Patrick Lindecker (F6CTE) the 8 of may 2004 (mail: f6cte@aol.com)

**In this paper, I will describe two digital modes "keyboard to keyboard" of PSK (Phase Shift Keying) type: the PSKFEC31 and the PSK63F, both provided with a correction error, this with the goal to show the type of problem that one can meet when creating a new digital mode.**
**These modes and many more are available in one of the software developed by the author, program which name is Multipsk, downloadable from the following WEB site: http: //members.aol.com/f6cte/**

## INTRODUCTION

Digital communications in HF are, sometimes, difficult, or impossible, because of:

* QRM of others radio-stations,
* fading (QSB),
* ionospheric Doppler modulation (for low modulation speeds as for PSK31 or, worst, PSK10),
* time overlap of identical signals coming from different paths in the ionosphere (particularly sensitive for quick modulation speeds),
* a too weak ratio Signal-to-Noise ratio (the signal is more or less drowned in the noise) .

The aim of these experimental slow or average modulation speed modes with error correction is to allow, in HF, a " chat " communication between Ham operators:

* without file or picture transmission (for instance...) with much less errors that using BPSK31 (basic mode being a sort of reference).

* a sensitiveness level superior to the BPSK31.

It is to be remembered that the BPSK31 mode is designed to assure communications with a ratio Signal-to-Noise down –11.5 dB, so a signal about 14 less powerful that noise (referenced to 3 kHz). The equivalent Morse speed is 37 words/minute in capital and 51 words/minute in small letters.

## COMMON CHARACTERISTICS

1) All these modes (including the BPSK31) are BPSK ("Binary Phase Shift Keying") modes. BPSK is a modulation with 2 phases: 0° / 180° (opposite). This modulation may also be seen as an amplitude modulation with 2 opposite levels. This is not the better way to transmit information but it has the advantage of being simple to work with, and relatively efficient, especially in noise.

To avoid any phase reference (because one must measure in comparison with some phase), it has be chosen a differential mode: what is detected is not the pure phase but the reversal or not of a symbol in comparison with its predecessor, according to this diagram:

* 0°--->180 ° or 180°--->0° for reversal,
* 0°--->0 ° or 180°--->180° for non-reversal.

Unfortunately, in this way, the bit error rate is doubled.

This modulation (" base band " called) acts on the AF carrier which itself modulates the RF carrier.

The detection may be " coherent ", in fact " quasi-coherent " because the " bit " synchronization is built from the signal itself and not from an external source (solution that would be besides imaginable).
The word " coherent " is employed because a sort of phase locked loop called " Costas loop " is used to follow the phase. In reality, the phase (at the start) is known with an ambiguity of 180 °, so this definition is not strictly applicable.
Note: the Costas loop is a solution (the one of the author) among others. For example, the original program PSK31SBW (from G3PLX - Peter Martinez) does not do any phase following but a simple measure of the average phase during a bit duration (non-coherent detection).

2) PSK31, PSKFEC31 and PSK63F use a character coding called "Varicode". This term is the opposite of "fixed length code". As in Morse, the length character in bits depends to the number of occurrences of a given character in the literature. It is necessary to note that the PSK31 Varicode is neither the same as the PSKFEC31 one nor the same as the PSK63F one. Each character is preceded by a bits sequence (separation code) known in advance, for example "011" in PSKFEC31 (with " 1 " for phase reversal and " 0 " for " no reversal "). Finally, one has less bits to be transmitted for a standard text, from which is the interest. Moreover, a fixed length code implicates to be able to detect a lack of bit (due, for example, to a bad " bit " synchronization). This problem is not simple and exposes to losses of characters in series.

3) The transmission may be done in LSB or in USB but to simplify, USB is generally used, including the low bands. This has the advantage to allow for two hams to agree on the same frequency which can be defined by its RF component and its AF component. For example, one will talk of a RF frequency of 7035 kHz with an AF frequency of 1000 Hz measured on the spectrum (" waterfall "), so, in fact, 7036 kHz in USB. In a general way, USB is in the process of generalization for all digital modes (except for the 45 bauds RTTY).

4) The modulation speed is almost 31.25 bauds or a multiple (for example 2 x 31.25=62 .5 bauds for the PSK63 (which is an extension of the BPSK31 to 62.5 bauds) and the PSK63F. This speed corresponds to a submultiple of 8000, the sampling frequency of 8000 samples/sec being common on the DSP cards and on sound cards (nevertheless Multipsk uses the sampling frequency of 11025 samples/sec, which is the base standard for sound cards).

5) The synchronization is extracted from the received signal. Its goal is to define when, precisely, the bit measure must be done. It is generally built from a non-linearity applied to the base band signal. The author, for example, squares the signal.

Note 1: the base band signal is obtained by demodulation of an AF signal, i.e. the suppression of the AF component (the carrier), this with a Costas loop and a matched filter.
Note 2: the digital processes are, of course, the same as the analogical processes, except that the calculation power permits more interesting solutions. For example, if one uses low-pass recursive filters of the first, second or fourth order as in analogical, one can also use linear non-recursive filters of order 500, see more...

6) the signal frequency may drift, so one must dispose of an automatic control of the frequency drift which calculates the average phase drift. As soon as a drift is detected, it is only necessary to shift the " VCO " frequency of the Costas loop.

7) The generated base band signal is filtered through a windowing filter. The window, by default, is the rectangular one where the signal acts on " step " on the AF carrier (as in CW, RTTY, AMTOR...). The occupied band is proportional to the speed modulation (in bauds) with an envelope pattern in sin (x)/x, which gives a very wide band. Upon the InterSymbol Interference (ISI) point, it is the best solution, i.e. a given bit does not interfere with its neighbor. By principle, in PSK mode, the transmission band is reduced by using a windowing filter with a soft transition, generating, nevertheless, a little ISI.

## SOME OTHER EXPERIMENTAL MODES

I originally proposed, first, a mode known as PSK10: the transmission is done at 10 baud in BPSK, as for the previous modes. A short set of characters is associated to this mode with a surer prefixe than the one of BPSK31. If this mode is very sensitive (minimum S/N = -17 dB), the bit duration (0.1 second) is too large relatively to the ionospheric Doppler modulation in short waves.
To reduce error rate, I have proposed a solution, through PSKAM10/31/50 modes, which consists to repeat each character (of fixed bit length) as in AMTOR FEC. This principle works very well. However, under bad conditions (for example QRM), there appears a problem of synchronization loss. As it is difficult to re-synchronize under bad conditions (for an 8 bits character, one must choose between 16 possibilities...), the decoding might be disjointed (with cycles of synchronization loss and re-synchronization). Except this problem, the error rate is very weak compared to PSK31.

## PSKFEC31 DESCRIPTION

I choose the short PSK10 set of characters for two reasons:

* to have a concise set of characters which reduce the errors rate: the more the choice is reduced, the less you have probability to produce an error,
* to have a sufficient speed for a ham (28 words per minute). With a PSK31 set of characters, the speed would be about 23 words per minute.

I have chosen sequences of bits providing a large number of transitions, so as to facilitate the synchronization.
The separation code is " 011 " (" 1 " for " phase reversal " and " 0 " for " no reversal "). " 011 " is also the idling character.
Further on, it will be found, for the example, the first characters (without the separation code):

| CHARACTER | CODE |
|---|---|
| Idling (>) | |
| Space | 1 |
| E | 0 |
| T | 1 1 |
| A | 0 1 |
| I | 1 0 |

The modulation speed (31.25 bauds) has been chosen to avoid PSK10 problem (duration of the bit too long in front of the ionospheric Doppler modulation in SW) and to be consistent with PSK31.

The minimum Signal-to-Noise ratio is –14.5 dB, for a 2% error rate.

To avoid the synchronization problem of PSKAM, I repeat the bits and not the characters, so the synchronization is much more easy (choice between only 2 possibilities).

The type of windowing filter used for sending bits is the one used for PSKAM: the window is rectangular but the connection between 2 successive bits is done by a decreasing sine then by a creasing sine (each of one during a quarter of period). This allows a not too large bandwidth without interference between symbols. Except during transitions, the level envelope is constant so the average power is 85 % of the maximum power (obtained for example with a "tune").

```
Example: 1011:   1    |  0   |  1    |  1
                R-R-R-D-C-R-R-D-C-R-R-R-R-R-R-R
             or C                          or D
R: rectangular (=1) of ¼ T. D: Sine decreasing of ¼ T (1 to 0)
C: Sine creasing of ¼ T (0 to 1)
```

## DESCRIPTION OF THE PROTOCOL

### *TRANSMITTING STATION*

Each bit is repeated 13 positions later, according to the same principle as in AMTOR FEC except that it is related to bit and not to character.
Suppose a "phase reversal bit" noted "1" and a "no phase reversal bit" noted "0". Suppose that instead of a difference of 13 bits we had a difference of 5 bits. We want to transmit 101110:

| DX | RX |
|---|---|
| 1 | x1 |
| 0 | x2 |
| 1 | 1 |
| 1 | 0 |
| 1 | 1 |
| 0 | 1 |
| y1 | 1 |
| y2 | 0 |

```
DX is the first position
RX is the second position
x1, x2 are the previous bits (0 or 1)
y1, y2 are following bits (0 or 1)
```

It will be transmitted (DX then RX):
`1 x1 0 x2 1 1 1 0 1 1 0 1 y1 1 y2 0`

***RECEIVING STATION***

At the reception, the DX and RX positions are unknown. As there are two possible sequences of bits, it is easy to find the good one making an autocorrelation from these two sequences (on two seconds duration, for example) and choosing the one with the largest autocorrelation.

**Example**:
I receive: `1 x1 0 x2 1 1 1 0 1 1 0 1 y1 1 y2 0`
Two sequences are possible: either starting with the first " 1 " or starting with the x1 bit (it is supposed a distance of 5 bits).

First hypothesis: sequence beginning with the first bit " 1 ":
1/1, 0/0, 1/1, 1/1, 1/1, 0/0

Second hypothesis: sequence beginning with the bit x1:
x1/1, x2/1, 1/0, 0/y1, 1/y2

As x1, x2, y1, y2 bits are worth 0 or 1, the autocorrelation will be maximum with the first sequence so the first "1" will be chosen as being the DX position.

The determination of the DX position will be done regularly (two times per second for example).

Once synchronized, the more probable bit will be the mean bit:
Mean bit = (DX bit +RX bit) / 2 .
Suppose I receive a " 1 ". By convention, I must receive a 1000 level (equivalent to a 180 ° phase reversal) for " 1 " (it would be -1000 for a " 0 "). I note that the received DX bit is worth -100 (instead of 1000, so I would conclude that it is a " 0 ") and the received RX is worth 700 (instead of 1000 but I would conclude that it is a " 1 "). There is an ambiguity. The mean bit is worth 400, so the decision would be " 1 " and it would be the good one.
In this example, the false DX bit is corrected by the good RX bit. So the error rate is reduced by a factor from 2 to 10 depending on the conditions (QRM, QSB, ionospheric Doppler modulation...).

## PSK63F DESCRIPTION

The author Nino Porcino IZ8BLY has chosen a long set of characters with the 256 ASCII and ANSI 256 characters. The separation code is similar to the one used for PSKFEC31.
The transmission speed of 42 words per minute is very comfortable.
The 62.5 bauds modulation speed has been chosen to be homogeneous with PSK31.

The minimum Signal-to-Noise ratio is –12 dB, for a 2% error rate.

The synchronization is similar to the one of PSKFEC31: one must choose between 2 bits, the choice not being done, here, with an autocorrelation but by the best choice between 2 Hamming distances (see further on).

The windowing filter used to transmit the bits is the one of PSK31: the window is not rectangular but with the pattern of a raised cosine (pattern allowing a soft transition). Besides, if one looks at a PSK31 (or PSK63F) signal, it will be seen that bits have a round form. This form considerably reduces the transmission band but introduces some InterSymbol Interference (bits recover each over), which is not really a problem, in fact. The average power is 79 % of the maximum power.

## DESCRIPTION OF THE PROTOCOL

### *TRANSMITTING STATION*

Each bit is introduced in a convolutive coder. This type of module will deserve a paper for itself. Without going into details, it can be said that each bit passes in a serial shifting register with two parallel outputs. Between the input and the two outputs, there is a special logic based on XOR gate, logic that very competent mathematicians have selected as being the best one.

Note: a XOR gate does not do else that adding 2 bits (in base 2), rejecting the carry bit: 0+0=0, 0+1=1, 1+0=1 and 1+1=0.
To resume, for an input bit, there are two output bits at the register output. The " diversity " level is proportional to the register length, which is 7 bits here, against 5 in QPSK31 (PSK31 with a " phase quadrature " modulation), but 9 in Pactor 2.
The two parallel bits (dibit) are then set in series and transmitted successively.

### *RECEIVING STATION*

At the reception, the respective positions of bits are unknown. As for PSKFEC31, there are two possible bit sequences.

Supposing having found the good sequence, one must do what it is called a "deconvolution". It is the reverse operation of the convolutive coder: one starts with a serie of dependant bits and the goal is to determine the initial bit. The logical solution would be:
* to determine all the possible sequences,
* to calculate the (Hamming) distance between the received sequence of bits and each of the possible sequences. The shortest distance will give the sequence to select.

Note: for example, the distance between " 01 " and " 00 " is 1, the one between " 01 " and " 10 " is 2, and so on.
This (optimum) method needs much calculation power (although now...).
The quite often used method is the Viterbi decoding algorithm (closed to the optimum) which at, each stage, get rid of the less probable sequences, which reduces the calculation need but not exempt from the Hamming distance calculation for the survivor sequences.

To come back to the previous problem, this Hamming distance is precisely used to determine which are the first and the second bit of a " dibit ". Between 2 possible bit sequences, the one which generates the minimum Hamming distance will be selected.

As soon as the bit is determined, the process is the same as for PSKFEC31:
* detection of the separation code between characters,
* character determination.

## DETECTION OF SIGNAL IN NOISE AND DISPLAY

It is difficult to display on the " waterfall " a very weak signal in noise. The only way to do it, is to average the spectrums. For example, in MULTIPSK, the displayed spectrum for PSKFEC31 is an average of 3 spectrums. As a signal (as PSKFEC31) is coherent and noise is non-coherent, the average of noise becomes nil and the average of the signal remains constant, so the more the number of spectrums averaged is, the best the signal comes out from noise.
By doing this, two problems appear:
1) a time delay is introduced,
2) as averaging is equivalent to a " low pass " filter, phenomena's becomes slow...particularly, the increasing and the decreasing of the signals (there is a sort of information persistence).

In conclusion, it's a compromise between keen detection and time delay...and it's one of the limits for detection of signal in noise: to reach a S/N ratio of 0.001 (-30 dB) for example, it will be necessary to have a baud rate of 1 baud so a CW equivalent speed of 1 to 2 words/min....and a time delay of 30 seconds or more !