1. Introduction

rk described in this report started out is an exploration of projects that could be used in laboratory courses for upper division computer engineering and signal processing courses; however, it has since taken on life of its own. The original intent was to evolve laboratory exercises in which students could experiment with embedded systems hardware and software analysis, design and implementation. While laboratory experiments involving audio signal generation and playback are used in existing courses, the limited processing capability of the personal computer (PC) hardware available precludes interesting experiments with more sophisticated signal processing. Since digital signal processing (DSP) chips have become readily available and widely used in embedded systems, emphasis was placed on experiments in which hardware and software design could be explored in the context of a PC, DSP evaluation board and a shortwave radio.

Most evaluation boards for modern microprocessors and signal processing chips are expensive and not easily interfaced to typical signals and test equipment likely to be found in a university teaching laboratory. Even the development software can require substantial investment in classroom time and computing resources. A search was launched to survey the available hardware and software resources available for such a project. The goal is the ability to do sophisticated signal processing and program development using nothing more than a stripped-down DSP board, a typical PC and inexpensive (free) development tools.

The DSP board chosen for evaluation is called the DSP-93 and is supplied in kit form by the nonprofit Tuscon Amateur Packet Radio (TAPR) oganization operating in partnership with the Amateur Satellite (AMSAT) organization. Prevously, TAPR developed the packet radio technology commonly used by for-profit firms manufacturing inexpensive VHF and UHF wireless modems. The DSP-93 consists of a Texas Instruments (TI) TMC320C25 signal processor chip and various interface chips for analog/digital conversion, radio control and serial input/output interface to the PC. It communicates with the PC using a universal asynchronous receiver/transmitter (UART) chip operating at 19,200 baud.

The DSP-93 has found widespread use in the AMSAT community as a codec for telemetry and communications with various satellites now in orbit. A suite of software development programs are available for Intel and Macintosh PCs, including an assembler, loader, debugger and development program that operates under Microsoft Windows. A number of software modems operating at data rates to 9600 baud have been developed for the DSP-93, including signal designs based on amplitude, frequency and phase modulation.

This report describes a software program that operates wtih the DSP-93 and radiotelegraph signals commonly used by amateur and commercial stations in the decametric (3-30MHz) radio spectrum. The primary reasons for choosing these signals are that signal propagation is via the ionosphere, which acts as an unreliable, multiple reflector, and that the noise process is very bursty. Thus, the transmission channel model is very much time-varying, sometimes badly distorted by multipath, often contaminated by interfering signals and atmospheric electrical noise, and sprinkled with dropouts due to strong adjacent channel signals. It hardly comes as a surprise that typical linear signal processing techniques learned from the books don't work very well in these conditions and that nonlinear techniques are often required.

Another reason for choosing these signals is that many sources of commercial modems for them are available, most implemented in analog form, but a few in digital form. For various reasons, these modems are designed using rather dated technology and capable of only fair performance. The digital modem design described in this report uses modern design techniques capable of substantially better performance than present analog and digital designs; in fact, the modem represents the optimal linear receiver for the class of signals used in this report. It is intended that this design serve as a proof of concept that these techniques can be adapted to systems used in everyday commerce for the fixed, aeronautical mobile and maritime mobile services of today.

Finally, the software design developed in this report represents a major departure from designs commonly used in DSP systems. Most systems used as examples in the DSP handbooks and practical designs are based on delicately timed decimation loops, where code path latencies must be carefully monitored and controlled. The digital modem described in this report ordinarily would be considered quite large compared to conventional designs. This can place very significant demands on the assembly langauge coding for each decimation schedule, and often results that the program breaks if just one more instruction is added to some routine. The more flexible and useful approach used in the modem program decouples the hardware and software operations using interrupts and a circular buffer holding input and output samples. The result makes practical the use of a high level structured programming language and portable software library. However, for this partiuclar project, the only tools available were an assembler, loader and text editor.

This report proceeds as follows. Following a description of the signals and propagation model, the basic hardware and software functional design is described. The detailed design and design rationale of the software is then described by each functional block. This includes the processing steps at the radio frequency (RF) and baseband stages of the modem. The major departures of this particular design relative to those commonly found in commercial equipment are in the signal decoders, of which there is one designed for ordinary asynchronous (start/stop) radiotelegraph (RTTY) signals and another for international synchronous teleprinter-over-radio (SITOR) signals used in the maritime mobile (ships at sea) services. When used in the amateur radio service, these signals are designated amateur teleprintor-over-radio (AMTOR). Following the functional block description, implementation features are described, including those believed novel in this application. Finally, an assessment of the analytical performance of the modem as compared with current designs is presented, along with examples of off-the-air comparisons using typcial signals corrupted by multipath, cochannel interference and noise. Appendix A contains operator notes, Appendix B contains an analysis of word error rates for asynchronous signals, and Appendix C contains an analysis and simulation of the data carrier detector (DCD) circuit.

Due to its large size, the program listing is not included in this revision of the report. The program distribution, including source and listing, can be found via the Collaboration Resources page of the author's home page http://www.eecis.udel.edu/~mills.

2. Design Approach

Decametric or high frequency (HF) radio signals are reflected by ionospheric layers ranging from 110 km to 350 km above the Earth. These layers vary in height and degree of ionization througout the day and seasons of the Earth and Sun. The height of each layer determines the reflection points and resulting path geometry and whether a propagation path exists between two points on Earth. The degree of ionization determines the maximum frequency and ionosperic absorption in the bands in which propagation is possible. While it is possible to predict with a fair degree of accuracy whether a propagation path exists between two points on Earth and the optimal frequency band to use, the ionosphere is notoriously unstable and tumultuous at best. Typically, two or more ray paths may exist which may add constructivly or destructively at the receiver, producing fades to 40 dB or more. Impulsive noise due to atomospheric electrical storms can overwhelm the recevier, producing signal dropouts lasting tens of milliseconds. In some services, strong interfering stations on the same or adjacent channels can be present. A successful digital or analog modem must perform well in spite of these signal impairments.

In order to demonstrate the utility of the hardware and software described in this report, a relatively simple signal design was selected, both because it is widely used in existing services, is relatively easy to understand, and because examples of competing technology are available to demonstrate performance. The signal design is based on frequency-shift keying (FSK) as used in conventional radiotelegraph services. Keying speeds are usually in the range from 45.45 baud to 100 baud, while frequency deviations are usually in the range from 170 to 850 Hz. Services using these signals usually operate using a 5-bit interchange code commonly called Baudot, but officially designated International Telecommunication Alphabet-2 (ITA-2) by the International Radio Consultation Committee (CCIR), a unit of the International Telecommunications Union (UIT). These signals can be transmitted in either asynchronous or synchronous modes as described later in this report.

The hardware and software described in this report operates with an external radio (receiver, transmitter or transceiver) and terminal or terminal program. Typically, an IBM or Macintosh PC is used for program development and on-the-air operation in a amateur radiotelegraph station. However, the design is completely compatible with radiotelegraph installations widely used in the maritime services. In the development of the hardware and software described in this report, the digital modem was developed and tested using several computer-controlled radios, a IBM compatible PC, and a Unix workstation. Shell scripts were used in the workstation to automatically collect performance data over many days of testing.

In some applications, the digital modem can operate as a regenerator, in order to reconstitute signals for an existing analog modem, such as a terminal node controller (TNC), which is sold by a number of manufacturers and widely used in amateur and shipboard stations. The design of most TNCs is based on analog technology using active filters and a microprocessor such as the Zilog Z80. In all significant ways, the digital modem described in this report provides the same functions as the typical TNC. A particular variant of analog TNC manufactured by Advanced Electronic Applications (AEA), Inc., and called the PK-232 is used in the performance comparisions described later in this report.

When used with radiotelegraph signals, a TNC operates using audio frequency tones in the range 1275-2295 Hz. The frequencies and signal levels are compatible with those used by typical single sideband (SSB) radio transceivers used by both the amateur radio and commercial services. Assuming the carrier suppression and opposite sideband suppression in the receiver and transmitter are sufficient, selection of the lower (mark) or upper (space) audio tone is equivalent to frequency modulating the carrier of an ordinary FSK transmitter. This technique is commonly called audio-FSK or AFSK.

3. Hardware Functional Description

The DSP platform used in this report, called the DSP-93, is built around the TI TMC320C25 [TI87], which is a 16-bit DSP chip operating with a 40-MHz clock and 10 MIPS instruction rate. The DSP-93 consists of a digital circuit board, an analog circuit board and and an external power supply transformer. The digital board includes the TMC320C25, 256K words of EPROM used for the

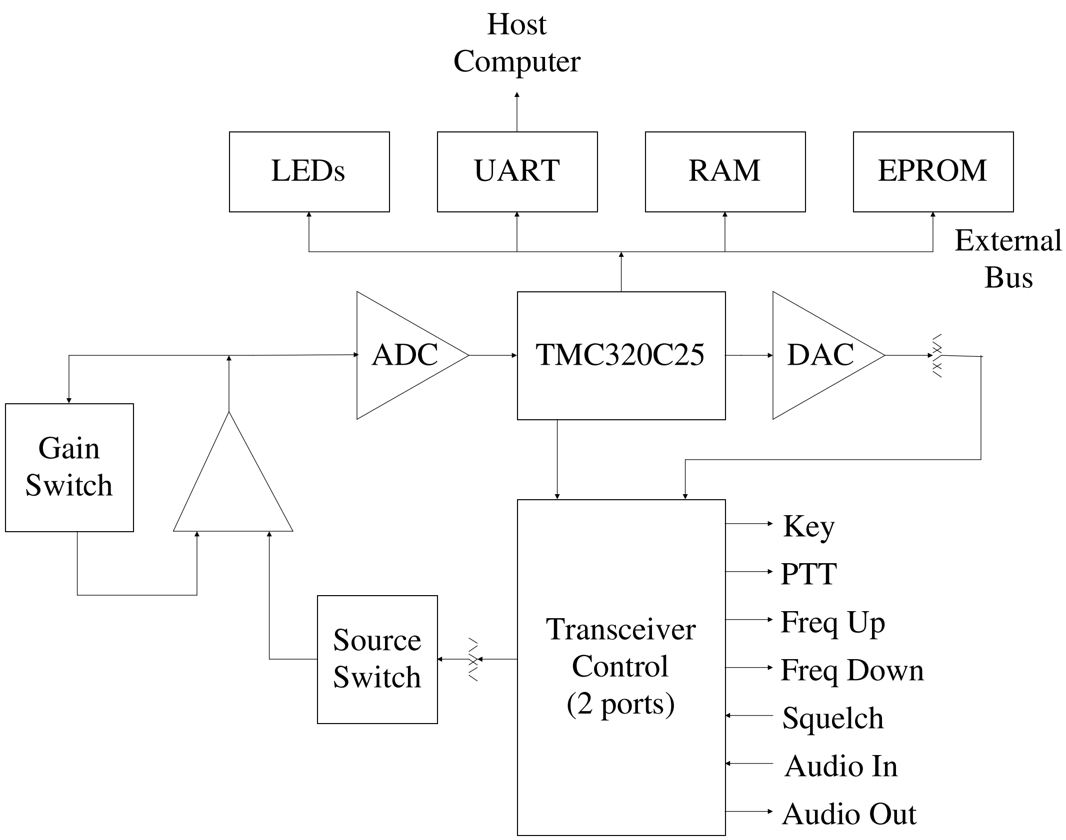


Figure 1. DSP-93 Hardware Functional Diagram

monitor program, 32K words of program RAM and 32K words of data RAM. The monitor program is used only during program development. The analog board includes a TI TLC32044 14-bit voiceband analog interface circuit [TI92], a 16550 UART and various chips for audio processing and transceiver control.

Figure 1 is a block diagram of the DSP-93 hardware components. The DSP program controls the front panel light-emitting diode (LED) display, the UART used to communicate with the PC terminal program, the analog/digital converter (ADC) and digital/analog converter (DAC) in the analog interface, two audio switches and the transceiver control interface. One of the audio switches selects which of several audio sources are input to the ADC, including the Audio In input of either transceiver port, the DAC output (for analog loopback), or the auxiliary (Squelch) input of either transceiver port. The other audio switch is used in the feedback path of a differential amplifier to implement a programmable gain function. Potentiometers are provided to set audio input and output levels for each transceiver port.

The DSP-93 program individually controls either of two transmitters, one connected to each radio port, using the push-to-talk (PTT) output, which serves the same function as on a typical mobile microphone. Most transceivers can connect the Audio Out directly to the microphone audio input; however, some transceivers equipped for direct FSK keying can use the Key output instead. Some transceivers are equipped to shift operating frequency in discrete steps up or down in response to a

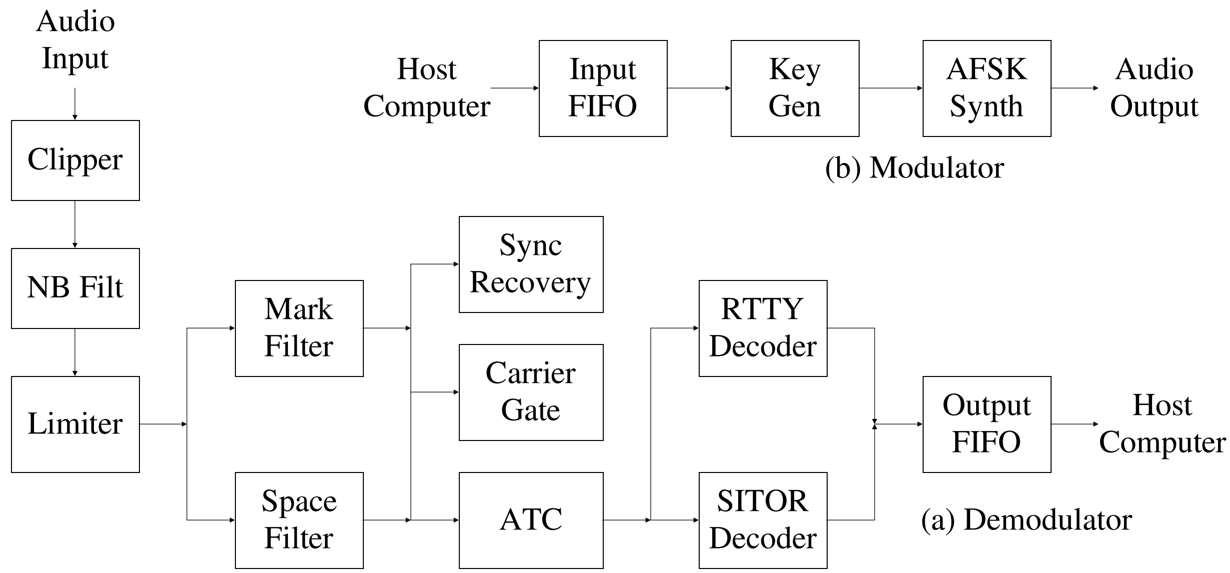


Figure 2. Software Block Diagram

button press on the microphone. The Freq Up and Freq Down outputs can be generated by the DSP-93 program in response to modem commands.

The analog interface includes, besides the basic ADC and DAC functions, a set of programmable dividers driven from the TMC320C25 clock. As used in the digital modem, these dividers are programmed to provide a basic analog/digital conversion clock rate as close to 8000 Hz as possible. This clock rate was chosen both for compatibility with standard telephone conventions, as well as a compromise between various sources of noise and available processing latencies.

The monitor program resides in EPROM on the digital board. It includes a rudimentary debugger, loader, a suite of prebuilt modem programs, and a set of utility programs, including a digital oscilloscope and spectrum analyzer. The loader is used to load programs developed on the PC, but otherwise the monitor is not used when the digital modem is in operation.

4. Software Functional Description

The following sections describe in detail the software functions implemented for the digital modem. These include the RF filtering and baseband processing functions and the RTTY and SITOR decoders. Following is an overview which may help in understanding the detailed processing steps to follow.

A software block diagram is shown in Figure 2. The demodulator functions are shown in the blocks labelled (a), while the modulator functions are shown in the blocks labelled (b). In the demodulator, the radio receiver audio input is connected to a spike clipper, narrowband filter (NB Filt), and limiter. These processing steps serve to condition the AFSK signal, reduce out-of-band interference, and reduce large amplitude variations due to multipath fading. There are two sets of channel filters, one for the lower (mark) tone, the other for the upper (space) tone. These serve to further reduce out-of-band interference and noise, as well as provide a frequency discriminator function. These signals are demodulated to produce the bipolar baseband signal that drives the baseband processing and decoding functions.

There are three functions provided by the baseband processing. The carrier gate function is used to suppress garbles due to noise in the RTTY decoder. The synchronization recovery function is used to provide bit synchronization for the SITOR decoder. The automatic threshold corrector (ATC) function is used to determine the slice level used to classify mark and space signals for both decoders. The RTTY and SITOR decoders represent the heart of the design and are based on theoretically optimum principles using matched filter and correlation techniques. The decoders produce a character stream which is translated to the ASCII code, buffered in a FIFO and delivered to the terminal program in the PC.

In the modulator, characters received from the PC terminal program are buffered in a FIFO, translated to the Baudot code, and used to generate the keying waveform, which is a bipolar signal controlling whether the mark or space tone is to be transmitted at a particular time. In the SITOR case, Baudot characters are translated to the CCIR 476 code before encoding for transmission. The keying waveform is then used to generate a phase-continuous AFSK signal, which is connected to the transmitter audio input. Control signals not shown are used to properly sequence transmit/receive switching and echo control.

The above design should be contrasted with that of a conventional analog modem, which consists of a set of narrowband active filters for the mark and space channels, followed by a limiter, discriminator and lowpass filter. A slicer processes the lowpass filter output to generate a binary signal that can be read by an embedded microprocessor. In the digital modem design, the limiter is placed before the channel filters, rather than after. The reasons for this choice are twofold: first, this maximizes the modem dynamic range; and second, this reduces the limiter intermodulation products that reach the detector. In following sections, the software functions implemented in the digital modem are described in some detail, along with comparisons with the conventional analog modem.

4.1. RF Bandpass Filtering and Limiting Functions

The audio signal produced by an ordinary SSB communications receiver tuned to a radiotelegraph signal consists of alternating tones of mark and space frequencies embedded in noise, cochannel interference and subject to multipath fading and dropouts. These signals are generally classified by transmission speed and frequency difference, or shift, between the mark and space tones. Most radiotelegraph systems of today use speeds to 100 baud at 170-Hz shift and are classed as narrowband direct printing (NBDP) systems. Wider shifts to 850 Hz are used in some older systems and in systems where the transmission schedule is shared between radiotelegraph and facsimile emissions.

An ordinary SSB communications receiver has a bandwidth of about 2100 Hz, which is appropriate for voice communications. On the other hand, a NBDP signal has an occupied bandwidth of about 500 Hz at 100 baud. In order to reduce extraneous signals due to aliasing, it is important to minimize the system response at frequencies outside the occupied bandwidth. To achieve this, the digital modem software implements two order-70 FIR filters, a 450-Hz bandpass filter for narrow 170 Hz shift (Figure 3) and a 1100-Hz bandpass filter for wider shifts to 850 Hz (Figure 4). These filters, which were designed using the Remez method and the Math Works, Inc., Matlab Signal Processing Toolkit, have steep skirts and a maximum stopband ripple of about 50 dB. While these characteristics are similar to those of the eight-pole, 0.5 dB-ripple Chebyshev filters used in some analog modems, the advantage of the FIR design is that it has a linear phase characteristic and does not distort the signal waveshape.

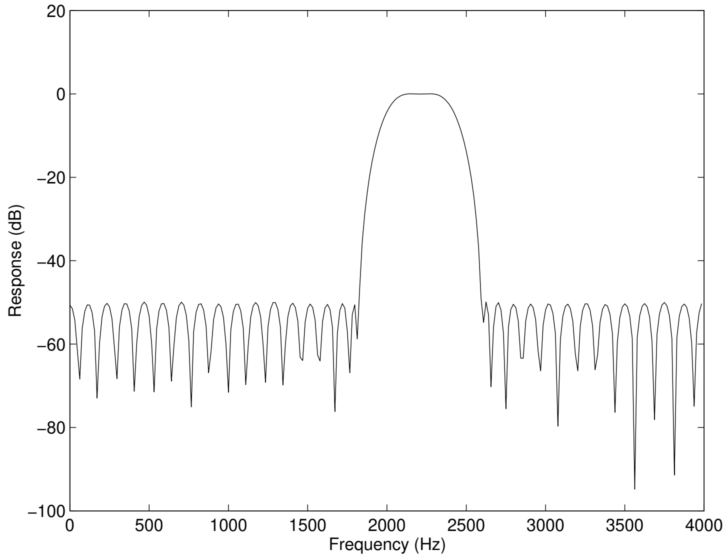


Figure 3. 170-Hz Bandpass Filter Response

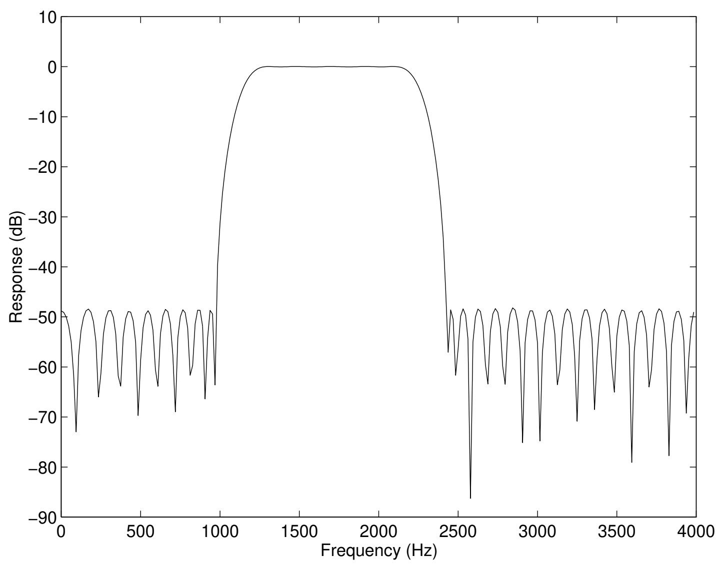


Figure 4. 850-Hz Bandpass Filter Response

Noise spikes due to unshielded ignition systems, electric light dimmers and similar sources are always a problem with weak-signal reception of decametric waves. In order to reduce interference due to these causes, a transient clipper is located at the input to the modem before the input bandpass filter. It is inteded to kill noise spikes that might ring and overload later stages in the modem, but has no significant affect at ordinary signal levels. A LED indicator flashes if the clipping level, which is about 10 dB below modem overload, is exceeded. The clipping level is chosen to minimize the loss in numeric significance in the various processing steps, yet provide some headroom for overload.

An adjustable limiter is included in the signal path after the narrowband filter to reduce variations due to multipath fading and other causes. A modem command can be used to select limiter gain from 0 to 30 dB in 6 dB steps. The limiter gain can be adjusted in response to prevailing noise,

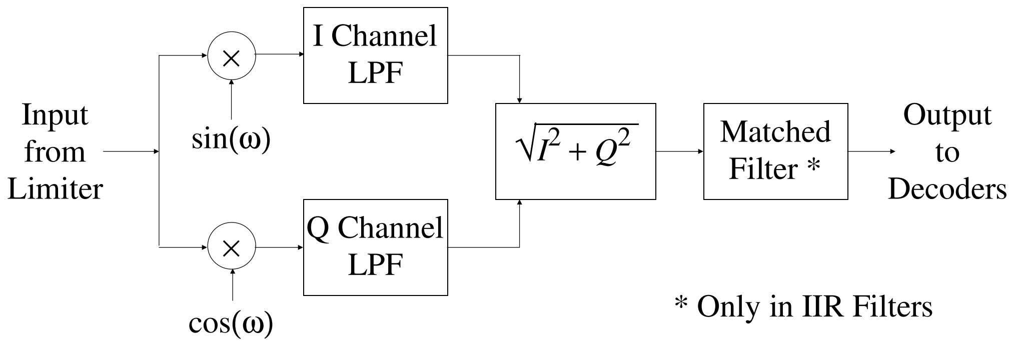


Figure 5. Mark/Space Channel Filters

interference and multipath conditions. In general, a relatively large gain can be used under most conditions of fair to good signal quality and moderate to severe multipath fading, in order to simplify receiver gain and tuning adjustments. A relatively small gain can be used under extreme low-signal conditions in order to achieve the maximum advantage of linear processing.

4.2. Channel Filters

The mark and space channel filters are the heart of any modem design. In ordinary analog modems, they are implemented as sixth- or eighth-order Bessel or Chebyshev filters using active devices and resistor/capacitor components. In the digital modem, there are two channel filters following the limiter, one for the mark channel, the other for the space channel. These are used both to reduce out-of-band interference and to provide a frequency discriminator function. Each of the two channel filters is implemented as a synchronous lowpass filter, as shown in Figure 5. The filter has two signal paths, one for the in-phase (I) component, the other for the quadrature-phase (Q) component. The I component is produced by multiplying the limiter signal by $cos(\omega)$, where ω is the mark (or space) frequency. The channel output signal is produced by squaring the I and Q components and extracting the square root with three iterations of the Newton-Rapheson algorithm.

There are two sets of channel filters implemented in the digital modem, one using infinite impulse response (IIR) filters, and the other using matched filters. These can be selected for various shifts and filter combinations. The order-2 Butterworth IIR lowpass filters are useful under most operating conditions with radiotelegraph signals and shifts wider than 170 Hz. These filters, which were designed using the bilinear-transform method and the Math Works toolkit, are narrow enough for good performance, but wide enough at about 300 Hz for easy tuning. In this case, postdetection matched filters are used in both the mark and space channels to optimize the transient response and as an interpolation filter. The calculated resonse of the IIR filter/discriminator is shown in Figure 6.

Alternatively, synchronous predetection matched filters can be selected for greater selectivity with RTTY and SITOR signals at all shifts. As described later, matched filters represent the optimum filter structure for binary signalling in AWGN channels. The matched filters are implemented using an interleaved shift register/delay line and four accumulators, one for the I and Q channels of the mark and space filters. In this case, postdetection filters are not necessary. The matched filters are matched to the baud rate and have bandwidths approximately equal to twice the reciprocal of the baud rate. The calculated response of the matched filter/discriminator for 45.45 baud is shown in Figure 7. Note that the zeros of the response are at multiples of the baud rate, which makes them

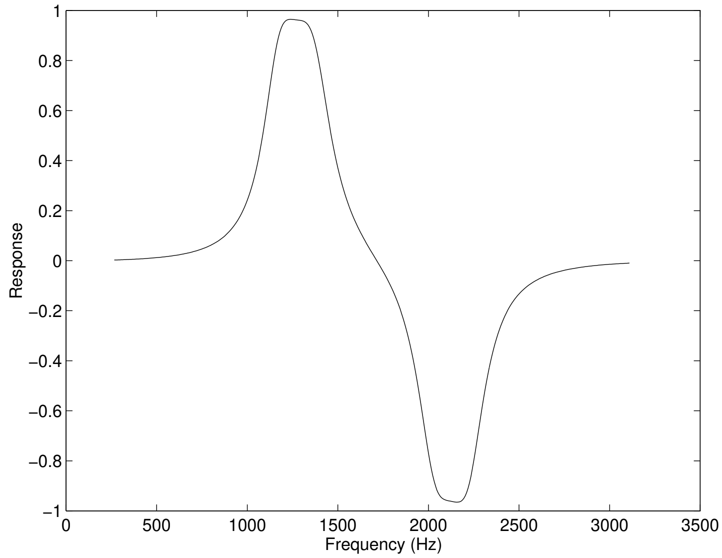


Figure 6. 850-Hz IIR Discriminator Response

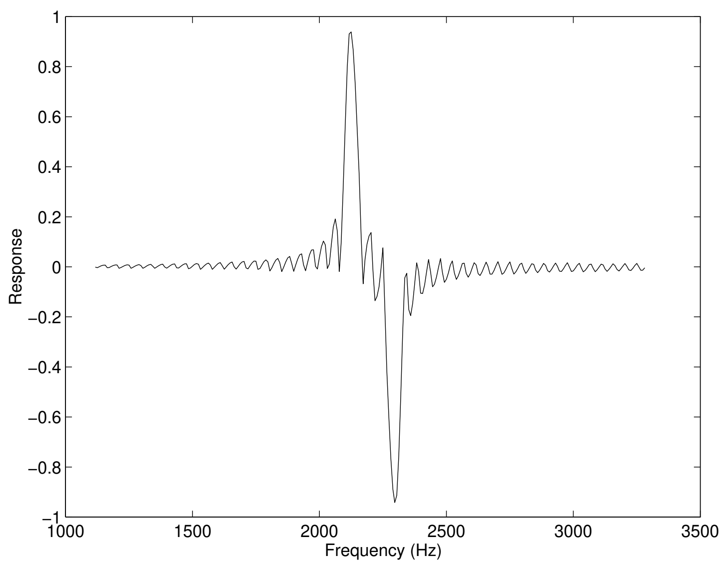


Figure 7. 170-Hz Matched Filter Discriminator Response

very sharp. Receiver tuning with these filters is very sensitive, especially at the lower baud rates, and requires, for example, an adjustment for older analog TNCs with compromise 200-Hz AFSK tone shift. At the lowest selectable baud rate of 10 baud, the receiver must be accurately tuned and remain stable to within a few Hz.

Various modem commands can be used to tailor the digital modem RF characteristics in response to special conditions. The analog gain of the modem can be adjusted from unity in steps of 6 dB using the analog gain switch shown in Figure 1. Another command inverts the detector signal for upright/inverted shifts. Another command disables either the mark channel or space channel, if required to reduce interference. Gain factors for all combinations of filters have been optimized for most conditions. However, the system gain following the limiter can be adjusted using a modem command. This may be useful for low level signals with the limiter switched off.

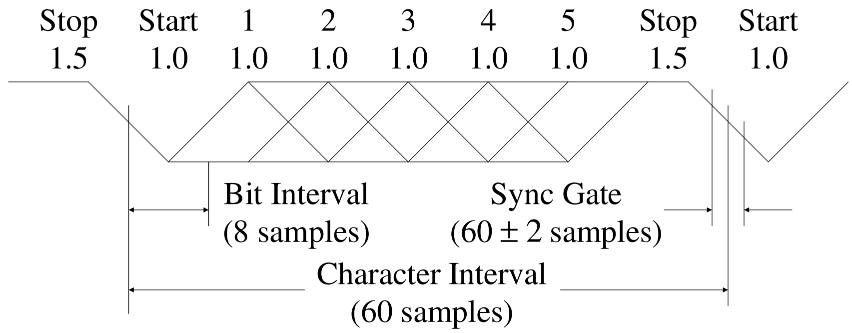


Figure 8. RTTY Signal Structure

4.3. Baseband Processing

Baseband processing includes those functions that develop signal quality estimates and slice signals used by the decoders. These signals include the carrier gate, which is used by the RTTY decoder to mute output, unless a carrier signal is present, and the slice level, which is used by both the RTTY and SITOR decoders to classify signals as mark or space. In analog modems, these signals are developed using active analog filters and diode-switched resistor/capacitor integrators. However, in conditions involving deep multipath fades, analog circuits often perform poorly. On the other hand, when these circuits are implemented using digital technology, much better results are possible. This section presents one possible approach that has been found to work well in practice.

Figure 8 shows the signal structure used in RTTY signalling. The duration of each signalling element is measured in baud intervals, or units. The asynchrous (sometimes called start-stop) RTTY signal consists of five one-unit information intervals preceeded by a one-unit start interval and followed by a 1.5-unit stop interval. The signal following the matched filter channel filters has a triangular shape, as shown in the figure. The optimum sample instants are at the triangle peaks, while the optimum slice level is midway between the maximum and minimum triangle peaks. A character begins at the first negative-going zero crossing, as shown in the figure, and ends at the stop bit, which must be a mark for a correctly formed character. The next character following can begin no earlier than after the 1.5-unit stop interval, but could occur later. In synchronous RTTY described later, the start bit of the next character following must fall in the sync gate interval shown in the figure.

The slice level signal is the threshold above which an input signal is considered mark and below which the signal is considered space. In modern estimation theory, this signal can be interpreted as the likelihood ratio, where the mark and space conditions are considered equally likely and the additive white Gaussian noise (AWGN) for each condition is the same. The optimum likelihood ratio is normally midway between the maximum (mark) and minimum (space) envelope signal amplitudes. In analog modems, the maximum and minimum envelope amplitudes are developed in a peak-detector circuit with a relatively long decay time constant.

Problems with the analog implementation of the ATC ciruit occur when the individual channel fading rate approaches the decay time constant, which is not an infrequent occurance. The circuit is required to provide a reliable slice level when one of the mark or space channels has faded completely into the noise and when strong interfering signals within the receiver passband, but

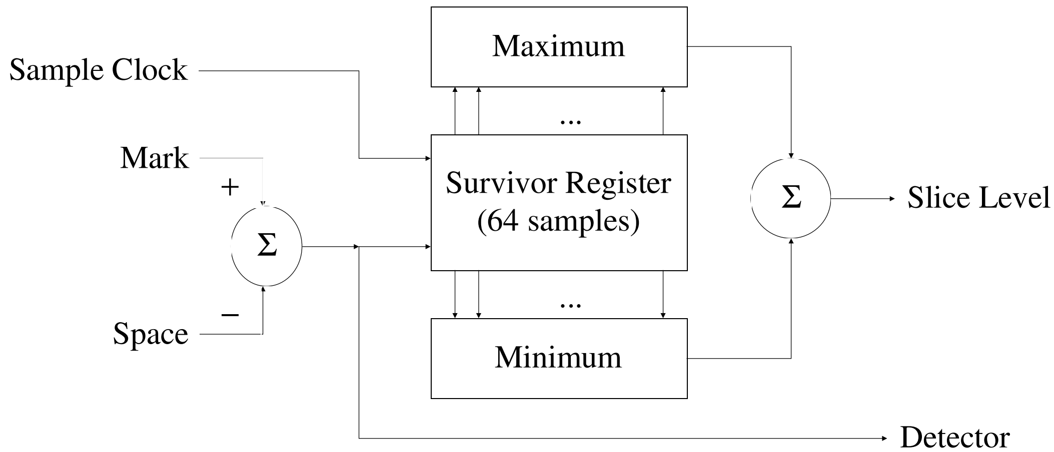


Figure 9. Automatic Threshold Compensator

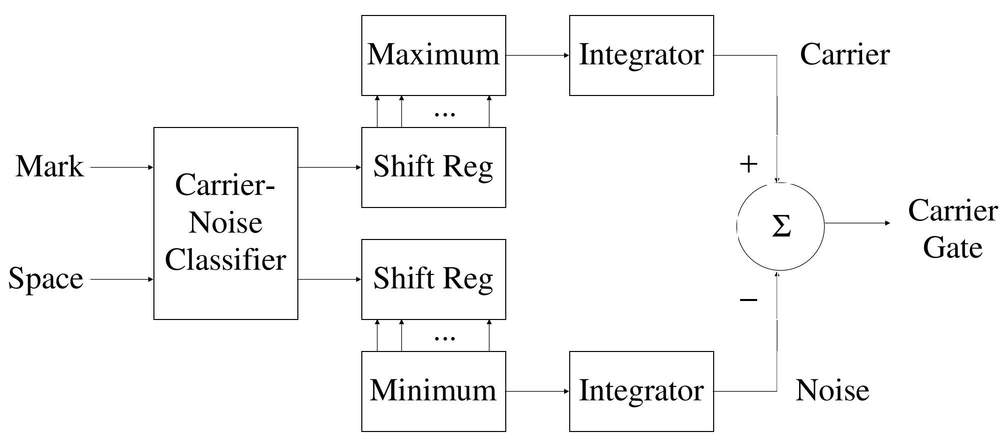


Figure 10. Carrier Gate

outside the modem passband, cause the receiver AGC cirduit to reduce the level of all signals in the receiver passband, including those within the modem passband.

In the digital modem, the ATC circuit shown in Figure 9 is used to estimate the optimum slice level. The detector signal (differential mark minus space) is sampled at eight times per baud interval and the samples saved in a 64-stage shift register (also used later in the RTTY decoder as the survivor register). The program searches for the maximum and minimum over these samples and generates a slice level equal to the sum of the (signed) maximum and minimum signals, which is by construction midway between the maximum mark and space envelope signals. Detector signals above this level are classified as mark samples, while those below this level are classified as space samples. The number of samples in the shift register is sufficient that at least one mark and one space sample must be in the register for any legitimate RTTY character, even one that has all data bits space. Thus, the slice level is individually determined for each character independent from all characters that may precede or follow that character.

The modem implements three signal quality estimators based on measured characteristics of the mark and space baseband signals and decoded characters. These are called the carrier distance, erasure distance and autostart distance. The carrier distance is developed directly from the baseband signal and recomputed at the sample rate, which is eight times the baud rate. The erasure and autostart distances are developed from measurements made by the RTTY and SITOR decoders and recomputed at the character rate. Associated with each of these three estimators is an adjustable threshold, which may be set by a modem command, and a gate, which indicates whether the distance is above or below the threshold.

The primary function of the carrier distance circuit shown in Figure 10 is to provide a muting signal for the RTTY decoder, in order to avoid extraneous signals which can yield no useful output. However, this signal is also available with the SITOR decoder as a tuning aid. The carrier distance is developed using two 192-stage shift registers operating as boxcar integrators and two four-stage shift registers operating as sample classifiers. At each sample time, the mark and space channel signals are compared. If the mark signal exceeds the space signal by a factor of at least two (6 dB), the mark signal is classified a carrier sample and the space signal classified a noise sample. Similarly, if the space signal exceeds the mark signal by a factor of at least two, the space signal is classified a carrier sample and the mark signal classified a noise sample. If neither of these two cases is true, both the mark and space signals are classified as noise samples.

The carrier samples and noise samples are shifted through separate classifier shift registers in order to delete outlyers and modulation products. At each shift, the maximum sample in the carrier classifier register represents the carrier signal, while the minimum sample in the noise classifier register represents the noise signal. The carrier energy (carrier signal squares) and noise energy (noise signal squares) are separately integrated over 192 samples (about three characters) using the boxcar integrators. Finally, the carrier distance is computed as the carrier energy minus the noise energy. If this distance exceeds the carrier threshold, the carrier gate is unblocked.

A character quality estimate called the erasure distance is developed by the RTTY and SITOR decoders. As each character or word is decoded, a quality function is developed based on the Viterbi algorithm (RTTY decoder) or correlation function (SITOR decoder), as described later. In addition, the detector signal squares are computed over all samples in the survivor register. The erasure distance is computed as the ratio of the mean quality function over the RMS detector signal. If this distance exceeds the erasure threshold, the erasure gate is unblocked.

In both the RTTY and SITOR decoders, an autostart gate blocks and unblocks the decoder output as determined by a set of sanity checks and the erasure distance. It operates to avoid garbles due to noise and signals other than radiotelegraph and is primarily intended for use in a crossband repeater, where reliable signal detection is essential. The RTTY decoder also uses the autostart gate to determine when to switch between asynchronous and synchronous modes. The SITOR decoder uses the autostart gate in order to distinguish between noise and a correctly synchronized character stream.

The autostart distance is computed from two signals generated by the RTTY and SITOR decoders, as described in the sections below for each decoder. One of these signals is derived from the quality function developed by the decoder, while the other is the detector signal squares developed during

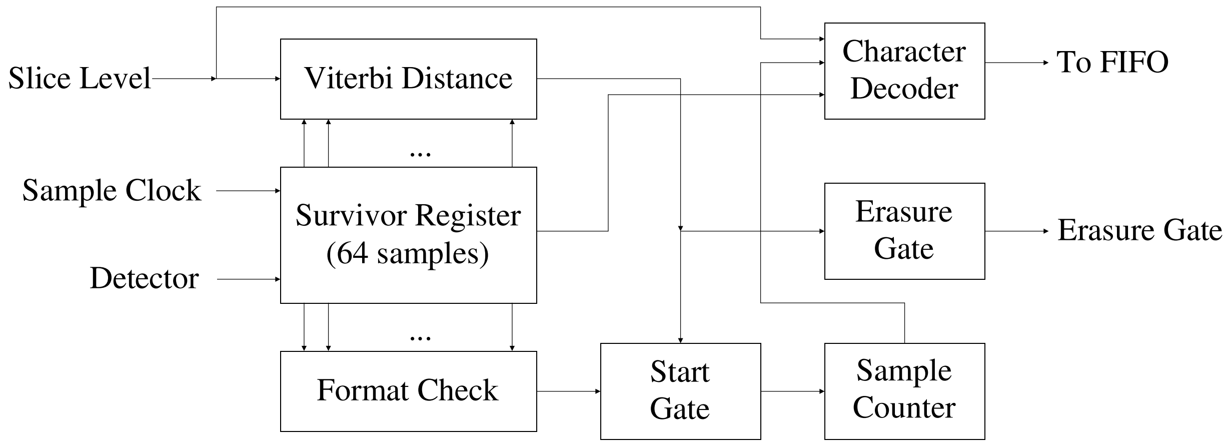


Figure 11. RTTY Decoder

the erasure distance calculation. The quality function is developed from these signals integrated over a period of 15 characters using a pair of shift registers operated as boxcar integrators. The autostart distance is computed as the ratio of the quality function over the RMS detector signal. If this distance exceeds the autostart threshold, the autostart gate is unblocked.

Including the integration performed while developing the erasure distance, the autostart distance is integrated over a total of 960 baseband signal samples, which gives it a fair degree of authority in separating valid characters from noise. In the RTTY decoder, a modem command selects the signals and gates used to control the autostart function, as described below. In the SITOR decoder, this command is not useful, since the extensive integration and correlation procedures are sufficient to avoid garbles without additional gating functions.

4.4. RTTY Decoder

The RTTY decoder shown in Figure 11 is compatible with the ITA-2 (Baudot) character format long used in the amateur radio and commercial services. The Baudot character format or codeword includes a start bit, which is always zero (space), five data bits and a stop bit, which is always one (mark) and usually 1.5 baud intervals in length. The decoder goes to some extremes in order to reliably extract RTTY characters under conditions of low signal levels, high noise levels and severe multipath conditions. Experience has shown the most important factor for good decoder performance is reliable detection of the start bit. This is done in the following way.

The decoder saves eight detector signal samples for each of eight baud intervals, a total of 64 samples, in the survivor shift register. These samples are used to determine the maximum and minimum amplitudes and slice level for the detector signal, as described previously. The oldest bit in the shift register is the stop bit of the previous character, the next oldest bit is the start bit of the current character, and the youngest is the stop bit of this character. A survivor is defined as the vector of samples beginning in the oldest bit and extending through the corresponding sample of each bit in turn to the youngest bit. There are eight survivors, one beginning for each sample in the oldest bit. The problem is to identify whether the oldest survivor begins a character and, if so, which of the other survivors represents the most likely transmitted signal.

The RTTY decoder can operate in two modes, asynchronous and synchronous, depending on signal quality and operator preference. (SITOR operation is always synchronous.) In asynchronous mode, which is the default, a character can arrive at any time and is decoded independently. As each new signal sample is received, a series of checks is performed to determine if the associated survivor begins a character. Only those survivors which meet the following criteria are eligible: (a) the carrier gate is unblocked, (b) in the last eight bit times, the oldest bit (stop bit of the previous character) is classified a mark, (c) the next oldest bit (start bit of the current character) is classified a space, and (d) the youngest bit (stop bit of the current character) is classified a mark. Unless all of these criteria are met, the decoder abandons further processing and waits for the next sample.

If the criteria are met, the distance for each survivor beginning in the start bit is computed according to the Viterbi algorithm. Survivors that do not show mark in the stop bits and space in the start bit are ignored. Of the remainder (there must be at least one), the one with maximum distance represents the most likely character as transmitted. Note that the distance as used here is the negative of the usual Viterbi distance, in that it increases with increasing probability. The maximum distance represents the quality function used in computing the erasure and autostart distances. This is the basis of the MAP claim, since of the 256 eight-bit words that might be received, only those 32 words which satisfy the above criteria have nonzero a-priori probability.

If the maximum distance over all survivors is below the erasure threshold, the decoder abandons further processing and waits for the next sample. Otherwise, the start gate shown in the figure is unblocked, the character counter started, and the survivor bits classified according to the slice level. The resulting character is delivered to the common output buffering routine described below. Once the character counter has counted out seven baud intervals (56 samples), the decoder resumes sample processing.

The operation of the carrier and autostart gates in RTTY asynchronous mode has been described previously. While the false-alarm rate using only the carrier gate can be rather high, the false-alarm rate using the autostart gate is much lower. For this reason, the operator should send a continuous burst of ASCII SYN characters (which are translated to Baudot LTRS characters) in RTTY asynchronous mode, or wait a couple of seconds in RTTY synchronous mode, when beginning transmission to allow a receiver known to be using the autostart function to be unblocked. The autostart gate is designed for long periods of inactivity, such as when the system is waiting for a selective call and it doesn't matter if a few characters are lost at the beginning of transmission. During a continuous period of activity, the autostart gate has only marginal usefulness.

In RTTY synchronous mode, the modem phase-locks to the start-bit transitions of a continuous sequence of characters. The decoder thus operates as a gated receiver, with the PLL signal derived only from the samples corresponding to start-bit intervals. Figure 8 shows the timing intervals involved, including the sync gate, which allows for a timing error of ± 2 sample intervals. The synchronous gated receiver improves performance, especially under multipath conditions resulting in deep fades of either the mark or space channels or both. In such cases, when valid character framing is lost, the decoder simply scraps the garble without losing valid framing for subsequent characters.

The phase error is the signal at the oldest sample in the start bit, which is exponentially averaged to form the error signal used to drive the critically damped PLL, which has a time constant of about two seconds. This assumes that the modem has first estimated the intercharacter time in asynchronous mode. To do this, the intercharacter time estimate is determined as the exponential average of the number of samples since the last start bit. The intent is to provide a seamless transition between asynchronous and synchronous operation. In order for the estimate to be updated, the intercharacter interval must fall within a window of four samples centered in the eighth bit, which provides protection against severe jitter due, for example, to extreme bias shifts in deep multipath fades.

The decoder starts in asynchronous mode and refines the intercharacter interval estimate as valid characters arrive in a continuous stream. Eventually, if the signal quality is good enough, the autostart gate is unblocked. If enabled, the modem automatically switches to the synchronous mode and an indicator LED is turned on continuously. Operation in synchronous mode continues until the autostart distance falls below the threshold, indicating the signal quality has deteriorated or phase lock has been lost, in which case the modem automatically switches to asynchronous mode. When transmitting in synchronous mode, the transmitter sends an idle character (LTRS) if the transmit buffer is empty and neither a ACK or EOT has been received from the terminal program. This helps the remote station maintain synchronization during pauses when no text is being transmitted.

4.5. SITOR Decoder

The SITOR decoder is compatible with the CCIR 476 Mode B (FEC) character format and protocol used in both commercial (SITOR) and amateur (AMTOR) operations. The decoder is an unusual design based on correlation and maximum-likelihood principles. In fact, the decoder never makes a decision based on classifying bits as marks or spaces. Rather, an approach based on soft decisions is carried through all processing steps, with the final decision based on a correlation of a 14-bit word, consisting of a received seven-bit character and its repetition, with each of the 35 CCIR codewords. The one with maximum correlation function wins and the function value (normalized by the RMS detector signal over the word) becomes the erasure distance. A design such as this represents the optimum receiver for equiprobable source symbols transmitted over nonfading or Rayleigh fading channels with additive white Gaussian noise.

The CCIR character format consists of seven bits, four of which must be one (mark) and three zero (space). In Mode B operation, the fifth character following a transmitted character is a repetition of that character. The error performance of this scheme can be analyzed as follows. Since all 35 seven-bit CCIR characters have weight four, the seven-bit CCIR code has minimum distance two. The 14-bit code, consisting of vectors of two identical CCIR characters, has minimum distance four, thus can correct all single-bit errors and in addition detect all two-bit errors.

An error-free codeword has a maximum correlation function of 14 with itself. A single bit error in one of the two characters results in a maximum function of 12 and is correctable. A two-bit error results in a maximum function of 10 and is not necessarily correctable. This could occur if each of the two characters had weight four, but were distance two from each other. In order to reliably correct single-bit errors, yet reject all others, the optimum decoder threshold should correspond to function values between 10 and 12, depending on the intended ratio of missed detections and false alarms.

A transmitted character and a replicated character from a previous transmission are transmitted as one 14-bit word. Words are transmitted continuously; if no characters remain to be transmitted, the two-character sync codeword RQ–α is transmitted. In addition, at intervals of about ten seconds, five repetitions of this word are transmitted as a retrain sequence, in order to allow reliable receiver resynchronization and avoid receiver buffer overrun.

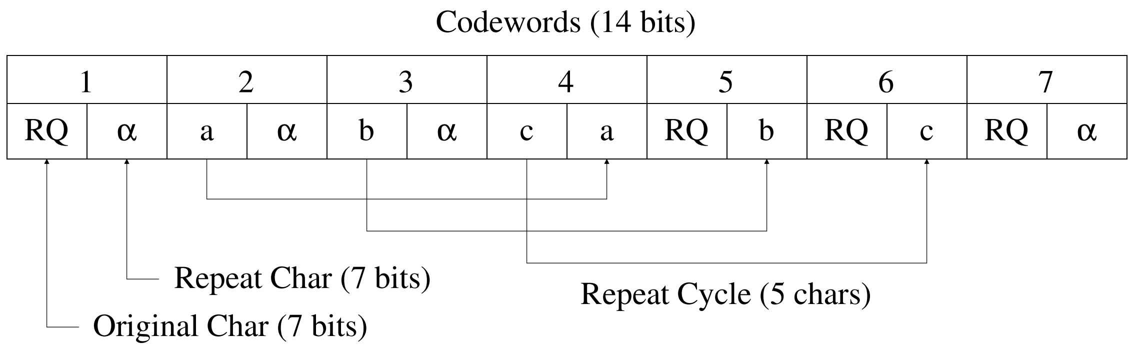


Figure 12. CCIR 476 Signal Structure

Figure 12 may help clarify the operations used to encode and decode the CCIR 476 signal. The encoder uses a three-stage shift register containing 14-bit CCIR 476 codewords. The least significant seven bits of each word represent an original character, while the most significant seven bits represent the repetition of the fifth character in the past. The figure shows the progression of the sequence “abc” begining and ending with the sync codeword RQ–α.

Note that a 14-bit codeword is 140 ms in length at 100 baud (10 ms per bit), while a standard 7.5-bit Baudot character is 150 ms in length at 50 baud (20 ms per bit). Thus, an unrestrained CCIR transmitter gains 10 ms per character over a standard Baudot printer operating at 50 baud. To avoid overrunning the receiver buffer, the transmitter inserts the retrain sequence after 70 ordinary codewords have been transmitted. Insertion at that rate continues indefinitely as long as characters remain to be transmitted.

Figure 13 shows a block diagram of the SITOR decoder. Bit synchronization is determined by a critically damped, type-I PLL with a time constant of about two seconds. When the SITOR decoder routine is called, the last eight bits (64 samples) are in the survivor shift register, in order to determine the slice level. The bit phase is extracted from the eight samples in the first bit of this register operating as a linear phase detector. If the signal polarities at the beginning and end of this bit are opposite, the signal value at the midpoint of the bit is likely a good measurement of bit phase. These quadrature-phase samples are exponentially averaged and used to derive the VCO signal, which is implemented by modulating the bit clock. The in-phase samples are corrected for slice level and shifted through a six-character (42 bit samples) shift register to be used later.

Word synchronization is determined by correlating characters in the original and repeat positions of received words with the retrain sequence, which is normally transmitted at the end of each line of text or 70 characters, whichever occurs first. The correlation function is continuously measured for each of 14 bit positions and averaged over all five words (70 bits) of the sequence. The value of the function is the quality function used to compute the erasure and autostart distances. When the autostart distance exceeds the threshold, correct character phase has been achieved and characters are output to the terminal program. The processing gain achieved by this method (over 18 dB) is enough to provide reliable synchronization without consideration of the carrier distance, even under marginal conditions with relatively high bit error rates.

Once the retrain sequence has been detected, the most likely decoded character is determined using a correlation process. In the normal case (other than the retrain sequence), the original and repeat

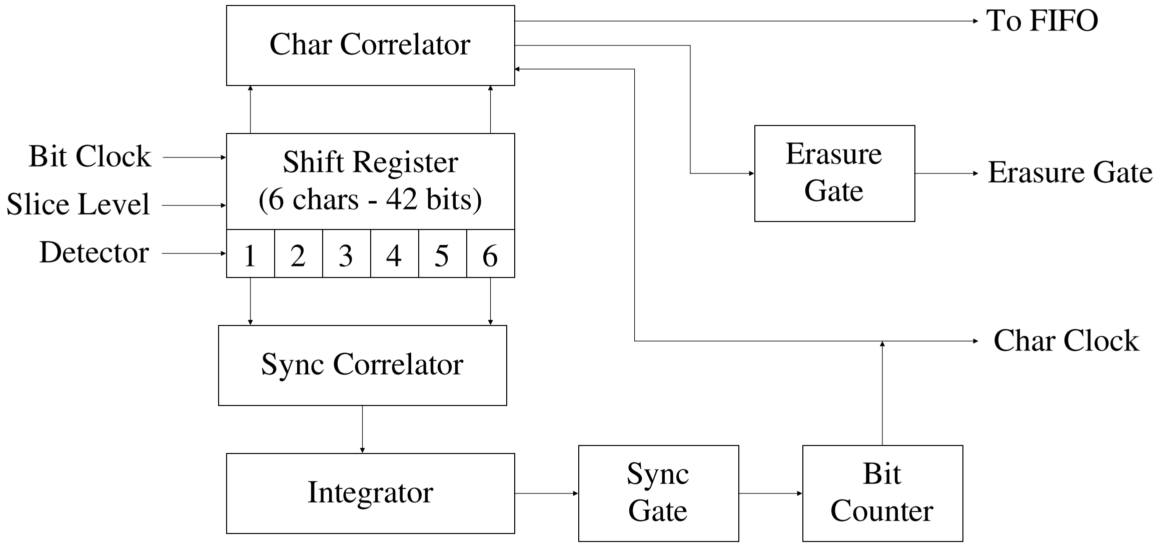


Figure 13. SITOR Decoder

characters are the same, so can be vector summed before correlation. This is done using the first and last character of the six-character shift register mentioned previously. Since the RQ–α word does not involve a replicated character, it must be correlated separately. The correlation process is somewhat tricky in view of the limitations of the TMC320C25 signal processor and is done as follows.

The 42-bit shift register developed by the baseband processing routines is stored in program memory in the coefficient page, while the correlation coefficients are stored in data memory. The decoding routine correlates the vector sum of the seven-sample original character code plus the seven-sample repeat character code (delayed five characters) with each of the 35 possible seven-bit CCIR characters and selects the one with maximum correlation function. Its value represents the quality function used to construct the erasure distance. If the erasure distance exceeds the threshold, the codeword is translated to Baudot and delivered to the common output buffering routines described below.

4.6. Transmitter Signal Generation

The transmitter signals for both RTTY and SITOR are synthesized AFSK audio sinewaves at the programmed mark and space frequencies. In addition, a PTT signal is developed to control the transmitter carrier. Signal generation involves three steps: switching and buffering data from either the terminal program or decoder, as described below under Half/Full-Duplex Modes, encoding the keying waveform, and generating the AFSK output signal.

Figure 14 is a block diagram of the encoder/synthesizers. Separate encoders are used for RTTY and SITOR, but they operate in much the same way. ASCII characters to be transmitted are translated to Baudot (and then to CCIR 476 code in case of SITOR) and encoded for transmission using a shift register. The mark/space signal is generated using the bit clock developed by the decimation schedule. Finally, the mark/space signal determines which of the mark or space frequencies is used to synthesize the AFSK output signal. The actual output waveform is generated using a table of sine-function values at three-degree increments.

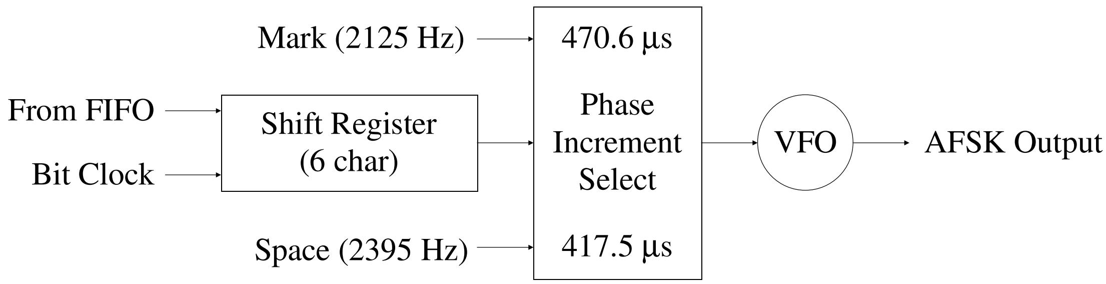


Figure 14. Transmitter Encoder/AFSK Synthesizer

Both encoders implement a request-to-send (RTS) delay, in order to allow receivers to synchronize in the synchronous-RTTY and SITOR modes, while the SITOR encoder inserts the retrain sequence every 70 transmitted Baudot characters to avoid overrunning the receiver buffer and to provide retrain opportunities. The encoders also provide idle fill, in order to provide continuous timing in synchronous modes - LTRS in the RTTY encoder, RQ–α in the SITOR encoder.

4.7. Character Buffering and Translation Operations

Baudot characters delivered by the RTTY and SITOR decoders are processed by a common output routine, which implements various gating and translation functions and switches the output to the input and output buffers, as described below. If the SITOR decoder is in use, or if the RTTY decoder is in use and operating in synchronous mode, output is blocked if the autostart gate is blocked. If unblocked and the erasure gate is unblocked, the character is translated to ASCII and delivered to the input or output buffer, as indicated by duplex mode. If the erasure gate is blocked, the erasure character "_" is delivered instead, unless suppressed by a modem command. In RTTY asynchronous mode, the gating protocol is controlled, as described previously. In this case, output is blocked if the erasure gate is blocked.

There are two sets of character buffers, one for input from the terminal program, the other for output to the terminal program. Each is implemented as a circular buffer of 1000 characters maximum size. The flow control functions depend on the number of characters remaining in the buffer. When less than one-third of the maximum size remains, the buffer is said to be in the red condition. In this condition the CTS line is dropped by the UART used by the terminal program. When more than two-thirds of the maximum size remains, the buffer is said to be in the green condition. In this condition the CTS line is raised, permitting the terminal program to send additional data. As noted previously, when the CTS line is dropped, the terminal program is blocked not only from sending data to be transmitted, but from sending commands as well.

As evident from actual operations, not all hardware and software configurations can respond to the CTS signal in time to prevent occasional buffer overrun when transmitting a long file, at least when operating at speeds of 19,200 baud with the Windows 3.1 Terminal program and a 50-MHz 486 processor. There is no provision in the present implementation to change the UART baud rate. An appropriate future enhancement would be an autobaud function similar to that implemented in some conventional computer modems.

The native interchange code used by both the RTTY and SITOR encoder/decoder is Baudot (ITA-2), while the native interchange code used by the terminal program is ASCII (ITA-5). The modem implements the Baudot and ASCII code translation tables to conform to conventions established by the international community as represented by tables in the ARRL Handbook, 1995 Edition. ASCII characters to be translated are first stripped of the parity bit, then mapped to upper case. Except for the following, all ASCII characters with undefined Baudot equivalents are discarded before translation.

ASCII	Baudot	Function
EOT	LTRS	close down transmitter immediately
ACK	LTRS	close down transmitter when buffer empty
ENQ	WRU	trigger answerback at receiver
SYN	LTRS	use for fill if necessary

On transmit, Baudot CR and LF characters are forced to LTRS shift, in order to improve copy under marginal conditions. If enabled, a SP (space) character is forced to LTRS shift for the same reason. On receive, if enabled, a SP character is interpreted as if the sequence LTRS-SP were transmitted. Another set of code translation tables is used to translate between the Baudot interchange code and the CCIR 476 transmission code used by the SITOR encoder/decoder. The Baudot-to-ASCII, ASCII-to-Baudot and Baudot-to-CCIR tables are hard coded in the modem program; however, the CCIR-to-Baudot table, which consists of 35 seven-sample code vectors, is built during reset processing directly from the Baudot-to-CCIR table.

4.8. Half/Full-Duplex Modes

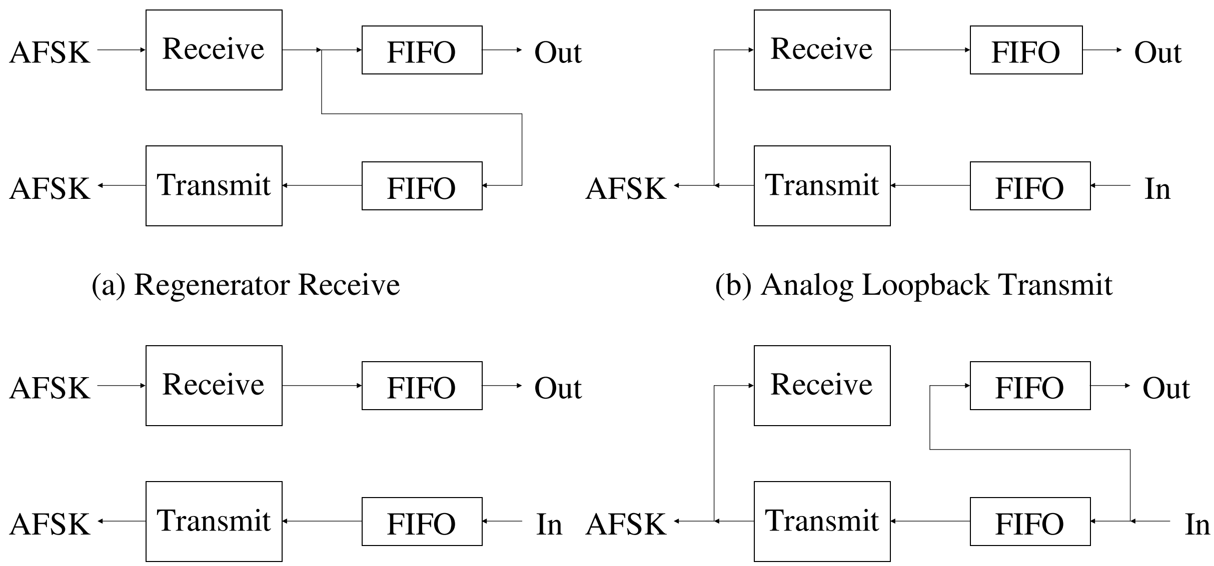
The modem can operate in four modes, as shown in Figure 15: regenerator receive (a), analog loopback transmit (b), digital loopback transmit (c), and full duplex (d). A modem command can be used to switch between these modes. The configuration of the modulator and demodulator and their associated buffers in each mode is described the following paragraphs.

Regenerator receive mode is used during receive for the half-duplex analog loopback transmit and digital loopback transmit modes. When operating as a regenerator, an ordinary TNC is connected to the AFSK output and used to demodulate and decode the regenerated signal. The regenerator demodulates the input signal and translates to ASCII, then translates back to Baudot and generates the output AFSK signal. The translations remove extraneous nonprinting characters and regenerate LTRS and FIGS control characters as described previously. The regenerated tones can also be used to drive a transmitter or crossband repeater.

In analog loopback transmit mode, characters are echoed as actually sent. There is a variable delay in this mode, depending on how far ahead the typist is of the transmitted signal. Note that there is an additional fixed delay of five characters in SITOR mode. In digital loopback transmit mode, characters are echoed as received from the terminal program. Note that the demodulator is not used, other than to drive the various indicator LEDs. In full-duplex mode, the modulator and demodulator, together with their buffers, are functionally separate. No provisions for echo are made in this mode. Note that in the analog and digital loopback transmit modes, the modulator and demodulator must be operated with the same shift and mark/space polarity, which is possible only with the 170-Hz and 850-Hz shifts.

5. Implementation Overview

This section contains a brief overview of the modem program implementation. It is designed to assist in understanding of the various operations and principles described herein, as well in possible modifications of the source code that may be useful in practice. The program listing is shown in



(a) Regenerator Receive

(b) Analog Loopback Transmit

(d) Full Duplex

(c) Digital Loopback Transmit

Figure 15. Duplex Modes

Appendix C. This program, which is in assembly code and richly commented, is the ultimate functional reference. As these programs go, this one is on the large side, with over 7000 words of program memory and over 7000 words of data memory.

The basic program design consists of a main program, which operates with interrupts enabled, a set of minimalist interrupt routines, and a circular buffer, which holds analog input and output samples. Input samples received from the A/D converter interrupt routine are saved in the circular buffer pending retrieval by the RF input routine, while samples generated by the RF output routine are saved in the same buffer pending retrieval by the D/A converter interrupt routine.

There are two pointers to the circular buffer, a get pointer used by the interrupt routines and a put pointer used by the RF input and output routines. The A/D interrupt routine stores an input sample at the current get pointer position and advances the pointer. The D/A interrupt routine reads the sample at the current get pointer position, which is delayed from the input sample by the number of samples in the buffer times the sample interval, 125 us.

The RF input routine reads the sample at the current position of the put pointer, which is usually not far behind the get pointer. For every input sample, the program produces an output sample, which is stored by the RF output routine at the put pointer position and the pointer advanced. While the put pointer can occasionally lag behind the get pointer due to code latencies, every input sample is replaced by a output sample corresponding to a fixed delay. Thus, the buffer serves as an elastic delay line, giving the main program considerable leeway in code latencies. Since the input and output interrupts are synchronous and every input interrupt results in exactly one output interrupt, no data are lost or duplicated, unless the latency exceeds the maximum delay in the buffer, in which case a slip equal to the entire buffer contents occurs.

The program operates at a basic rate of 8000 Hz, corresponding to the sample rate of the A/D and D/A converters. Due to the choice of master oscillator frequency and the programmed divider configuration of the converters, the nearest actual rate is 7947 Hz; therefore, it is necessary to slip an occasional sample interval in order to closely approximate the 8000 Hz sample rate with a resolution better than 15 PPM. At this rate, the program implements the predetection filtering, limiting and demodulation functions to develop the mark and space channel signals used by the baseband routines. The RTTY and SITOR AFSK signals are also generated at this rate.

The output of the filter/limiter stages is multiplied by sine and cosine functions separately for both the mark and space frequencies, making four filter channels in all. Identical lowpass filters are used for each channel, one for the in-phase (I) component, the other for the quadrature-phase (Q) component. After filtering, the two components for the mark channel are squared and summed, then the two components for the space channel are squared and summed. The output signals for the mark and space channels are generated by a square-root routine using three iterations of the Newton-Raphelson algorithm. For the predetection matched filters, the output of each channel is taken at this point. For the IIR filters, the output at this point is filtered by a matched filter operating at baseband, both to provide the required impulse response and also to act as a decimation filter.

The predetection matched filters and the postdetection matched filters used with the IIR channel filters are matched to a rectangular pulse of duration equal to a bit interval. In the predetection case, four filters are used, one each for the I and Q phase of the mark and space channels. They are implemented using a circular buffer with the four samples interleaved in sequence. There are two reasons for this. First, the filters can grow quite large - up to 3000 samples at a 10-Hz baud rate - which would make a transverse filter of the FIR type impractical. Second, only a limited amount of memory is available for shift registers and coefficients in the TMC320C25. The same structure is used for the postdetection matched filter, but only two samples are necessary for each cycle.

There are three decimation clocks used in the modem, one at the nominal sample rate, which is eight times the baud rate, another at the baud rate and a third at the character rate of the decoder. The baseband processing routines operate using the sample clock to operate the survivor shift register, determine the maximum mark and space channel signal estimates, and calculate the carrier distance. The RTTY and SITOR decoders also operate using the sample clock to decode the characters and provide the character clock. In the case of the SITOR decoder, the sample clock is modulated to provide the VCO function for the PLL, which explains the use of “nominal” above.

The baseband signals and signal quality estimators are determined using the sample clock, in order to realize convenient filter structures and reduce the processing burden. The functions provided at this level are described above under Baseband Processing above. However, as the sample clock is variable over a small range, in order to track SITOR symbol phase, this clock is inappropriate to drive the SITOR encoder, since this needs a precision source of timing. Thus, the bit clock for the SITOR encoder is developed separately.

The autostart functions based on Viterbi distance use the character clock developed by the decoders themselves. The character translation, character buffering and related functions are driven by the decoders and thus also operate at this rate.

The 2x14 integrators used for SITOR character phase resolution consist of two sets of 14 interleaved matched filters stored in a circular buffer. One set consists of 14 matched filters, where the impulse response is matched to the 70-bit CCIR retrain sequence consisting of five repetitions of the sync codeword RQ–α. The other set consists of 14 boxcar integrators used to sum the squares of 70 carrier samples for use as a normalizing function. As each bit is received, the erasure distance is calculated as the ratio of the matched filter output to the square root of the boxcar integrator output.

The matched filter operations in the SITOR decoder are implemented using correlation techniques. The correlation function is stored in the form of an array of vectors in the order corresponding to the Baudot code equivalents. Each vector has seven bipolar components selected according as to whether the CCIR character has a one or zero in the corresponding bit position. The components are scaled so that the peak of the correlation function is equal to the maximum of the input signal. The organization is such that a pipelined multiply-and-add function can be embedded in a tight loop that scans the vectors one after the other looking for a maximum. This loop takes 35 iterations of 18 machine cycles for a total of 630 cycles per character, or about 5.6% of the available cycles.

The 42-bit shift register used in the SITOR decoder is implemented in program memory in the coefficient page, rather than in data memory as would normally be expected. The reason for this is that there are too many correlation coefficients (230) to fit in the internal data memory and the CPU architecture requires the coefficients and data memory to be on different busses for maximum pipeline efficiency. Therefore, the coefficient page is mapped to data memory in order to shift new samples into the delay line and mapped to program memory before correlation.

There are generally spare cycles left over when all processing for an input sample is completed. The program uses these cycles for housekeeping functions, such as running the intricate LED display program, looking for terminal program input and output data, searching for command input and so forth.

Many functions of the program make use of what are called boxcar integrators. These data structures consist of a circular buffer, a buffer pointer and an accumulator. A new signal sample is introduced by first subtracting the old sample value in the buffer at the pointer position from the accumulator and storing the new sample at this position. Then, the new sample value is added to the accumulator and the buffer pointer advanced. The accumulator thus contains the sum of all samples in the buffer.

Boxcar integrators are used in the program to implement the matched filter operations and to compute the various distance functions. They are simple to implement, can be made quite large, e.g., 3000 words in the predetection matched filters, and use relatively few processor cycles. They can be easily multiplexed, e.g., 2x2 in the predetection matched filters and 2x14x5 ways in the SITOR word synchronization matched filter.

Most of the signals processed are in fractional q15 format, where the decimal point is immediately behind the sign bit. The processor is operated in sign extension mode (SXM), overflow mode (SOM), where overflows are clamped to the largest or smallest integer representation, and product mode (SPM 1), where the results of a multiplication are left-shifted one bit before transfer to the accumulator. These conventions simplify the arithmetic operations and help preserve significance. Product accumulations, VCO frequencies and some critical data are maintained in multiple precision (32 bits).

Only a few interrupt-related program variables are in internal data memory block B2, while filter coefficients and some delay lines are in internal memory block B0, which is usually mapped to program memory. The remaining delay lines are in internal data memory block B1. Most of the working variables are in the user page (8) accessed by direct addressing modes. Arrays, shift registers and accumulators accessed by indirect addressing modes follow the working variables.

6. Performance Assessment

In this section the performance of the digital matched-filter digital modem is compared to a conventional TNC, which uses analog filters. Comparisons are made both analytically and experimentally using off-the-air signals.

6.1. Analytical Performance Analysis

There are three analytical performance metrics of interest, signal-to-noise ratio (SNR), bit error rate (BER) and word error rate (WER) [COU93]. In the following, SNR is the ratio E_s (energy per symbol) to N_0 (noise spectral density per Hertz). For the purpose of the development, all bit intervals of the seven-bit word of Figure 8 are assumed equal, so the BER is equal to the symbol (bit) error probability P_e and the WER is equal to the word error probability P_w. Following standard conventions, the noise is considered additive, white and Gaussian, in spite of the practical experience that noise on decametric radio circuits is usually far more bursty than Gaussian.

In a matched filter, the output SNR is related to the input SNR

$$\mathrm{SNR_{out}} = 2T_sB\ \mathrm{SNR_{in}} ,$$

where B is the input bandwidth over which $\mathrm{SNR_{in}}$ is measured, $B = 2100$ Hz for a typical SSB communications receiver, and T_s is the integration time. In the RTTY decoder at 50 baud with filters matched to the bit interval, $T_s = 20$ ms, so the processing gain is 10 log(2 .02 2100) = 19.2 dB. In the SITOR decoder at 100 baud with filters matched to the 14-bit CCIR codeword as described previously, $T_s = 140$ ms, so the processing gain is 10 log(2 0.14 2100) = 27.7 dB. The method of finding the character phase in the SITOR decoder involves coherent integration over five repetitions of the CCIR sync codeword, $T_s = 700$ ms, so the processing gain is a whopping 10 log(2 0.7 2100) = 34.7 dB. These are impressive figures, but probably misleading.

A more revealing comparison may be the following. For the purposes of comparisons, the baseband signal is assumed 100 baud for both the digital modem and conventional TNC and the limiter is switched off. By Carson's rule, the bandwidth of a FSK signal with modulation bandwidth B and modulation index β can be approximated by

$$B_t = 2(\beta + 1)B .$$

For the 170-Hz shift used in SITOR and amateur RTTY, $\beta = {}^{170}\!/_{100} = 1.7$. The required bandwidth can be approximated by the first null in the baseband spectrum, which occurs at the baud rate for bipolar signals, so $B = 100\ Hz$. Thus, $B_t = 2\ (1.7 + 1)\ 100 = 540\ Hz$. Figures somewhat less than this can be assumed if some degree of pulse shaping (ideally, raised cosine) is used at the transmitter. Assuming a 500-Hz predetection filter is used by the conventional TNC, the SNR following the filter is improved about 6.2 dB. A similar improvement, but with better phase response characteristics, is provided by the FIR predetection filter in the digital modem. Therefore, subsequent comparisons will be developed on the assumption the receiver bandwidth is 500 Hz.

Conventional TNCs use discriminator filters with bandwidths in the 300-Hz range, which is also the case with the IIR filters in the digital modem. There is no appreciable improvement with these filters in either modem. A good postdetection lowpass filter used in a conventional TNC might have a 100-Hz cutoff for 100-baud signals, possibly less if pulse shaping is used at the transmitter. Thus the best improvement that can be expected of the lowpass filter is 10 log(500 / (100 sqrt(2))) = 5.5 dB. However, the optimal postdetection matched filter provides a processing gain of 10 log(2 .01

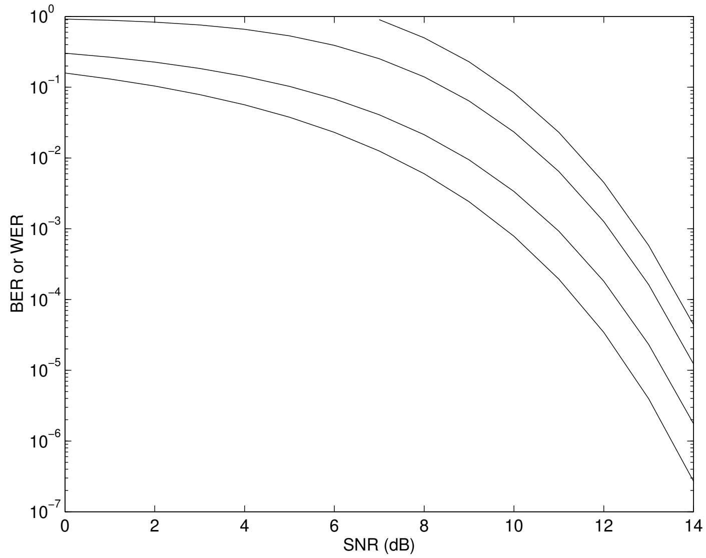


Figure 16. Computed Bit and Word Error Probabilities

500 / sqrt(2)) = 8.5 dB, 3.0 dB better than the conventional TNC. On the other hand, the optimal matched predetection filter has a processing gain of 10 log(2 .01 500) = 10.0 dB, 4.5 dB better than the conventional TNC.

The bit error probability for a noncoherent FSK channel is given by

$$BER = p = \frac{1}{2}\exp(\frac{-E_s}{2N_0}) \; ; \tag{1}$$

and $q = 1 - p$. Solving for SNR,

$$\mathrm{SNR} = \frac{E_b}{N_0} = 2\ln(\frac{1}{2p}) \; . \tag{2}$$

Figure 16 shows the BER and WER as a function of SNR over the probability range of interest. Numbering from the top of the figure, the top two traces show the WER for two cases considered later, while the third shows the BER for the noncoherent FSK channel and the fourth shows the BER for the ideal coherent FSK channel. In the range $P_e < 10^{-5}$ of interest, the noncoherent SNR is only about one decibel inferior to the coherent SNR; so, for most purposes the difference can be ignored.

A word (character) error rate (WER) of 10^{-4}, corresponding to one character error for two lines of print, can be considered an acceptable level of performance for amateur service. In Appendix B, expressions are developed for WER given the bit error probability p and each of three decoder models. In the case of the asynchronous RTTY decoder, the WER includes components due to errors on the data bits, as well as errors on the start and stop bits. The Markovian model called Case 3 in Appendix B applies to the asynchronous RTTY decoder. The composite WER due errors of all causes is found to be

$$\mathrm{WER} = (1 - q^7)(1 + \frac{4.78}{2} + \frac{2.33p + 3.75q}{4}) \approx 3.97(1 - q^7) \; ,$$

A graph of this equation is shown in the top trace of Figure 16. According to the graph, in order to achieve a WER of 10^{-4}, a SNR greater than about 13.5 dB is required. The conventional TNC would require a SNR of 13.5 - 5.5 = 8.0 dB in a 500-Hz bandwidth to achieve this BER; however, the digital modem would need only a 13.5 - 10.0 = 3.5 dB SNR in the same bandwidth to achieve the same WER. This is about the limit of detectability for a modern communications receiver. In other words, if the operator can find the signal, it will probably yield fair copy.

In the case of the RTTY synchronous decoder, the digital modem operates as a gated receiver. If errors due to loss of phase lock can be ignored, the only errors that affect the WER are on the data bits; therefore the composite WER due these causes is:

$$\mathrm{WER} = 1 - q^5 .$$

A graph of this equation is shown in the second trace from the top of Figure 16. At a WER of 10^{-4}, the digital modem requires a SNR of 13.1 dB, which is only a marginal improvement over the asynchronous case. However, at the same SNR as the asynchronous case, the synchronous decoder has a WER less than half that of the asynchronous decoder.

In the case of SITOR, a few comments on the CCIR 476 code are in order. The code has minimum distance four, so can correct one error and detect two errors; however, the equivalent rate is only five information bits in 14 code bits. If t is the number of correctable errors, a (n, k) binary code must satisfy the Hamming, or sphere-packing bound [WIL96, p. 432],

$$\sum_{j=0}^{t} C_j^n \le 2^{n-k} .$$

For example, a (14, 6) code satisfies the bound for $t = 2$ errors:

$$\sum_{j=0}^{2} C_j^n = 196 \le 256 = 2^{14-6} .$$

This code has twice the throughput of the CCIR code for the same bit rate and can correct twice the number of errors.

The actual performance of the CCIR code can be estimated in the following way. The WER for uncoded transmission of a 14-bit word can be determined directly from (1) with the substitution $p = 1 - q^{14}$. At reasonably high SNR levels, the coding gain of the CCIR 476 code can be approximated by the asymptotic coding gain ACG, where [WIL96, p. 521]

$$ACG = 10\ log(d_{\mathrm{min}}R) = 10\ log(4 * 1/2) = 3dB,$$

expressed in decibels. This amounts to shifting the uncoded curve of Figure 16 left by this amount. In contrast, for the (14, 6) code above, $ACG = 10\ log(5 * 6/14$~=~3.31 dB.

The soft-decision characteristic of the maximum-likelihood decoder adds an additional increment of coding gain. As discussed in [WIL96, p. 522], for coherent detection, reasonably large distances and SNR levels, the soft decision decoder enjoys a coding gain of approximately 3 dB. In other cases the coding gain could be diminished as much as a decibel. From Figure 16, it is apparent that the curve for the uncoded 14-bit symbol gives away about a decibel in SNR relative to the

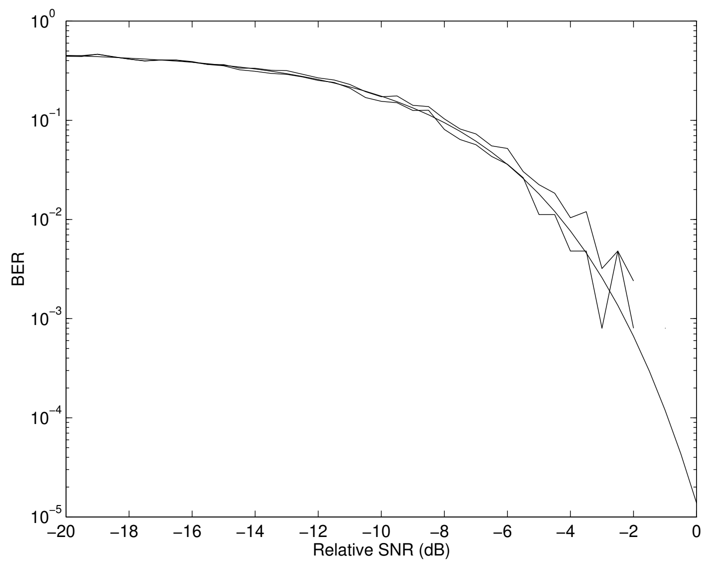


Figure 17. Simulated Predection Filters and Demodulator

noncoherent FSK BER curve. If this is the case, the combination of the CCIR code and maximum-likelihood, soft decoding improves the SNR by approximately 4 dB, at least in the WER range of interest.

6.2. Simulation Results

The effectiveness of the demodulator and decoder designs was determined using three signal processing simulators and the Pro-Matlab system [MOL90]with the Signal Processing Toolbox {LIT88]. One simulator models the RF predetection and detection functions, while the other two model the postdetection filtering and decoding functions. The RF simulator samples the audio input signal at a 8000-Hz rate, processes it through a bandpass filter, optional limiter, synchronous matched filters and noncoherent detectors for each of the mark and space channels. There are two postdetection simulators, one for the RTTY decoder and the other for the SITOR decoder. The RTTY decoder operates in either asynchronous (start/stop) mode or synchronous mode with a 7-bit code, consisting of a start bit, 5-bit Baudot character, and a stop bit, where the 32 Baudot code combinations are equally likely. The SITOR decoder operates in synchronous mode with a 14-bit code, where the 35 CCIR 476 code combinations are equally likely.

In the RF simulator, the modulated signal is generated by an alternating sequence of 2125-Hz mark and 2295-Hz space signals, where each alternation has an interval of 10 ms, corresponding to the bit interval for 100-baud FSK. The noise signal is generated by a Gaussian random noise generator followed by a 400-2500-Hz bandpass filter. The resulting 2100-Hz noise process approximates the characteristics of a conventional SSB receiver. The signal and noise powers are both normalized, then combined in the ratios required for each signal/noise simulation.

The input signal is further filtered by a bandpass filter with nominal bandwidth 375 Hz, for a computed processing gain of 7.5 dB. The hard-limiter receiver includes a 30-dB limiter after this stage and before the channel filters; the linear receiver has no limiter. The resulting signal is processed by a synchronous matched filter with an integration time of 10 ms followed by a noncoherent FSK demodulation, for a processing gain of 5.75 dB. (Note that the noncoherent

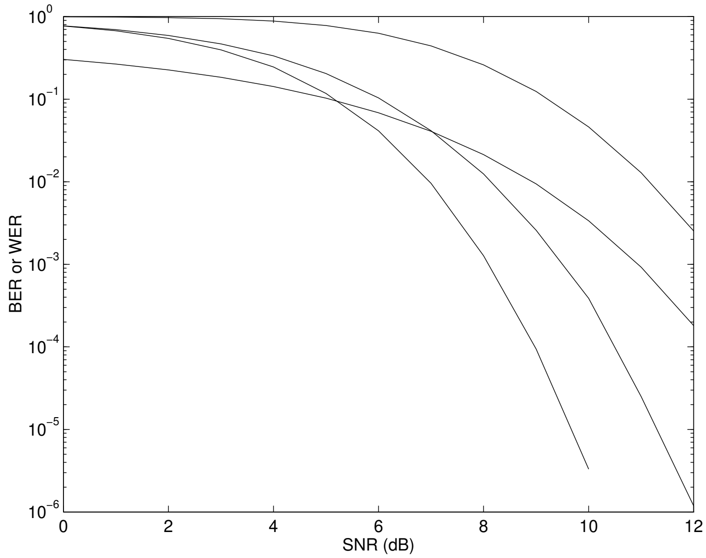


Figure 18. Simulated SITOR Decoder

demodulator loses 3 dB over the coherent one.) The computed total processing gain relative to the 2100-Hz SSB receiver is the sum, 7.5 + 5.75 = 13.25 dB.

The simulated performance of the RF predetection processing steps is evident in Figure 17 for both the linear and hard-limiter receivers. The figure shows the calculated and measured bit error rate (BER) for input SNR from -20 dB to 0 dB. The calculated curve is for the ideal noncoherent FSK demodulator corrected for the processing gain determined as above. While not always clear from the figure, the smooth line is the calculated characteristic, the top line the characteristic measured with the hard-limiter receiver and the bottom line the characteristic measured with the linear receiver. Clearly, the performance of both receivers is very close to ideal and for most purposes could be considered ideal. Where it might be necessary to distinguish between the two receivers, the hard-limiter one gives away probably less than a decibel in threshold and less than 50% in BER, relative to the linear receiver, throughout the useful SNR range.

The simulated performance of the SITOR decoder correlator and decision algorithms is shown in Figure 18. The uppermost two traces in this figure show the BER for the ideal noncoherent FSK demodulator (bottom), together with the uncoded WER for the same demodulator and 14-bit word (character) code (top). The remaining two traces show the WER of the coded CCIR signal using both hard-decision (top) and soft-decision (bottom) decoding. The latter two traces clearly show the effects of the FEC coding, in which the reduction in WER due to the FEC code at low SNR is only marginal, but the improvement at moderate to high SNR is spectacular. Assume for example a WER of 10^{-4}, which is considered acceptable for HF radio services. According to the figure, this is achieved with an input SNR of about 9.2 dB using soft-decision decoding. At this same SNR, the WER with hard-decision decoding is about 3×10^{-3}, some 30 times worse. The uncoded performance is over 0.1, which is clearly unacceptable.

The simulated performance of the RTTY decoder is shown in Figure 19. The uppermost of the three traces shows the WER for the asynchronous mode, the middle trace the WER for the synchronous mode and the bottom the WER for the ideal noncoherent FSK demodulator and 5-bit word (character) code. The WER is considerably worse than the CCIR decoder, both because of the lack of coding and also because of the need for threshold correction (see below). In fact, the asynchronous

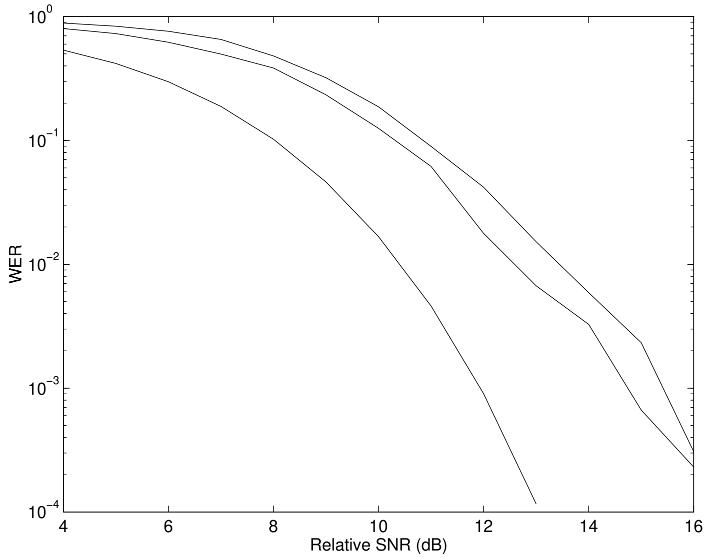


Figure 19. Simulated RTTY Decoder

mode codec gives up about 4 dB relative to the ideal along with a whopping two orders of magnitude increase if WER over much of the useful SNR range. In any case, the advantage of the synchronous mode over the asynchronous one is minor, about a decibel of processing gain and about 30% reduction in WER in the useful SNR range.

6.3. Proof of Performance

Following is an example comparison of a SITOR broadcast from AT&T Miami Radio WOM, which sends ship traffic lists, information and weather forecasts on a regular basis. This particular broadcast was purposely selected as representing marginally acceptable copy for amateur service. There are two copies of the same message, the first (Figure 20) decoded by the digital modem described in this memorandum, the second (Figure 21) decoded by the AEA PK-232 TNC, which is typical of multipurpose analog modems. Both modems were connected to the same SSB communications receiver. In both cases, the "_" character represents an erasure, where an uncorrectable error was detected. The messages have not been edited in any way, except some lines that spilled past the right margin have been folded to fit on these pages.

Most would probably agree that the digital modem copy is significantly more intelligible than the analog modem copy. A count of erasures shows 60 in the digital copy, 224 for the analog copy. In most cases where obvious errors have occurred in the analog copy, the digital copy is apparently correct; however, there are a few cases where obvious errors have occurred in the digital copy and the analog copy is apparently correct.

7. Conclusions

The above analysis and simulation results demonstrate the processing gain due to the predetection matched filter and noncoherent FSK demodulator is about 13.25 dB relative to the canonical SSB receiver. For a WER of 10^{-4} with the uncoded 14-bit word, the noncoherent FSK demodulator needs 13.49 dB of SNR, whilc the CCIR 476 soft-decision correlation decoder needs only about 9 dB for the same performance. This represents a conventional coding gain for the soft-decision decoder of about 4.5 dB over the ideal at the chosen WER. In contrast, the hard-decision decoder has a coding gain about 1.5 dB less than this, but still equivalent to a doubling in transmitter power.

```
CQ CQ CQ DE WOM WOM WOM
NATIONAL WEATHER SERVICE FORECAST RELAYED BY AT+T STATION WOMAQ
FOLLOWS:

_PY___AOOT 04:41:10 UTC
NATIONAL WEATHER SERVICE MIAMI FL
1130 PM EDT TUE JUN 27 1995
CARIBBEAN SEA AND SW N ATLC BEYOND 50 NM FROM SHORE.
.CARIBB_AN SYNOPSIS...NO SIGNIFICANT WEATHER FEATURES.
NW CARIBBEAN N OF 15N AND W OF 75W
.TONIGHT...S _F 17N WIND E 15 TO 20 KT. S__S 5 TO 7 FT. N L 17N
_____QRND E 10 TO 15 KT. SEAS 3 TO 5 FT.
.WED...WIND E 15 KT. SEAS TO 5 FT. ISOLATED TSTMS.
.WED NIGHT...WIND E TO SE 10 TO 15 KT. SEAS 3 TO 5 FT. ISOLATED
TSTMS. SW CARIBBEAN S OF 15N AND W OF 75W.
___TONIGHT....S OF 11N AND W OF 80W WIND VARIAB__ 10 TO 1_ KT.
SEAS 5 FT. ELSEWHERE WIND NE 20 KT. SEA___I FT. WIDELY SCATTERED
TSTMS S OF 13N.
.WED...S OF 11C AND W OF 80W WIND VARIABLE 1___59 15 KT. SEAS 5 FT.
ELSE_HERE WIND E 20 KT. SEAS 8 FT. WIDELY SCATTERED TSTMS.
.WED NIGHT...WIND E TO _E 10 TO 15 KT. SEAS 5 TOV___!5. WIDELY
SCATTERED TSTMS MAINLY S OF 12N.
E CARIBBEAN_ _F _UTW.
.TONIGHT_MMMN OF 15N WIND E
_PD KT. SEAY 5___5. S OF _
15 TO 20 KT. SEAS 5 TO 7 FT.
.WED...WIND E 15 KT. SEAS 5 TO 7 FT. WIDELY SCATTERED TSTMA MAINLY N
OF 15N./
#____3 ,8&#5.6___4, 3 10 TO _5 KT. SEAS TO 6 FT. ISOLATED TSTMS.
.OUTLOOK THU THROUGH SUN...LITTLE CHANGE.
SW N ATLC S OF 32N AND W OF 65W
.SYNOPSIS...WEAK HIGH P___ RIDGE FROM 22N 65W TO S FLORIDA __LL
XR__REAT WWD INT_ GULF OF MEXICO AS BIOAD LO PRES AREA NEAR _EPEN 72W__
DRIFTS SB_M
S.TONIGHT__MMN OF 27N E OF 70W WIND E TO SE 10 KT. S_A___T FT.
ELSEWHERE N OF RIDGE WIND W TO NW 15 KT. SEAS 6 FT. S OF RI_G_ IND
SE 0 KT EXCEPT V_CIABLE W OF 75W. SEES 4 FT. ISOLAT_D TSTMS.
.WED...N OF 25N E OF 70W WIND S TO SE 10 KT. SEAS 5 FT. ELSEWHERE N
OF RIDGE WIND W TO NW 10 TO 15 KT. SEAS TO 6 FT. S OF RIDGE WIND E
10 KT. SEAS 4 FT. ISOLATED TSTMS.
.WED NIGHT...N OF 25N AND W OF 77W
WIND VARIAB___ 59 10 KT. SEAS 4
FT. E OF 77W WIND N TO NW 10 KT. SEAS 5 _T. S OF 25N WIND VARIABLE
5 TO 10 KT EXCEPT W OF 75W WIND NE 10 TO 15 KT. SEAS 3 TO 5 FT.
SCATTERED SHOWERS AND ISOLATED TSTMS MAINLY W OF 75W AND N OF 25N.
.OUTLOOK THU THROUGH SUN...HIGH PRES RIDGE WILL RESTABLISH ALONG 22N
AS LO PRES CENTER MOVES SLOWLY INTO N ATLC. N OF RIDGE WIND SW TO W
ABOUT 15 KT AND S OF RIDGE E ABOUT 10 KT. SEAS 3 TO 5 FT.

NNNN

-AR-
```

Figure 20. Test Message - Digital Modem

This analysis, while conventional, is probably somewhat misleading, since the actual comparison should be between the 14-bit CCIR 476 system running at 100 bps and the conventional 7.5-bit Baudot (start plus data plus stop bit) system running at 50 bps, since both have the same throughput. Assuming the predetection filters are optimized for each speed, the predetection processing gain with the latter system is 3 dB more and the 10^{-4} threshold is 0.28 dB less. However, the above simulation results for the RTTY decoder operating in asynchronous mode show a processing loss 4 dB at a WER of 10^{-4}. Thus, the RTTY decoder operates at a net processing gain of about -0.7 dB

```
CQ CQ CQ DE WOM WOM WOM
NATIONAL WEATHER SERVICE FORECAST RELAYED BY AT+T STATION________
FOLLOWS:

06___-_995 04:41:10 UTC
NATIONAL WEATHER SERVICE MIAMI FL
1130 PM EDT TUE JUN 27 1995
CARIBBEAN SEA AN_ SW N ATLC BEYOND 5_ NM FROM SHORE___
._AR__B_RN__YNOPSIS...NO SIGNIFICANT WEATHER FEATURES.
NW CARIBBEAN N OF 15N AND W OF 75W
__MTONIGHT...S OF 17N WIND _ 15 TO 20 KT._ ____5 TO 7 FT._N _F_17N
____R_Q__D E 10_ 59 15 KT. SEAS 3 TO 5 FT_
.WED...WIND E 15 KT. SEAS TO 5 FT. ISOLATED TSTMS.
.WED NIGHT...W_ND E TO SE 10 TO 15 KT. SEAS 3 TO 5 FT. ISOLATED
TSTMS. SW CARIBBEAN S OF 15N AND W OF 75W.___/_4_59,8&#5.__._ OF 11N AND
OF 80W WIND VA_IA_LR 10 TO 15 KT.
SEAS _ FT. ELSEWHERE WIND NE 20 KT. S___QI_ FT. WIDELY SCATTERED
TST_S S OF 13N.
.WED...S OF ____4$__ 2 9! 80W WIND VARIABLE____4_9_15 KT. _EAS 5 FT.
ELSE_HERE WIND E 20 KT. SEAS 8 FT. WIDELY SCATTERED TSTMS.
.WED NIGHT...WIND E TO _E 10 TO 15 KT. SEAS 5_ _____?&!5. WID_ZY__
SCATTERED TSTMS MAINLY S OF 12N.
E CARIBBEAN_ _F _UTW.
.TONIGHT_MM_N___ 15N WIND E__YW_ KT. SEAS_5___. S OF 15__ 28,$ 3
15 TO 20 KT. SEAS 5 TO 7 FT._
.WED...WIND E 15 KT. SEAS 5 TO 7 FT. WIDELY SCATTERED TST_A__AINLY N
OF 15____TM__E_ NIGHT____QG
_D E _QP TO 15 KT. S_AS TO 6 FT. ISOLATED TSTMS.
.OUTLOOK THU THROUGH SUN...LITTLE __A_GE._
__ , -5): 9! 32N AND W OF 65W
.SYNOPSIS...WE_K_HIGH_PR_H_RIDGE FROM 22N 65W TO S FLORIDA RPG___
RO_REAT W_N _NT_ GULF OF MEXICO AS _ROA_ _O PRES AR_A__EAR___R_N
72________
DRIF_S SDTM_
__MITONIGHT_._-_ 9! 27N E OF 70W WIND E TO S___P_KT. SEAR___ FT.
ELSEWHERE N OF RIDGE WIND W TO NW 15 KT. SEAS 6 FT. S OF__I_GE_WI_D
SE _Q_ KT EXC__ _V___BLE W OF 75W. SE_S 4 FT. ISOLAT_D TSTMS.
.WED...__ 9! 25N E OF 70W WIND S TO SE 10 KT. SEAS 5 FT. ELSEWHERE N
OF RIDGE WIND W TO NW 10 TO 15 KT. SE_S_TO 6 FT. S OF RIDGE WIND E
10 KT. SEAS 4 FT. ISOLATED TSTMS.__.WED N_G_T_..___9! 25N AND W OF _UUW
WIND _AR_________ TO 10 KT. SEAS 4
FT. E OF 77W WIND N TO NW 10 KT. SEAS 5_ _._ 9! 25N WIND VARIABLE
5 TO 10 KT EXCEPT W OF 7_W WIND NE 10 TO 15 KT. SEAS 3 TO 5 FT.
SCATTERED SHOWERS AND ISOLATED TSTMS MAINLY W OF 75W AND N OF 25N.
.OUTLOOK THU THROUGH SUN...HIGH PRES RIDGE WILL RESTABLISH ALONG 22N
AS LO PRES_CENTER MOVES SLOWLY INTO N ATLC. N OF RIDGE WIND SW TO W__
ABO_T 15 KT AND S OF RIDGE E ABOUT_10 KT_ SEAS 3 TO 5 FT.

NNNN

-AR-
```

Figure 21. Test Message - Analog Modem

over the ideal and -5.2 dB relative to the SITOR decoder. In other words, the Baudot transmitter needs about three times the power as the CCIR transmitter for the same WER.

Closer inspection of the Baudot simulation shows the cause for most of the WER degradation is the the threshold corrector, which is necessary in order to reliably find the start bit and decode the data bits. The technique used in the DSP modem is similar in most respects to that used in conventional analog modems, in that the mark/space decision (slice) level is set midway between the maximum and minimum differential signal values over some past sample of the demodulated signal. Obviously, noise spikes can produce unwanted bias in the decision threshold, which cause timing errors in start-bit detection and misaligned sample intervals when decoding the data bits.

However, bias in the decision threshold is much less of a problem in the SITOR decoder for two reasons. First, the CCIR 476 code includes only those code combinations with four bits mark and three bits zero, thus has constant weight. Therefore, the bias shows up as a constant in the correlation function which compares the receuved vector with each of the 35 CCIR code vectors. Since only the maximum of the correlation function is used, a constant bias does not affect the decision. On the other hand, the Baudot code includes all 5-bit code combinations, so does not have constant weight. Second, the CCIR codeword phase is determined by a correlation function of some 70 intervals, which amounts to some 18 dB of processing gain. However, the Baudot codeword phase is determined on a bit-by-bit basis, at least in asynchronous mode. The simulations show a modest processing gain when using synchronous mode, but the gain is small compared to the loss due to threshold correction.

Contrary to some opinions expressed in the past, there is good reason to praise the CCIR 476 code design with respect to more efficient FEC codes that have been suggested. The constant-weight codebook avoids threshold errors, as demonstrated in the above simulations. Every codeword has at least one transition, which enhances bit phase recovery in degraded conditions of electrical and cochannel interference. The interleaved FEC transmission format, which does not affect the simulation results, provides additional improvement in fading conditions common in HF radio communication. The code is designed to be self-synchronous by transmitting a heroic unique word at least once every print line. Considering the results of the simulations described in this report, especially the reduced transmitter power required, the choice of CCIR over conventional Baudot transmission format is compelling.

8. References

[TI87]Texas Instruments, Inc. *Second Generation TMC320 User's Guide*. Document Number SPRU0014. Texas Instruments, Inc., 1987.

[TI92]Texas Instruments, Inc. *Linear Circuits, Volume 2 - Data Conversion*, DSP Analog Interface and Video Interface. Texas Instruments, Inc., 1987.

[LIT88]Little, J., and L. Shure. Signal processing toolbox for use with MATLAB. MathWorks, Inc., 1988.

[MOL90]Moler, C., J. Little, and S. Bangert. Pro-Matlab User's Guide. MathWorks, Inc., 1990.

[COU93]Couch, L.W. II, *Digital and Analog Communication Systems*. Macmillan, New York, 1993.

[WIL96] Wilson, S.G. *Digital Modulation and Coding*. Prentice Hall, Upper Saddle River, NJ, 1996.