High Performance BPSK31

Ideas for a New Generation

John A. Gibbs, NN7F
NN7F@ARRL.net

Abstract: Advancements in Personal Computing capabilities and signal processing theory present an opportunity to improve Amateur Radio's most popular digital mode, BPSK31. Error reduction methods are investigated and simulations are compared to theory. Areas for further investigation are suggested.

Keywords: BPSK31, DSP, Digital Radio

Introduction

This paper is intended to stimulate discussions and encourage improvements in Amateur Radio Digital communications. Rather than introducing another digital mode, the focus is on improving existing modes through the development of new error reduction methods. We will show there is both motivation and opportunity for this effort. To lend credibility and encouragement, the results of some receiver signal processing improvements are presented for the most popular Amateur Digital Radio mode, BPSK31. This short paper is by no means comprehensive, but some areas for further research for all 'chat' modes are suggested.

Why investigate BPSK31? To answer this question we need to look at what has happened in the last 15 years. In the December 1998 issue of RadCom Peter Martinez, G3PLX, introduced soundcard-based PSK31. Moe Wheatley, AE4JY, followed with many improvements; perhaps the most significant was providing the program as a Dynamically Linked Library (DLL). PSKCore.dll has been used in many PSK31 programs (Table 1) and enabled developers to focus on the user features and operational enhancements without having to understand the details of communication theory and signal processing.

However, the availability of PSKCore could be viewed as a double-edged sword. Certainly the wide range of PSKCore-based software has been a major contributor to the rapid growth of PSK31. But perhaps the ease of using a canned signal processing solution has slowed the advancement of PSK31 signal processing technology[1].

Since the introduction of PSK31, new many soundcard digital modes have been developed that optimize various combinations of modulation, bandwidth, data rates and error correction. But none of these innovative solutions to the challenges of HF data communication have managed to dethrone BPSK31 as the most popular mode.

WinPSK	Zakanaka
WinPSKse	Logger32
WinWarbler	DXPSK
HamScope	W1SQLPSK
MultiPSK	PSK31Deluxe
QuickPSK	SmartPSK
WO-PSK	RCKRtty
DX4WIN	YPLog
N1MM Logger	

Table 1: Some PSKCore Programs [Ref 1]

Why is BPSK31 so much more popular than these newer modes? Ed Sack, W3NRG, compared digital modes in a DCC 2007 paper [2]. He showed that performance of BPSK31 is excellent, one of the top digital modes. For these ratings he measured the actual transmission performance of forty amateur digital modes under weak signal conditions. In comparing transmission rate, bandwidth and copy percentage he found that by his metrics BPSK31 came out best in overall performance. These excellent

---

[1] Moe has made PSKCore source code freely available, so the responsibility for lack of further innovation lies elsewhere.

results may partially explain the popularity of BPSK31. But one should also note he showed that BPSK31 had more copy errors than the FSK modes. *A fundamental question this paper addresses is whether a significant improvement in BPSK31 decoding accuracy is possible and practical*.

Opportunity

We shall see that such improvements are possible because of two major trends during the past 15 years. First is the well known Moore's Law; a 1965 observation by Gordon Moore that the progress of technology has grown the number of transistors on an IC at an exponential rate. This trend has continued over the decades and has averaged a doubling of transistors every two years. Equally important to DSP is the fact that as the area of transistors shrinks, the reduced capacitance allows clock speeds to increase without increasing power dissipation. David House, another Intel executive, noted that the combination of these two effects would cause computer performance to double every 18 months [3]. That is more than a *thousand times increase* in DSP power since PSK31 introduction!

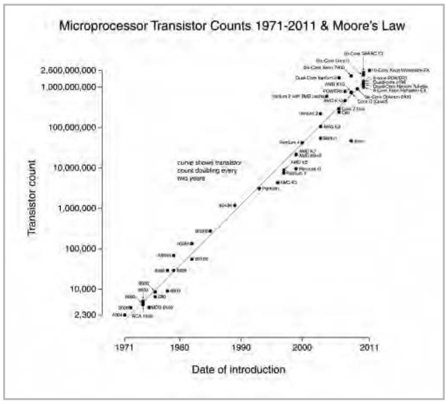

Figure 1: Moore's Law [ref 3]

The second major trend is the advancement of signal processing algorithms for correcting real world propagation conditions. Moore's Law has also driven this trend as increased DSP capability made possible real-time implementations of decoding algorithms with ever increasing complexity. Seemingly hundreds of articles and dissertations have been written on the subject. Most address the problems of reducing the destructive effects of noise and multipath in the mobile, high-speed data environment. But even with the improvements of Moore's Law, the computational requirements of these algorithms often exceed current DSP capabilities. So much of the work has been focused on reducing the complexity of these new DSP algorithms. Reduced computational complexity means one can process real time data at higher baud rates with lower battery power consumption and hence provide a competitive advantage in high-speed file transfer applications like Web pages, high-resolution pictures and video.

Motive

While the commercial market has an insatiable appetite for speed, the agencies regulating Amateur radio have limited the data rate that we can use at HF. Therefore the current direction of research is not directly applicable to HF Amateur Radio. Stated more positively, *there is both motive and opportunity for Amateur Radio to once again to make technical contributions to the communication field, this time in data communication over noisy, fading HF ionospheric channels.*

There are other ways that our low-speed HF data communications differ from the current research directions.

- File transfers require Bit Error Rates (BER) in the part-per-million region. Amateur chat modes work very well at a 1% Character Error Rate (CHER) as the operator can correctly interpret many transmission errors.
- Commercial applications are restricted by regulations and licensing to specific frequencies. Amateurs can change to different bands for improved propagation.
- Commercial applications typically have a direct propagation path and possibly several reflected paths, which yields Ricean fading. Except for nearby communications, Amateur Radio HF propagation does not have a direct path. The lack of a direct path leads to Rayleigh fading which requires different receiver signal processing.

- File transfers can tolerate a relatively long latency compared to chat modes. To combat multipath fading, commercial applications often spread the data over a time interval much longer than the fading rate by a technique called interleaving. Then forward error correction techniques can be used to reconstruct an accurate copy of the transmitted file. In Amateur Radio when copy is poor due to QRM or QSB we repeat an important message (*de NN7F, NN7F*) or use the equivalent of ACK/NACK (*How copy? -- Copy 100%*).
- At the GHz frequencies used in commercial applications, it is practical to use antennas that are widely spaced in terms of wavelength. Research is invested heavily in the Multiple-Input, Multiple-Output (MIMO) field that optimally combines multiple transmitter and receiver antennas. As discussed later, modifications of this technology could be used to reduce fading using vertical and horizontal receiving antennas and low-cost DC-IF receivers.

Scope

This paper will be limited to investigating BPSK31 receiver DSP. Limiting the scope to receiver changes also allows us to maintain backwards compatibility with the current BPSK31 operations. Fortunately, we shall see that there are several areas for significant improvement in the receiving software.

Computer simulation work by Daniel Crausaz, HB9TPL, is used extensively in this paper [4]. In his QEX article, he used MultiPSK and simulated ionospheric communication using PathSim software. In this paper these results will be compared to theory and used as a baseline for the proposed signal processing improvements.

Some of these proposed DSP algorithms have been recently implemented and the results are presented below. Moe's WinPSK open source code was used as the development platform. The code was modified to send a continuous series of 8's in the demo mode. (This was the test mode used in Daniel's work[2].) Access to Moe's source code also allowed adding routines to compute and display of Character Error Rate (CHER) as well as the ability to speed up the simulation. It was therefore practical to measure CHER over 10,000 characters as compared to 200 characters in Daniel's work and thereby reduce the statistical uncertainty in the results.

We begin discussing receiver performance improvements under the best propagation conditions; what communication engineers call the AWGN channel. Most of the remainder of this paper is devoted to this important case. Then we will address ways to improve performance under QSB conditions. With appropriate modifications, we will find that much of what is developed for the AWGN case is applicable to the fading channel case.

AWGN Channels

Communication engineers call the total path of our transceivers, antennas and the propagation path a "channel". An Additive White Gaussian Noise (AWGN) channel is a benign environment, no QSB or QRM, just QRN that is heard as the background noise. This happy state of affairs happens rather often. This is fortunate as the signal processing in existing BPSK31 software was designed to work in this environment. We shall see that our current BPSK31 software works poorly or not at all in the presence of rapid fading. Improvements for the case of rapid QSB are covered in the next section.

---

[2] Note that the average number of bits per character in BPSK31 is 6 bits, whereas the character '8' has 11 bits. Therefore the following graphs show about twice the expected average message error rate.

To understand the performance of BPSK31 receivers, we must first understand that BPSK31 bits are sent differentially. For BPSK31, if a ‘0’ is sent, the phase is reversed. If a ‘1’ is sent, the phase between the two bits is unchanged. To detect this phase shifting, PSKCore uses a scheme called Differential Detection where the phase of the most recent bit is compared to the phase of the previous bit. This simple detector is robust, but basically twice as noisy as more sophisticated techniques. We shall see that with new algorithms and the PC signal processing power available today, this noise penalty can be greatly reduced.

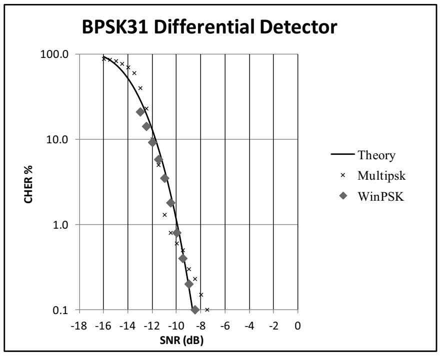

Figure 2: Current Performance

Figure 2 shows that in the presence of AWGN, Daniel’s MultiPSK measurements and new simulations with WinPSK behave in a manner consistent with theory[3]. Particularly over the CHER range of interest there is excellent agreement between theory and results.

It has long been recognized that the ideal way to demodulate PSK is for a ‘genie’ to somehow have an exact copy of the (suppressed) carrier phase. This is called Coherent Detection and is often used as a benchmark. In Figure 3, Daniel’s AWGN measurements are compared to the results of a theoretical coherent detector. It is clear that there is room for improvement in the error rate.

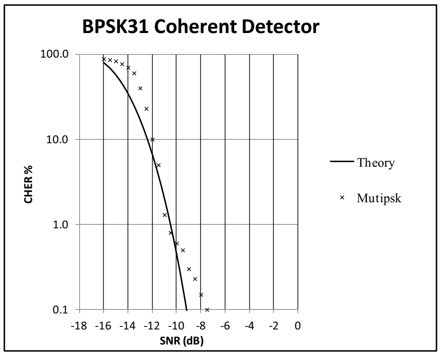

Figure 3: Ideal Performance

MLSE – A practical way to approach the performance of coherent detection is to compare the received sequence with all possible sequences that could have been sent by the transmitter and select the most likely sequence. This is called Maximum Likelihood Symbol Estimation (MLSE) [5]. While the complexity of these comparisons grows exponentially with the length of the sequence, adding terms has diminishing returns so complexity can be kept reasonable. Figure 4 shows measurements made with the modified WinPSK software and the theoretical results. They show that 4 bits is sufficient to approximate Coherent Detection [6]. For low additional complexity we have improved the error rate 2-3 times with the largest improvement coming at high error rates where it is most needed.

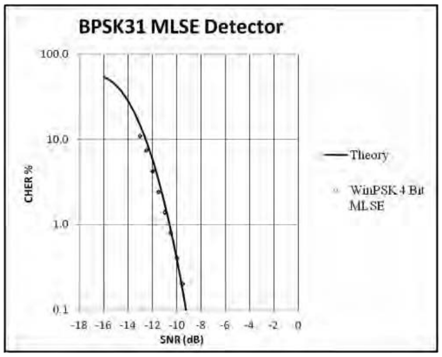

Figure 4: Improved Performance

Estimated Data Feedback - Another practical technique sometimes used is to feedback the previously estimated data to remove the modulation by multiplication. A particularly appealing version of this technique uses the recursive least squares algorithm to predict the carrier phase. This technique can not only improve the AWGN and fading performance but can also remove tuning errors. [7]

Is coherent demodulation of BPSK31 the fundamental performance limit? If like high speed commercial applications discussed earlier, we were sending random data then we would have reached a limit. But the text we are sending is not random, and the encoding we are using has properties that we can exploit. This information we have about the data we are receiving is called by communication engineers ‘a priori

---

[3] The scales used in this and the following figures are adopted from Crausaz. The horizontal scale is the signal to noise measured in a 3kHz bandwidth, the same as what we hear from the receiver when listening to PSK31. The Character Error Rate on the vertical scale is the percentage errors we see in the output text (as opposed to the traditional BER measurement.)

information'. Useful a priori information might include the statistical frequency of letters and letter combinations in the English language (the lingua franca of Amateur Radio) and the properties of Varicode, the coding used in BPSK31.

Varicode compresses the message by using short codes for the most commonly used letters (highest character frequency) as seen in Figure . In the sample text analyzed for this graph, the space character occurs most often (16% of the characters are spaces) and therefore is encoded with only 3 bits. [8] Likewise, the character "e" occurs almost 10% of the time and is encoded with 4 bits. 'Z' occurs only 0.02% of the time and so is encoded with 12 bits.

Varicode is a variation on a Fibonacci code, one of the 'prefix free' codes. Coding theory shows that for the statistical distribution of characters in the English language, the minimum average length of an ideal prefix free code is 5 bits. Varicode comes in one bit longer at average length of just over 6 bits. This extra bit occurs because the beginning of a Varicode character is indicated by sending a '1' after at least two consecutive '0's of the previous character[4].

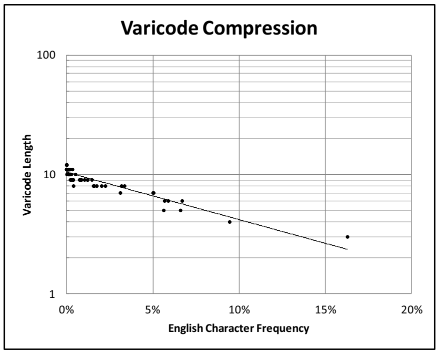

Figure 5: Varicode Compression

An inspection of the code words shows that no (non-extended) printable Varicode character has more than nine consecutive '1's. Similarly, no printable character has more than ten bits before detecting the "00" that ends every character. If these limits are exceeded, we must have a decoding error earlier in the codeword. But how can we decide which bit is in error?[5]

To answer that question we need to know the reliability of each of the bit decisions the decoder makes. Probably all the current BPSK31 programs and certainly all that are based on PSKCore.dll use what is called a hard output decoder. That is, the decoder decides if a '1' or a '0' was sent and passes that binary information on to a routine that identifies the character sent. In contrast, a soft output decoder not only makes a decision on the bit sent, but sends a number representing the confidence level of that decision.

Specific errors that could be detected and corrected with soft decoding include:

- Decoded as special ASCII character. (Except LF, CR and backspace.)
- Exactly three zeros in a row (likely to be decode error on first or last "1" of Varicode char.)
- Greater than ten bits before "00" (if extended characters are ignored)
- Greater than nine consecutive '1's

In addition, the statistical properties of Varicode and the English language could be used to detect and correct probable errors. For instance, an inspection of Varicode shows roughly equal probabilities of "1" or "0" so no a priori information is available. But the conditional probabilities contain extra information. Specifically, a "1" is twice as likely as a "0" to follow a "0". This information could be used by the detector in deciding what bit was probably sent after a "0" was detected in the previous bit.

Message Sent	100 (spc) + 1100 (e)
Bit 2 error	1101100 (l)
Bit 3 error	1011100 (s)
Message Sent	11010101100 (8)
Bit 2 error	100 (spc) +10101100 (h)
Bit 3 error	11110101100 (j)

Table 2: Possible Decoding Errors

---

[4] This extra bit allows idle message string of zeros to be sent if there is a pause in the live keyboard typing. This continuous flow of bits is important to keep the receiver synchronized with the transmitter.

[5] PSKCore (ver 1.21) ignores invalid Varicode characters and does not attempt to correct the errors.

The statistics of the English language can also be used to reduce the error rate. The longest Varicode characters represent uncommon letters. If a bit is decoded incorrectly, two more common letters could be interpreted as one less common letter. Similarly, if a less common letter is decoded incorrectly, it may be interpreted as two more common (short) characters. Examples are presented in Table 2

In some cases we can decrease the chance of these errors because we know the probability of these letters and we can estimate the error probability of each bit in the message using soft detection and noise power estimation. The degree to which this statistical processing can actually improve reception needs to be investigated.

Fading Channels

A much larger issue for PSK31 than the AWGN channel case is communicating over a Rayleigh fading channel. Rayleigh fading occurs when there are a large number of paths through the ionosphere that have approximately the same loss, but different path lengths. These multiple paths cause time varying phase shifts and deep amplitude fades to the narrow bandwidth PSK31 signals.

Condition	***Freq. Spread (Hz)***
Flat Fading	0.2
Good	0.1
Moderate	0.5
Poor	1.0

Table 3: CCIR Recommendation 520-1

Daniel has shown that even for a CCIR ‘Good’ channel, BPSK31 performance is *degraded 15dB* from an AWGN channel. Further, he states that it is unusable during Moderate and Poor conditions [4].

As expected from theoretical work, Daniel’s simulations also show that because of the low PSK31 data rate, the difference in the time delay of these multiple ionospheric paths does not contribute to CHER under propagation conditions normally encountered. Instead his simulations show that Doppler Spread is the cause of BPSK31 reception problems under fading conditions [4].

The theory and simulations in academic articles agree with these observations. They show that as Doppler spread increases, the conventional differential detector used in today’s PSK31 software has an error floor that is independent of signal strength. This means that *no matter how much you increase the transmit power, the CHER will not be reduced* (Figure). The faster the fading, the higher the error floor will be. This explains Daniel’s observation that Moderate and Poor propagation conditions are unusable for BPSK31. Fortunately, detectors that are only slightly more complex than the detectors used in today’s PSK31 give far better fading performance.

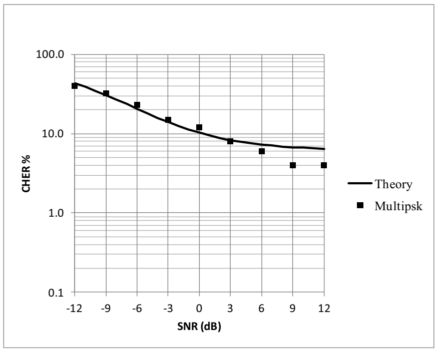

Figure 6: CCIR ‘Poor’ Conditions

The ionospheric Doppler spreading error floor is caused by the carrier phase changing from bit to bit. This spreading, of course, also widens the signal bandwidth. Therefore the matched filter used in the AWGN case is no longer optimum and the sample rate is too low compared to the widened signal.

A solution is to remove the frequency spreading before the bit filtering, decimation and detection operations. This requires real time estimation/prediction of the channel phase shifting. The literature shows that first-order fading models can effectively eliminate the error floor under moderate and poor channel conditions. These simple models can be incorporated in the MLSE and data feedback solutions presented for the AWGN case.

Diversity Reception

Unfortunately, while the CHER floor can be eliminated with modern DSP, the 15dB degradation in SNR noted by Daniel under fading conditions still occurs. To minimize deep amplitude fades, diversity techniques are needed. The principle is that the message reaches the receiver(s) via multiple paths and the receiver(s) combine the information to give the best estimate of the data sent. For BPSK31 the paths can be different antennas, either by using vertical and horizontal antennas or widely spaced antennas with the same polarization. When one of the paths is in a deep fade, the other one may not be, giving a higher resistance to fading than if only a single path was used.

Maximum performance when using multiple antennas requires multiple receivers. If a single receiver is switched between antennas, the receiver could lose synchronization because of the different propagation path lengths. Rather than just voting on which receiver has the best signal, the BPSK31 software should optimally combine the signals. This could be done using techniques developed for MIMO technologies recently being adopted for commercial high speed data transmission.

It would be difficult and expensive to use multiple synchronized Amateur superheterodyne receivers. But a Software Defined Radio (SDR) can have low hardware costs and its technological strengths are well matched to narrow bandwidth digital modes. BPSK31 signals are by agreement only transmitted in fixed 3kHz bands, so it is easy to implement inexpensive, fixed-frequency local oscillators with very low phase noise[6]. Low noise gain at low frequency IFs can be easily implemented. With proper attention to minimizing distortion throughout the entire receiver, hardware AGC is not needed. This is important because the AGC circuit can cause phase shifts that vary with the signal level.

Conclusion

Advancements in theory and PC processing power present us with the opportunity to improve our most popular digital mode and advance the state of the art. These improvements can be implemented in the popular open source PSKCore.dll and immediately incorporated into a wide range of PSK31 software. These improvements can also be extended to other Amateur digital communication modes including QPSK31 and RTTY. This paper has shown that such improvements are possible and that there are many opportunities today for Amateurs to make technical contributions to the communications field.

References

1. Richard Ferch, VE3IAY, http://storm.ca/~ve3iay/digital.html#PSK31
2. "Ranking Digital Modes for a 'Stealth' QTH", Ed Sack, W3NRG, DCC 2007
3. From Wikipedia, "Moore's Law" entry, downloaded June 26, 2013
4. "Signal Resilience to Ionospheric Distortion of HF Digital Chat Modes", Daniel Crausaz, HB9TPL, QEX, Nov/Dec 2007
5. "Multiple Symbol Differential Detection of Uncoded and Trellis Coded MPSK", Divsalar, Simon, Shahshahani, JPL Publication 89-38, Nov 1989
6. "Digital Communications over Fading Channels: A Unified Approach to Performance Analysis", Marvin K. Simon, Mohamed-Slim Alouini, 2000, John Wiley & Sons
7. "Prediction-Based Decision-Feedback Differential Detection for MDPSK", Robert Schober et al.
8. Telegraphic alphabet for data transmission by phase shift keying at 31 Bd in amateur and amateur-satellite services, Recommendation ITU-R M.2034, 02/2013

---

[6] A return to crystal oscillators instead of noisy frequency synthesizers would reduce blocking dynamic range issues.