

# An All-software Advanced HF Modem for Amateur Radio'

Matthew Ettus, N2MJJ<sup>2</sup>  
Carnegie Mellon University  
733 Maryland Avenue, Apt. #3  
Pittsburgh, PA 15232  
mne@cmu.edu  
<http://www.andrew.cmu.edu/user/mne/>

## Abstract

The need for an inexpensive and robust replacement for 300 baud FSK on the amateur HF bands is apparent. A modem was developed which allows for greatly increased data bandwidth (up to 500bps), while at the same time allowing for increased reliability through the use of advanced modulation and coding methods. The entire system runs on a standard PC with a soundcard, under Linux. The only necessary hardware is a mechanism for keying the radio.

## Introduction

The main goals of this design were to create a modem which is suitable for use in the amateur HF bands, which would provide high throughput, and, at the same time maintain low error rates. Also important was making the system accessible to all amateurs without the purchase of additional hardware. The code had to be simple and efficient enough for the additional processor load to be unnoticeable to the user.

Most amateur radio HF modem designs have concentrated on confining transmitted signals to narrow bandwidths, usually 500 Hz or less. In order to obtain adequate throughput, higher order modulation schemes, such as 8- or 16-PSK are necessary, as in Clover-II. These non-orthogonal modulations require significantly more power to obtain the same bit error rates (BER).

Modern communication theory dictates the opposite approach -- spreading the signal over a significantly higher bandwidth, coupled with the extensive use of FEC (Forward Error Correction) coding. While this may seem counter-intuitive, it is a direct result of Shannon's channel capacity theorem:

$$C = B \log_2(1 + S/N)$$

C, channel capacity, increases linearly with B, bandwidth, but only with the log of the received power (a function of S/N, the signal to noise ratio). Thus, throughput can be increased by exponentially raising power to get more bits per second per hertz (for bandwidth efficiency), or by increasing bandwidth, for more bits per unit energy (for power efficiency).

## Theory

### Modulation

- 
- 1 This paper is submitted for use in the Proceedings of the 16th ARRL and TAPR Digital Communications Conference
  - 2 This project was originally started as a class project for EE55 1 in the Spring of 1997. The team consisted of Matthew Ettus, Micheal **Batz**, and Michael Lu. Continuing development has been by M. Ettus

**RTTY**, **AMTOR**, and other common HF modems use binary frequency shift keying (FSK) for data transmission, which is, unfortunately, not very bandwidth or power efficient. Phase shift keying (PSK), and its many variants, have become popular recently in such modes as **Clover-II** and **Pactor**. PSK, however, suffers on the HF channel due to such factors as doppler shift, especially at very high data rates.

MFSK is a generalization of FSK in which one of  $m$  different tones is sent every symbol period, where  $m = 2^n$ . Each symbol conveys  $n$  bits of information. In binary (standard) FSK,  $m=2$  and  $n=1$ . As  $m$  increases past 4, the number of tones ( $m$ ) increases faster than the number of bits sent per tone ( $n$ ), so the bandwidth necessary for a given throughput increases also. The power necessary to maintain a given BER, on the other hand, decreases with increasing  $m$ . The symbol error rate is dependent on the energy transmitted for each symbol, but each one carries multiple bits. Thus less energy per bit is necessary. Under moderate  $E_b N_0$  ratios, with higher  $m$  ( $m \geq 8$ ), MFSK actually performs better than coherent binary PSK (BPSK).

Another advantage is the ability to perform demodulation noncoherently. This is a very valuable property on ionospheric paths, because strong multipath and doppler make it difficult to recover absolute phase information from received signals.

### Forward Error Correction (FEC)

Through the addition of controlled redundancy, Forward Error Correction allows received signals to be decoded successfully, even the presence of limited errors. Thus, less energy per bit need be expended by the transmitter. Lower power transmitters may be used, or higher throughputs can be accommodated.

There are many different FEC methods in use, each with different properties. For this modem, we chose to use Reed-Solomon codes.<sup>3</sup> Reed-Solomon codes are very powerful codes based on Galois field arithmetic.

One property of MFSK is that when an error does occur, it will affect (on average) half of the bits in a given receive symbol. This does not fit well with many other FEC schemes, because they do not handle burst errors well. Because RS codes are based on symbols (groups of bits), rather than individual bits, multiple errors in the same symbol have no more detrimental effect than a single error. This burst error correcting capability also helps on the HF path because many of the sources of noise (i.e. interfering carriers, fading, and burst noise) are bursty in nature.

In our design, we augment the burst error correcting capabilities of the RS codes by interleaving them. Thus consecutive errors are spread across different code blocks, rather than being concentrated in the same one.

## System Design

### System Specifications

For our modem, we chose to use **16-FSK**, in a bandwidth of 2000 Hz (this will fit through the **2.7kHz BW** available on most HF SSB radios). Due to the requirement for orthogonality of

---

<sup>3</sup> The Reed-Solomon encoding and decoding software used was adapted from code written by **Phil Karn, KA9Q**, and others.

symbols, and the noncoherent demodulation which will be discussed below, each tone must be separated in frequency from the next by an amount equal to the symbol rate. To equally space the 16 tones in 2000 hertz, each must be separated by 133.3Hz. 125 was chosen as it is the closest factor of 8000, a convenient sampling rate on PC sound cards. Thus, the symbol rate is 125 baud, and the (raw) bit rate is  $4 \times 125 = 500$  bits per second.

We are using shortened RS codes over  $GF(2^8)$  of block length 32 bytes, with 8 of these blocks in a packet. In each block, the modem can be set to use from 2 to 12 redundancy bytes (30 down to 20 data bytes). This allows from 1 to 6 errors to be corrected per block (1 for every 2 redundancy bytes). Each byte is composed of two 4-bit symbols (16 FSK gives 4 bit symbols). The actual data rates are then 312 to 469 bits per second.

The 8 blocks are interleaved in a packet, meaning that the first byte of each block is sent, then the second byte of each block, etc. Thus, a burst of errors is spread into separate blocks. With the highest amount of redundancy, 96 consecutive symbols could be wiped out, and the packet still recovered. A sequence of 8 symbols (the synch vector) is sent as a header for each packet. This allows the receiver to have a known sequence for it to use in synchronization with the transmitted packet.

## Modulation

Modulation is performed by generating the appropriate sine wave corresponding to the symbol to be transmitted. This is done by stepping through a table of sine values, and outputting the values to the sound card.

## Demodulation

In order to demodulate the incoming signal, synchronization must be acquired first. By correlating the synch vector to the received signal, the modem determines when to start demodulating. The actual demodulation is done by using the fast Fourier transform (FFT). The FFT returns the energy in each frequency band. The band with the highest energy is chosen as the best estimate of the received signal. This is equivalent to having 16 separate filters, and choosing the one with the highest output.

## Results

This modem has been tested off of the air, so far, using a baseband simulator running on another PC. Computer to computer file transfers have been accomplished. The next step is on-the-air testing. In testing our modem with a channel simulator, we can see that our results match up well with our theoretical calculations.

The modem performs very well in gaussian noise environments. In order to maintain a 10<sup>-5</sup> BER, 16-FSK requires 6dB less energy per bit than 2FSK, and 1 dB less than BPSK. With the addition of the Reed-Solomon coding, between 3 and 6 dB of additional gain is achieved (depending on the level of redundancy used), giving an overall gain of 9 to 12dB. At the high redundancy rate, 312 bps is achieved with about 12db (1/15) of the power of 300 baud FSK.

We are currently in the process of expanding the channel simulator from simple AWGN (additive white gaussian noise), to include fading and other HF channel effects. The properties of FEC should allow our modem to experience less degradation in these channels than other modems. The FEC can cope with large amounts of incorrect data due to temporary fades.

## Implementation

The code for this modem was written in C, for the Linux operating system. It should be portable to any other OS, with minor changes in the sound interface code. The code, when running on a Pentium Pro 150 took less than 5% of the processing time of the computer, so it should function well on much slower computers. A floating point unit is recommended.

## Conclusions

We have shown that it is possible to implement a modem entirely on a PC, to save costs, while at the same time, improving on the available options for HF communications. Our design allows higher throughput than the freely available options (AMTOR, etc.) with less than one tenth of the transmitted power. Conversely, one could communicate effectively in conditions more than 10 times worse than is currently possible.

## Areas for further study

Continuing work on this modem will expand it in several ways. First, we will further modify the RS encoding and decoding software to allow for »on-the-fly« reconfigurability of the amount of redundancy in each block. Thus, smaller fraction could be used when channel conditions are good, and then increased when conditions get poorer. We also plan on modifying the code to allow the modem to change to higher or lower order MFSK (i.e. 8-, 32-, 64- or even 128-FSK) depending on channel conditions.

Other future work will include integrating the code into a driver so that the Linux system can use it in conjunction with its other networking features, such as TCP/IP and AX.25. This would **make** for a complete HF networking system

## References

McDermott, Tom. "Wireless Digital Communications: Design and Theory." Tucson Amateur Packet Radio Corporation, 1996.

Rappaport, Theodore S., "Wireless Communications: Principles & Practice." Prentice Hall, 1996.