい。JT65の“normal”メッセージ(平文)も同様なことがいえる。本当に多くのユーザーが発見しているように、JT65のショートハンドメッセージはオペレータの耳または目によっても解読が可能であり、コンピュータによっても同様である。

8. Synchronization and Modulation

JT65は1分間のT/R(送信/受信)のシーケンスを使用して送信機と受信機との厳しい同期を要求する。典型的なアマチュア局の装置はオープンループの中では十分な精度のタスクを成し遂げることは出来ない。このため、擬似ランダム“同期ベクトル: Sync Vector”を符号化された情報ビットの間に分散させている。それは相対的な時間と周波数誤差を正確に較正することができ、デコーダー(解読器)が働くことのできる範囲内の厳密な構成要素を作り上げることができる。加えてそれはある送信状態の中でそれを成し遂げるにはあまりにも弱い信号の時さえその解読が可能であり、連続的な送信の平均化が可能である。信号の同期は非常に重要で(ショートハンドメッセージは除く) 各々の送信の半分は同期信号を送るために捧げられている。

JT65の送信は126の隣接している時間間隔に分けられる。各々の長さは0.372秒(11025サンプル/秒の時サンプル数が4096)となる。各々の時間間隔以内の波形は増幅が一定のサイン波で、65波の周波数が前もって割り当てられておりその中から1波が取り出される。周波数は周波数間隔に対して位相が連続している方法で行われる。送信は通常、UTCの秒目盛がT=1秒から始められてt=47.8秒に終了する。同期信号は1270.5Hzであり、通常、Fig.3のトップに示されるように、擬似ランダムシーケンスの1で示すインターバルで送信される。これらのシーケンスは数学的な特性を持っており、基準化された自動調整機能はゼロでない遅延(lags)を1から殆ど0にする。この結果のように、それは優れた同期ベクトルを作る。

符号化したユーザー情報を同期信号には使用されない163個のインターバルの間で送信される。

各々のチャンネルシンボルは $1270.5 + 2.6917(N+2)$ mHzの周波数トーンを発生させる。ここでNは0 ~ 63の範囲の整数であり、mはJT65のサブモードA,BとCに対して1,2,と4を割り当てる。シグナルリポートの"000"は擬似ランダムシーケンスのなかの同期とデータ位置を逆にすることによって伝達される。平文は厳しい同期に頼っているため、それらはUTC分の始まりにのみ解読される。

```

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,
0,1,1,1,0,0,1,1,1,1,0,1,1,0,1,1,1,1,0,0,0,1,1,0,1,0,1,0,1,1,
0,0,1,1,0,1,0,1,0,1,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,
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

```

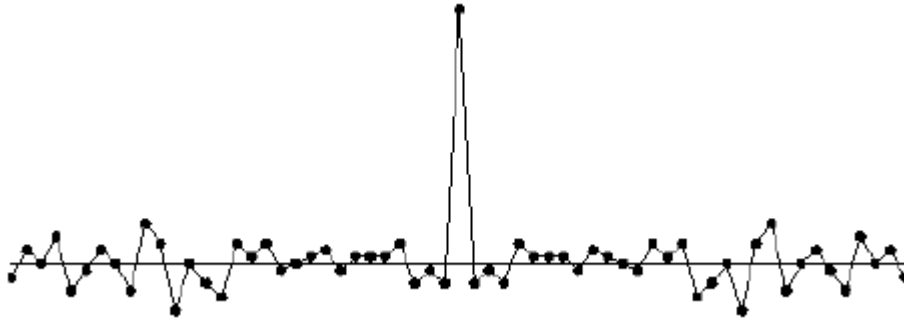


Fig.3 JT65の擬似ランダムシーケンスは同期ベクトルを使用している。その自己相関機能をグラフで表現している。中央の際立った関連スパイク波は送信と受信局の同期時間と周波数を現している。

ショートハンドメッセージは同期ベクトルを使用しない。そして交互に変化するトーン1.486秒(16384サンプル)のその間隔を使用する。その下側の周波数は1270.5Hzであり常に同期トーンと同じである。その周波数間隔は26.917nmHzでありR0,RRR,と73に対して $n = 2,3,4$ の値を持っている。そのショートハンドメッセージがQS0に関連してくるまで送信機と受信機のその周波数オフセットは高い精度で常に計測されている。この結果、これらのメッセージはその局のコールサインが事前に解読されているかしっかりとオペレータにより識別されていることができる。正確な時間同期はショートハンドメッセージでは必要は無く、だからそれらは送信を通じて何時でもスタートすることが可能である。

今の時点で、JT65はモールスコードのように文字によるメッセージを送信するのではないと言うことをハッキリさせるべきである。代わりに、全体のメッセージを72ビットの単一の記号列に置き換えている。これらの記号列を63組 - 6ビットシンボルのシーケンスにしている。これらのシンボルはラジオチャンネルで送信される：それらのある部分は完全に受信されるが、その他はノイズによって破壊される。もしそのシンボルが十分であれば(確率的に設定した範囲内であれば)正しく、完全な72ビットに圧縮されたメッセージは正確に回復させることができる。解読されたビットはその時人が読めるようにメッセージを送ったものに翻訳される。この符号化技術と強力なFECはメッセージが決してバラバラに受信されないように保証する。メッセージのコンポーネントはお互いの間では間違えることは出来ない。コールサインは誤って、もしくは数文字失われて表示されることは無い。コールサインの文字O又はRに対してシグナルリポートまたは了承などを混同するようなチャンスはない。N8CQまたはEM73のようなグリッドロケータになるような誤った解読のチャンスも無い。もし貴方がSKDしたパートナーのコールサインが示されず逆に他の局のコールサインがあれば、あなたは意図したようにスケジュールが保たれてい

るという結論は決してしないだろう。

9.Reception and Demodulation (受信と復調)

WSJTにより受信したJT65信号はベースバンドに変換されて良く知られたDSP技術のシーケンスを使用して分析される。このプロセスはおよそ0-3kHz範囲のオーディオ信号を11025サンプル/sのデジタル化した信号を用いて始まる。

デジタル信号はローパスフィルターされて、数2で割られることによりサンプル数を下げる。パワースペクトルは2048-サンプルのブロックからディスクリットフーリエ変換を用いて計算され、そして擬似ランダム同期パターンをテストする。同期パターンからのピークを見つけて要求された周波数と時間誤差を確立する。それはドップラシフトとEME通信の遅延を含み、さらに周波数較正と時間セッティングの誤差が含まれる。その同期を確立するための精度は周波数においては1.5Hz付近であり時間精度は0.03秒である。一度同期が確立されると、プログラムは小さなグループのトーン間隔上を再測定する。滑らかなカーブをその結果にフィットさせる。このため、小さな周波数の変動の追跡と補正が可能となる。シンボル間の位相合否の追跡は要求されない。

正確な同期情報の支配下に、プログラムは63チャンネルのシンボル毎に64 - ビンスペクトルを計算する。これらのスペクトルは分解能2.7mHz (即ちサブモードJT65Bでは $m = 2$ なので5.4Hzとなる)であり、非常に弱い信号にたいして、それらは本質的にノイズのように見える。

個々のデータトーンはそのノイズ上では見つけることが出来ないかもしれない。

平均上では、しかしながら、各々のトーン間隔のなかに信号を含んだひとつの周波数ピンは他よりも大きな増幅度を持っている。統計的な既知のランダムガウスノイズ分布を使用してWSJTは送信された各々の可能値を持つシンボルの確率を計算する。この確率情報は同期したシンボル数のスペクトル計測に基づいており、受信した情報の基礎となる。グレーコード化し、シンボルをインタリーブ(データの配置を変えてアクセスを高速にする)し後に削除される。その確率データは解読器に送られる。

10.Reed Solomon Decoder (リードソロモン解読器)

(RS63, 12)のような小さなエラー訂正コードでも "invert" 逆にするか又は符号化効率を上げるのは非常に困難である。基本的な問題がこれである: 63チャンネルのシンボルの各々に対して測定したスペクトルを与えることは、ユーザーメッセージとして確実に識別することが出来る唯一の72ビットのシーケンスがあるだろうか? 原理的に、1つは受信したスペクトルに対してその結果を関連させるために 2^{72} の符号化したユーザーメッセージとの組合せの照合を行わねばならない。そのようなアプローチは全く不可能である。しかしながら: 現在の3GHzのコンピュータと制限の無いメモリーと非常に効率的なプログラムを用いても、この方法で受信した1つのメッセージを解読するためにはおよそ2億年の時間がかかってしまう。

リードソロモンコードは経済的に重要であり、それはメッセージを解読するアルゴリズムが数学的に非常に良く定義されているからである。このアルゴリズムは複雑性を変化させてさらにこの方式の理想的な感度を追求するように動作する。プログラムバージョン4.5以降のWSJTはリードソロモン解読方式に最新式のアルゴリズムを採用している。それはRalf Koetterと Alexander Vardy⁸の論文に論拠している。この使用しているプログラムは彼等の会社である

CodeVector Technologiesからライセンスを受けている。受信されたシンボル値のソフト解決統計情報を与えて、この解読器は各々の分析された伝達情報に対してハッキリした結果を出すことが出来る。非常に高い信頼性を持って、それは72ビットの送信されたメッセージか、または結果の無いフラグの何れかに変える。

WSJT解読器のエラーレートは信号レベルの関数として注意深く計測されている。この結果はAppendix C.に要約している。簡単に要約すると、K-Vデコーダ(解読器)は“ほぼ常に解読出来る”から“殆ど解読に失敗する”までカーブが急激に変化することを示しており、この間の信号対雑音比はWSJTのスケールで示すとおよそ-23から-25dB (JT65B)に減少する。この結果の“クリーンデータ”(ガウス分布のノイズを加えて、フェーディングも含み、他の信号の混信はないものとする)を更に示し、RS(63,12)のアルゴリズムを解読しているK-V曲線からエラーレートを求めることは非常に稀であり、あなたも見たことが無いと思われる。

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.

11. Deep-Search Decoder

(ディープサーチ解読器)

何がK-V解読器の解読に失敗させたのか？何か更に出来ることはないのか？ 2^{72} の組合を実行してユーザーメッセージを解読するには時間がかかりすぎて人生はあまりにも短いことに気が付くことになる。しかしながら実際のEME QSOで送信されるメッセージ数は 2^{72} よりも非常に小さく貴方が思っているよりも更に小さい。もしもっともらしく興味あるメッセージが最初にテストされるならば-多かれ少なかれ人が非常に弱いCW信号をコピーするのと同じ方法である - そしてサーチアルゴリズムが “時間切れ” を実行したならば、もし実行時間が過ぎた後に組み合わせがないということが分かったならば上記のような強制的なアプローチが事実上行われる。WSJTでは、丁度これを行おうとするアルゴリズムが “Deep search” と呼ばれる手順である。このディープサーチはもっともらしいコールサインとグリッドロケータのリストを使用してスタートする。このようなリストは長いあいだ殆どのEMEオペレータによって記憶または記録によって維持されている。これらはどの局が、微弱なCQをだしているのか、あなたのCQに答えているのか、貴方のQSOの最終に答えているのかを決定する時に大きな助けとなる。WSJTディープサーチ解読器において、エントリーした個々のリストは “CQ” とWSJTユーザーのホームコールサインのペアであり、それによって仮に作られてテスト用のメッセージである。もしNcコールがそのリスト内に存在するならば、およそ2Ncメッセージを発生させてこれを完全に符号化する。そしてそのチャンネルシンボルが受信したスペクトルと良く一致するかをテストする。貴方がともかく選んだそれらしいコールサインのリストから(その局を)言い当てることが出来る。WSJTでも実例ファイルを用意しており、VHF/UHFバンドでweak signalの通信で活動している世界中のほぼ5000局のコールサインを網羅している。良く知られているJT65ユーザーはかれら自身のファイルを持っており、必要に応じてコールサインに加えるか、削除するかメンテナンスをしている。効果があるのは、貴方のコールサインが組合せフィルターのセットを定義する。そしてそれは貴方にあった設定になっており、貴方が受信を期待するメッセージの集合に感度が最高になるように調整している。このディープサーチはデータベースに無いコールサインを含んだメッセージまたは平文、CQの傍に何かあるもの、またはファーストメッセージ欄のあなたのコールサインに対しては感度が高くない。そのようなメッセージ類はK-Vアルゴリズムの注目に値する感度で解読することができる。しかしながら、先に定義した少集合以内の如何なるメッセージにはディープサーチ解読器が低いエラーレートを保っている間では4dB

以上の感度が得られる。これらの4 dBの感度は“スケジュールゲイン”として広く認められているものと等価であるということは明白である。これはCWオペレータが前もってコンタクトをアレンジしているか、または良く知っているコールサインをコピーしている時と同じような経験をすることが出来るものと同じである。

12. Decoding Shorthand Messages (短縮メッセージの解読)

さらに、期待される擬似ランダムパターンを持った同期信号を探すために、WSJTはJT65の設定されたショートハンドメッセージを持つ信号が変化するのがサーチする。幾つもの周波数が計測されて先に送信された同期信号の中身を比較する。その変調波が決められた矩形波サイクルに従う変調波を確かめるテストを行う。もしその周波数類と変調波が一致すればそしてもしその増幅度が予めセットした閾値を超えれば、あるショートハンドメッセージが決定されることになる。近接した周波数と時間の許容度の理由から、非常に低い解読不良率を維持している間は更に低い検出限界をセットすることが可能である。ショートハンドメッセージにたいする計測された感度曲線はAppendix Cのなかにあり、K-Vアルゴリズムとディープサーチ解読器に関する情報も含んでいる。

13. Operator Responsibilities and Message Integrity

(オペレータの責任とメッセージの完全性)

JT65を含むWSJTモードのQS0は全ての段階での活動的なユーザーの参加を要求する。パーティ、QRM, QRNまたは、マルチパスによる信号の歪み発生などの変則的な事態の中でオペレータを巻き込むことがプログラムの出力の誤りを避けるために必要である。殆どのオペレータは、JT65のQS0を初めの数回行うことで必要な熟練を容易に獲得していることが分かる。§ 2 で述べているように有効なQS0を行うためのガイドラインを見ると、JT65の特別の特性に言及したところを参考にすると良い。WSJTによるコンタクトは本質的にその内容を記録している。JT65QS0が成功裏に完結した時に、両者は必須の情報が交換されていることを知っている。さらにもし望むならばその記録されたWAVEファイルを見ればそれを証明することが出来る。これらのファイルは誰でもそれを証明ができるように、その完全なコピーはビット追跡が可能ないように記録されている。特に興味があるか、困難であったQS0に対しては記録されたWAVEFORMとスクリーンイメージはEメールによってしばしば交換が行われている。私は私自身のQS0やそのバンドのモニターから、または他人が私に数多く送ってくれたデータからJT65のWAVEファイルを数多く蓄積してライブラリーを作っている。これらのファイルはWSJTの感度を最適化したり、エラーレートを最小限にするためのアルゴリズムの検討のために実際のコンディションデータが非常に役立っていることが証明できている。更なる発展はこの分野で、数年以内にやってくることは確かである。

14. On-the - Air Experience

(QS0での経験)

JT65の初めての实用バージョンは2003年11月に完成した。初期のQS0テストをN3FZと実施したがJT65はVFH/UHF Weak-signalの愛好家の新しい大きな武器になるということを直ちに確信した。エラー訂正コードの実利はweak-signalのアマチュア無線通信にたいして非常に明白である。NASAの厳しい情報の符号化と強力なFECを用いて深遠な宇宙の写真を地球に送ってくるということに、殆ど驚きも無く実感している。深遠な宇宙通信において、1dBでも感度を改善することは数100万ドルのコストを削減することが出来る。これは大きなアンテナと送信パワーに莫大な費用がかかっているからである。

JT65プロトコルの仕様は最初の開発段階からマイナーな方法で発展してきている。その間に、解読器は堅実に改良されてQS0能力の相当大きな進展を見た。どの位多くのEME QS0がJT65を用

いて実行されたか知る由も無いがその回数は数千回を上回るのは確かである。ユーザーはプログラムのバグや操作上の改良の報告に躊躇しないのでWSJTはそのようなフィードバックによって大きな利益を受けている。新しいEME愛好家のグループが多数発生した。これはJT65 QSOは従来の方式が要求する設備よりも非常に控えめの設備によって遂行できること分かったからである。数100のJT65 EME QSOは2mバンドで、150wの送信電力とシングル八木アンテナによって作られている。そして“ピックガン”と呼ばれるステーションは送信電力5wでEME通信が出来ている。多分50MHzでのEME QSOは長い間最も困難な離れ技と考えられてきたが、通常の通信となった。

15.Looking Ahead 将来への展望

JT65の技術仕様の拡大または大きな改訂の必要性についての将来の見通しは持っていない。しかしながら、JT65の遂行能力についての改良するかもしれない多くの方法を考えることは出来る。スタートとして、受信したオーディオデータは全体の受信時間が完了してからのパッチモードよりも、受信しながら適宜処理すべきだと考える。これはリアルタイム処理のディスプレイが必要になる。そしてこれは受信したデータを20ないし30秒後に信号の早い解読が必要になると考えている。あるサウンドカードはそれらのサンプリングレートの誤差が0.6%の大きさを持つということが分かった。WSJTで使用している現在のJT65解読器はそのようなエラーを訂正する機能は無く感度が不必要に影響を受ける。信号を見つけることと混信の抑圧は巧くいつている。必要な信号の周波数変動を追跡するためのアルゴリズムは今のところ改善することが出来る。特に432と1296MHzにおいてドップラーが起こす周波数の変化をしっかりと追跡できることは確かに望ましい。送信/受信シーケンスのもっと正確なタイミングコントロールは必要と思われWindowsのもとでも可能かもしれない。解読手順の処理スピードは改善が出来る...そのリストが相当数ある。たぶんその他はこれらのある改善またはもっとハッキリした改良の考えなどをチャレンジとして取り上げることになるだろう。

Appendix A: Detail of Message Encoding 付録A：メッセージ符号化の詳細

幾つかの段階で発生するJT65メッセージの符号化は§4-6で説明している。ユーザーメッセージは初めに符号化され72ビット小さな集団になる。このビット集合は12-6ビット情報シンボルに詰め込まれる。そしてリードソロモンコードの51パリティのシンボルが加えられる。63チャンネルのシンボルがインタリーブ（並び替えられて）され、グレーコード化され、64-トーンのFSKによって送信される。同期ベクトルは65番目の周波数で送られる。2つのトーン間隔は最も低いデータトーンよりも低い。

ある任意の選択として、メッセージをパッキングしチャンネルシンボルに並べる。これらの更なる詳細を示す。JT65を実行する内容を簡単に理解するために、実際のソースコード例で示すのがベストである。次に付け加えたのはフォートランプログラムでありLinux下で簡単にコンパイルすることが出来る。ここではメインプログラムをリストしている；フルソースコードであり必要なサブルーチンを含んでいる。そしてLinuxでファイルを作ることが出来る。これは pulsar.Princeton.edu/~joe/K1JT/JT65code.tgz からダウンロードが出来る。コンパイルしたプログラムはJT65メッセージ(コマンドライン上にエンコードして引用している)を受け付ける。そしてバックしたメッセージに答える。チャンネルシンボルは6ビット値である。プログラム出力例としてFig.3に示し§5で説明している。

```

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: Support Callsign Prefixes and Suffixes

コールサインのプレフィックスとサフィックスはJT65によってサポートされている。そのリストはAppendix Aの中で紹介しているように、ソースコードアーカイブ pulsar.Princeton.edu/~joe/K1JT/JT65code.tgz 中のファイルpfd.fにある。サポートしているサフィックスは/Pと/Oから/9を含み、プレフィックスリストは下記に示している。付加的なプレフィックスとサフィックスは将来このリストに加えることが出来る。北極の5°以内の如何なるグリッドロケータはサポートしていない代わりに450個のプレフィックスが用意されている。

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	PO
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

(計測された感度とエラーレートの関係)

JT65のプロトコルは全てに対して1回で定義することが出来る。通信上の処理能力は特に復調器のソフトウェア実行能力によって決まる。WSJTバージョン4.9の§9-12で概略を説明しているがJT65の解読については3つの段階に分けられる：ソフト処理によるリードソロモン解読器、ディープサーチ解読器、そしてショートハンドメッセージの解読器である。§13で強調しているのは、バーディ、空電雑音、その他の混信、そしてオペレータの介在が解読過程の本質的な部分である。オペレータは“Zap”を有効にするとバーディを取り除くことが出来る。“Clip”機能は広い帯域にわたりスパイク性ノイズを低減することが出来る。“Freeze”は同期信号のサーチ範囲を制限する。このような支援機能とプログラムのグラフィックスおよび数学的な表示などを使用することによって解読器からのスプリウス出力を捨てて本物を抽出することが出来る。

送信周波数に、単純な減衰、ホワイトガウシアンノイズを加えて、レイレイスフェージング係数を掛けることによって通常の状態を作り出し、その時の感度と解読器のエラーレートを実際に計測することが出来る。このような方法によるシュミレータのソフトウェアはLinuxベースで作られており、初期段階(非常に本質的な部分)のWSJTプログラムを発展させた。このシュミレータはWSJTのすべてのデジタル信号を発生させることが出来る。そしてバンドを制限した

ガウシアンノイズと、これにS/Nを設定して特性を決めて、そしてフェーディングを加味したこれらの環境の中にWSJTの信号を挿入している。この結果のオーディオファイルはWAVフォーマットで記録することが出来る。その時ファイルをオープンしてWSJTを解読することが出来る。それらはまたシュミレータ内で直接解読が可能である。使用するコードはLinuxでコンパイルされるがWSJTと完全に一致する。

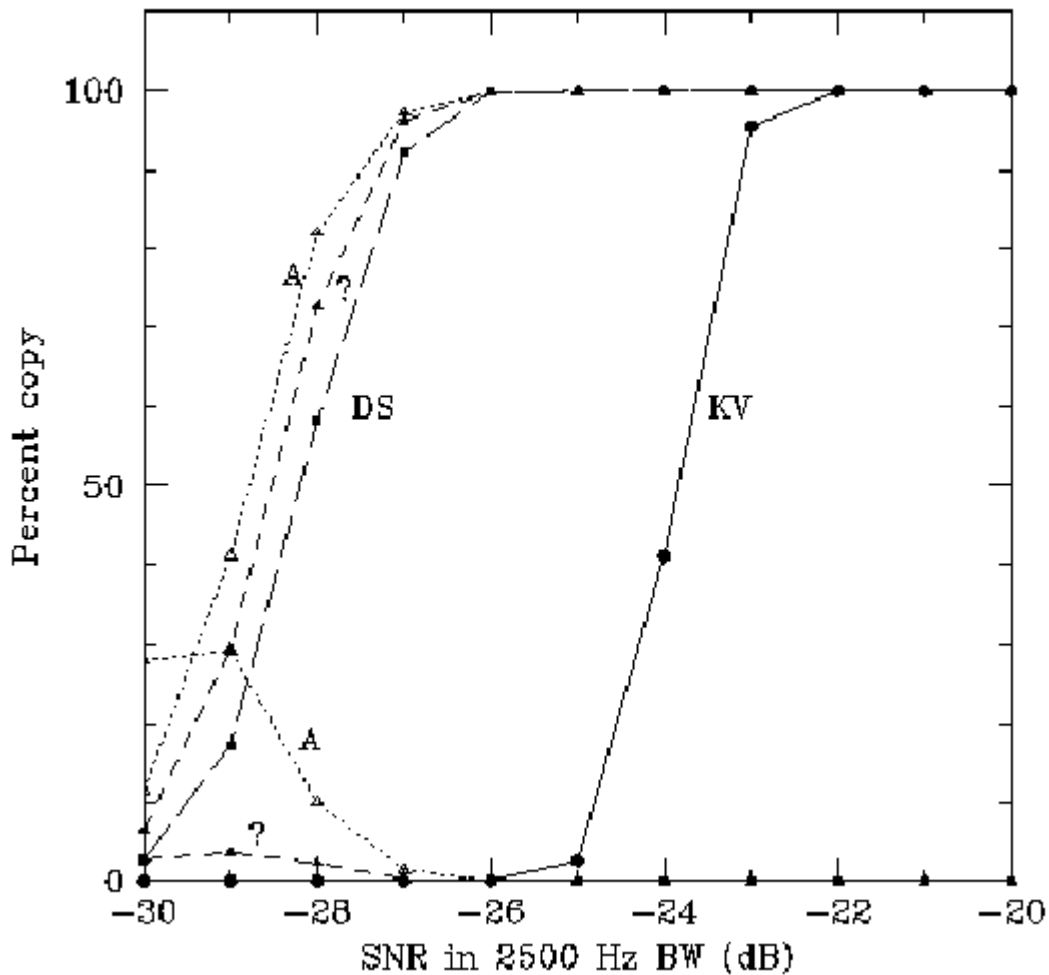


Fig. 4-JT65Bに対するSNR特性を計測した結果を示している。KVラベルの曲線はKoetter-Vardyアルゴリズムを示し、DSはディープサーチアルゴリズムの結果を示している。KVアルゴリズムの誤解読のレートは計測するためには非常に小さすぎる。DSアルゴリズムに対しては“ハードエラー”レートは0.03%であり、グラフに示すにはあまりにも小さすぎる。“?”と“A”が付いた曲線は解読メッセージに?がある場合のディープサーチのソフトエラーレートを示し、図中の左下に示す。Aは積極的な解読がリクエストされたことを示している。

数十万回のJT65の送信についてのシュミレーションがこのような方法で行われている。前段ではデバックと解読器のファインチューニングのために、後段では完成したプログラムの感度とエラーレートを計測するために行われた。シュミレーションの結果はFig. 4と5に要約されている。

る。Fig.4を求めるために1000回ほど送信が繰返されて、SNRのレベルを-30、-29、...-20dBのレベルで行われ、2つのコールサインとグリッドロケータを持った標準JT65メッセージを用いた。全てのWSJT解読器 (Versionは4.9) は11,000回夫々テストランが行われた。Fig.4の中の実線と丸ドットの曲線はKoetter-Vardy解読器の結果を图示している。本質的な結論として96%の正解解読率時のSNRは-23dBであり、41%では-24dBになり3%では-25dBとなる。誤りのない解読はKV解読器を用いることで如何なるテストでも再現が可能であった。

ディープサーチアルゴリズムについては、四角い点と長い点線を用いて曲線を示している。92%の解読率では-27dBであり、58%では-28dB、17%では-29%となる。3つの“ハードエラー” (? フラグの無い解読エラー) は11,000回の送信テストの中ではおよそ 2.7×10^{-4} (Fig.4で示すには非常に小さい) であった。もし1つが ? マークを持ったメッセージが解読されると、そのナンバーに対して正しいコピー率は93%、73%および29%とでは-27、-28、-29dB (印と短い点線の曲線) と増加する。WSJTの“ Aggressive decoding ”を選択すると解読率が向上して、97%、82%および41%が-27、-28、-29dB (印と短い点線の曲線) となる。しかしながら解読不良率も増加し、特に-28dBとそれ以下-29dBでは29%となる。

同様の計測はサブモードであるJT65AとCでも同様である。その結果はFig.4に示したようにJT65Bと同じ傾向である。JT65Aに対しては約1dBほど左にシフト (JT65Bより感度が良い) しておりJT65Cに対しては約1dB右にシフトしている。

JT86の平文はその同期ベクトルがはっきり見つけられないと解読ができない。WSJTにおける同期手順はサブモードJT65A,BとCでは全く同一である。Fig.4で図解しているようにSNRが-29dBよりも小さい時には同期不良が解読作業の多くの失敗の原因になっている。同期を掛けることは他の理由でも非常に重要である。同期を正しくすることは蓄積された平均メッセージを解読するのが可能になる。送信されたメッセージがディープサーチアルゴリズムで解読可能であるかは無関係である。Fig.5では同期レートが計測されている。SNRが10dBレンジを越える値を用いてシュミレーションのための送信を1000回繰返した。同期が93%達成ではSNRは-28dB、74%で-29dB、44%で-30dBと19%では-31dBとなった。これらの計測結果から、メッセージを平均することは例えば3回の送信を平均すると-26dBになり、8回の送信では-28dBになり、20回となれば-29dBとなる。これらの結論はWSJTのQSO経験からも一致している。

シュミレータはJT65ショートハンドメッセージの解読レートの計測にも使用された。これもFig.5で図解している。各々のSNRで1000回繰返された。ショートハンドメッセージは88%の解読率ではSNRは-31dBとなり、60%で-32dBそして26%では-33dBとなる。1100回の試行でショートハンドメッセージの解読が正しくなかったのは5回であった。5回すべては注意深いオペレータによってスプリアスによるものとして判断された。原因としては計測された周波数のズレが通常の許容範囲よりも大きかった。

他にも多くの理由がありJT65のQSOの遂行能力はここで示したシュミレーションの結果とは幾分異なるかもしれない。Fig.4と5で示した計測結果は付加したホワイトガウシアンノイズ (AWGN) とフェーディング無しの理想的な条件下で行われている。(付加的なシュミレーションとしてはレイレイスフェーディングの効果が追加されるが、その結果は同様な傾向になり、その曲線は数dB右へシフトされる)。バーディの影響、その他の混信そして ノン・ガウシアンノイズは数量化するのが難しい。私はしばしばWSJTをモニターモードにして時には数日間そのままにしておく私の受信機は144.100から144.160の間を任意に受信する。私は人口の多い地域に住んでおり、2mバンドでは他の信号と同様に数多くのバーディが現れたり続いたりしている。典型的な誤解読レートはバンドが静かな時は平均して1時間に1回ないしは2回よりは多くは無い。

スプリアスを解読したファイルを検証してみると常に背の高いパルスがその付近に見えるのでオペレータは有効でない情報を見つけ出して避けることができる。このように意図した方法で使用するときに、WSJTは高く正確な通信プロトコルとなる。

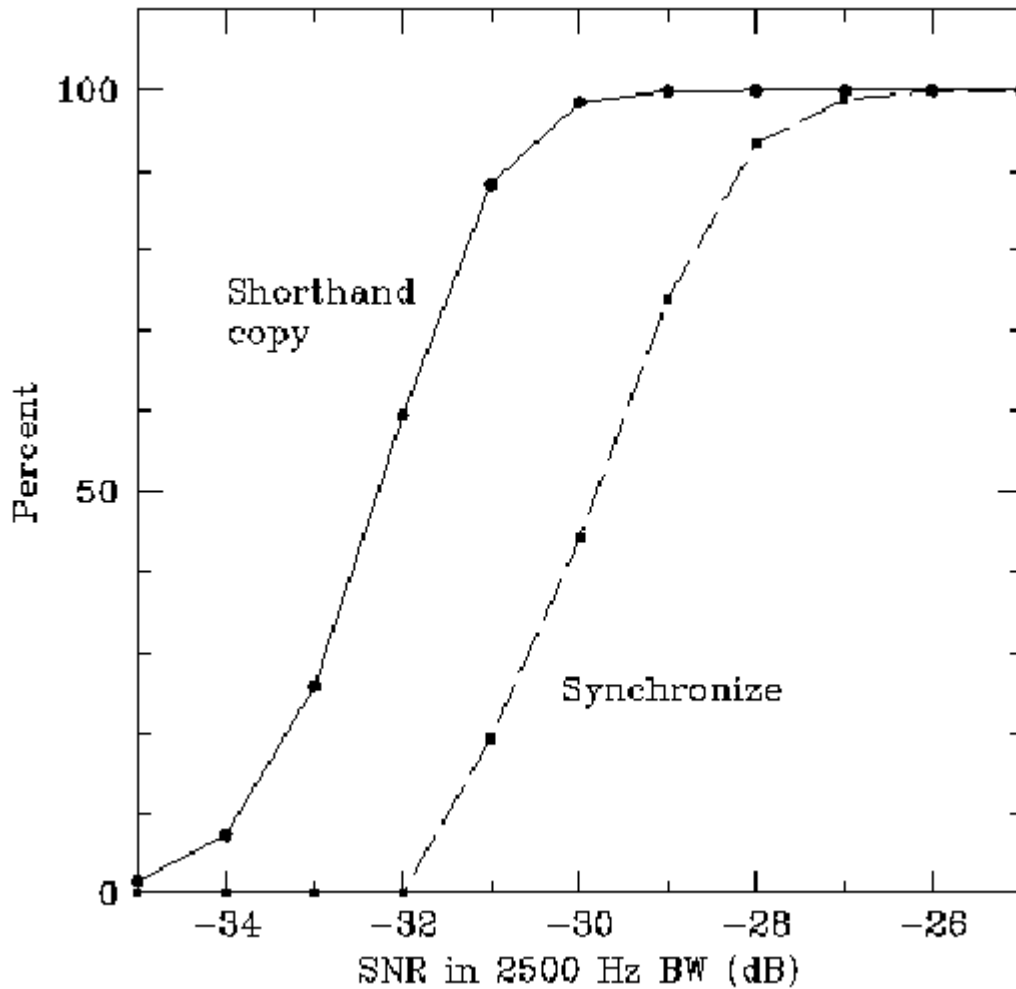


Fig.5 SNRを関数としてのメッセージの同期とショートハンドメッセージのコピーレート