

🍁 目次 🍁

🍁 アマチュア無線 3 級・4 級の点数が面白いほどとれる@富沢いずみ	2
🍁 アマチュア無線の交信@部員 K	6
🍁 東北大周辺のお手軽移動運用スポット@JP7VTF	11
🍁 自作ログソフトを作ってみた@部員 K	16
🍁 デジタル変調方式について@リー	19
🍁 誤り訂正符号について@JP7VTF	22
🍁 MATLAB を用いた WSPR の生成:通信路符号化の例@部員 W	26
🍁 編集後記	40

アマチュア無線、東北大アマチュア無線部とは？

🍁 アマチュア無線？

趣味で使うことのできる無線で、日本全国はもちろん世界中と通信することができます。運用方法などにも様々な方法があり、色々な楽しみ方が出来る無線趣味のこと。

🍁 東北大アマチュア無線部？

アマチュア無線の運用や各種大会出場、電子工作などを行っている学友会の部活動。

川内キャンパスと片平キャンパスの両方の部室を拠点に活動中です。

🍁 当部誌について

今年度から部の活動と部員の研究成果を共有、発表するために刊行を始めました。今号はまだ二号目、うまれたばかりの部誌です。バックナンバーは公式 HP で公開中なので是非そちらもご覧下さい。

アマチュア無線 3 級・4 級の点数が面白いほどとれる

富沢いずみ

1. はじめに

大学に入り何のサークルに入ろうかと悩んでいる時に本部誌やポスターを見て一念発起、アマチュア無線をやってみようと思ったときまず何を必要があるか……そう。まずは見学に来て下さい。と、まあ冗談はさておきアマチュア無線を始めようとした時に最初に必要になってくるのがアマチュア無線技士の免許です。アマチュア無線に使用される電波も限りのある資源のためそれを利用できるのは免許を持った人に限られて来るといわけです。似た例では自動車の運転免許が挙げられるかもしれませんね。

そんなわけで最初に必要になってくる免許ですが、これを取得するには基本的に試験を受け合格する必要があります。本稿ではそんな試験の概要と問題を出来るだけ簡潔に分かりやすく解説しアマチュア無線の世界の第一歩を踏み出す足掛かりになればと思います。なお本稿ではアマチュア無線 3 級を中心に扱います。また以降では 3 級アマチュア無線技士を 3 アマ、4 級アマチュア無線技士を 4 アマと略します。

2. アマチュア無線免許試験とは

はじめにで触れたようにアマチュア無線免許を取得するためには基本的には試験を受け合格しなければなりません。¹まずは試験の概要と受け方について解説します。当試験は日本無線協会の本部・支部ごとに年 4~12 回東京大阪名古屋に加え札幌仙台金沢広島松山熊本など各地域の代表的な都市にて開催しています。²試験問題は 4 択から選ぶ方式で内容は工学と法規の 2 部門に別れています。3 アマの場合は工学 14 問・法規 16 問の計 30 問で試験時間は 1 時間 10 分、4 アマの場合は工学・法規ともに 12 問の計 24 問で試験時間は 1 時間です。また合格点は 3 アマの場合工学の満点 70 点のうち 45 点、法規の満点 80 点のうち 55 点を取得することで合格になります。4 アマの場合は工学満点 60 点

¹ 正確には e ラーニング受講後修了試験を受ける、養成過程講習会を受講し修了試験を受けるなど別の方法もあるが、本稿では一番一般的な試験についてのみ触れる。

² 開催日、開催場所等については公式 HP を適宜確認して下さい。

のうち 40 点、法規 60 点のうち 40 点を取得することで合格になります。

受験には特定の受付機関に申請する必要があり、試験手数料と受験票送付料の計 5463 円 (3 アマ)もしくは 5163 円(4 アマ)を支払います。また受験時には送付されてきた受験票及び受験整理票と写真(前述したものに貼り付け)を持っていく必要があります。受験会場、申請方法、受験に必要な費用、試験方法などは変わる可能性がありますので受験前に公式 HP で確認下さい。

3. 試験対策

お待たせしました、本題の試験対策に入ります。本章では試験に出題された問題を元に勉強法と最小限学んでおくべきことについて解説していきます。なおこの勉強法や内容につきましてはあくまで個人の経験と見解に基づくものであることをご理解下さい。

3.1 法規

3 アマの場合法規の出題は 16 問かつ 80 点と大きな割合を持つため、ここで確実に点数を取ることが合格の重要な鍵になります。また内容も典型的なものが多いため条文の暗記と徹底的な過去問の演習が必要です。

内容は無線の設備・従事者・免許・書類関係、運用の規則や符号、モールス信号の約 3 つに分けることができます。

・設備・従事者・免許・書類

無線設備の決まりや無線従事者としての決まり、免許についての法令などが聞かれる内容です。過去問などで演習しながら適宜条文を確認し、「○のときは、△しなければならない」と繋がるように勉強しましょう。

・運用規則・符号

運用時の決まりや手順、用いる符号について聞かれる問題です。ここについては実際の運用で使用することが多い内容のため出来るだけ丁寧に覚えて使えるようにすべきです。符号については一覧を確認する、運用手順などがまとまったものを参考にするなどして覚えましょう。

・モールス信号

数字やアルファベットが含まれた文を正しいモールス信号にしたものを選ぶ問題のため消去法が活躍するところでもあります。全てのモールス信号に自信がない場合でも数字と

いくつかの符号さえ確実に覚えていれば正しい答えに近づくことが可能でしょう。そのためにも少なくとも0から9までの数字や特徴的で覚えやすい符号については必ず覚えておくべきです。(アルファベットについても語呂合わせや規則性を見つけるなどして出来るだけ覚えておくと楽になります)

3.2 工学

こちらは法規に比べると出題数は14問と少ないものの電気系の専門要素が含まれたり簡単な計算があったりと時間が掛かると思われます。理系、特に光学系で学んだ方は触れたことがある内容も多いと思いますが、確実に解ける問題を落とさないことに気をつけながら出来るだけ解けるようにする必要があります。内容は電気・回路、受信送信電源、アンテナ・電波・電波障害の約3つに別けることが出来ます。

・電気・回路

電圧、電流、抵抗、電力、回路素子、リアクタンスなどの電気の基礎に関わる問題が多いです。基礎的な公式、ダイオード・トランジスタなどの素子の名称などの暗記と理解、公式等を利用した計算を出来るようにおく必要があります。

・受信・送信

無線機の構成や波形などを問う問題が多いです。基本的な原理の理解と周波数などの計算を出来るようにする必要があります。

・アンテナ・電波・電波障害

アンテナの構造や原理、指向性を問う問題や電波伝搬の原理などを問う問題が多いです。良く用いられるアンテナの構造や指向性を図で確認するとともに電波伝搬の仕組みについて過去問や参考書を用いながら理解する必要があります。

3.3 勉強に役立つ本、サイト

法規、工学のどちらを勉強するにあたって過去問の存在は大きいです。出題形式の把握や理解不足、暗記不足の場所を洗い出すためなど勉強の中心になってくると思われるので出来れば一冊は手元においておきたいところです。なお問題については近年大きな変化はないと思われるため過去のものの中古で調達することも選択肢として考えられます。また要点マスターや技師試験テキスト、解説付き問題集などが販売されているためそれらを利用することでより理解した上での勉強が可能になると思います。工学分野については電気・電波系の参考書を勉強することでより理解が深まりますが3級突破が目標であればそ

ここまで高度な知識は要求されないと思います。

ネット上にも様々な試験勉強に役立つサイトが存在します。過去問を元に一問一答形式で何度も問題を解けるサイトなどは勉強の友となると思われます。

- ・ <http://srz.a.la9.jp/hamtest/>
- ・ <https://soft.taprix.org/web/mn2/ham3/>
- ・ <http://get-q.net/>

4. まとめ

本稿では3アマ(4アマ)の免許を取得する際の勉強に役立てればと筆者なりの試験問題の分析と勉強方法について掲載しました。筆者も約1週間の勉強で無事3アマ免許試験を合格することが出来たように基本的な内容の暗記と理解があれば簡単な試験であると思われるので是非この記事を参考に(しなくてもいいですが)免許を取得し、アマチュア無線の道を踏み出して下さい。

参考文献

公益財団法人日本無線協会 <http://www.nichimu.or.jp/index.html>

(閲覧 2020/09/10)

アマチュア無線の交信

部員 K

はじめに

アマチュア無線では毎日多くの方が、電波を使って交信を行っています。アマチュア無線の交信と聞いてどんな内容を話しているの？と疑問を持たれる方も多いのではないのでしょうか。今回は、通常交信とコンテストにおける交信の様子を紹介していきたいと思いますので、ぜひアマチュア無線に興味を持っていただければと思います。

交信例

通常交信の交信例を示します。交信の中で登場しているレポートや、様々なアマチュア無線で使われる言葉についても交信例のあとに解説しております。

自分：CQCQCQ こちらは、JA7YAA/7³ 宮城県石巻市上品山移動です。ご入感局おられましたらコールください。受信します。どうぞ。

相手：JA7YAA/7。こちらは、相手のコールサインです。どうぞ。

自分：相手のコールサイン。こちらは、JA7YAA/7⁴です。こんにちは、お声がけありがとうございます。レポートは 59 お送りいたします。大変強力に入感しています。QTH は宮城県石巻市です。JCC コード 0602 です。名前は、雀の「す」、雀の「す」に濁点、切手の「き」、鈴木と申します。相手のコールサイン。こちらは JA7YAA/7 です。どうぞ。

相手：JA7YAA/7。こちらは、相手のコールサインです。

(相手の RS レポート、QTH、QRA、天気、リグ、アンテナ、等の話題。)

お返しします。JA7YAA/7。こちらは、相手のコールサインです。どうぞ。

自分：相手のコールサイン。こちらは JA7YAA/7 です。

59 のレポートありがとうございます。こちらの天気は晴れです。リグは、IC-910 を使っております。

カード交換を、JARL ビューロー経由でされているようでしたら JARL ビューローにお返しします。

³ /はポータブルもしくはストロークと読む。

⁴ コールサインを確認しあったら、フォネティックコードを使う必要はない。

相手のコールサイン。こちらは JA7YAA/7 です。どうぞ。

相手：JA7YAA/7。こちらは、相手のコールサインです。

了解しました。天気、リグの紹介ありがとうございます。こちらの天気も晴れです。リグは、IC-7100M を使っております。

こちらからもカードは JARL ビューローにお返しします。

お返しします。JA7YAA/7。こちらは、相手のコールサインです。どうぞ。

自分：相手のコールサイン。こちらは JA7YAA/7 です。

カード交換ありがとうございました。また、聞こえておりましたら QSO お願いいたします。

今日は QSO ありがとうございました。73。失礼します。

相手：こちらこそありがとうございました。失礼します。

RS レポート

RS レポートとは上に示した交信例に示した 59 に当たります。相手の電波の受信の状況を相手に伝えるために使用します。

R 了解度(Readability)	
5	完全に了解できる
4	実用上困難なく了解できる
3	かなり困難だが了解できる
2	かろうじて了解できる
1	了解できない
S 信号強度(Signal Strength)	
9	きわめて強い信号
8	強い信号
7	かなり強い信号
6	適度な強さの信号
5	かなり適度な強さの信号
4	弱いが受信容易
3	弱い信号
2	大変弱い信号
1	微弱でかろうじて受信できる信号

フォネティックスコード

フォネティックコードはアルファベットを正確に相手に伝えるために使用し、主にコールサインをはじめに相手局に伝えるときなどに使用します。

A	アルファ	N	ノベンバー
B	ブラボー	O	オスカー
C	チャーリー	P	パパ
D	デルタ	Q	ケベック
E	エコー	R	ロミオ
F	フォックストロット	S	シアラ
G	ゴルフ	T	タンゴ
H	ホテル	U	ユニフォーム
I	インディア	V	ヴィクター
J	ジュリエット	W	ウィスキー
K	キロ	X	エックスレー
L	リマ	Y	ヤンキー
M	マイク	Z	ズールー

和文通話表

和文通話表は、日本語を正確に相手に伝えるために使用し、主に自分の名前や、QTHを相手に伝えるときなどに使用します。

朝日のア	いろはのイ	上野のウ	英語のエ	大阪のオ
為替のカ	切手のキ	クラブのク	景色のケ	子供のコ
桜のサ	新聞のシ	雀のス	世界のセ	そろばんのソ
煙草のタ	千鳥のチ	鶴亀のツ	手紙のテ	東京のト
名古屋のナ	日本のニ	沼津のヌ	ねずみのネ	野原のノ
はがきのハ	飛行機のヒ	富士山のフ	平和のヘ	保険のホ
マッチのマ	三笠のミ	無線のム	明治のメ	もみじのモ
大和のヤ		弓矢のユ		吉野のヨ
ラジオのラ	リンゴのリ	るすいのル	れんげのレ	ローマのロ
わらびのワ	尾張のヲ	おしまいのン		

コンテストとは

コンテストとは、規定の時間内に交信した局数に応じて得られるポイントに加えて、マルチプレイヤーというポイントを乗じた得点を競い合うアマチュア無線の競技であり、様々なコンテストが開催されており、弊部も毎年数回参加しています。

主なコンテストとして、4月のALL JA コンテスト、7月の6m and down コンテスト、8月のField Day コンテスト、10月の全市全郡コンテストなどがあります。コンテストは多くの局と交信する必要があるため、交信はコールサインとコンテストナンバーのみを交換し、素早く済ませます。

コンテストのルールと注意点

コンテストはいつでも開催されているわけではありません。開催日と時間が決まっていますので、調べて確認します。また、コンテストにより、部門、種目、コンテストナンバー、マルチプレイヤーの計算方法が異なりますので、注意が必要です。

コンテストの交信において交換するものは2つです。コールサインとコンテストナンバーです。交信例を以下に示しますが、このような形で、素早く交信を済ませることで、多くの局数を稼ぐのがポイントです。

ここで、注意点があります。コンテストによっては、ある一部の地域との交信しか得点にならないことがあります。コンテストで使用周波数も決まっていますので、注意してください。また、同一バンドで、同じ局と交信しないようにすることにも注意します。なぜならば、同じ局と同じバンドで重複交信をしても得点にならず、もはや時間の無駄となってしまうからです。コンテストは時間が限られており、重複交信は相手局の迷惑にもなってしまうためしないようにしてください。

コンテストの得点は、交信数にマルチプレイヤーというポイント乗じた得点になります。この計算方法から、いかにマルチを稼ぐかということが重要になるかがわかるかと思えます。

交信例

コンテストにおける交信例を示します。

自局：CQ コンテスト CQ コンテスト JA7YAA/7 コンテスト どうぞ

相手局：相手のコールサイン どうぞ

自局：相手のコールサイン こちらから 5907M どうぞ

相手局：5907M 確認しました こちらから 5901L どうぞ

自局：5901L 確認しました ありがとうございます

相手局：ありがとうございます

まとめ

弊部では毎年数回、コンテストに参加しています。特に、6m and down コンテストはOBの方の力も借りながら運用しています。ぜひ、弊部のメンバーとアマチュア無線の交信の面白さを感じていただければと思います。

東北大周辺のお手軽移動運用スポット

JP7VTF

1. 初めに

アマチュア無線にはさまざまな楽しみ方がありますが、最もポピュラーな楽しみ方は他の無線局との**交信**でしょう。せっかく交信するのなら、なるべく遠くまで電波を飛ばし、どれだけ遠くにある無線局と交信できるかチャレンジしてみたいものです。遠くまで電波を飛ばす方法にはさまざまありますが、手っ取り早く遠くまで電波を飛ばす方法に、**電波を遠くまで飛ばしやすい高台や見通しの開けた場所に自分が移動する**という方法があります。これをアマチュア無線では「**移動運用**」と呼ばれます。今回は、東北大の周辺で手軽に移動運用ができるスポットをいくつか紹介します。

2. 移動運用時の持ち物・注意点

2.1 運用をするための最低限の持ち物

無線機

144 MHz/430 MHz 帯のハンディ機(2万円～)があるとよいでしょう。

ペン、ノート、時計

相手の無線局のコールサイン、交信日時、交信内容などを記録します。

無線従事者免許証、身分証明書

場合によってはお巡りさんと遭遇する可能性があります。悪いことをしていないことを証明するためにも、必ず所持しておきましょう。

2.2 注意点

以下に紹介する場所には、観光客や散歩、散策目的の一般の人たちも来ます。そのため、他の人たちの迷惑にならないように運用することを心がける必要があります。

3. 移動運用スポットの紹介

3.1 八木山動物公園駅 てっぺんひろば

仙台駅から東北大学川内キャンパス・青葉山キャンパスに行くには、地下鉄東西線が便利です。仙台駅から乗ると、川内駅→青葉山駅の次に東西線終着駅の八木山動物公園駅が

あります(図1)。この駅にある立体駐車場の屋上には「てっぺんひろば」という広場があります(図2)。ここからは仙台の南側を一望することができます(図3)。この場所は、駅から動物公園に行くお客さんの人通り一定数あるので、絶対に目立たないように端っこでゲリラ的に運用するのがよいでしょう。



図1 八木山動物公園駅の位置



図2 てっぺんひろばと八木山動物公園



図3 てっぺんひろばから見た仙南のパノラマ写真

3.2 大年寺山公園

東北大学片平キャンパスから八木山方向に行ったところに大年寺山があります(図4)。駐車場があるので車やバイクでのアクセスが便利です。大年寺山には地上デジタル放送用の送信アンテナが3本生えており、仙台市随一の電波スポットです(図5)。これらに加えて仙台藩主伊達家の墓所である無尽灯廟があります。運用はこの裏側の仙台市中心街を見渡せる場所から行くとよいでしょう(図6)。北は利府町や大崎市、南は巨理町と交信できました。



図4 大年寺山の位置



図5 地上デジタル放送の送信アンテナ



図6 仙台市中心街と筆者の無線機

3.3 太白山山頂

八木山動物公園駅から4 kmほど南下していくと、標高320 mの太白山があります(図7、図8)。太白山はふもとから山頂まで1時間程度で登ることができます。登山の難易度は高くないため、気軽に(準備は必要ですが)登ることができます。頂上直前の急登を登る際にはグリップの効くトレッキングシューズがあると便利です。頂上からは仙南を見渡すことができます(図9)。私が運用をした際には仙南に限らず仙北(泉区)の方とも交信できました。



図7 太白山の位置



図8 太白山

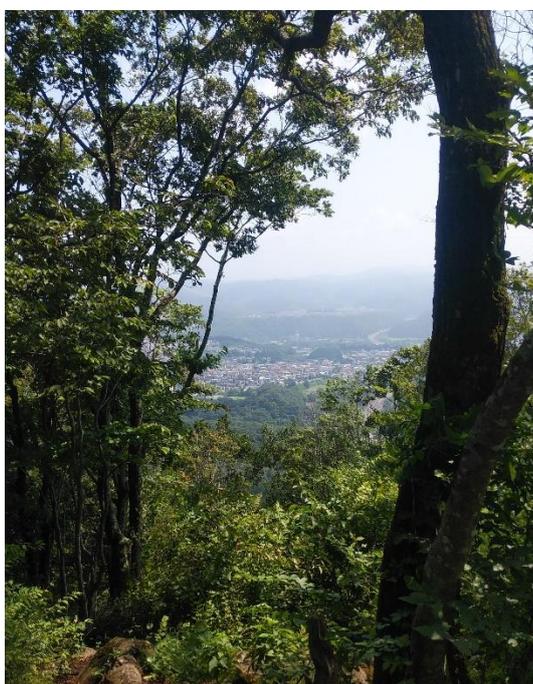


図9 山頂からの眺め

3.4 青葉山キャンパス

青葉山キャンパスがある青葉山も熊さんが出没するくらいには山です。今回は農学部(図10)の先、ユニバーシティ青葉山の近くをご紹介します(図11)。おそらく、東北大の敷地内の中では比較的高い場所なのではないでしょうか?ただ、ここはイマイチのような

気がします。仙台市役所付近の局と通信できましたが、先に紹介した場所に比べると電波の飛びは悪いようです(紹介するほどでもないのに地図も載せていない)。



図 10 農学部の位置



図 11 運用場所でのパノラマ写真

4.まとめ

本稿では東北大周辺の手軽な移動運用場所について紹介しました。お散歩・観光がてら訪れてみると、楽しいのではないのでしょうか。また、移動運用場所を自分で探してみるのも面白いかと思います。これらを通して、自分の知らなかった仙台を知ることができたと筆者は感じています。

自作ログソフトを作ってみた

部員 K

はじめに

ログソフトとはアマチュア無線の交信を記録するために使われているソフトウェアのことです。基本的には、交信した局のコールサインや RS レポート、相手局の様々な情報を入力し保存し、表示する、編集するといった機能が実装されています。現在も様々なログソフトが使われており、通常交信用やコンテスト用などさまざまなソフトウェアがあり、多くのソフトウェアが無料でダウンロードして使うことができます。今回は、そのようなログソフトウェアを自作してみたのでまとめたいと思います。

試作

私自身、工学部でもなく、大学の授業では、情報基礎で少しだけ、C 言語について学習し、趣味でプログラミングをしているくらいで、GUI のソフトウェアについては作ったことがありませんでした。多くの、GUI のソフトウェアは visual studio などを使って作られていると思うが、今回は php という言語を使ってソフトウェアを作成してみました。php は主にウェブ制作において使われているプログラミング言語ですが、phpdesktop というソフトウェアによりデスクトップソフトウェアと似たように使用できるようなのでそのような形で制作を進めてみました。php を使った理由としては、GUI のソフトウェアが他の言語に比べ容易に作れるといわれていることや、これまでにウェブアプリケーションを作る際に使ったことがあるからです。

実装した機能としては、ログの入力と保存、保存したデータの表示と編集、様々な形式への書き出し、過去のログデータの検索と表示です。またコンテスト用のソフトウェアにはマルチや得点を集計する機能を実装しました。

初めに制作に要した時間は 1 日程で、その後、気の向いたときに時々改良をし 1 カ月ほどで現在のような形になりました。以下に制作したソフトウェアの画面を示します。

K KLOG

Year Month Day Time Callsign HisRS 59 MyRS 59

Mode Freq PWR Name QTH Code

Remarks Remarks2 OP QSL J

[ログの表示](#)
[カード](#)
[ハムログ](#)
[MMQSL](#)
[ADIF](#)

図 1 通常交信用のソフトウェアの画面の様子

Callsign にコールサインを入力し、登録をクリックすれば、コールサインの検索結果を表示するとともに、日時が入力されるようにしました。ログの表示をクリックすれば、新しいウィンドウが開き、データが表示されます。

K KLOG(contest)

Call RCVD 59 SENT 59 Mode Freq 1.9 ▼ MEMO OP

ログの表示 1.9 ▼

[合計点数](#)
[データの結合](#)
[提出用ログ形式](#)
[MMQSL](#)

図 2 コンテスト用ソフトウェアの画面の様子

コンテスト用のソフトウェアについては、コンテストによって、マルチや得点の集計方法が異なるため、コンテストごとにプログラムを変更することで対応する必要があります。多くのソフトウェアはすでに様々なコンテストの集計方法がプログラミングされており、インストール後に選択できるようになっているが、私の自作したソフトウェアでは書き換えが必要であり、その点が面倒な点であると思っています。

実際に使ってみた

8月に行われた、フィールドデーコンテストにおいて、コンテスト用のソフトウェアを使用しました。その際の感想をまとめたいと思います。まず、使用して、1点問題が発生しました。コンテストにおいて、コンテストナンバーというものを交換することによって、そのナンバーによってマルチが発生するが、マルチを集計するプログラムが間違っておりうまく集計されませんでした。それ以外の機能は正常に機能しました。ただ、周波数の入力のミスや、保存し忘れなどがあり、注意しなければならないと思いました。

ログデータの提出については提出用ログ形式から出力しておき、合計点数を事前に出しておき、その後、JARL ホームページにある、JARL コンテスト 電子ログサマリー作成ページのツールを利用し、ログを提出しました。JARL のツールを使わなくても、提出用ログを出力できるように今後改良したいと考えています。

まとめ

ログソフトの自作の様子についてまとめてみました。自分で使いやすいように改良できることが自作の最大のメリットだと思います。まだまだ、基本的な機能しかなく、一般的に使われているログソフトには全然勝てませんが、今後も改良をしていきたいと思ます。

参考文献

PHP とはなんでしょう? <https://www.php.net/manual/ja/intro-what-is.php> (閲覧 2020/09/02)

PHP Desktop <https://github.com/cztomczak/phpdesktop> (閲覧 2020/09/02)

デジタル変調方式について

リー

1. デジタルによる変調の利点とは

アナログ変調は、送りたい信号を規格(搬送波のこと)にのせて使うことでノイズを軽減するなどの正確に情報の伝達を行うことが主目的でした。しかし、デジタル変調は0、1の情報しか元々所持しておらず、正確に復調するのではなく0、1を判別すれば良いため、ノイズはあまり問題になりません。そのためアナログ変調と異なり、「正確に情報を伝達すること」に加え、デジタルでは標本化定理⁵に則って信号が復元されることから「高速に大量の情報を送ること」も重要になってきます。(アナログの時、速度は元の信号に依存する。例えば「言葉を話す」など)そのため、アナログの時とは異なる考え方を学ぶことが重要になってきます。次に主な変調方式4種類を説明します。

2. ASK(Amplitude Shift Keying)変調

これは、AM変調と同じように振幅を変化させることで情報を伝達する方法です。“0”と“1”にあたるものとして信号を“送る”・“送らない”を割り当てる方法(On Off Keying)がわかりやすいと思います。しかし、信号が送信されていないのか送信機に異常があるのかを判別することが難しいため、モールス信号のように短音と長音の2音を“0”、“1”に割り当てることが多いようです。このように、正確に情報を送ろうとすると高速に大量の情報を送信することが性質を持っています。また、フェージングなどにより波形が歪むと復調が難しくなるためあまり使われることが多くない方法になります。

3. FSK(Frequency Shift Keying)変調

これは、FM変調のように周波数を変更することで“0”、“1”の区別をつけて変調する方法になります。

⁵ 標本化定理とは、信号の2倍より高い周波数でサンプリング(一定間隔で信号の大きさを測定すること)を行えば、元の信号を完全に復元することができる定理のこと。

この方法だとフェージング⁶の影響も少なく安定した通信を行うことが可能になります。しかし、この方法の問題点は高速通信可能なデータの量が制限されてしまうという点です。標本化定理により信号の周波数が大きければ大きいほど多くの情報を伝えることが可能なのですが、FSK では2つの判別しやすい周波数を使用することになります。そうすると周波数の低い信号が出てきてしまい、早い方に統一してある次の PSK 変調方式よりデータ量の観点で劣るためこれも使用されることが少ないです。

4.PSK(Phase Shift Keying)変調

アナログでは PM 変調の名前で使用される変調方式で、位相を変化させることで“0”、“1”の区別を行います。この方式は FSK の安定性を有しつつ周波数は変えないという良い所どりをしたものになっています。また、この変調方式には多くの種類が存在する。単に、元の信号を“1”、位相を n だけ変化させた信号を“0”とする BPSK(Binary PSK)だけでなく、位相を $n/2$ だけ移動させた4つの信号(2ビットを表す信号で次の4つにそれぞれ対応する、“00”、“01”、“10”、“11”)を使用する QPSK(Quarter PSK)などである。このように安定性と高速通信の両方を実現させる PSK は利便性が高くデジタル変調でよく使われる方法になります。

5.QAM(Quarter Amplitude Modulation)変調

これは先ほどの QPSK に振幅の要素を組み合わせたもので、日本語では直角位相振幅変調と言われるものになります。QPSK を基本とするのは、位相を細かく分けると似た信号間で内容の取り違えをしてしまうことが多くなる割に情報量があまり増えない。つまり、効率が非常に悪いためです。この QAM により、1つの信号で図のような3ビットや4ビットの信号を1周期の信号で表すことができるようになります。位相の取り違えは起こりやすいのですが、振幅の場合は大きさが電力に等しいため電力を大きくするだけで正確に判別できる振幅の数が増えるため PSK よりも使い勝手の良い変調になります。これにも非常に多くの種類が存在し、3ビットの信号を扱う QAM を 8QAM、4ビットの信号を扱う QAM を 16QAM などのように名前がつけられています。

⁶ フェージングとは、電波が干渉しあうことでもともと送信した電波と受信した電波の強さが変動してしまうこと。移動している時はフェージングによるエラーが起きやすい。

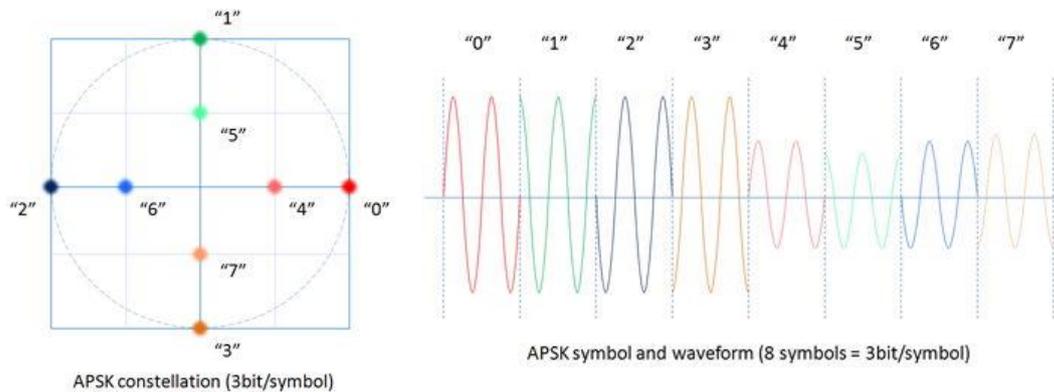


図 8QAM の振幅と位相の位置(左)とその波形の様子(右)
 (引用:silex technology 「変調の話(1)」, <http://www.silex.jp/blog/wireless/2013/03/post-4.html>,2020/9/17)

アナログの世界ではよく使われる AM や FM はデジタルではあまり使われないといったこともあり、変調にはデジタルの特徴というものが現れていると思っています。本記事で、デジタルとアナログの違いを理解してもらえたら嬉しいです。

参考文献

DENGYO 「第 3 回 無線通信に必要な変調技術」, URL <https://www.dengyo.com/labo/kouza/radio03.html>,2020/9/17

Silex technology 「変調のはなし(1)」, URL <http://www.silex.jp/blog/wireless/2013/03/post-4.html>,2020/9/17

オリックス・レンテック 「デジタル変調方式」, URL https://www.orixrentec.jp/helpful_info/detail.html?id=67

誤り訂正符号って？

JP7VTF

1. 身近で使われる誤り訂正符号

みなさんの中で CD や DVD、BD を使ったことがない人はほとんどいないのではないのでしょうか。これらの光ディスクは、表面に極めて微小なくぼみが彫られており、このくぼみのパターンによって情報を記録しています。再生機(CD,DVD,BD プレイヤー)はこのパターンにレーザ光を照射し、その反射を見ることで情報を読み出しています。では、ディスクの表面に傷があったらどうなるのでしょうか？おそらく、レーザ光の反射の仕方が変わり、期待していたものとは違う情報が読み出されてしまい、音楽や映像を正しく再生できなくなります。でも、たとえば、みなさんのお家にある古くて傷がついてしまっているような CD やレンタルビデオ屋から借りてきた、使い古されて傷のついた DVD でも再生できたという経験があるのではないのでしょうか？もしかしたら、それは**誤り訂正符号**のおかげかもしれません。

2. 誤り訂正符号とは

皆さんは、「誤り訂正符号」とはなんぞや？と思ったのではないのでしょうか。誤り訂正符号を簡単に説明すると、**情報に少しづつこいくらいに余分な情報を付け加えておいて、誤りが発生した際にはその余分な情報を頼りに必要な情報を復元する仕組み**といえるでしょう。仮に間違っ情報を読み出しても、誤り訂正符号を用いることで、正しいものか正しくないものかある程度までは判断して修正することができます。誤り訂正符号にはさまざまなものが考案されています。たとえば CD ではリードソロモン符号と呼ばれる誤り訂正符号が使われています。光ディスクに限らず、誤り訂正符号は通信にも使われています。たとえば電波で長距離通信すると、途中でノイズが入り、間違っ情報を受信してしまうことがあるためです。例えば地デジでは畳み込み符号やリードソロモン符号が、携帯電話や衛星通信などでは LDPC(Low density parity check)やターボ符号など様々な種類の誤り訂正符号が使われています。身近な場所で活用されている誤り訂正符号ですが、その理論は高度な数学の上に成り立っており、非常に難解です。そこで、ここでは誤り訂正符号の雰囲気をごんやりとでも知っていただけるように、誤り訂正符号の中でも理解しや

すい**繰り返し符号**について説明したいと思います。

3.誤り訂正符号の一例:繰り返し符号

まず仮定として、ここではデジタル信号を扱うことにします。デジタル信号とは平たく言うと、「0」と「1」の組み合わせで情報を表すような信号です。

例えば、「0」というデータをなんらかの媒体に記録したとします。しかし、この媒体が汚れてしまい、それを次に読み出したら「1」と誤って認識してしまう可能性があります。そこで、このような誤りをなくすために繰り返し符号を付与することで誤り訂正を行うことを考えます。以下に繰り返し符号の**生成法**を記します。とても簡単です。

リスト1 繰り返し符号の生成法

「0」を表す繰り返し符号は、「000…」と奇数個の「0」から成る
「1」を表す繰り返し符号は、「111…」と奇数個の「1」から成る

つまり、「0」を記録したときは「0」を奇数個だけ繰り返し書き、「1」を記録したいときは「1」を奇数個だけ繰り返し書けばよいのです。ここでは奇数個の「0」または「1」の塊のことを**符号語**と呼ぶことにします。

書き出しにルールが必要ならば、読み出しにもルールが必要になります。そこで、以下に読み出すときのルールである**復号則**を記します。これも簡単です。

リスト2 繰り返し符号の復号則

1つの符号語(奇数個の「0」または「1」の塊)のうち、多数決を取って多かった方を読み出したい情報であると判断する

リスト1でなぜ1つの符号語の「0」または「1」の総数を奇数個にしたかということ、多数決を取ったときにちょうど半分にならないようにするためだったのです。まあ、とっても簡単ですね。

では、試しに、繰り返し符号を使った誤り訂正の過程を見ていきたいと思います。ここでは「0 1 0 1」という情報を記録することにします。この「0 1 0 1」から5桁の繰り返し符号を生成すると、以下のようになります。

リスト3 「0 1 0 1」から生成した繰り返し符号

00000 11111 00000 11111

リスト3の数列をなんらかの媒体に記録します。とりあえず、黒板にチョークで書いておきましょうか。ではこれを3日後ぐらいに読み出したとしましょう。上記のような意味不明な羅列は案の定、誰かによって黒板消しで消されています。まあ、かろうじて消した跡から以下の数列を読み出せたとします。

リスト4 読み出した数列

00100 11001 01010 11110

さて、リスト3とリスト4を比較すると、いくつかの桁の数字が間違っていて読み出されていることがわかります。では、リスト2の復号則を用いてリスト4から元の数列を復元してみます。1個目の符号語は「0」が4つに対して「1」が1つなので元の情報は「0」です。2個目は「0」が2つに対して「1」が3つなので「1」です。残りの符号語も同様に復号すると、「0 1 0 1」となります。これは、最初に記録しようと考えていた情報に一致します！おそらく最初から黒板に「0 1 0 1」と書いていたら、消された後には何が何だか分からなくなっていたでしょう。このように、繰り返し符号を用いると、間違えて情報を読み出してもある程度までは訂正することができます。

4. 誤り訂正符号の欠点

上記に示したように、誤り訂正符号を用いると間違えて情報を読み出してもある程度までは正しい情報に訂正することができます。このように便利な誤り訂正符号ですが、都合のいいことばかりではなく欠点もあります。そのうちの3つについて説明したいと思います。

1. ある程度までしか訂正できない

どのような種類の誤り訂正符号でも、「ある程度」までしか訂正ができず、**読み出した情報の欠損ぐあいがひどすぎると訂正できなくなります**。たとえば、前節で紹介した符号語長が5桁の繰り返し符号の場合、その過半数である3つ以上の桁を誤って読み出した場合、正しく訂正ができなくなります。たとえば、「1」を記録したかったために「11111」を媒体に書いても、「00011」と読み出してしまうと、元の情報は「0」であ

ると間違えてしまいます。

2. 「余分な情報」を付与する分だけ容量を必要とする

どのような種類の誤り訂正符号でも、多かれ少なかれ「余分な情報」を付与する必要があります。**誤り訂正符号が誤り訂正できるのはこの余分な情報のおかげですが、あまりにもこれが大きいと正味で記録できる情報が少なくなってしまう**。たとえば、前節で紹介した符号語長が 5 桁の繰り返し符号の場合、「1」 という 1 桁の情報を記録するのに「11111」という 5 桁もの情報を必要とします。これはもともとの情報の 5 倍に当たり、あまり効率がよくありません⁷。繰り返し符号は誤り訂正符号の中でも効率が悪い部類に入り、実際に使用されることはほとんどありません。

3. 符号化・復号化に手間と時間がかかる

どのような種類の誤り訂正符号でも、もともとの情報と誤り訂正符号の間とで変換・逆変換を行う必要があります⁸。繰り返し符号であれば多数決を取っているだけなのでそこまで手間はかかりませんが、この他の高度な誤り訂正符号においては、**符号化・復号化にそれなりの手間を必要とします**。

以上のように、誤り訂正符号にはよいこともあればよくないこともあります。実際に誤り訂正符号を使用する際には、**これらのトレードオフを考慮しつつ、最大限の利益が得られるようにシステムを設計します**。

5.まとめ

本稿では、誤り訂正符号とは何かについて簡単に述べ、誤り訂正符号の一例である繰り返し符号について、誤り訂正の過程を示しました。また、誤り訂正符号の欠点についても、繰り返し符号の場合の実例を踏まえて述べました。情報通信において重要な役割を担う縁の下の力持ちである誤り訂正技術について興味を持っていただければ幸いです。

参考文献

和田山 正, “誤り訂正技術の基礎”, (2010), 森北出版.

⁷ 元の情報の桁数を k 、符号語の桁数を n としたとき k/n を符号化率と呼ぶ。符号化率は誤り訂正符号の効率を表す。たとえば上記の繰り返し符号では $k=1$ 、 $n=5$ で符号化率は $k/n=1/5$ である。

⁸ 誤りが発生していない場合には逆変換の手間が省けるように工夫された誤り訂正符号もある。

MATLAB を用いた WSPR の生成: 通信路符号化の例

部員 w(草)

1. 近年のアマチュア無線のトレンド: WSJT 系

アマチュア無線というと、音声による通信やモールス符号による通信(通称 CW: Continuous wave)などを思い浮かべるかと思います。実は、アマチュア無線にはそれ以外にも PC を用いたデータ通信も盛んに行われています。近年、宇宙物理学者 Joseph Taylor 氏(1993 年ノーベル物理学賞)を中心に開発された WSJT 系(Weak Signal Communication by Joe Taylor)モードがアマチュア無線家の間で流行しています。これらのモードは地球上の無線局から月面に電波を照射し、その反射波を受信することで通信を行う月面反射通信(EME: Earth-moon-earth)用に開発されましたが、近年は短波帯(HF: High frequency)を用いた遠距離通信にも用いられるようになりました。

WSJT 系モードが流行している理由は、他の音声通信や CW では交信ができないような長距離間(たとえば地球の裏側)の通信がいつも簡単に実現できるからでしょう。そもそもなぜ長距離通信が困難であるかといえば、距離が増大すると電波が減衰してしまい、まわりの雑音に埋もれてしまって信号を識別できなくなってしまうためです。音声通信や CW では人間の耳の良さに依存しています。これに対して、WSJT 系モードは、あらかじめ送信する信号に強力な誤り訂正符号を付与しており、送受信は通常パソコンによって行われます。誤り訂正符号とは、あらかじめ送信する情報に冗長性を持たせておき、誤りが発生した際にはその冗長な情報を元に情報を修正する符号です。WSJT 系モードで長距離通信を簡単に実現できる理由の 1 つに強力な誤り訂正符号を使用しており、雑音に対する耐性が強いことが挙げられます。

2. WSPR とは ?

WSJT 系のモードにはさまざまなものがあります。最もポピュラーなものとしては、HF での遠距離通信に用いられる「FT8」や、より短時間に交信ができるようにした「FT4」が挙げられます。また、FT8 登場以前に EME や VHF(Very high frequency)、UHF(Ultra high frequency)帯、HF 帯で使用されていた「JT4」、「JT9」、「JT65」などの

「JT シリーズ」があります^[1]。これらのモードは双方向通信を行うためのものであり、お互いにコールサインを交換しあいます。

一方、WSPR(ウィスパー:英語で「ささやき」の意)は、放送のように一方向通信を行うための WSJT 系モードです。WSPR は「Weak signal propagation reporter」の略で微弱な信号の伝搬を報告するためのモードです。世界中には WSPR の信号を受信している局が多数あります。これらの局は無線機で受信した信号をパソコンで復調し、「WSPR.net^[2]」というサイトに自動で報告します。WSPR.net 上では世界中からの受信報告がデータベース化されています(図 1)。WSPR の特徴として、前述の FT8 などに比べて誤り訂正能力が高く、より低い送信電力でより遠くまで信号を伝えられることが挙げられます。本稿では、この WSPR 信号の生成方法について説明します。

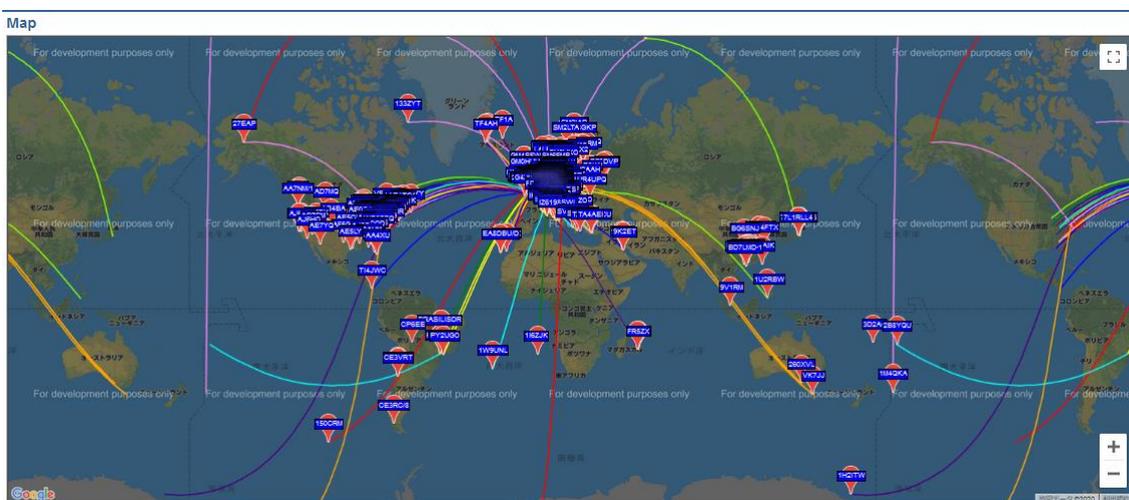


図 1 WSPR.net 上で共有されている受信報告例。

線で結ばれた無線局の間で電波伝搬が報告されている。

3. MATLAB を用いた WSPR 信号の生成

WSPR のプロトコルは参考文献^[3]に公開されています。また、このプロトコルを満たす通信路符号化のアルゴリズムの解説も有志によって公開されています^[4]。これらに対して、本稿はプログラミング言語 MATLAB を用いた実装例を踏まえながら、プロトコルとアルゴリズムを解説する内容になっています。WSPR の信号生成は大きく分けて 6 つの処理に分けることができます。

- | | |
|--------------|-----------|
| 0 送信したい情報の準備 | 1 情報の可逆圧縮 |
| 2 畳み込み符号で符号化 | 3 インターリーブ |
| 4 同期信号の付与 | |

以下にそれぞれについて順を追って説明します。

3.0 送信したい情報の準備

WSPR の標準メッセージ構成では、①送信者の呼出符号(コールサイン)、②送信場所、③送信電力の情報を信号に乗せて送信します。①の呼出符号(コールサイン)は無線局を識別するための世界にただ一つの符号です。WSPR で送ることのできる呼出符号は、大文字英数字とスペースから構成される 6 文字の文字列に制限されています。これらは可逆圧縮の仕様による制約であるため、詳細は 3.1 で述べます。②送信場所は送信機のある場所を表します。アマチュア無線ではグリッド・ロケータとよばれる、経度と緯度から算出される 6 桁の英数文字列を用いて場所を表すことがあります^[5]。例えば、アマチュア無線部のアンテナ群がある東北大学片平キャンパスを表すグリッド・ロケータは「QM08KG」です。WSPR ではこのうちの上 4 桁を使用します。③送信電力はアンテナに入力する電力を表します。WSPR では単位として「dBm」を使用します。dBm は基準電力 1 mW との電力比を常用対数で表した物理量です。こちらも可逆圧縮の仕様上、0~60 dBm の範囲の整数値のみを送信できます。これは真数に換算すると 1 mW から 1 kW までの範囲に該当しており、アマチュア無線局の送信電力を表すのには十分な範囲でしょう⁹。

さて、ここでは例として、①を「K1ABC」、②を「FN42」、③を「37」とします。なぜこのようにしたかという、前述の参考文献^[3]ではこの例での符号化の結果を掲載しており、本稿もそれに合わせることで両者の対応関係をわかりやすくするためです。

3.1 情報の可逆圧縮

情報の可逆圧縮に先立って、送信できる情報の制約について説明します。まず、送ることのできる呼出符号は、数字、大文字英数字、スペースから構成される 6 文字の文字列に制限されています。また、上から 2 桁目についてはスペースを使うことはできません。上から 3 桁目については数字のみを使うことができます。残りの下 3 桁については数字を使うことはできません。2 文字目に数字が来るようなコールサイン(まさに今回例に使っているような「K1ABC」のようなコールサイン)では、最初の 1 文字目をスペースにし(「_K1ABC」)、必ず上から 3 桁目に数字が来るようにします。これらをまとめたものを図 2 に示します。

⁹ 日本国内のアマチュア無線局に免許される空中線電力の上限は 1 kW とされています。ちなみに、JA7YAA ではバンドによっては空中線電力 1 kW が免許されています。

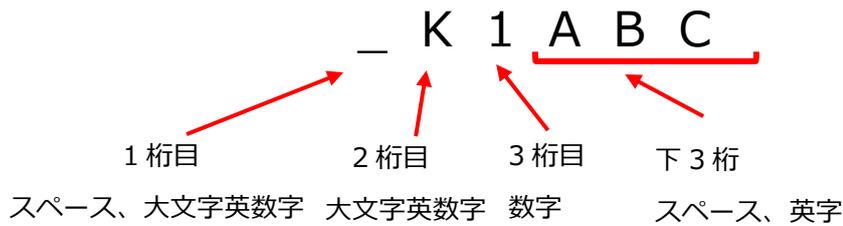


図 1 送信できる呼出符号の制約

さて、この制約条件を踏まえて、呼出符号について可逆圧縮を行います。リスト 1 に圧縮の手順を示します。[n 文字目]とは、呼出符号の上から n 文字目を表す整数値のことであり、呼出符号の文字とこの整数値の対応は右に示す説明の通りです。これを MATLAB で実装するとリスト 2 のようになります。

リスト 1 呼出符号の可逆圧縮の手順

$N_1 = [1 \text{ 文字目}] : \text{数字 } 0\sim 9 \text{ を } [0] \sim [9]、A\sim Z \text{ を } [10] \sim [35]、\text{スペースを } [36] \text{ に割り当て}$
 $N_2 = N_1 \times 36 + [2 \text{ 文字目}] : \text{数字 } 0\sim 9 \text{ を } [0] \sim [9]、A\sim Z \text{ を } [10] \sim [35] \text{ に割り当て}$
 $N_3 = N_2 \times 10 + [3 \text{ 文字目}] : \text{数字 } 0\sim 9 \text{ を } [0] \sim [9] \text{ に割り当て}$
 $N_4 = N_3 \times 27 + ([4 \text{ 文字目}] - 10) : A\sim Z \text{ を } [10] \sim [35]、\text{スペースを } [36] \text{ に割り当て}$
 $N_5 = N_3 \times 27 + ([5 \text{ 文字目}] - 10) : A\sim Z \text{ を } [10] \sim [35]、\text{スペースを } [36] \text{ に割り当て}$
 $N_6 = N_3 \times 27 + ([6 \text{ 文字目}] - 10) : A\sim Z \text{ を } [10] \sim [35]、\text{スペースを } [36] \text{ に割り当て}$

リスト 2 MATLAB における呼出符号の可逆圧縮の実装例

```

%% 設定
callSign = ' K1ABC';
loc = 'FN42';
pwr = 37;
%% コールサインのコーディング
callSign = upper(callSign); % すべて大文字に統一
callSign_int = uint64(unicode2native(callSign,'Shift_JIS')); % ASCIIに変換
% 数字をASCIIから固有の整数値に変換
callSign_int((48 <= callSign_int) & (callSign_int <= 57)) ...
    = callSign_int((48 <= callSign_int) & (callSign_int <= 57)) - 48;
% 英字をASCIIから固有の整数値に変換
callSign_int((65 <= callSign_int) & (callSign_int <= 90)) ...
    = callSign_int((65 <= callSign_int) & (callSign_int <= 90)) - 55;
callSign_int(callSign_int == 32) = 36; % スペースを固有の整数値に変換
callSign_int(1,4:6) = callSign_int(1,4:6) - 10; %先に10を減算してしまう
N = callSign_int(1,1).*36 + callSign_int(1,2); % 1-2文字目
N = N.*10 + callSign_int(1,3); % 3文字目
N = N.*27 + callSign_int(1,4); % 4文字目

```

```
N = N.*27 + callSign_int(1,5); % 5文字目
N = N.*27 + callSign_int(1,6); % 6文字目
```

リスト 2 では、圧縮の結果が変数 N に格納されます。N の取りうる場合の数は $37 \times 36 \times 10 \times 27 \times 27 \times 27 = 262177560$ 通りとなり、これは 28 ビット分の情報 ($2^{28} = 268435456$ 通り) に相当します。コールサイン「_K1ABC」を圧縮すると 10 進数で「259047992」となります。大した計算ではありません。

次にグリッド・ロケータの情報を圧縮します。使用するグリッド・ロケータは 4 桁の英数字列で、上 2 桁は A~R の英文字(整数値「0」~「17」に割り当て)、下 2 桁は 0~9 の数字(整数値「0」~「9」に割り当て)となります。グリッド・ロケータはリスト 3 に示す手順で圧縮されます。

リスト 3 グリッド・ロケータの圧縮

```
M1 = (179 - 10*[1桁目] - [3桁目]) * 180 + 10 * [2桁目] + [4桁目]
```

続いて、送信電力の情報を圧縮します。送ることができるのは 0~60 dBm までです。これは、リスト 4 に示す手順でグリッド・ロケータの情報とマージされます。リスト 3、リスト 4 によって計算される M は 22 ビットに相当する情報になります。

リスト 4 送信電力の圧縮

```
M = M1*128 + [パワー] + 64
```

では、リスト 3、リスト 4 を MATLAB で実装した例をリスト 5 に示します。アホがばれるので自分の書いたコードをあまり掲載したくはないのですが。

リスト 5 MATLAB におけるグリッド・ロケータ・送信電力の圧縮の実装例

```
%% グリッドのコーディング
loc = upper(loc);
loc_int = uint64(unicode2native(loc, 'Shift_JIS'));

loc_int(1,1:2) = loc_int(1,1:2) - 65;
loc_int(1,3:4) = loc_int(1,3:4) - 48;
M1 = (179 - 10.*loc_int(1,1) - loc_int(1,3)).*180 + 10.*loc_int(1,2) +
loc_int(1,4);

%% パワーのコーディング
M = M1.*128 + pwr + 64;
```

「FN42」と「37」を圧縮すると 10 進法で「2896997」となります。

最後にコールサインの情報 N(28 ビット分)とグリッド・ロケータおよび送信電力の情報 N(22 ビット分)をマージします。N の LSB(Least significant bit, 最下位ビット)の後ろに M の MSB(Most significant bit, 最上位ビット)を続けるように N と M をくっつけます(図 2)。これにより、28+22=50 ビットの情報になりました。なお、これに加えてさらに下位ビットに 31 ビットだけ「0」を加えます。最終的には、81 ビットの情報になります。これを MATLAB で実装するとリスト 6 となります。なお、リスト 6 では畳み込み符号での符号化に先立って変数型の変換(uint64 型から logical 型配列)を行っています。今回の例では、上位 50 ビットが[1,1,1,1,0,1,1,1,0,0,0,0,1,1,0,0,0,0,1,0,0,0,1,1,0,0,0,1,0,1,1,0,0,0,1,0,1,1,0,0,0,0,1,1,0,1,0,0,0,1,1,0,0,1,0,1]となります。

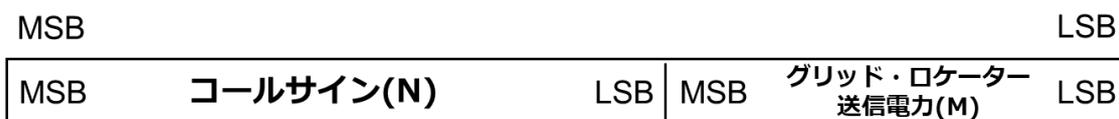


図 2 コールサイン、グリッド・ロケータおよび送信電力をマージ後の情報

リスト 6 MATLAB における各種情報のマージの実装

```
data_int = N .* (2.^22) + M;
data = logical(unicode2native(dec2bin(data_int), 'Shift_JIS')-48) % 50 bitのデータ
```

3.2 畳み込み符号で符号化

次に、可逆圧縮を施した情報を畳み込み符号で符号化します。畳み込み符号とは、名前の通り畳み込み演算を用いた誤り訂正符号です。本節では、はじめに 2 元有限体における畳み込み演算について説明し、それを踏まえて WSPR で用いられる畳み込み符号について説明します。

まず、「2 元有限体」というよくわからない言葉が出てきましたが、筆者も正直よくわかりません¹⁰。文献^[6]を参考にじっくり説明すると、2 元有限体とは①使われる数字は「0」と「1」のみ¹¹で、②足し算と掛け算が定義されている、集合と演算の組ということにしておきましょう¹²。この話で私たちが一番気にしなければいけないことは、②の 2 元有限体における 2 項演算子の定義です。リスト 7 に 2 元有限体における加法と乗法の定義を示します。2 元有限体における加法は、ブール代数でいうところの排他的論理和(EX-OR)と等価になります。

¹⁰ 無責任な、だったら書くな。学部 3 年の授業で符号理論([6]はその時の教科書)を取ってはいたもののほとんど覚えていません。

¹¹ 有限の集合を扱うので「有限体」。

¹² 理学が専門の人から絶対怒られるやつ。「詳しい理屈はわからないが、実用上使えれば OK」という工学的な考え方のもと、今回はこれで許してください。

通常の整数に対する加法(整数環における加法)では、「1+9」のような演算をすると和は「10」となり繰り上がりが発生しますが、2元有元体ではそうはならず、「1+1=0」となります。また、2元有限体における乗法は、ブール代数でいうところの論理積(AND)となります。

リスト 7 2元有元体における 2項演算子の定義

(a) 加法

x+y		x	
		0	1
y	0	0	1
	1	1	0

(b) 乗法

x · y		x	
		0	1
y	0	0	0
	1	0	1

次に、畳み込み演算について説明します。フーリエ解析を勉強したことがある方は畳み込み演算はご存じかとは思いますが、ここでは2元有限体における畳み込み演算の方法について説明します。2元有限体のベクトルが2つあるとします(例えば[0, 1, 0, 1]や[1, 0, 0, 1]など)。この2つのベクトルをそれぞれ h, x としたとき、畳み込みの結果 y は(1)式で定義されます。(1)式では、インデックス n, k を用いて各ベクトルの要素を表しています。また、各ベクトルで定義されていない要素についてはすべて0であるものとします。加法および乗法はリスト7の定義に従います。

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{1}$$

いきなり数式を見せられてもよくわからないかと思しますので、実際に計算例を示します。畳み込み演算は筆算ですること可能です。例として、図3に[1, 0, 1, 0]と[1, 1, 0, 1]の畳み込み演算を示します¹³。やり方は簡単で、普通の乗算の筆算と同じように、かける数とかけられる数を書きます。この時、[1, 0, 1, 0]なら「0 1 0 1」、[1, 1, 0, 1]なら「1 0 1 1」というように上位と下位をひっくり返して書きます。以後も、通常の乗算の筆算と同じように計算を進めます。ただし、加法と乗法の定義はリスト7に従います。最後に得られた「0 1 0 0 1 1 1」を上位と下位でひっくり返した[1, 1, 1, 0, 0, 1, 0]が畳み込み演算の結果となります。

¹³ 畳み込み演算は書籍によっては「*」という演算子で表されることがあるので図3でもこれを用いています。ただプログラミング言語では乗法を「*」と書くものが多いので紛らわしい。

				0	1	0	1	
			*	1	0	1	1	
				0	1	0	1	
		0	1	0	1			
	0	1	0	1				
	0	1	0	0	1	1	1	

図3 筆算を使った畳み込み演算の例

2元有限体における畳み込み演算について MATLAB で実装した例をリスト8に示します。本当は MATLAB にそういう関数が実装されているのかもしれませんが、簡単だったので自分で作ってみました。関数 F2conv2 は引数として logical 型配列を2つ与えるとそれらの畳み込み演算の結果を返します。

リスト9 MATLAB における2元有限体上の畳み込み演算の実装例

```
function out=F2conv2(imp, input)
% 2元有限体の畳み込み
% 入力
% imp 1 x ? インパルス応答 logical
% input 1 x ? システムへの入力 logical
% 出力
% out 1 x ? システムの出力 logical

%% 関数本体
imp = fliplr(imp);
buffer = [false(1, length(imp)-1) input false(1, length(imp)-1)];
for m=1:(length(imp)-1+length(input))
    out(1, m) = logical(mod(sum(buffer(1, m:(m+length(imp)-1)) & imp), 2));
end
end
```

さて、畳み込み符号というのは、(1)式において、 h にはある決まったベクトルを、 x には符号化したいデータを用いて畳み込み演算を行い、得られた結果を符号とする誤り訂正符号です。なお、 h は生成多項式と呼ばれます。

WSPR では、 h として $h_1 = [1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1]^{14}$ と $h_2 = [1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0]^{15}$ の2つが用いられます¹⁶。

¹⁴ MSB と LSB が逆になるように反転して16進数表記にすると「F2D05351」になります。参考文献^[4]にはこちらの値が掲載されています。なお、Joseph Taylor 氏のプログラム(なんと FORTRAN で記述)では、これをさらに10進数表示にした「-221228207」が使われています。

¹⁵ 同様に MSB と LSB が逆になるように反転して16進数表記にすると「E4613C47」、10進数では「-463389625」となります。

¹⁶ なんでこんな0と1の羅列になるのかと思うかもしれませんが、これは「Layland-Lushbaugh polynomials」と呼ばれる「well-known」な生成多項式のようなものです(筆者もよくわかっていませんが、書いてありました^[7])。大元の論文はこちらのようです^[8]。この論文の筆者は NASA のジェット推進研究所の

(1)式において、入力信号 x の長さを m 、生成多項式 h の長さを K (拘束長と呼ぶ)とすると、畳み込みの結果 y の長さは $m + K - 1$ となります。さて、WSPR の場合 $m = 50$ となります。また、 h_1, h_2 はともに $K = 32$ となり、 $50 + 32 - 1 = 81$ となります。また、WSPR では異なる生成多項式 h_1, h_2 を用いた畳み込みの結果を組み合わせ使っています。このため、最終的に得られる畳み込み符号は $81 \times 2 = 162$ ビットになります。

ちなみに、畳み込み符号においては拘束長 k が大きいほど誤り訂正能力が高くなります。他の畳み込み符号の実装例を見てみると、NASA のボイジャー計画で使われたものは $K = 7$ 、カッシーニなどで使われた符号は $K = 15$ などがあります^[9]。WSPR の拘束長というのはだいぶ長い部類に入るでしょう。ところで、畳み込み符号の復号にはビタビアルゴリズム¹⁸が使われるのが一般的ですが、ビタビアルゴリズムの計算量は拘束長に対して指数関数的に増加するため、あまりにも拘束長が長いと実用的ではありません。このため、WSPR の復号には、誤り訂正能力が若干落ちるものの計算量を抑えられる逐次型復号アルゴリズム(ファーノアルゴリズム)が用いられます。

3.3 インターリーブ

畳み込み符号の欠点として「バースト誤り」に弱いという特徴があります。バースト誤りとは、短い区間に誤りが多数現れるような誤りを表します(誤りが散らばって現れるものはランダム誤りと呼ばれます)。畳み込み符号は、入力された情報の手がかりを連続する複数の符号(拘束長の長さ分)にわたって記録する誤り訂正符号です。このためバースト誤りがひどいと、誤り訂正を行うための手がかりが十分に得られなくなり、訂正は困難となります。そこで、通信路中でバースト誤りが生じても、それが1箇所に集中しないように符号の順番を入れ替える「インターリーブ」という手法が用いられます。

WSPR におけるインターリーブは次のようなアルゴリズムで実行されます。まず、 $0, 1, 2, 3, 4, 5, 6, 7, \dots, 160, 161$ と 0 から 161 まで 1 ずつ増加する数列を考えます。また、これらの数は符号なし 8 ビット整数として考えます。ではこの数列のそれぞれの数について、MSB と LSB をひっくり返した数に対応づけるような数列を考えます。これは、 $0, 128, 64, 192, 32, 160, 96, 224, 16, \dots$ となります。次にこの数列から 162 以上の数を消去します。すると、 $0, 128, 64, 32, 160, 96, 16, \dots$ となります。最後に、この数列をも

¹⁸ ビタビアルゴリズムの発明者であるビタビ博士は世界的な通信技術・半導体設計開発会社 クアルコム社の創業メンバーの1人です。ビタビ博士はビタビアルゴリズムをはじめとして情報・通信工学の分野に大きな影響を与える一方で、お金儲けも上手というわけです。なんと母校に 5200 万ドルの寄付をして学部で自分の名前を付けてしまうほど。

上記の同期信号とインターリーブ後の2元有元体ベクトルはリスト12のように合成されます。同期信号との合成前は0と1の2値しかとらないのに対し、同期信号と合成後は0~3の4値を取るようになります。また、これをMATLABで実装するとリスト13のようになります。

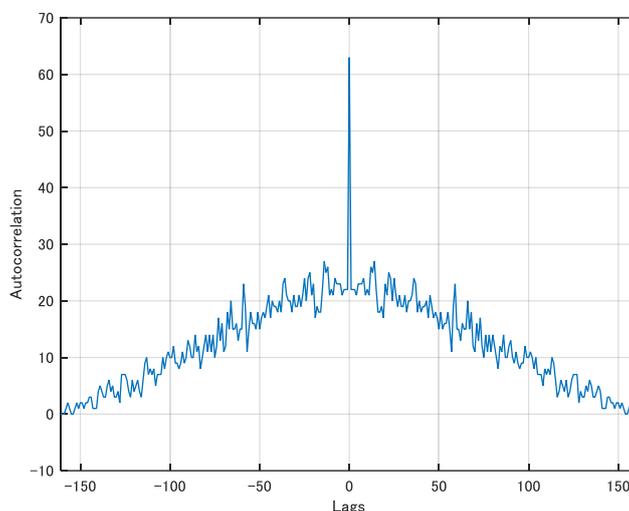


図5 同期信号の自己相関関数

リスト12 同期信号との合成方法

$$[\text{合成後のベクトル}](n) = 2 \times [\text{インターリーブ後のベクトル}](n) + [\text{同期信号}](n)$$

リスト13 MATLABにおける同期信号との合成の実装

```
sync_pat-
tern=[1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1,
0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
bb = il.*2 + sync_pattern
```

本稿で示している例における同期信号と合成後の結果は、[3,3,0,0,2,0,0,0,1,0,2,0,1,3,1,2,2,2,1,0,0,3,2,3,1,3,3,2,2,0,2,0,0,0,3,2,0,1,2,3,2,2,0,0,2,2,3,2,1,1,0,2,3,3,2,1,0,2,2,1,3,2,1,2,2,2,0,3,3,0,3,0,3,0,1,2,1,0,2,1,2,0,3,2,1,3,2,0,0,3,3,2,3,0,3,2,2,0,3,0,2,0,2,0,1,0,2,3,0,2,1,1,1,2,3,3,0,2,3,1,2,1,2,2,2,1,3,3,2,0,0,0,0,1,0,3,2,0,1,3,2,2,2,2,2,0,2,3,3,2,3,2,3,3,2,0,0,3,1,2,2,2]となります。

最後に、上記のベクトルで4FSK変調します(FSKについては本誌の他記事参照^[10])。上記のベクトルも変調方式もちょうど4値なので、1回の変調で1つのシンボルを送ることができます。

4. 今後の展望

本稿では WSJT 系モードの 1 つである WSPR の通信路符号化について MATLAB のコーディング例を交えて説明しました。

一方、これまでに私たちは 135 kHz 帯の免許交付を目指して 135 kHz 帯 WSPR 送信機の試作について報告していました^[11]。この送信機はパソコンで通信路符号化を行う方式を採用していました。しかしながら、WSPR の通信路符号化はそこまで計算コストを必要とせず、マイコン等でも実装することが可能です。そこで、今後はマイコンに WSPR の通信路符号化を行うプログラムを実装し、送信機単体で運用ができるように改良します。これによってパソコンが不要になるため、より簡便に運用を行うことができます。

参考文献

- [1] "https://www.jarl.org/Japanese/2_Joho/2-2_Regulation/communication-software.htm," 2020/9/19 閲覧.
- [2] "<http://wsprnet.org/drupal/>," 2020/9/19 閲覧.
- [3] "https://physics.princeton.edu/pulsar/K1JT/WSPR_2.0_User_Japanese.pdf," 2020/9/19 閲覧.
- [4] G4JNT, "WSPR Coding Process," "http://www.g4jnt.com/Coding/WSPR_Coding_Process.pdf," (2009) 2020/9/19 閲覧.
- [5] JARL, "グリッド・ロケータとは," "https://www.jarl.org/Japanese/1_Tanoshimo/1-2_Award/gl.htm," 2020/9/20 閲覧.
- [6] 和田山 正, "誤り訂正技術の基礎," (2010), 森北出版.
- [7] ziggurat29, "Encoding WSPRs," "<https://hackaday.io/project/166875-careless-wspr/log/167301-encoding-wsprs>," 2020/9/22 閲覧.
- [8] J. Layland, W. Lushbaugh, "A Flexible High-Speed Sequential Decoder for Deep Space Channels," IEEE Transactions on Communication Technology 19(5), pp.813-820 (1971).
- [9] "畳み込み符号," "<https://ja.wikipedia.org/wiki/%E7%95%B3%E3%81%BF%E8%BE%BC%E3%81%BF%E7%AC%A6%E5%8F%B7>," 2020/9/22 閲覧.

[10] リー, “デジタル変調方式について,” 東北大学アマチュア無線部誌 Vol.2 学祭号, p p.19-21 (2020).

[11] JP7VTF, “E 級アンプを用いた 135 kHz 帯 FSK/GFSK 送信機の試作,” 東北大学アマチュア無線部誌 Vol.1 新入生歓迎号(“<http://www.ja7yaa.org.tohoku.ac.jp/contents/bushi2020.pdf>”),pp.23-27 (2020).

編集後記

今回デジタル変調についての記事をのせたりーです。前回に引き続き変調の話を書きました。このような内容に興味があり、送受信機を作ってみたいと思っています。そのために勉強したことを私なりの解釈で概要を説明するといった形式にしてみましたがいかがだったでしょうか。今は、このような勉強関係のことしか書けませんが、追々「作ってみた」という記事を作ればと思います。最後に私たちの部誌を読んでいただきありがとうございました。また次の部誌にも掲載いたしますので読んで頂ければ幸いです。

部員 K です。現在、アマチュア無線局の数は減少傾向にあります。しかしながら、アマチュア無線においてできることは増えており、例えば、d-star といったデジタルモードや、新しくアマチュアバンドにおけるモードが許可されたりしています。アマチュア無線の楽しみは尽きない程存在しているのではないかと考えております。私自身、去年、従事者免許を取得したばかりの初心者ではございますが、どんどん新しいこと挑戦し、面白さを発信したいと考えております。最後までお読みいただきありがとうございました。

JP7VTF です。今回は大学周辺の移動運用場所の紹介と誤り訂正符号についての記事を投稿しました。今回は私なりのアマチュア無線の楽しみ方を記事にしましたが、アマチュア無線にはさまざまな楽しみ方があります。この部誌を読んでアマチュア無線に少しでも興味を持っていただければ幸いです。部誌をご覧いただきありがとうございました！

MATLAB を用いた WSPR の生成を書いた部員 W です。デジタルモード WSPR のプロトコルについて記事を書きました。「アマチュア無線」とは言ってもプロ顔負けの技術が導入されていて、こんなに複雑で面白い仕組みが詰まっていることをしていただければと思います。今回は部誌を読んでくださりありがとうございました。

初めての方ははじめまして、そうでない方はまた読んで下さりありがとうございます。富沢いずみです。今回は「アマチュア無線 3 級・4 級の点数が面白いほどとれる」と某センター試験のアレっぽく免許試験についての記事を書かせて貰いました。本記事でも力説しましたが無線免許もアマ 3 までなら簡単に取れるので是非取得してみても如何でしょうか?あと今回もまた編集も担当させて頂きました。今後も部誌の執筆、発行をしていく予定ですのでまた部誌で会いましょう!!

