

Patrick Lindecker (F6CTE) the 8 of may 2004 (mail: f6cte@aol.com)
Deutsche Bearbeitung: Dieter Zimmermann (DL2RR)

In diesem Artikel beschreibt der Author zwei digitale PSK-Betriebsarten (Phase Shift Keying) für den "Tastatur zu Tastatur"-Betrieb, um die Probleme aufzuzeigen, die bei der Entwicklung neuer digitaler Betriebsarten auftreten können:

PSKFEC31 und PSK63F, beide mit Fehlerkorrektur ausgestattet.

Diese Betriebsarten und noch weitere sind in einer Software enthalten, die vom Verfasser entwickelt wurde. Das Name des Programmes ist Multipsk und kann von folgender WEB-Seite heruntergeladen werden: <http://members.aol.com/f6cte/>

Einführung

Digitale Kommunikation auf Kurzwelle ist manchmal schwierig bis unmöglich, und zwar wegen

- QRM von anderen Funkstationen,
- Fading (QSB),
- Ionosphärische Doppler-Modulation (für niedrige Übertragungsgeschwindigkeiten wie bei PSK31 und noch gravierender bei PSK10),
- Zeitüberschneidung identischer Signale, die auf verschiedenen Wegen über die Ionosphäre ankommen (besonders empfindlich bei hohen Übertragungsgeschwindigkeiten),
- Ein zu niedriges Signal/Rausch-Verhältnis (das Signal geht mehr oder weniger im Rauschen unter).

Das Ziel dieser experimentellen Betriebsarten mit langsamer bzw. durchschnittlicher Übertragungsgeschwindigkeit und Fehlerkorrektur ist es, eine unmittelbare Kommunikation zwischen OMs auf Kurzwellen zu erlauben, und zwar

- ohne Daten- oder Bildübertragung (zum Beispiel...) mit viel weniger Fehlern als bei Benützung von BPSK31 (eine grundlegende Betriebsart, die eine Art Referenz darstellt),
- durch eine erhöhte Empfindlichkeit BPSK31 überlegen zu sein,
Es sei daran erinnert, dass der BPSK31 Modus entworfen wurde, um Kommunikationen mit einem Signal-zu-Rausch-Verhältnis bis zu -11,5 dB sicherzustellen, so dass noch Signale, die bis zu ca. 14 mal schwächer als das Rauschen sind, erkannt werden (bezogen auf eine Bandbreite von 3 kHz). Dies entspricht einer Morsegeschwindigkeit von 37 Worten/min bei Versalien und 51 Worten/min bei Kleinbuchstaben.

Allgemeine Beschreibung

- 1) Alle diese Betriebsarten (einschliesslich BPSK31) sind BPSK-Modi. BPSK ist eine Modulation mit zwei Phasen: 0° und 180° (gegenüberliegend). Diese Modulation kann auch als eine Amplitudenmodulation mit zwei inversen (entgegengesetzten) Pegeln angesehen werden. Dies ist zwar nicht die beste Art, um Informationen zu übertragen, aber sie hat den Vorteil, einfach zu funktionieren und besonders in Rauschen verhältnismässig effizient zu sein.

Um jede Phasenreferenz zu umgehen (da man abhängig von den Phasenlagen messen muss) wurde ein differentialer Modus gewählt:

ausgewertet wird nicht die reine Phase, sondern deren Änderung, also die Invertierung bzw. Nichtinvertierung eines Bits im Vergleich mit seinem Vorgänger, entsprechend diesem Diagramm:

$0^\circ \rightarrow 180^\circ$ oder $180^\circ \rightarrow 0^\circ$ für Phasen-Umkehrung

$0^\circ \rightarrow 0^\circ$ oder $180^\circ \rightarrow 180^\circ$ für Nichtumkehrung.

Die Bitfehlerrate wird leider auf diese Art und Weise verdoppelt.

Dieses Modulation erfolgt auf einem NF-Träger ("Basis Band" genannt) der wiederum die Modulation des HF-Trägers bewirkt.

Die Demodulation kann kohärent sein, tatsächlich ist sie aber "quasi-kohärent", weil die Bit-Synchronisierung vom Signal selbst durchgeführt und nicht aus einer externen Quelle abgeleitet wird (eine Lösung, die auch vorstellbar wäre).

Das Wort "kohärent" wird benutzt, weil eine bestimmte PLL (phase locked loop), "Costas-Loop" genannt, benutzt wird, um der Phase zu folgen. Allerdings beginnt die Phase mit einer Mehrdeutigkeit von 180° , so dass diese Definition nicht streng anwendbar ist.

Anmerkung: diese "Costas-Loop" ist eine Lösung unter mehreren (so auch eine des Verfassers). Das ursprüngliche Programm PSK31SBW (von G3PLX - Peter Martinez) zum Beispiel folgt nicht jeder Phasenlage, sondern misst einfach die durchschnittlichen Phasenzustand während einer Bitdauer (nicht-kohärente Erkennung).

- 2) PSK31, PSKFEC31 und PSK63F verwenden einen "Varicode" genannte Zeichenkode. Dieser Begriff ist das Gegenteil von "fixed length code" (ein Kode mit fester Länge). Wie beim Morsekode hängt die Länge eines Zeichens (in Bit) von zur Häufigkeit des Vorkommens in der Literatur ab. Es ist notwendig darauf hinzuweisen, dass der PSK31-Varicode weder dem des PSKFEC31- noch des PSK63F-Kodes gleicht. Vor jedem Zeichen steht eine bekannte Bitfolge als Trennungskode (separation code), zum Beispiel **011** in PSKFEC31 (mit **1** für

die Phasen-Umkehrung (phase reversal) und **0** für keine Phasenänderung (no reversal). Man braucht schliesslich weniger Bits, um einen standardmässigen Text zu übertragen. Allerdings ermöglicht ein fester Längencode das Entdecken einen fehlerhaften Bits (zum Beispiel bei einer schlechten “Bit-Synchronisierung”). Dieses Problem ist nicht ein fach und führt zu einer Reihe von Zeichenverlusten.

- 3) Die Übertragung kann in LSB oder in USB durchgeführt werden, aber zur Vereinfachung wird im allgemeinen USB benutzt, auch auf den niedrigen Bändern. Dies hat der Vorteil, dass durch Angabe des HF- und der NF-Wertes immer die gleiche Frequenz definiert ist. Zum Beispiel, man nennt eine HF-Frequenz von 7035 kHz (Anzeige des Transceivers) mit einer im Spektrum bzw. Wasserfall gemessenen NF-Ablage von 1000 Hz, dann ist die tatsächlich ausgesendete Frequenz 7036 kHz in USB. Im allgemeinen ist USB der Standard für alle digitalen Betriebsarten (ausgenommen bei 45-Baud-RTTY).
- 4) Die Übertragungsgeschwindigkeit ist meist 31.25 Baud oder ein Vielfaches (zum Beispiel 2 mal 31.25=62.5 Baud bei PSK63 (eine Erweiterung von BPSK31 auf 62.5 Baud) und PSK63F. Diese Geschwindigkeit entspricht einer Teilgrösse von 8000, da sowohl bei DSP- als auch in Sound-Karten die Abtastung mit 8000 samples/sec erfolgt (trotzdem verwendet Multipsk eine Abtastrate von 11025 samples/sec, die bei Sound-Karten Standard ist).
- 5) Die Synchronisation wird aus dem empfangenen Signal gewonnen, damit wird der genaue Zeitpunkt bestimmt, bei dem die Bitmessung fertig sein muss. Sie beruht im allgemeinen auf einer im Basisbandsignal angewandten Nichtlinearität. Der Verfasser quadriert zum Beispiel das Signal.

Anmerkung 1: das Basisbandsignal wird durch Demodulation eines NF-Signals erhalten, d. h. durch Unterdrückung des NF-Trägers mit einer Costas-Schleife und einem darauf abgestimmten Filter.

Anmerkung 2: die digitalen Prozesse sind natürlich die gleichen wie die analogen, ausser dass der Einsatz eines Computers interessantere Lösungen ermöglicht. So zum Beispiel kann man rekursive Tiefpass-Filter erster, zweiter oder vierter Ordnung entsprechend denen in analoger Technik einsetzen, aber auch lineare nichtrekursive Filter 500. Ordnung benutzen, siehe später.

- 6) Die Signalfrequenz kann driften, so dass man über eine automatische Steuerung der Frequenzdrift verfügen muss, welche die durchschnittliche Phasendrift

Anmerkung 1: das Basisbandsignal wird durch Demodulation eines NF-Signals erhalten, d. h. durch Unterdrückung des NF-Trägers mit einer Costas-Schleife und einem darauf abgestimmten Filter.

Anmerkung 2: die digitalen Prozesse sind natürlich die gleichen wie die analogen, ausser dass der Einsatz eines Computers interessantere Lösungen ermöglicht. So zum Beispiel kann man rekursive Tiefpass-Filter erster, zweiter oder vierter Ordnung entsprechend denen in analoger Technik einsetzen, aber auch lineare nichtrekursive Filter 500. Ordnung benutzen, siehe später. berechnet. Sobald ein Offset entdeckt wird, muss nur noch die "VCO"-Frequenz der Costas-Loop nachgezogen werden.

- 7) Das erzeugte Basisbandsignal ist durch ein per Vorgabe rechteckiges Fenster gefiltert, in welches der NF-Träger des Signales gesetzt wird (wie in CW, RTTY, AMTOR...). Die eingenommene Bandbreite ist proportional zur Schrittgeschwindigkeit (in Baud) mit einer $\sin(x)/x$ -Hüllkurve, das eine sehr grosse Bandbreite ergibt. Auf dem InterSymbol Interference Punkt (ISI) gesetzt, ergibt dies die beste Lösung, d. h. ein gegebenes Bit beeinträchtigt seinen Nachbar nicht. Durch dieses Prinzip wird im PSK-Modus der Durchlassbereich reduziert, wenn man ein Fenster-Filter mit einem weichen Übergang benutzt, das trotzdem ein kleines ISI erzeugt.

EINIGE ANDERE EXPERIMENTELLE BETRIEBSARTEN

Ursprünglich hatte der Verfasser eine Betriebsart namens PSK10 vorschlagen: die Übertragung erfolgt mit 10 Baud in BPSK, wie bei den bisherigen Betriebsarten. Eine kurzer Zeichensatz verbindet sich bei diesem Modus mit einem Prefix, der sicherer als der von BPSK31 ist. Wenn dieser Modus auch sehr empfindlich ist (minimales S/N = -17 dB), die Bitdauer (0.1 sek) ist doch zu gross im Verhältnis zur ionosphärischen Doppler-Modulation auf Kurzwellen.

Um die Fehlerrate zu reduzieren, hat der Verfasser für die PSKAM 10/31/50-Modi eine Lösung vorgeschlagen, bei der jedes Zeichens wiederholt wird (bei fester Bitlänge) wie in AMTOR FEC. Dieses Prinzip funktioniert sehr gut. Allerdings kann unter schlechten Bedingungen (zum Beispiel bei QRM) die Synchronisation verloren gehen. Da ein erneutes Synchronisieren unter schlechten Bedingungen schwierig ist, (bei einem 8-Bit-Zeichen muss man zwischen 16 Möglichkeiten auswählen ...), könnte die Dekodierung aussetzen (mit Zyklen von Synchronisationsverlust und Resynchronisieren). Abgesehen von diesem Problem ist die Fehlerrate im Vergleich zu PSK31 sehr gering.

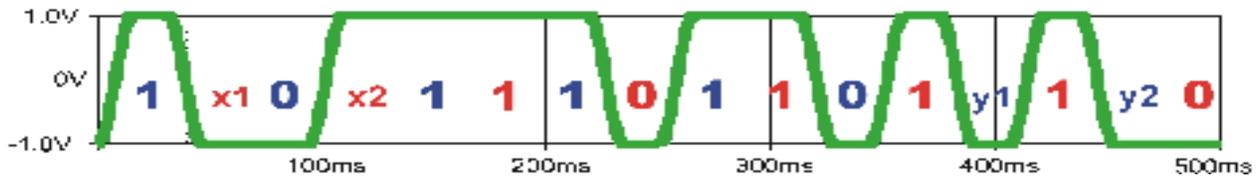
BESCHREIBUNG DES PROTOKOLLS

DER SENDETEIL

Jedes Bit wird um 13 Positionen versetzt wiederholt, nach dem gleichen Prinzip wie in AMTOR FEC, mit dem Unterschied, dass der Versatz bit- und nicht zeichenbezogen ist.

Nehmen wir an, dass eine **1** für ein "Phasen-Umkehrungs-Bit" (phase reversal bit) und eine **0** für ein "Nicht-Phasen-Umkehrungs-Bit (no phase reversal bit)" steht. Nehmen wir weiterhin an, wir hätten statt eines Abstandes von 13 Bit nur einen von 5 Bit und wir wollten **101110** übertragen:

DX	RX	DX ist die erste Position
1	x1	RX ist die zweite Position
0	x2	x1, x2 sind beliebige vorherigen Bits (0 or 1)
1	1	y1, y2 sind beliebige nachfolgenden Bits (0 or 1)
1	0	
1	1	
0	1	
y1	1	dann wird folgendes Muster gesendet (erst DX dann RX):
y2	0	



DER EMPFANGSTEIL

Beim Empfang ist die Position von DX und RX unbekannt. Da es zwei mögliche Bitfolgen gibt, es ist leicht, die richtige zu finden, indem man eine Autokorrelation von beiden Folgen (zum Beispiel über eine Dauer von zwei Sekunden) durchführt und wählt die mit der grössten Übereinstimmung (= Autokorrelation).

Beispiel: Ich empfangen **1 x1 0 x2 1 1 1 0 1 1 0 1 y1 1 y2 0**

Zwei Folgen sind möglich: entweder ab der ersten **1** oder ab **x1** (es soll wieder ein Abstand von 5 Bit angenommen werden).

erste Hypothese	?	?	?	?	?	?									
1 x1	0 x2	1	1	1	0	1	1	0	1	y1	1	y2	0		
zweite Hypothese	?	?	?	?	?										

Erste Annahme: die Bitfolge beginnt mit dem Bit **1**? : 1/1, 0/0, 1/1, 1/1, 1/1, 0/0

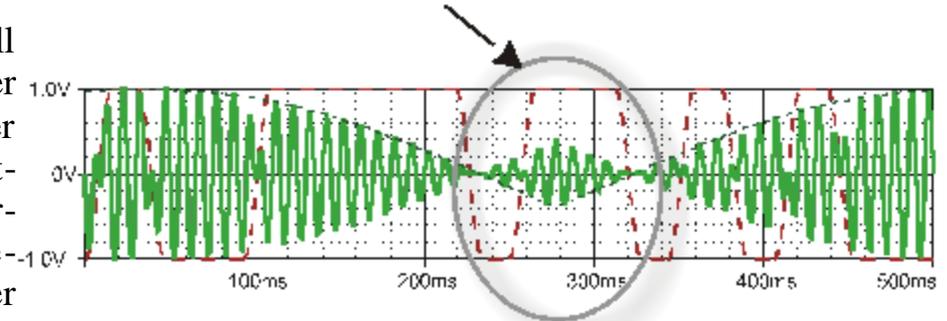
Zweite Annahme: die Bitfolge beginnt mit dem Bit **x1**? : x1/1, x2/1, 1/0, 0/y1, 1/y2

Da die Bits x_1 , x_2 , y_1 den Wert **0** oder **1** haben können, wird die Autokorrelation bei der ersten Folge ein Maximum ergeben, so dass die erste **1** als DX-Position gewählt wird. Die Bestimmung der DX-Position wird regelmässig durchgeführt (zum Beispiel zwei mal pro Sekunde). Wenn einmal synchronisiert wurde, ist das **Durchschnittsbit = (DX + RX) / 2** mit grosser Wahrscheinlichkeit das richtige Ergebnis.

Angenommen, eine **1** soll empfangen werden. Per Definition muss dies einer 180° -Phasendrehung entsprechend einen normierten Pegel von 1000 ergeben, bei einer **0** würde er -1000 betragen).

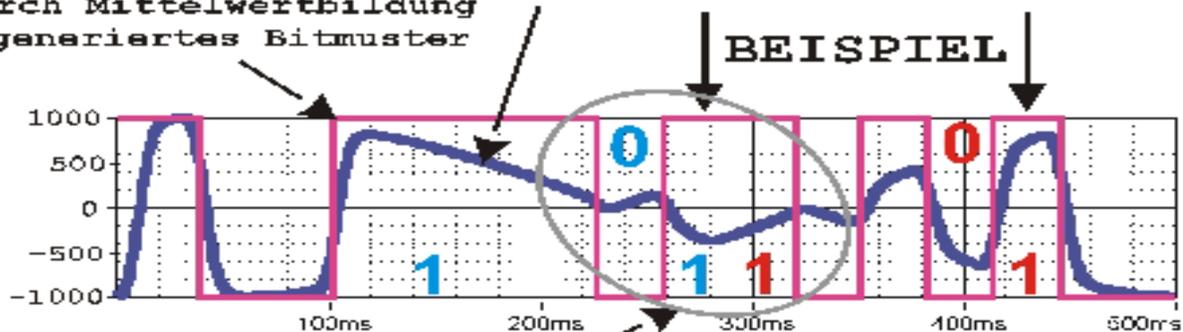
Nehmen wir weiter an, das empfangene DX-Bit hat aber nur den Wert -300 und wird dementsprechend als **0** dekodiert, das empfangene RX-Bit hat den Wert 800, ist also per Festlegung eine **1**. Es besteht somit eine Mehrdeutigkeit. Der gemittelte Wert der beiden Bits ergibt 250 und entspricht somit einer **1**, was in diesem Fall auch richtig ist.

Phasenfehler durch Mehrwegübertragung



dekodiertes Signal

Durch Mittelwertbildung regeneriertes Bitmuster



Bitfehler durch Mehrwegübertragung

In diesem Beispiel wird das falsche DX-Bit vom richtig erkannten RX-Bit korrigiert.

Die Fehlerrate wird so, je nach den Bedingungen (QRM, QSB, ionosphärische Doppler-Verschiebung...), um einen Faktor von 2 bis 10 reduziert.

PSK63F BESCHREIBUNG

Der Verfasser dieser Betriebsart, Nino Porcino IZ8BLY, wählte eine lange Zeichenfolge mit dem 256-ASCII- bzw. ANSI-Zeichensatz. Der Trennungskode ist ähnlich dem, der für PSKFEC31 benutzt wird und die Übertragungsgeschwindigkeit von 42 Worten pro Minute ist sehr komfortabel.

Die 62.5 Baud Übertragungsgeschwindigkeit wurde gewählt, um der von PSK31 zu entsprechen.

Der minimale Signal/Stör-Abstand ist -12 -dB für eine 2%-Fehlerrate.

Die Synchronisierung ist der von PSKFEC31 ähnlich: man muss hier zwischen zwei Bits wählen, die hier nicht mit einer Autokorrelation, sondern durch die beste Auswahl zwischen zwei Hammingdistanzen (siehe weiter vorn).

Das für die Bitübertragung benutzte Filter-Fenster ist das gleiche wie bei PSK31: das Fenster ist nicht rechteckig sondern mit kosinusförmigen Flanken (dieses Muster erlaubt einen weichen Übergang). Wenn man ein PSK31-Signal (oder PSK63F) betrachtet, merkt man, dass die Bits eine rundliche Form haben. Diese Form reduziert den Durchlassbereich ganz besonders, erzeugt aber Interferenzen zwischen den Zeichen, sogenanntes. "InterSymbol Interference". Das stellt aber kein wirkliches Problem dar, da die Bits nach jeder Übertragung regeneriert werden. Die durchschnittliche Leistung ist 79% der maximalen Leistung.

PROTOKOLBESCHREIBUNG

DER SENDETEIL

Jedes Bit durchläuft einen 'Convolute'-Kodierer. Die Beschreibung dieses Modules würde einen eigenen Artikel für sich verdienen. Ohne ins Detail zu gehen, kann man sagen, dass jedes Bit ein seriellen Schieberegister mit zwei parallelen Ausgängen passiert. Zwischen dem Eingang und den beiden Ausgängen befindet sich eine auf einer XOR-Verknüpfung basierende Logik - einer Logik, die von sehr fähige Mathematikern als die beste ausgewählt wurde.

Anmerkung: ein XOR-Gatter macht nichts weiter als die binäre Addition zweier Bits ohne Übertrag: $0+0=0$, $0+1=1$, $1+0=1$ und $1+1=0$.

Zusammenfassend bedeutet dies, dass es pro Eingabebit zwei Ausgabebits am Registerausgang gibt. Dieser "diversity"-Pegel ist proportional zur Registerlänge, in diesem Fall 7 Bit, gegenüber 5 Bit bei QPSK31 (PSK31 mit einer "Phasen-Quadratur"-Modulation), und 9 Bit bei PACTOR 2.

Die zwei parallelen Bits (Dibits) werden dann in Reihe gesetzt und aufeinanderfolgend übertragen.

DER EMPFANGSTEIL

Beim Empfang sind die jeweiligen Position der Bits unbekannt. Wie bei PSK-FEC31 sind zwei mögliche Bitfolgen möglich. Angenommen, dass die richtige Folge gefunden wurde, muss man das tun, was "deconvolution" genannt wird. Es handelt sich dabei um den umgekehrten Betrieb des "Convolution"-Kodierers: man startet mit einer Serie von 'abhängigen' Bits mit dem Ziel, das Startbit zu bestimmen. Die logische Lösung würde wie folgt sein:

- alle möglichen Folgen bestimmen
- berechnen des (Hamming) Abstandes zwischen der empfangenen Bitfolge und jeder der möglichen Folgen. Der kürzeste Abstand ergibt die zu wählende Folge.

Anmerkung: beispielsweise ist der Abstand zwischen **01** und **00** gleich 1, der zwischen **01** und **10** gleich 2, und so weiter.

Diese (optimale) Methode benötigt viel Rechenleistung (obwohl jetzt...).

Eine sehr oft benutzte Methode ist der Viterbi Dekodierungs-Algorithmus (nahezu das Optimum), welcher an jeder Stufe die unwahrscheinlicheren Folgen entfernt, was den notwendigen Rechenaufwand reduziert, aber nicht von der Berechnung der Hamming-Distanz befreit.

Um zum bisherigen Problem zurückzukommen, diese Hammingdistanz wird genau dazu benutzt, um zu entscheiden, welche das erste und das zweite Bit eines "Dibit" ist. Zwischen zwei möglichen Bitfolgen wird diejenige gewählt, die die minimale Hammingdistanz erzeugt .

Sobald das Bit bestimmt ist, ist der Prozess der gleiche, wie bei PSKFEC31:

- Erkennen des Trennkodes zwischen den Zeichen
- Zeichenbestimmung.

SIGNALERKENNUNG UND -DARSTELLUNG IM RAUSCHEN

Ein sehr schwaches verrauschtes Signal ist im "Wasserfall" immer schwierig darzustellen. Die einzige Art, dies zu tun, ist die Mittelwertbildung mehrerer Kurzzeit-Spektren. Zum Beispiel ist das dargestellte Spektrum bei PSKFEC31 in MULTIPSK der Durchschnitt von 3 Spektren. Da ein Signal (wie PSKFEC31) kohärent und Rauschen nichtkohärent ist, geht der Mittelwert des Rausches gegen Null und der des Signales bleibt konstant, je länger also die Spektren gemittelt werden, desto besser kommt das Signal aus dem Rauschen heraus.

Wird dies getan, entehen jedoch zwei Probleme:

1. eine Zeitverzögerung wird eingeführt,
2. da Mittelwertbildung einer Tiefpass-Filterung gleichkommt, werden alle Vorgänge langsam...insbesondere die ansteigende und abfallende Flanke des Signals. (es gibt eine Art Informations-Persistenz).

Schlussbemerkung: man muss einen Kompromiss zwischen exaktem Erkennen und Zeitverzögerung treffen ... und es gibt Grenzen beim Erkennen von Signalen in Rauschen: um zum Beispiel ein S/N-Verhältnis von 0.001 (-30 dB) zu erreichen, wird eine Übertragungsrate von 1 Baud notwendig sein, entsprechend einer CW-Geschwindigkeit von 1 bis 2 Worten/Min. ... und eine Zeitverzögerung von 30 Sekunden oder mehr!