Modulation Recognition for HF Signals

James E. Giesbrecht[a,b,c], Russell Clarke[c], and Derek Abbott[a,b]

[a]Centre for Biomedical Engineering, The University of Adelaide, SA 5005, Australia
[b]School of Electrical & Electronic Engineering, The University of Adelaide, SA 5005, Australia
[c]Ebor Computing, 147 Henley Beach Rd., Mile End, SA 5031, Australia

ABSTRACT

High-frequency (HF) communications is undergoing resurgence despite advances in long-range satellite communication systems. Defense agencies are using the HF spectrum for backup communications as well as for spectrum surveillance applications. Spectrum management organizations are monitoring the HF spectrum to control and enforce licensing. These activities usually require systems capable of determining the location of a source of transmissions, separating valid signals from interference and noise, and recognizing signal modulation. Our ultimate aim is to develop robust modulation recognition algorithms for real HF signals, that is, signals propagating by multiple ionospheric modes.

One aspect of modulation recognition is the extraction of signal identifying features. The most common features for modulation recognition are instantaneous phase, amplitude, and frequency. However, this paper focuses on two feature parameters: coherence and entropy. Signal entropy and the coherence function show potential for robust recognition of HF modulation types in the presence of HF noise and multi-path. Specifically, it is shown that the methods of calculation of coherence and entropy are important and that appropriate calculations ensure stability in the parameters. For the first time a new metric, called *Coherence-Median Difference* (CMD), is introduced that provides a measure of the dominance of coherence at specific frequencies to coherence at all other frequencies in a particular bandwidth.

Keywords: Modulation Recognition, Entropy, Coherence, Compression, HF.

1. INTRODUCTION

The HF radio band (nominally 2 MHz to 30 MHz) can be used for long-distance wireless communications, because the ionosphere and its various layers refract transmissions in this band. Such refraction enables signals to propagate beyond the horizon to distant receivers unable to be reached by higher frequency (VHF and above) signals. This fact makes the HF band attractive for private and commercial interests as well as for defense forces spread across the globe. For example, spectrum management agencies monitor the HF band for unlicensed operators and military agencies use the HF band for communications.

In the past, HF communication systems were analog. This meant that signals received by an antenna were down-converted to baseband using filters, oscillators, and many discrete components. Receivers were generally constructed in a super-heterodyne configuration.[1] The baseband signals were then passed through demodulators to extract the information content. Often special demodulators had to be switched in to demodulate signals with different modulation schemes.

Today there are so many different modulation techniques that monitoring and detection of such signals using traditional methods would require numerous HF receivers. Monitoring and detection would also require some prior knowledge of the signals so as to choose the correct receiver. However, with the advent of software radio this task is much easier.

Software radio aims to replicate hardware functions in software running on a generic platform. In so doing many of the problems associated with hardware implementations are avoided. In addition, the receiver and

Send correspondence to:
The University of Adelaide - E-mail: james@eleceng.adelaide.edu.au; Phone: +61 (8) 8303 6296
Ebor Computing - E-mail: james.giesbrecht@ebor.com; Phone: +61 (8) 8238 0300; Fax: +61 (8) 8238 0304

Smart Structures, Devices, and Systems II, edited by Said F. Al-Sarawi,
Proceedings of SPIE Vol. 5649 (SPIE, Bellingham, WA, 2005)
0277-786X/05/$15 · doi: 10.1117/12.582077

transmitter chains can easily be changed to accommodate various modulation schemes. Here is where modulation recognition plays an important role. Modulation recognition enables a receiver to automatically switch in the correct software defined demodulator for a particular signal.

This paper continues the work presented earlier by Giesbrecht, Clarke, and Abbott.[2,3] In the previous work we discussed the preliminary application of coherence[4] and entropy[5] to real HF signals. This work discusses in more detail the fundamentals of coherence and entropy and how they might be used in modulation recognition.

2. MODULATION RECOGNITION

The process of determining the modulation type of a signal with no foreknowledge of the signal modulation characteristics is known as modulation recognition. Fundamental processes of modulation recognition are feature extraction and classification. Feature extraction determines unique characteristics of the signal so that a classifier can establish the modulation type. Common features include instantaneous amplitude, variance of phase, spectral symmetry, transmission models, and higher order statistics. Classification associates features with modulation types. These associations are normally made through threshold detection logic, artificial neural networks (ANNs), or pattern recognition algorithms.

Previous work[2,3] on real HF signals has shown that most recognition methods do not adequately rationalize the choice of thresholds and feature functions. Moreover, they do not assume appropriate noise and channel models for application of their methods to the HF band, and they do not apply their methods to real signals. Figure 1 illustrates power spectra of typical HF signals.

Nevertheless, the most common features are statistical measures of instantaneous amplitude, phase, and frequency.[6–9] Others have considered zero-crossings, covariance, and higher-order statistics (e.g. m$^{\text{th}}$ -order cumulants and kurtosis).[10–14] This work concentrates on the fundamentals of coherence and signal entropy as they apply to modulation recognition.

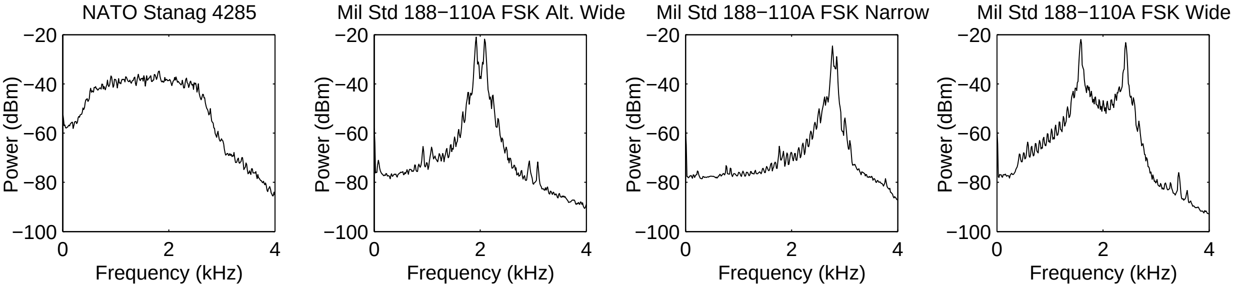

Figure 1. Power spectra of various real HF signals. The NATO Stanag 4285 is an 8-PSK signal while the other plots are of 2-FSK signals.

3. SIGNAL FEATURES

3.1. Signal Entropy

Shannon's[15] work in communication shows that *entropy* is the uncertainty in one's ability to predict the output of a data source and is equal to the information provided by that source when an event occurs. Moreover, entropy is defined as

$$H(\mathbf{X}) = \sum_{i=1}^{N} P(x_i)I(x_i) \tag{1}$$

where $P(x_i)$ is the probability of the i^{th} character in an alphabet of N characters that makes up the data source represented by the random variable $\mathbf{X}$, and $I(x_i)$ is the self-information provided by the i^{th} character. Self-information is defined as

$$I(x) = -\log_N P(x). \tag{2}$$

Hence for a binary digital signal with characters 0 and 1 the entropy is

$$H(\mathbf{X}) = -P(0)\log_2 P(0) - (1 - P(0))\log_2 (1 - P(0)) \tag{3}$$

where $P(0)$ is the probability of a zero-bit and $1 - P(0)$ is the probability of a one-bit.

For entropy computed over the length of a message Shannon shows that, provided successive symbols are statistically independent, (1) can be estimated by

$$\hat{H}(\mathbf{X}) = \frac{1}{L}\sum_{i=1}^{L} \hat{P}(x_i)\hat{I}(x_i), \tag{4}$$

where L is the number of symbols in the message that contains a distribution of all possible symbols, $\hat{P}(x_i)$ is an estimate of the probability of each symbol in the set of all possible symbols, and $\hat{I}(x_i)$ is an estimate of the information provided by each symbol, such that

$$\lim_{L\to\infty} \hat{H}(\mathbf{X}) = H(\mathbf{X}). \tag{5}$$

Benedetto, Caglioti, and Loreto[5] present a method whereby the similarity or dissimilarity of two information sources, $\boldsymbol{A}$ and $\boldsymbol{B}$, is defined by their relative entropy. They compress a long sequence A from $\boldsymbol{A}$ and subtract the compressed length, L_A, from the compressed length (L_{Ab}) of the sequence $A+b$, where $A+b$ is the concatenation of A and a small sequence b from $\boldsymbol{B}$. This difference is normalized by the length of b (represented by $|b|$), to arrive at the entropy of $\boldsymbol{A}$:

$$S_{Ab} = \frac{L_{Ab} - L_A}{|b|}. \tag{6}$$

The entropy of $\boldsymbol{B}$ with respect to $\boldsymbol{A}$ is similarly calculated. However in the work presented here we consider only S_{Ab} in relation to $H(\mathbf{X})$.

In the context of modulation recognition why bother with this entropy-compression technique? Firstly, it is a relatively new method that has not previously been thoroughly compared to Shannon's method and may provide some benefit for modulation recognition. Secondly, as shall be seen later, it is sensitive to the message structure whereas Shannon's method is not. And lastly, it is a subject of ongoing investigation as to whether or not signal entropy provides a useful parameter for recognition of signal modulation.

This work attempts to answer the following questions. How does S_{Ab} compare with $H(\mathbf{X})$ for various probabilities of a zero-bit? Does the compression method used for S_{Ab} play a role? What is the affect of $|b|$ on the entropy calculations? And, how can entropy be used in modulation recognition?

3.2. Coherence

Another measure that appears useful for modulation recognition is the coherence function. In general, the coherence function is analogous to correlation coefficients in the frequency domain. Consider the power-spectral density of a signal X. It is defined as

$$P_{xx}(f) = \int_{-\infty}^{\infty} \rho_{xx}(\tau)e^{-j2\pi f\tau}d\tau \tag{7}$$

where f is frequency, τ is time delay, and $\rho_{xx}(\tau)$ is the auto-correlation function of X. In a similar manner, the cross-spectral density is defined by

$$P_{xy}(f) = \int_{-\infty}^{\infty} \rho_{xy}(\tau)e^{-j2\pi f\tau}d\tau \tag{8}$$

where $\rho_{xy}(\tau)$ is the cross-correlation of signals X and Y. From these definitions the coherence function is the cross-spectral density normalized by the respective power-spectral densities;

$$C_{xy}(f) = \frac{|P_{xy}(f)|^2}{P_{xx}(f)P_{yy}(f)}. \tag{9}$$

For each frequency, the coherence function varies between 0 and 1 and indicates the similarity between X and Y. If the value of the function at a particular frequency is close to unity, it indicates that X and Y are similar at that frequency. On the other hand, if the value of the function is near zero it implies that the two signals are dissimilar at the particular frequency. Furthermore, if the coherence is unity it necessarily means that X and Y are correlated.

Carter's[4] excellent treatise on the estimation of coherence shows that the coherence between X and Y is also related to the signal-to-noise ratio (SNR) of Y assuming that X is the transmitted signal and Y is the noisy received signal. In this description

$$C_{xy}(f) = \frac{\mathrm{SNR}_y(f)}{1+\mathrm{SNR}_y(f)}, \tag{10}$$

where

$$\mathrm{SNR}_y(f) = \frac{P_{xx}(f)\,|H(f)|^2}{P_{nn}(f)}, \tag{11}$$

and where $H(f)$ is the transfer function of the transmission channel and $P_{nn}(f)$ is the power-spectral density of the received noise. Note that with this definition, a coherence of unity not only demonstrates complete correlation of X and Y but also implies an infinite SNR since in the limit as $\mathrm{SNR}_y(f) \rightarrow \infty$ the coherence tends to 1.

Despite the simplistic appearance of (9) and (10) the calculation and measurement of coherence can be quite difficult. For example Proakis[16] demonstrates that the calculation of the power spectral densities of FSK signals, and therefore coherence, is not an easy task. Therefore coherence is usually estimated. Carter[4] points out that the coherence function between two signals is extremely sensitive to misalignment in time. He suggests that the best methods to estimate coherence are *weighted overlapping segment averaging* (WOSA) (e.g. Welch's periodgram method[17]) and *lag reshaping.* For each of these techniques the signals are broken into windowed overlapping segments. As the number of segments increase, the variance of the coherence estimate decreases. Moreover, the percentage of overlap between segments affects the bias. The bias in the estimate of coherence decreases as the percentage overlap increases. However there is a point of diminishing returns (i.e. about 50% - 60% overlap) where further increase in overlap provides minimal reduction in the bias. Consequently, an accurate estimate of coherence depends on the choice of number of segments, percentage overlap, and windowing method.

So, how does coherence vary with the message in the signal? How does the coherence of a signal vary with SNR and frequency? Furthermore, how can coherence be used to identify a signal's modulation?

3.2.1. Coherence-Median Difference

Related to coherence we define the coherence-median difference (CMD) for FSK signals as the difference between the mean of the coherences at the symbol frequencies less the median coherence over a bandwidth of interest. CMD is described by

$$\mathrm{CMD} = \frac{1}{M}\sum_{i=1}^{M} C_{xy}(f_i) - \mathrm{median}(C_{xy}(f)\forall f \in \mathrm{BW}) \tag{12}$$

where f_i is the i^{th} symbol frequency, M is the number of modulation levels (or symbol frequencies), and BW is the bandwidth over which the CMD is computed. For a BFSK signal with two symbol frequencies (the so-called mark and space frequencies), (12) reduces to

$$\mathrm{CMD} = \frac{C_{xy}(f_{\mathrm{m}}) + C_{xy}(f_{\mathrm{S}})}{2} - \mathrm{median}(C_{xy}(f)\forall f \in \mathrm{BW}) \tag{13}$$

where f_{m} and f_{S} are the mark and space frequencies respectively.

Positive CMD indicates that the coherence at the symbol frequencies is above the coherence at all other frequencies in the bandwidth of interest. Negative or zero CMD implies that the coherence at other frequencies dominate the coherence across the bandwidth as a result of 1) poor selection of the number of overlapping segments, 2) poor selection of the amount of overlap, and/or 3) truly poor coherence between X and Y. Later it will be seen that the CMD can be used to provide a measure of certainty that the peak coherence is indeed that of the mark or space frequencies.

4. METHOD AND RESULTS

4.1. Signal Entropy

Two binary sequences A and B, with equal probability of a zero-bit, were created to compare the entropy as defined by (1) with that of (6). The compression algorithm used for (6) was the LZW algorithm with 12-bit codes. The probability of a zero-bit was varied and plotted in Figure 2.

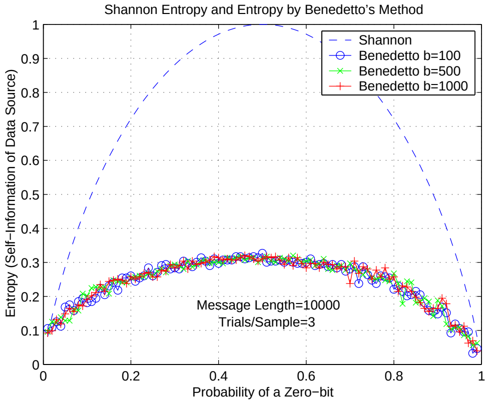

Figure 2. Shannon's entropy versus entropy computed with (6) for various zero-bit probabilities and lengths of appended sequences. The message length is 10,000 bits with the probability of a zero-bit as shown on the abscissa. The probability of a zero-bit in the appended sequence is equal to the probability of a zero-bit in the message. Each data point is the average entropy of a 3-trial experiment in which each trial consists of generating a random message of the required length and zero-bit probability and then computing the entropy of the message for each of the appended sequence lengths.

As can be seen, the length of b does not greatly affect S_{Ab}. In fact, S_{Ab} greatly underestimates $H(\mathbf{X})$. At a probability of 0.5, S_{Ab} underestimates $H(\mathbf{X})$ by 70%! Now if $|b|$ is held constant and the number of samples varied, as in Figure 3, S_{Ab} comes much closer to $H(\mathbf{X})$ but with slight skewing. The variance of the samples on the curves also appears to increase. For message lengths of 100 and $|b| = 25$ the entropy S_{Ab} approaches Shannon's entropy curve and at times surpasses it.

If S_{Ab} is plotted (see Figure 4) as a function of the length of the message and $|b|$ it becomes clear that entropy is relatively flat for long messages and large $|b|$. Only when the message length decreases and $|b|$ decreases is there a "hill" in the entropy at the origin. In addition, the entropy is slightly more sensitive to changes in $|b|$ as the message length decreases than for changes in message length as $|b|$ decreases.

In part, this behavior is explainable by considering the operation of the compression algorithm. As the length of A decreases the dictionary being built by the compression algorithm becomes smaller because there are fewer message bits. And because the dictionary is smaller, when the compressor begins compressing the part of the sequence attributed to b, it finds new words that are not in the dictionary and therefore requires longer codes to encode the new words. This in turn gives rise to a larger entropy measure because the algorithm compresses $A+b$ less efficiently than it does for longer sequences of $A+b$. Consequently, there is a rise in the entropy as the length of A decreases for most lengths of b. For the case of small lengths of b, the randomness of the message begins to affect the entropy more pronouncedly. This is evidenced by the "spikes" in the surface at low $|b|$.

The entropy "hill" noted above is also partly a result of inefficiencies in the compression algorithm. If one considers the performance of the LZW algorithm as compared to another compression algorithm, say Zip 2.3[*], it is apparent why the entropy rises at the origin in Figure 4. Figure 5 shows that the 12-bit LZW algorithm is generally less efficient than the Zip 2.3 algorithm and it is also less efficient at compressing 1's than it is at compressing 0's[†]. It is this skewing effect that yields the skewing in entropy seen previously. When (6) is recomputed using Zip 2.3 a much more consistent entropy results. Figures 6 and 7 illustrate this.

[*] Zip 2.3 is based on the search algorithm of Rabin and Karp[18] and the compression algorithm of Fiala and Greene.[19]

[†] A 14-bit LZW algorithm is similarly inefficient.

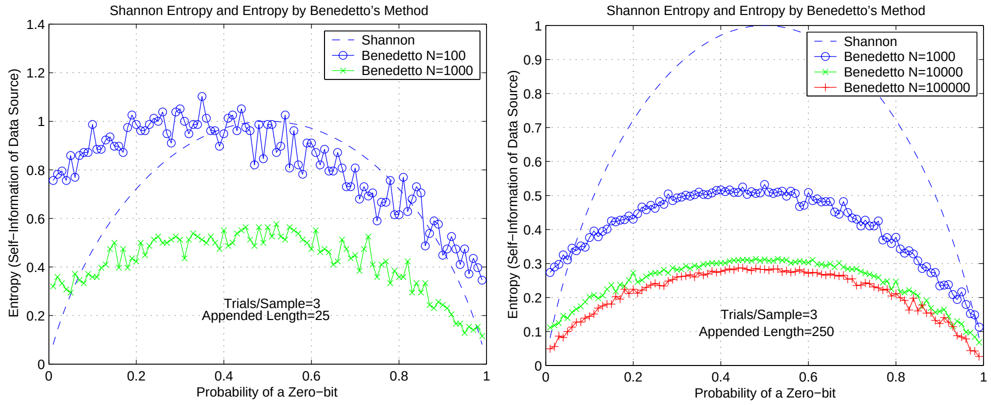

Figure 3. Entropy for various message lengths (N) and $|b| \in \{25, 250\}$. Entropy is computed in the same fashion as Figure 2 with the exception that the appended sequence is constant for varying message lengths. As the message length decreases the entropy curve via Benedetto's method approaches Shannon's entropy curve. The "roughness" of the curve increases as the appended length decreases (cf. left against right). Skewing is due to the LZW compression algorithm.

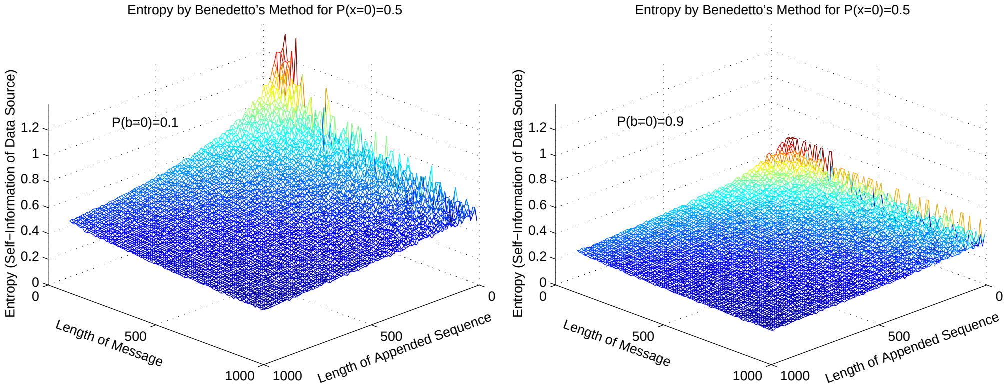

Figure 4. Surface plot of entropy as determined by (6) for various message lengths (N) and $|b|$. In both diagrams the probability of a zero-bit in the message is $P(\boldsymbol{X} = 0) = 0.5$. The probability of a zero-bit in the appended sequence is $P(\boldsymbol{B} = 0) \in \{0.1, 0.9\}$. The rise in the surface at the origin is due to the LZW compression algorithm.

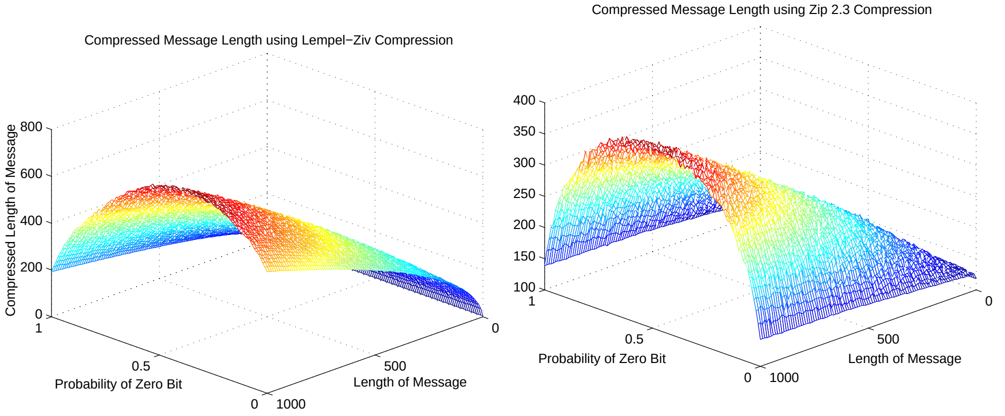

Figure 5. Surface plots of 12-bit LZW and Zip 2.3 compression algorithms for various message lengths and zero-bit probabilities.

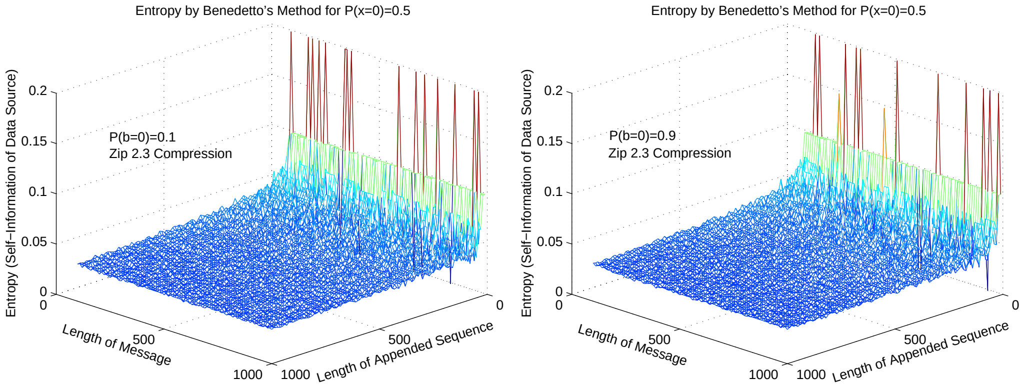

Figure 6. Surface plot of entropy, as determined by (6) using the Zip 2.3 compression algorithm, for various message lengths, N, and $|b|$. In both diagrams the probability of a zero-bit in the message is $P(\boldsymbol{X} = 0) = 0.5$. The probability of a zero-bit in the appended sequence is $P(\boldsymbol{B} = 0) \in \{0.1, 0.9\}$. The rise at low $|b|$ is consistent with the observation in Figure 3 while the "spikes" are attributed to the randomness of the message.

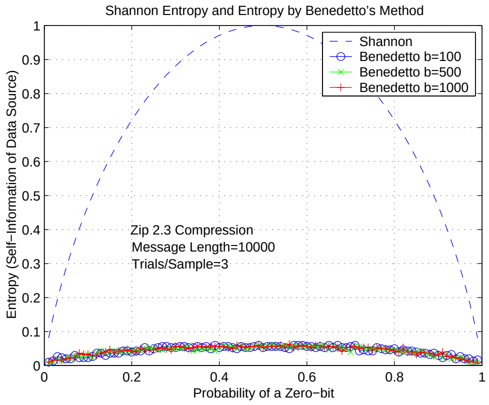

Figure 7. Comparison of entropy of (6) computed with Zip 2.3 compression and Shannon's entropy formula. Compare this with that of Figure 2. Though the overall entropy is lower, the skewing present in Figure 2 is not present here.

Next, consider two binary sequences of length L consisting of $\frac{L}{2}$ binary 1's and $\frac{L}{2}$ binary 0's and having identical probability distributions. Suppose that one sequence, S_1, has a completely random arrangement of the 1's and 0's, and the other sequence, S_2, has a structured arrangement such that the first $\frac{L}{2}$ binary digits are consecutive 0's and the second $\frac{L}{2}$ binary digits are consecutive 1's (the reverse can likewise be supposed). In such a case, Shannon's method in (4) will compute identical entropies for S_1 and S_2 while Benedetto's method in (6) will not. Figure 8 shows that Shannon's method is insensitive to the structure of the message as would be expected from the definition of the entropy. For large L, Benedetto's method shows a marked difference in entropy between S_1 and S_2. The reasons for this behavior are not immediately apparent and are still a subject of investigation.

The conclusion of all of this is that the compression algorithm used in computing (6) plays a critical role. If an inefficient compressor is used, the measured signal entropy can vary with not only the message but the compressor as well! An efficient and consistent compressor is needed for good estimates of entropy. However, the usefulness of entropy in modulation recognition is still unknown because entropy can vary with the message.

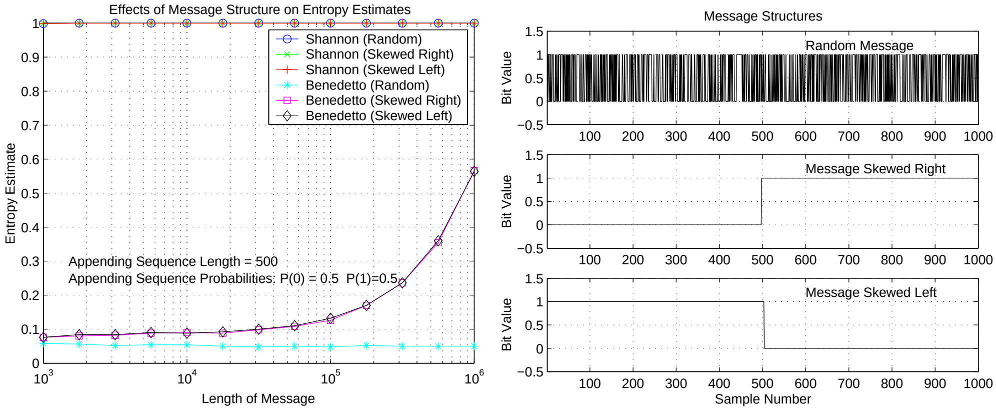

Figure 8. Entropy (left) as computed by Shannon's method and Benedetto's method for a message with a random arrangement of bits and a message with a structured arrangement of bits (right). Zip 2.3 compression is used for the calculation of entropy by Benedetto's method.

4.2. Coherence

4.2.1. Coherence vs. Hamming Distance

Two arbitrary BFSK signals were produced to study the effects of the message on the coherence between a transmitted signal and its noisy received counterpart. The mark frequency ($f_{\rm m}$) for each signal was 3 kHz and the space frequency ($f_{\rm s}$) was 1 kHz. During spectral processing $f_{\rm m}$ and $f_{\rm s}$ did not necessarily fall in specific frequency bins so $C_{xy}(f_{\rm m})$ and $C_{xy}(f_{\rm s})$ were estimated using linear interpolation between the coherence computed at adjacent frequency bins. To simulate bit errors in the received signal Y, the Hamming distance between Y and the original transmitted signal X[‡], was varied and the coherence computed using methods suggested by Carter.[4] X was generated with a uniform distribution of marks (1's) and spaces (0's). For each Hamming distance $Y = f(X, d)$ where $f(X, d)$ was a function that returned X with a random selection of d bits (the Hamming distance) inverted. For example, when d was zero $f(X, d)$ returned X, and when d was ten, $f(X, d)$ returned X with 10 randomly chosen bits inverted.

Figure 9 illustrates the effects of Hamming distance on coherence between X and Y. We immediately observe that the curvature of the coherence decreases as the length of the message increases (in this study message length is equal to the maximum Hamming distance shown). Furthermore, the CMD indicates that the peak coherence is generally dominated by the coherence at the mark and space frequencies and that as the length of message increases the peak coherence averages out above 0.6. This suggests that for long messages the coherence function can be used to identify BFSK signals with a thresholding technique.

The peak coherence also follows closely the coherence at $f_{\rm s}$. The fact that the peak coherence is larger than the coherence at $f_{\rm s}$ indicates that the interpolation used to compute the coherence is underestimating the true coherence at that frequency. A parabolic interpolator would provide a better estimate.

Finally, the CMD gradually increases from zero because at low Hamming distances the average coherence at the mark and space frequencies is near the median coherence.

Now, why does the coherence vary with the message? Consider the power spectrum of two BFSK signals with no noise or interference. If each BFSK signal consists of the first half being marks and the second half spaces there would be two relatively "clean" tones in each of the respective power spectrums. The coherence between these two BFSK signals will be near unity. However, as the distribution of marks and spaces of the second BFSK signal becomes more uniform the resulting numerous symbol transitions spreads the signal energy over many more frequencies. So even though the second BFSK signal contains the same tones as the "clean" signal, the overall coherence is affected because the signal energy is spread.

[‡] An alternative interpretation is that Y is an arbitrary BFSK signal being compared against an arbitrary BFSK reference signal X, with varying levels of correlation.

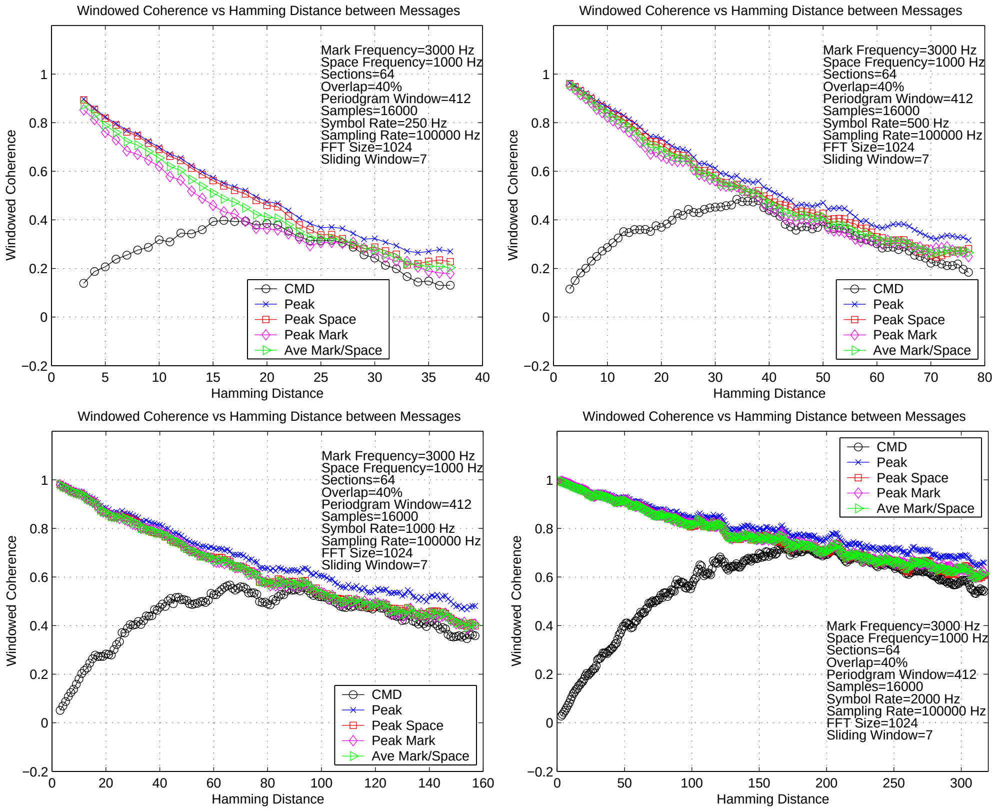

Figure 9. Coherence and CMD of two BFSK signals as a function of Hamming distance between the signals. The figures show coherence and CMD for two BFSK signals each having 40 symbols (upper-left), 80 symbols (upper-right), 160 symbols (lower-left), and 320 symbols (lower-right). Shown are coherences for the mark and space frequencies, the mean of the coherence of the mark and space frequencies, the peak coherence across the bandwidth of interest, and the CMD. As the number of symbols increase the slopes of the coherence curves decrease. In each figure the CMD is positive indicating that the coherence at the mark and space frequencies dominate the coherence function across the bandwidth of interest. A sliding boxcar window of 7 samples is used to smooth the plots.

In all cases as the Hamming distance increases the coherence decreases. Moreover, as the Hamming distance increases the power spectra of the two BFSK signals become less and less similar. Eventually a Hamming distance is reached that makes the two BFSK signals very different and hence the coherence is low. However, the coherence tends toward unity as the message length increases because each incremental change in the Hamming distance becomes less and less significant.

4.2.2. Coherence vs. SNR

In this experiment the coherence was computed for a 1 kHz tone with varying levels of additive white-Gaussian noise (AWGN) to study the effects of SNR and frequency. Although AWGN is not representative of noise in the HF spectrum it will suffice for the sake of the following discussion that shows why coherence estimates need to account for the signal bandwidth.

Ideally, at an arbitrary frequency coherence versus SNR appears as in Figure 10. This curve represents the coherence as a function of SNR where the noise is confined to an infinitesimal bandwidth about the arbitrary frequency.

In reality, SNR can never be defined over an infinitesimal bandwidth. So, if coherence is calculated using (11) over, say, a 1 kHz bandwidth around the 1 kHz tone we find that the S-curve of Figure 10 shifts right or left depending on whether the f in $C_{xy}(f)$ is above or below the 1 kHz tone. This behavior is simply a result of the fact that the SNR at each frequency in the 1 kHz bandwidth is different. For example, the SNR at the 1

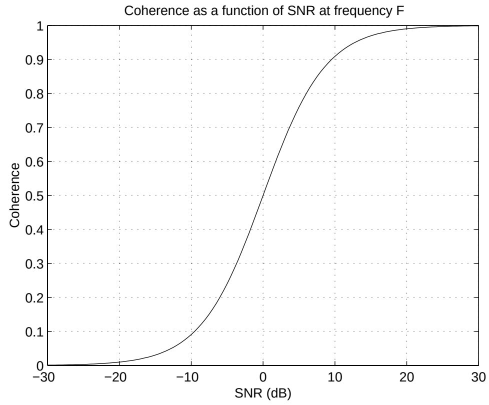

Figure 10. Coherence versus SNR at an arbitrary frequency where the noise is confined to a very small bandwidth about that frequency. This is simply a graphical representation of (10).

kHz tone will be much higher than the SNR at $1 \pm \Delta f$ kHz since the magnitude of the 1 kHz tone at $f = 1 \pm \Delta f$ kHz will be near zero and the noise will be something larger than zero.

If we now specify a bandlimited sinc function (which can be interpreted as a series of tones in the 1 kHz bandwidth) centered at 1 kHz and then restrict the AWGN to a 600 Hz band around the center we find that the coherence S-curves in the 600 Hz band fall on top of each other, while the coherence S-curves outside the 600 Hz band are shifted away from the 0 dB SNR position (see Figure 11). Coherence curves outside the 600 Hz band (i.e. $C_{xy}(f)$ for 500 Hz $\leq f \leq$ 700 Hz and 1300 Hz $\leq f \leq$ 1500 Hz) are not greatly affected by noise and as a result are shifted left because the SNR at each frequency is very high.

Consequently, it is obvious that coherence is affected by noise bandwidth, signal bandwidth, frequency, and SNR. It is also apparent that for strong signals and weak noise at a particular frequency the coherence S-curve moves left. Conversely for weak signals and strong noise at a particular frequency the coherence S-curve moves right. So, to get a reasonable measure of coherence for an arbitrary signal, the coherence must be computed over the bandwidth of the signal. If compared against noise, the SNR must be specified over the bandwidth. If another signal, that signal must also be specified over the bandwidth.

Thus coherence is indeed a "brute-force" method, as proposed previously,[3] since it is only useful for determining whether or not a signal is the same type as a reference signal. For example, if the reference signal is a BFSK signal with $f_{\mathrm{m}} = f_1$ and $f_{\mathrm{s}} = f_2$ and the received signal is a BFSK signal with $f_{\mathrm{m}} = f_3$ and $f_{\mathrm{s}} = f_4$ the coherence will only indicate that the two signals are not the same. Inferring that the received signal is a BFSK signal, with different mark and space frequencies, from knowledge of the reference signal and the calculated coherence is unlikely.

If coherence S-curves are plotted for various Hamming distances between two BFSK signals, as in Section 4.2.1, a series of curves result (see Figure 12). For negative SNR, the curves merge and the coherence drops to zero which implies that the average coherence at the mark and space frequencies does not dominate the coherence function. That is, as the noise power increases the two BFSK signals become less and less similar. For positive SNR the average coherence for the mark and space frequencies is primarily affected by the Hamming distance.

If one considers the CMD curves, the CMD tends to go negative at very low SNR (less than -10 dB) which indicates that the coherence at frequencies other than the f_{m} and f_{s} dominate the coherence function. The CMD remains positive for SNR greater than -10 dB as the Hamming distance increases. But, for a Hamming distance of zero the two BFSK signals are the same and consequently the CMD is zero since the median coherence over the signal bandwidth is the same as the average coherence between mark and space frequencies.

The high level of coherence for all the noted Hamming distances and for SNR greater than approximately -5 dB, as well as the positive CMD for SNR above -10 dB suggests that the coherence function is a robust "yes/no" measure to determine whether a received signal is the same modulation as a reference signal.

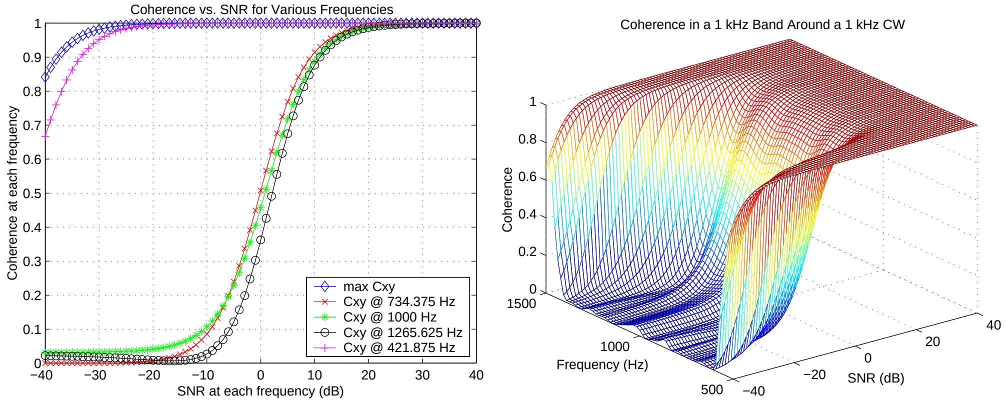

Figure 11. Coherence versus SNR and frequency about a bandlimited sinc function (right). The bandwidth of the sinc function is 1 kHz and the bandwidth of the AWGN is 600 Hz. The projection (left) shows that the S-curves in the 600 Hz band are nearly identical. Curves outside the band are shifted left.

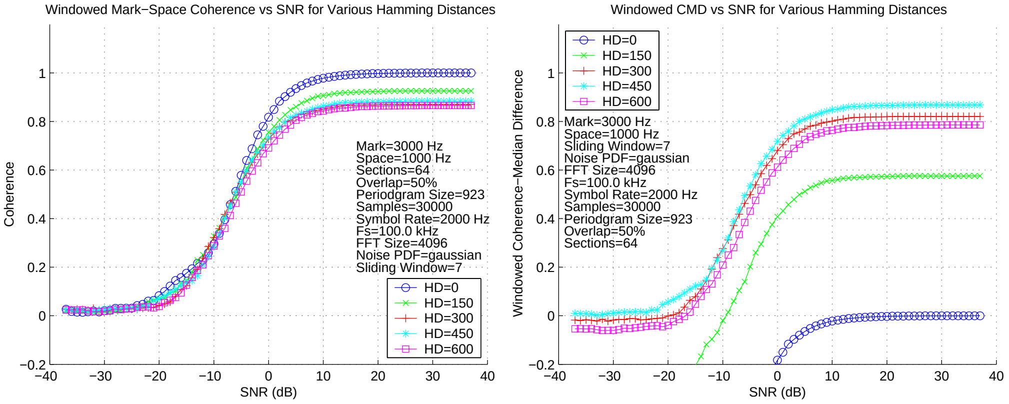

Figure 12. Average coherence $\left(\frac{C_{xy}(\mathrm{fm})+C_{xy}(\mathrm{fs})}{2}\right)$ at the mark and space frequencies versus SNR (left) and CMD at the mark and space frequencies versus SNR (right) for various Hamming distances. AWGN is bandlimited to 3 kHz around $f_\mathrm{c} = \frac{\mathrm{fm}+\mathrm{fs}}{2}$. As Hamming distance increases coherence decreases at positive SNR. At negative SNR coherence decreases to zero because the two BFSK signals become less and less similar with increasing noise. Positive CMD indicates that coherence at the mark and space frequencies dominate the 3 kHz bandwidth. For a Hamming distance of zero, the CMD drops to zero because the average mark-space coherence is the same as the median coherence for high SNR.

5. SUMMARY AND FUTURE WORK

Two methods for recognizing HF modulation schemes are discussed. One technique measures the entropy between pairs of modulated HF signals. The other method attempts to identify similarities between pairs of HF signals based on estimates of the coherence function.

Entropy, as calculated with a compression technique, is shown to greatly depend on the compression method, is susceptible to inefficiencies in the compression method, and can vary with the message. A clear application of this method of entropy estimation to modulation recognition is yet to be determined.

The concept of the coherence-median difference (CMD) is introduced and the coherence function, with the help of CMD for BFSK signals, shows great promise in being able to identify whether or not a received signal has the same modulation as a reference BFSK signal. For BFSK signals with long messages, the coherence function tends to rise above 0.6 and therefore a "brute-force" thresholding technique is a possible classification algorithm.

Future work will apply the knowledge presented here to the sample sets of *real* HF signals collected earlier.[2,3] The current research focus is building the analysis portion of the research platform presented in the previous work.

ACKNOWLEDGMENTS

Financial support for this work is provided by the The University of Adelaide and Ebor Computing.

REFERENCES

1. J. Smith, *Modern Communication Circuits*, McGraw-Hill Series in Electrical Engineering, McGraw-Hill Inc., New York, 1st ed., 1986. pp. 1-13.
2. J. E. Giesbrecht, R. Clarke, and D. Abbott, "Improved techniques for monitoring the HF spectrum," in *Proceedings of SPIE*, **5274**, 2003.
3. J. E. Giesbrecht, R. Clarke, and D. Abbott, "Monitoring the HF spectrum in the presence of noise," in *Proceedings of SPIE*, **5649**, 2004.
4. G. C. Carter, *Coherence and Time Delay Estimation: An Applied Tutorial for Research, Development, Test, and Evaluation Engineers*, IEEE Press, Piscataway, NJ, 1993.
5. D. Benedetto, E. Caglioti, and V. Loreto, "Language trees and zipping," *Physical Review Letters* **88**(4), p. 048702, 2002.
6. A. O. Hero III and H. Hadinejad-Mahram, "Digital modulation classification using power moment matrices," in *Proceedings of the IEEE 1998 International Conference on Acoustics, Speech, and Signal Processing*, **6**, pp. 3285–3288, (Seattle), 1998.
7. H. Ketterer, F. Jondral, and A. H. Costa, "Classification of modulation models using time-frequency methods," in *Proceedings of the IEEE 1999 International Conference on Acoustics, Speech, and Signal Processing*, **5**, pp. 2471–2474, (Phoenix, Arizona), 1999.
8. M. L. D. Wong and A. K. Nandi, "Automatic digital modulation recognition using spectral and statistical features with multi-layer perceptrons," in *Proceedings of the IEEE International Symposium on Signal Processing and its Applications*, **2**, pp. 390–393, (Kuala Lumpur, Malaysia), 2001.
9. K. E. Nolan, L. Doyle, P. Mackenzie, and D. O'Mahony, "Modulation scheme classification for 4G software radio wireless networks," in *Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA 2002)*, pp. 25–31, (Crete, Greece), 2002.
10. J. Aisbett, "Automatic Modulation Recognition Using Time Domain Parameters," Tech. Rep. ERL-0367-TR, Electronic Research Laboratory DSTO, Department of Defence, Commonwealth Government of Australia, November 1986.
11. S.-Z. Hsue and S. S. Soliman, "Automatic modulation classification using zero crossing," *IEE Proceedings* **137 Pt. F**, pp. 459–464, December 1990.
12. A. K. Nandi and E. E. Azzouz, "Modulation recognition using artificial neural networks," *Signal Processing* **56**, pp. 165–175, 1997.
13. A. K. Nandi and E. E. Azzouz, "Algorithms for Automatic Modulation Recognition of Communication Signals," *IEEE Transactions on Communications* **46**, pp. 431–435, April 1998.
14. W. Akmouche, "Detection of multicarrier modulations using 4th-order cumulants," in *MILCOM 1999 Military Communications Conference*, **1**, pp. 432–436, Institution of Electrical and Electronic Engineers, October 1999.
15. C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal* **27**, pp. 379–423, 623–656, July, October 1948.
16. J. G. Proakis, *Digital Communications*, Communications and Signal Processing, McGraw-Hill Book Company, 2nd ed., 1989.
17. P. D. Welch, "The use of the fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics* **AU-15**, pp. 70–73, June 1967.
18. R. Sedgewick, *Algorithms*, Addison-Wesley, Reading, Massachusetts, 2nd ed., 1988.
19. E. R. Fiala and D. H. Greene, "Data compression with finite windows," *Communications of the ACM* **32**, pp. 490–505, April 1989.