PART III

Signal Features for Modulation Recognition

Chapter 9

Introduction to Modulation Recognition

MODULATION recognition is a process whereby the modulation type of an unknown signal can be identified. There are a three fundamental steps in this process: parameter extraction, feature selection, and classification. Classification is a vast area of research with a wealth of techniques that can be applied to identify a signal type. Parameter extraction and feature selection, on the other hand, are comparatively under-represented. A critical question for these processes is: what parameters and features provide the greatest separation for a classification algorithm? What do these parameters and features mean in a physical sense? This chapter provides an overview of the modulation recognition process and current research, and identifies parameters that are analyzed in later chapters.

9.1 A Context for Modulation Recognition

Modern HF communications use a plethora of digital and analog modulation techniques so that even a conventional HF receiver that employs modulation recognition can be used for but a few modulations. Fortunately, the advent of software radio is reducing the scale of this problem. The goal of software radio is to replace hardware functions with equivalent software implementations running on a generic platform. By doing this, the receiver chains can be easily modified to cater for varied sets of modulation schemes. A typical software radio directly samples the incoming radio-frequency (RF) signal and performs downconversion in generic digital hardware, while demodulation is performed in software.

Many researchers of communication systems commonly use a Gaussian distribution to model noise and interference. In fact, as previously shown, the Gaussian assumption is not generally applicable for the HF environment. Furthermore, researchers often apply their methods to synthetic signals only and ignore the complexities of dealing with real HF signals, which include deep fades, impulsive noise, Doppler shifts, and multipath components that can be delayed by milliseconds. It is in this context that this thesis proposes signal identifying parameters for modulation recognition of real HF signals, that is, signals propagating by multiple ionospheric modes accompanied by co-channel signals and non-Gaussian noise. The ultimate aim is to develop algorithms that are capable of recognizing the modulation of real HF signals.

Modulation recognition is a process that determines the modulation of a signal with no prior knowledge of that signal. The process usually consists of three steps: parameter extraction, feature selection, and classification; though, sometimes parameter extraction and feature selection are combined and called feature extraction. Parameter extraction attempts to identify characteristics of the signal. Basic characteristics are frequency, phase, and amplitude, but often statistical measures (*e.g.* standard deviation, n^{th} order moments, n^{th}-order cumulants) are added to the list of parameters. The number of parameters in the list form an N-dimensional vector. The feature selection step is a transformation from an N-dimensional parameter space to an M-dimensional feature space, where $M \leq N$. Implicit in this transformation is the mapping of the N-dimensional vector into M orthogonal basis vectors. This may not be possible for a suboptimal parameter list, but the more orthogonal the basis vectors the easier a classifier

can separate the signal types. Classification identifies a modulation based on its associated features. It does this by grouping unique features that are, ideally, mutually orthogonal. Common methods for grouping of features include decision theoretic methods, artificial neural networks (ANNs), pattern recognition algorithms, and statistics. Classification is a vast field and will not be discussed further here. There are numerous references (Landgrebe 1997, Ma, Theiler & Perkins 2004, Kuo & Landgrebe 2004, Wang & Paliwal 2003, Distante, Leo, Siciliano & Persaud 2002, Choi & Lee 2003, Tang, Tao & Lam 2002, Choi & Kim 2000, Sun, Bebis & Miller 2004, Cao, Chua, Chong, Lee & Gu 2003, Fragoulis, Rousopoulos, Panagopoulos, Alexiou & Papaodysseus 2001, Leone, Distante, Ancona, Persaud, Stella & Siciliano 2005, Mitchell & Westerkamp 1999) for those interested in classification as it relates to modulation recognition.

The focus in this work is the feature extraction step. What are useful features of real HF signals? This question is critical for the development of robust modulation recognition techniques. As a step towards that goal, the remaining chapters investigate three features of some known modulations. However, a review of current research is warranted before addressing these features.

9.2 Literature Review

Hero III & Hadinejad-Mahram (1998) describe a method that uses spatial moments, of the m^{th} order, to classify m-ary phase shift-keying (PSK), frequency shift-keying (FSK), and quadrature amplitude modulation (QAM). Specifically, the authors use linear combinations of joint phase and magnitude moments (with orders greater than 100), and a *de-noising* procedure to extract the signal from a noise contaminated one. They show that the spatial moments of the signal alone are determined from an eigenvalue decomposition of a whitened moment matrix of the noise contaminated signal. The model breaks a noise-contaminated signal into in-phase (I) and quadrature-phase (Q) components before processing and plotting the results in the I-Q plane. The results clearly show that the method discriminates between 4-PSK and 4-QAM, and that the method is invariant to arbitrary phase rotations and scale variations of the received signal due to unknown carrier phase angle or signal amplitude. Hero and Hadinejad-Mahram assume additive white Gaussian noise (AWGN) and an ideal bandpass filter

in their study. A Gaussian process can represent real noise, but it is not always a valid assumption. Moreover, an ideal bandpass filter is not feasible in practical scenario. Furthermore, the authors do not present results that show the method discriminating between real signals (*i.e.* those corrupted by the physical transmission medium).

Nolan *et al* (2002) also consider higher-order statistics for a classification method for fourth-generation (4G) software radio. The authors assume an AWGN channel with Rayleigh fading, Doppler effects, and uniformly distributed phase noise. They further assume that all symbols are equally likely and that the expected signal space is approximately symmetrical. The researchers use 8^{th} order moments and normalized matrices to create graphs of the signals in terms of moments versus signal-to-noise ratio (SNR). These form the basis for a modified pattern-recognition/phase-classification approach. Modulation types can easily be distinguished at SNRs where the moments have zero slope. However, below some threshold SNR the moments become unstable and the modulation types are not easily determined. For bipolar phase-shift-keying (BPSK) and quadrature phase-shift-keying (QPSK) the threshold SNR is about 9 dB. They also try the approach on Gaussian minimum-shift-keying (GMSK), but performance degrades due to the inter-symbol interference (ISI) inherent in GMSK. Nevertheless, GMSK appears to be reliably detected at an SNR above approximately 25 dB. Nolan *et al* more accurately model the transmission medium than Hero and Hadinejad-Mahram, but still do not address the performance of their method when real signals are applied to it.

Ketterer *et al* (1999) apply statistical methods to tackle the modulation recognition problem. The Cross Margenau-Hill Distribution (CMHD) (Hippenstiel & De Oliveira 1990) and an auto-regressive (AR) covariance method is used to extract frequency, phase, and amplitude feature functions from an incoming digital signal. The extracted feature functions are then passed into a classification system that uses threshold detection logic to identify, offset orthogonal-keying (OOK), amplitude shift-keying (ASK), 2-FSK, 4-FSK, 2-PSK, 4-PSK, 8-PSK, 16-PSK, and 8-QAM. The feature functions for each of the modulation schemes possess particular characteristics. For example, the feature function for 16-PSK and 8-PSK has segments of constant amplitude. For m-ary FSK, the AR method accentuates symbol frequencies. The researchers show that in the presence of synthetic or real noise the algorithm correctly identifies the modulation at least

97% of the time. The authors use a *real-world* short-wave radio signal with symbols identical to the synthetic data to demonstrate the robustness of the algorithm. Critical to the method is an accurate determination of the signal center frequency. This can be a problem particularly for broad low-power signals. Moreover, the method has no obvious rationale for the choice of detection thresholds, and the feature functions for PSK, ASK, and QAM change depending on the symbols in the message. Though the feature functions show the presence of periods of constant phase levels for the various modulation types, the distribution of those levels in time varies with the symbols. Yet, this is one of few papers that consider the performance of the algorithms with real signals.

In a method for detecting multi-carrier modulations, Akmouche (1999) shows that a modified kurtosis function provides some separation of BPSK, QPSK, 16-/256-QAM and OFDM-32 on synthetic data with an SNR of 0 dB. The modified kurtosis appears useful for clearly distinguishing OFDM from single carrier modulations, but does not generally apply to cyclostationary signals. Furthermore, the modified kurtosis is sensitive to Gaussian noise and the number of samples on which the modified kurtosis is computed. Thus the apparent advantage of the modified kurtosis for real signals is in the discrimination of multi-carrier modulations[24] from single-carrier modulations. Since the modified kurtosis is sensitive to noise, albeit Gaussian noise, real signals may require noise removal prior to computation of the modified kurtosis. One method for noise removal uses wavelets (Mallat 1999, Ferguson 2001). The process, called *wavelet de-noising*, approximates the noise-free signal with as few wavelets as possible. This is done by choosing a wavelet family that maximizes the number of wavelet coefficients near zero.

Hsue & Soliman (1990) describe a simple zero-crossing method that separates synthetic continuous-wave (CW), *m*-ary PSK, and *m*-ary FSK signals. They also show that carrier frequency, carrier-to-noise ratio (CNR), and symbol rate can be estimated using zero-crossings. There are a two main drawbacks to this method: the assumption of a Gaussian noise model, and the result showing that the CNR must be greater than 15 dB to be effective on synthetic signals. In practice, the CNR for a real signal can be less than 15 dB. Nevertheless, the zero-crossing method and parameter extraction

[24] Multi-tone PSK is a common HF signal.

functions are attractive because of their simplicity, but appear unsuitable for real signals that are noisy. The classification method, which uses histograms, a decision tree and a pattern recognition algorithm, appears suitable for classifying real signals.

Others take a hybrid approach to modulation recognition. Nandi *et al*. (1997, 1998) and Wong & Nandi (2001) describe a feature extraction and classification method for discriminating ten different modulation types. Feature extraction is obtained through statistical methods, while classification is based on neural networks. From an incoming signal they extract features that are similar to features collected by many other researchers. These features include the maximum value of the power spectral density of the instantaneous amplitude, the standard deviation of the instantaneous phase, the standard deviation of the instantaneous amplitude, as well as the standard deviation of the instantaneous frequency. The authors also propose feature parameters based on the expectation of the second, third, and fourth order cumulants at zero lag. Once extracted from the signal of interest the feature set is fed to an artificial neural network with a hidden layer of perceptrons[25]. Without noise and with suitable training, the neural network is capable of discriminating 4-ASK, BPSK, QPSK, 2-FSK, 4-FSK, 16-QAM, V.29 (a PSK & QAM 600 baud modem modulation), V.32 (an FSK 2400 baud modem waveform), and 64-QAM with no error. The researchers also test the method with Monte-Carlo simulations to study the effect of noise on the classifier. With simulated SNRs as low as -5 dB the method correctly recognizes the modulation type 89% of the time. The authors, however, do not mention the noise model for the simulations and do not appear to consider the effects of real-noise, multipath, or Doppler shifts on the performance of the system. All of these strongly influence HF communications and so it is doubtful that the quoted performance figures are achievable for practical HF transmissions. Furthermore, the methods require oversampling of the signal by as much as eight times, which is often impractical in digital receivers where the RF signal is sampled directly. The large oversampling factor is not necessarily a problem for narrowband receivers as the overall data rate is low (the data rate is proportional to the bandwidth). However, the wider the bandwidth the greater the data rate. Increasing the the samping rate far above the Nyquist criterion will only increase the data rate even more.

[25] A perceptron is a simple feedforward neural network. It was invented in 1957 by Frank Rosenblatt at Cornell Aeronautical Laboratory.

Waller & Brushe (1999) have a different approach to modulation recognition. Instead of pattern recognition, parameter estimation, or neural networks, the authors construct a model the transmitter based on observations of the noisy received signal. They focus on frequency modulation (FM) and phase modulation (PM) and estimate the transmission system giving rise to the received signal. This is based on the assumption that another recognition system determines that the signal is some form of angle modulation (FM or PM). For the estimation, the researchers pass the signal through parallel FM and PM demodulators (see Figure 9.1). The output signal from each of the demodulators is re-modulated with parallel FM and PM modulators using an estimate of the carrier frequency of the received signal. The outputs of the four modulators are then correlated with the original received signal to determine the type of modulation. Phase modulation is identified if correlation peaks appear in either or both of the *straight-through* paths (*i.e.* FM demodulator-FM modulator, or PM demodulator-PM modulator paths) and a peak is observed in the FM demodulator-PM modulator *cross*-path while none appears in the other *cross*-path (PM demodulator-FM modulator). Frequency modulation is determined if there are correlation peaks in either or both of the *straight-through* paths and none in the *cross*-paths. The method seems somewhat troublesome though it is simple and innovative. Firstly, the method suggests some message dependency. In the case of an FM signal that is passed through a PM-demodulator and then an FM-modulator the claim is that the lack of a correlation peak is due to the particular message signal not correlating with an integral of itself. Secondly, the classification of an FM signal is based on no correlation peaks being present in the *cross*-paths. In fact, the simulation results do show a small correlation peak for this case. So a question arises as to how high a correlation peak must be before it is recognized as useful information. Clearly, the identification of correlation peaks relies on appropriate threshold detection and in this sense, the method is no different from many other researchers (Hero III & Hadinejad-Mahram 1998, Nolan *et al* 2002, Tan, Sakaguchi, Takada & Araki 2002).

Tan *et al* (2002) describe a direction-finding (DF) and null-steering (NS) system that employs modulation recognition based on threshold detection. In this system, signals from a receiver array are passed to an estimator of directions-of-arrival, followed by a signal combiner, and finally a modulation recognizer and demodulator. The modulation recognition algorithm identifies a single side-band (SSB) signal if the received

NOTE: This figure is included on page 134 of the print copy of the thesis held in the University of Adelaide Library.

Figure 9.1. A parallel FM/PM recognizer. Waller & Brushe (1999) pass the unknown signal (assumed to have a form of angle modulation) through parallel FM and PM demodulators. The output signal from each of the demodulators is re-modulated with parallel FM and PM modulators using an estimate of the carrier frequency of the received signal. Outputs of the modulators are correlated with the unknown input signal and the presence or absence of correlation peaks are used to identify the modulation type as FM or PM.

spectrum is not symmetric about a carrier and the difference between the highest and lowest peaks around the carrier is greater than a threshold value. It classifies a signal as FM if the maximum instantaneous amplitude is less than a threshold. As with many other modulation recognition techniques, this method is only simulated and its effectiveness with real signals is unknown. The authors also assume channel noise can be modeled as AWGN. Though this is a generally accepted noise model for simulation it does not adequately represent the effects of real HF noise. This is especially true for present-day complex digital modulations and for HF transmissions (Sevgi & Ponsford 1999, Goris 1998).

In a related field, Benedetto et al (2002) present a method for extracting information from generic sequences that is general enough to apply to modulation schemes. The authors describe a way to measure relative entropy between two information streams $\mathbb{A}$ and $\mathbb{B}$. Using a data compression algorithm, they compress a long data sequence, $\bar{A}$

from $\mathbb{A}$ and subtract the length of the compressed sequence from the length of the compressed sequence $\overline{A+b}$, where $\overline{A+b}$ is the concatenation of $\overline{A}$ and a small sequence $\overline{b}$ from $\mathbb{B}$. This is defined as the *entropy* of $\mathbb{A}$ (designated by Δ_{Ab}). In a similar manner they compute the *entropy* of $\mathbb{B}$ as Δ_{Bb}. Relative entropy (discussed in more detail in the next chapter) between the two information sequences is then

$$S_{AB} = \frac{\Delta_{Ab} - \Delta_{Bb}}{|b|}. \tag{9.1}$$

Benedetto *et al*. apply the algorithm to language recognition, authorship attribution, and classification of sequences. In each application the measure of relative entropy is successful in identifying a language, author, or language classification. For example, the method is able to successfully distinguish between Dutch, Danish, English, French, Finnish, German, Italian, Portuguese, Spanish, and Swedish. It was also able to correctly identify the author of a text 93.3% of the time. And for language classification, the method is able to correctly classify the Romance, Celtic, Germanic, Slavic, and Baltic languages. This technique seems suitable for the identification and classification of signal modulations. It is conceivable that an entropy measure could be calculated for a signal with unknown modulation and then compared against a database of entropy measures to identify the modulation type. What remains is to determine how well the method copes with HF signals. This is a topic of the next chapter.

Aisbett (1986) discusses a parameter extraction function for measurement of signal-to-noise ratio (SNR), discrimination of constant envelope and varying envelope signals, as well as the detection of the number of states of multi-level amplitude modulation. The function is the product of the expectation of signal X and the expectation of signal Y less their covariance. It appears able to separate synthetic CW, AM, and FM signals to some extent, but not convincingly. Nevertheless, the method for determining SNR is interesting and is addressed in detail later.

It is clear that current research on modulation recognition concentrates on trial-and-error and statistical methods, threshold detection logic, pattern recognition techniques, or artificial neural networks. Furthermore, rationalization of thresholds and feature

functions in current research is weak, and common assumptions about the HF channel are inappropriate (Sevgi & Ponsford 1999, Goris 1998). Most methods in the literature perform two basic functions: feature extraction and signal classification; and all suffer a common problem. The problem is that of practical validity. Many researchers claim success of their methods based on assumptions of additive-white-gaussian-noise, propagation characteristics, simulated data, and somewhat arbitrary thresholds.

Validation raises many questions, and one of the intentions of this thesis is to answer some of them in whole or in part. What is the best way to determine the modulation type? Are the chosen correlation thresholds valid for HF communications? Are brute-force methods better than statistical methods? If a recognition technique fails in the field, why did it fail, and how can it be improved to operate successfully? Which recognition method is the most robust and able to handle real data? Is there a benchmark signal or benchmark noise distribution that could be used to verify modulation recognition methods for the HF band? This last question highlights the importance of the discussion in Part II. As we shall see, it appears that a good benchmark is the modified Bi-Kappa distribution.

9.3 Summary

This chapter provides an overview of current research in the field of modulation recognition. The key question to be addressed is: what orthogonal parameters and feature functions can be extracted from a real HF signal to ease the classification of the signal. At present, most recognition techniques depend on threshold detection algorithms and statistical measures, of which many have little physical justification. Indeed, much research does not apply proposed methods to real signals, yet success is claimed based on simulations alone. It is not uncommon for previous research to make inappropriate assumptions about the transmission medium for the sake of computational convenience or as a result of inadequate knowledge of real HF propagation conditions.

Nevertheless, there are some that provide potentially useful signal features for the automatic recognition of real signals. The remaining chapters of this book address three

parameters: namely entropic distance, coherence, and signal-to-noise ratio (SNR). Entropic distance is based on Benedetto *et al*'s (2002) work, and SNR is based on Aisbett's (1986) research. The coherence function is a spectral analysis tool that, having no representation as a signal feature in the surveyed literature, is a plausible feature given that spectral analysis (*e.g.* Fourier transform) is common in signal processing.

The next chapter describes these features and methods for applying them to HF signals.

Chapter 10

Signal Features for Modulation Recognition

THERE are many parameters under investigation for the purposes of modulation recognition (see Chapter 9). They include characteristics such as power-spectral density (PSD), signal-to-noise ratio (SNR), bandwidth, instantaneous frequency and amplitude, and statistical measures (including high-order moments and cumulants).

Research described in the literature suffers problems related to assumptions about the transmission medium and methods of validation. In many cases an assumption of Gaussian noise is made, which is not always valid—especially at low-noise sites. Few researchers apply modulation recognition algorithms to *real* signals, where *real* implies signals propagating by multiple modes with co-channel interference and non-Gaussian noise.

This chapter presents three signal features: coherence, entropic distance, and signal-to-noise ratio (SNR). Coherence is analogous to correlation in the frequency domain, entropic distance is a distance metric based upon Shannon's information entropy, and the SNR measure is computed with Aisbett's *hash* function. In a subsequent chapter these features are extracted from real and synthetic signals.

10.1 Introduction to Methods

The literature survey of the previous chapter reveals that numerous parameters are under investigation for a variety of modulations. A consensus on the best method for identifying a modulation appears distant. However, some parameters are interesting and worth further investigation. The three features of interest are: coherence, entropic distance, and signal-to-noise ratio (SNR). Coherence is a cross-spectral analysis tool that measures the similarity of two signals in the frequency domain. Based on concepts of information entropy, the entropic distance measures the difference in entropy between signals. Signal-to-noise ratio can be computed in many ways. Aisbett's (1986) *hash* function is re-introduced and used to provide estimates of the true power of a signal contaminated by noise. With a slight algebraic modification, the function can also be used to estimate SNR. It is the intention of this part of the thesis to apply these parameters to a received signal (real or simulated), which is described in the following general way.

Common to all the feature extraction methods hereinafter, is the supposition that X is either a synthetic or real reference signal and that Y is a noisy, attenuated, and distorted version of X captured by a receiver. The model appears in Figure 10.1.

In general, the channel transfer function, $G(p_1, p_2, p_3, \ldots)$ can represent any mix of functions with any number of variables, p_i. In the following discussion of methods, G depends on parameters appropriate for the particular signal feature such as: time,

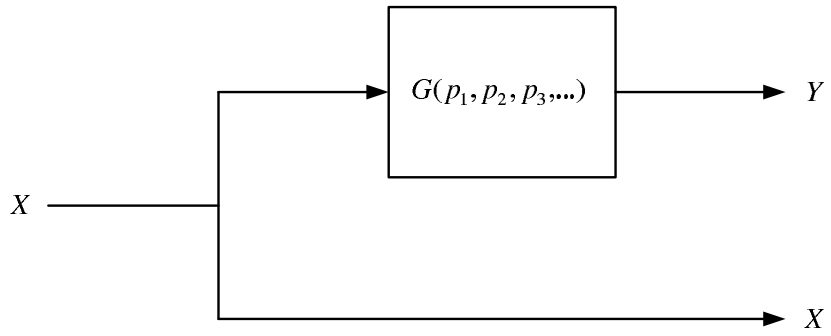

Figure 10.1. Common model for analysis of signal features. The common model for analysis of signal features provides two paths. One path includes the channel transfer function, $G(p_1, p_2, p_3, \ldots)$, that is dependent on numerous variables. The other path passes a reference signal uncorrupted to a receiver.

frequency, Hamming distance[26], and additive noise. The signals, X and Y, are drawn from the set of signals listed in Table 10.1 and Appendix B.

The table lists synthetic and real HF signals that form a sample set for procedures in subsequent sections. The convention in the remaining chapters, for referring to a signal in this table, is to name the signal followed by its category. For example, a real FSK Alternate Wideband signal would be referred to as "FSK Alt. Wide/R". The category is necessary to distinguish a real signal from a synthetic one. Figure 10.2 shows the power spectra of the real HF signals in the table. Chapter 11 describes the test setup for collection of the signals.

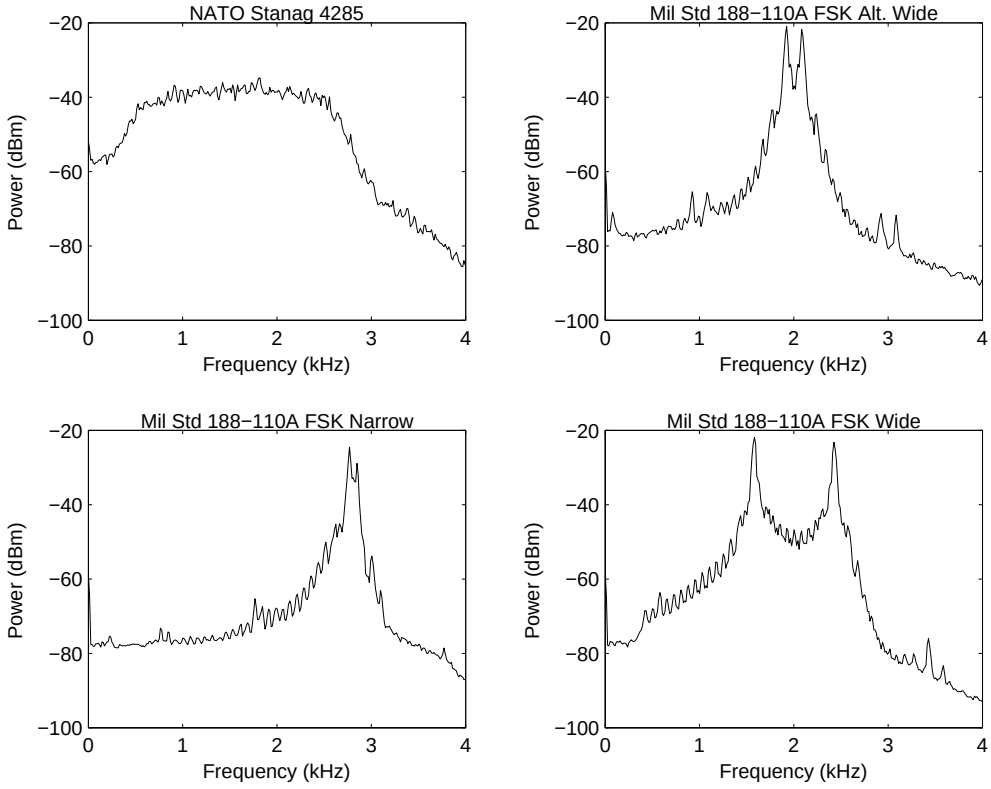

Figure 10.2. Power spectra of various real HF signals. Baseband power-spectra of various real HF signals measured with the research platform described in Section 6.3 (*upper-left*: Stanag 4285/R; *upper-right*: FSK Alt. Wide/R; *lower-right*: FSK Wide/R; *lower-left*: FSK Narrow/R;—see Table 10.1).

[26]Recall that the Hamming distance is the number of differing bits between two binary sequences.

Table 10.1. HF signals used for modulation recognition experiments. Various HF signal types used for modulation recognition experiments. A synthetic (*i.e.* simulated) signal is indicated by an "S" in the Type column. A real (*i.e.* recorded off air) signal is indicated by an "R" in the Type column. Signals I to IV are generated by an BAE Systems Adaptive Radio Modem ARM-9401 (BAE Systems 2002) configured to operate in a BER test mode. In this mode, the modem modulates the chosen waveform with a 511-bit pseudo-random code. Signals I to IV are military standard modem waveforms (U.S. Dept. of Defense 1991, NATO 1989). Signals V to VII are synthetic imitations. All signals carry a random message.

No.	Type	Modulation	Standard	Characteristics
I	R	8-PSK (scrambled)	NATO Stanag 4285	75 baud Long Interleaving Sub-Carrier 1800 Hz Channel 600-2400 Hz
II	R	FSK Alt. Wide	Mil-Std-188-110A Sec. 5.1.1	75 baud No Interleaving Mark: 1915 Hz Space: 2085 Hz Channel 1870-2130 Hz
III	R	FSK Wide	Mil-Std-188-110A Sec. 5.1.2	75 baud No Interleaving Mark: 1575 Hz Space: 2425 Hz Channel 1530-2470 Hz
IV	R	FSK Narrow	Mil-Std-188-110A Sec. 5.1.3	75 baud No Interleaving Mark: 2762.5 Hz Space: 2847.5 Hz Channel 2717-2892 Hz
V	S	2-FSK (noiseless)	-	1200 bit/s Sub-Carrier 2000 Hz Mark: 1915 Hz Space: 2085 Hz
VI	S	2-PSK (noiseless)	-	1200 bit/s Sub-Carrier 2000 Hz
VII	S	8-PSK (noiseless, scrambled)	Stanag 4285	1200 bit/s Sub-Carrier 1800 Hz Channel 600-3000 Hz

10.2 Coherence as a Signal Feature

The coherence function is a ratio of power spectral densities and provides a measure of the similarity of two signals at specific frequencies. It is analogous to correlation in the frequency domain. Consider that the power-spectral density of a signal, $x(t)$, is defined as

$$P_{xx}(f) = \int_{-\infty}^{\infty} \rho_{xx}(\tau) e^{-j2\pi f\tau} d\tau \tag{10.1}$$

where f is frequency, τ is time delay, and $\rho_{xx}(\tau)$ is the auto-correlation function of $x(t)$. Similarly, the cross-spectral density of two signals, $x(t)$ and $y(t)$, is

$$P_{xy}(f) = \int_{-\infty}^{\infty} \rho_{xy}(\tau) e^{-j2\pi f\tau} d\tau \tag{10.2}$$

where $\rho_{xy}(\tau)$ is the cross-correlation of signals $x(t)$ and $y(t)$. The coherence function is then the quotient of the cross-power spectral density and the product of the respective auto-power spectral densities;

$$\gamma^2(f) = \left| \frac{P_{xy}(f)}{\sqrt{P_{xx}(f) P_{yy}(f)}} \right|^2 . \tag{10.3}$$

Sometimes this definition is referred to as the magnitude squared coherence (Carter 1993). For each frequency, the coherence function indicates the similarity between $x(t)$ and $y(t)$. If the value of the function at a particular frequency is close to unity, it indicates that $x(t)$ and $y(t)$ 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 that frequency. Furthermore, if the coherence is unity it necessarily means that $x(t)$ and $y(t)$ are correlated.

Carter's (1993) excellent treatise on coherence shows that the coherence between $x(t)$ and $y(t)$ is also related to the signal-to-noise ratio (SNR) of $y(t)$ assuming that $x(t)$ is the transmitted signal and $y(t)$ is the noisy received signal. For this interpretation

$$\gamma^2(f) = \frac{\mathrm{SNR}_y(f)}{1 + \mathrm{SNR}_y(f)}, \tag{10.4}$$

where

$$\mathrm{SNR}_y(f) = \frac{P_{xx}(f)\,|H(f)|^2}{P_{nn}(f)}, \tag{10.5}$$

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 $x(t)$ and $y(t)$ but also implies an infinite SNR since in the limit as $\mathrm{SNR}_y(f) \rightarrow \infty$ the coherence tends to 1. In the context of Figure 10.1, the reference signal $X = x(t)$ and the received signal $Y = y(t)$.

Despite the simplistic appearance of Eq. (10.3) and Eq. (10.4) the calculation and measurement of coherence can be relatively simple or quite difficult (see worked examples in Appendix A) . For example Proakis (1989) 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 (1993) 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), for example Welch's periodgram method (Welch 1967), and *lag reshaping*. For each of these techniques the signals are broken into windowed overlapping segments. Each overlapping section is detrended and smoothed by an N-point window (*e.g.* Hanning, Hamming) before an M-point Fast Fourier Transform (FFT) is applied, where $M > N$ to ensure a resolveable coherence estimate. 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.

In either definition of coherence, the numerator and denominators are real and one would be inclined to accept that phase does not affect coherence. Appendix A provides examples where this is the case. As shall be seen, however, the coherence function is sensitive to the message in some digital signals; a result that agrees with Carter. That is, visual similarities between the spectra of two signals (real or synthetic) do not

necessarily imply high coherence. The prospect of the coherence as a signal feature for modulation recognition is not good. Nevertheless, it is worth investigating.

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, if at all?

Coherence-Median Difference

Before answering these questions, it is necessary to define a new parameter specifically to aid analysis of coherence of FSK signals. The coherence-median difference (CMD) is the difference between the mean of the coherences at the symbol frequencies less the median coherence over a bandwidth of interest. The CMD is described by

$$\text{CMD} = \frac{1}{M}\sum_{i=1}^{M}\gamma^2(f_i) - \text{median}(\gamma^2(f) \quad \forall f \in \text{BW}) \tag{10.6}$$

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. The median term represents the "noise-floor" of the coherence function. For a 2-FSK signal with two symbol frequencies, Eq. (10.6) reduces to

$$\text{CMD} = \frac{\gamma^2(f_\text{m}) + \gamma^2(f_\text{s})}{2} - \text{median}(\gamma^2(f) \quad \forall f \in \text{BW}) \tag{10.7}$$

where f_m and f_s are the mark and space frequencies respectively.

The CMD varies between -1 and +1. A positive CMD indicates that the coherence at the symbol frequencies is above the coherence at all other frequencies in the bandwidth of interest. A 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. Another way of describing the CMD, is the difference between the height of the symbol frequencies and the "noise floor" of the coherence function (see Figure 10.3).

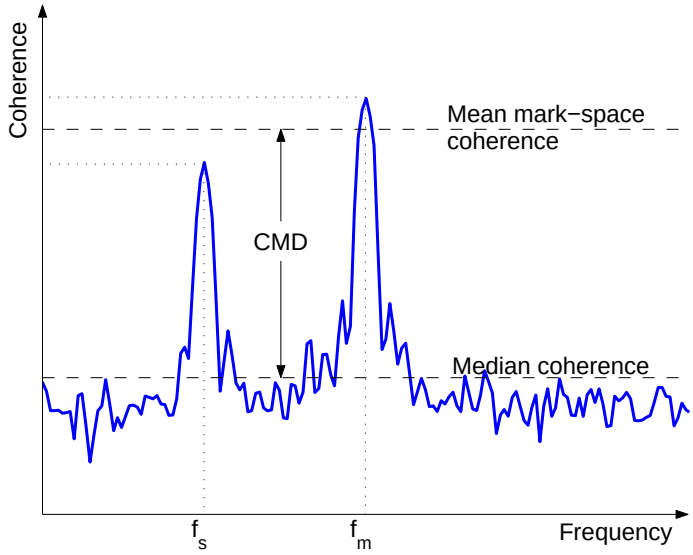

Figure 10.3. Pictoral representation of the CMD for m-ary FSK. The Coherence-Median-Difference (CMD) is defined for m-ary FSK signals and is described generally by Eq. (10.6). For 2-FSK (as shown), the CMD is the distance from the median coherence across a bandwidth to the mean coherence of the mark and space frequencies (f_{m} and f_{s} respectively). The CMD measures the dominance of the coherence of the symbol frequencies in the bandwidth of interest. The median coherence measures the "noise-floor" of the coherence function.

Later it will be seen that the CMD is useful for determining coherence thresholds for the identification of FSK signals and that it can provide a measure of certainty that the peak coherence occurs at the mark or space frequencies.

Experiments with Coherence

Four experiments will attempt to answer the questions previously posed. The first addresses the importance of choosing an appropriate number of segments and overlap for the calculation of coherence. The second study illustrates the relationship between coherence and SNR using Eq. (10.4). The third investigates the relationship between coherence and Hamming distance[27] for 2-FSK signals and 2-PSK signals. The last experiment verifies the sensitivity of coherence to misalignments in time through comparisons between simulated and real signals.

[27] Hamming distance is the number of bits different between two binary sequences.

Coherence versus Number of Segments and Overlap

Two arbitrary tones, X and Y, are simulated to understand the relationship between coherence and the number of segments and overlap. Both signals are without channel effects such that the transfer function $G(p_1, p_2, p_3, \ldots)$ consists of only one parameter, X. The output of the transfer function is a tone, Y, having a frequency near that of X such that $\omega_y = m\omega_x$ where ω_y is the angular frequency of Y, ω_x is the angular frequency of X, and m is a constant from the real number line.

It can be shown (see Appendix A) that given $m = \frac{\omega_y}{\omega_x}$ and

$$X(t) = A\cos(\omega_x t + \theta), \text{ and} \tag{10.8}$$

$$Y(t) = B\cos(\omega_y t + \beta), \tag{10.9}$$

the coherence of $X(t)$ and $Y(t)$ is

$$\gamma^2(\omega) = \frac{R(m,n,\theta)}{\sqrt{1 + 2\cos(2\beta)\,\mathrm{sinc}(2\pi mn) + \mathrm{sinc}^2(2\pi mn)}}, \tag{10.10}$$

where n is an arbitrary integer and where

$$\begin{aligned} R(m,n,\theta) = {} & \mathrm{sinc}^2(\pi[m+1]n) + \mathrm{sinc}^2(\pi[m-1]n) \\ & + 2\,\mathrm{sinc}(\pi[m+1]n)\,\mathrm{sinc}(\pi[m-1]n)\cos(2\theta). \end{aligned} \tag{10.11}$$

With an assumption that $\theta = 0$ and $\beta = 0$, Eq. (10.10) collapses to

$$\gamma^2(\omega) = \frac{\left(\mathrm{sinc}(\pi[m+1]n) + \mathrm{sinc}(\pi[m-1]n)\right)^2}{1 + \mathrm{sinc}(2\pi mn)}, \tag{10.12}$$

which does not depend on frequency, but only on the ratio of the frequencies of $X(t)$ and $Y(t)$. This is an example of a separable sinusoidal process described by Nuttall (1958). For the special case of $m = \pm 1$ (*i.e.* $\omega_y = \pm\omega_x$), the coherence function is one (the expected result) irrespective of phase. The coherence function returns zero for $m \in \{0, \pm 2, \pm 3, \ldots\}$. For all other m, the coherence takes on a value between zero and one. Figure 10.4 illustrates this.

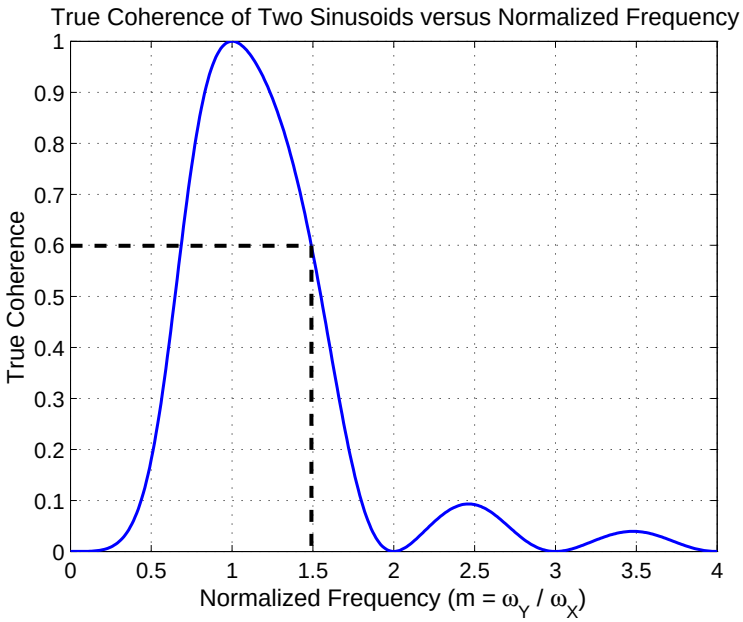

Figure 10.4. Theoretical coherence of two arbitrary sinusoids. The true theoretical coherence of two sinusoids of arbitrary frequency is related to a sinc function of normalized frequency, $m = \frac{\omega_y}{\omega_x}$. Note the null coherence at multiples of ω_x (*i.e.* integer values of m). Also note the local maxima at non-integer values of m. Clearly the greater the frequency difference (*i.e.* $\omega_y - \omega_x$), the lower the coherence. In this example $n = 1$. A mirror image of this plot exists for $m < 0$.

Recall that the coherence of two processes is estimated with the *Weighted Overlapped Segment Averaging* (WOSA) method. This method breaks each sequence of samples into overlapping segments. These segments are averaged in the calculation of the power-spectra prior to the calculation of coherence. The WOSA method can be summarized as follows:

i) break $X(t)$ into overlapping R segments of length δ samples;

ii) break $Y(t)$ into overlapping R segments of length δ samples;

iii) apply a suitable window to each segment of $X(t)$ and $Y(t)$;

iv) compute the auto-power spectrum of $X(t)$ and $Y(t)$ using
$P_{uu}(\omega) = \sum_{r=1}^{R} U_r(\omega)U_r^*(\omega)$ where r is the segment number and $U_r(\omega)$ is the Fourier transform of $X(t)$ or $Y(t)$;

v) compute the cross-power spectrum of $X(t)$ and $Y(t)$ using
$P_{xy}(\omega) = \sum_{r=1}^{R} X_r(\omega)Y_r^*(\omega)$ where r is the segment number and $X_r(\omega)$ is the

Fourier transform of the r^{th} segment of $X(t)$ and $Y_r(\omega)$ is the Fourier transform of the r^{th} segment of $Y(t)$;

vi) calculate the coherence according to Eq. (10.3).

The number of segments and segment overlap in the WOSA method are related by

$$R = \frac{K - \delta}{J - \delta}, \tag{10.13}$$

where R is the number of segments, K is the total number of samples, J is the number of samples in each segment, and δ is the number of samples of overlap for each segment. Appendix A describes this fundamental relationship in more detail.

The experiment progresses by varying the number of segments and segment overlap in the coherence calculation to observe the effect on coherence. Multiple points on the curve in Figure 10.4 are chosen. The coherence is estimated for the these cases and the error between the true coherence and the average of the estimated coherence is plotted. The variance of the error is also analyzed.

For this experiment $\omega_x = 1$ rad/s and $\omega_y = m$ rad/s. The sampling rate is chosen to be 7958 Hz so that 50,000 samples are contained within the period of ω_x. A large number of samples are required so that a statistically significant number of samples are present in each overlapped segment. Estimation of the coherence is repeated for various normalized frequencies, number of segments, and overlap.

Coherence versus SNR

At an arbitrary frequency, f_a, coherence versus SNR (assuming Gaussian noise) appears as in Figure 10.5. This *S*-shaped curve represents the coherence as a function of SNR (hereafter referred to as the *S-curve*) where the noise is confined to the bandwidth of the tone at f_a.

In reality, SNR can never be defined over an infinitesimal bandwidth, as suggested by Eq. (10.4). So, if coherence is calculated using Eq. (10.4) over an arbitrary bandwidth around f_a what happens to the *S-curve* of Figure 10.5? One would expect it to shift right or left depending on whether the frequency of interest is in-band, out-of-band,

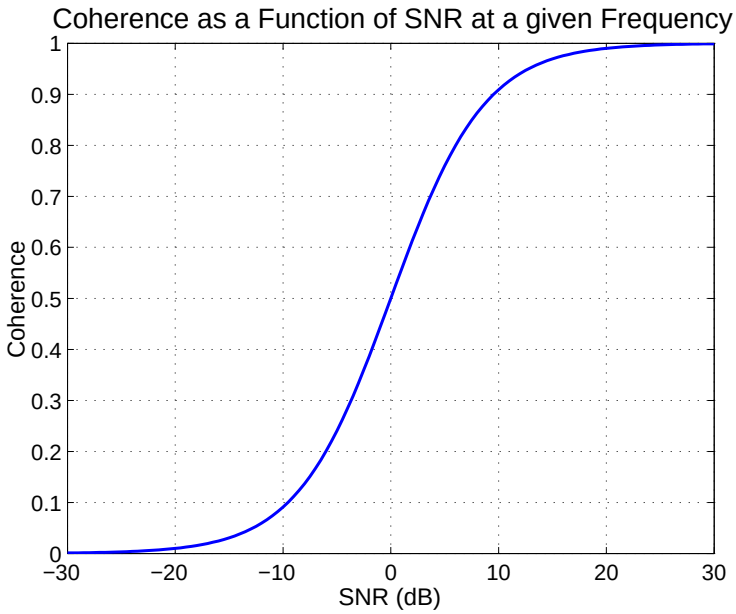

Figure 10.5. Coherence versus SNR at a single frequency. Coherence versus SNR at an arbitrary frequency where the noise is confined to the bandwidth of the tone at that frequency. This is simply a graphical representation of Eq. (10.4).

or in the transition bands. This behaviour is simply due to a different SNR at each frequency. For example, the tone at f_a has the vast majority of its energy at f_a and little elsewhere. Therefore, the SNR at f_a will be much higher than the SNR at $f_a \pm \Delta f$ since the magnitude of the tone at $f_a \pm \Delta f$ will be near zero and the noise will dominate.

This experiment applies Eq. (10.4) to a bandlimited sinc function centred at 1 kHz, to which is added bandlimited Gaussian noise. In the frequency domain this signal appears to have a flat response across the defined bandwidth (in this case 600 Hz). Using the logic above, one would expect that the *S-curve* remains stationary within the bandwidth and shifts outside the bandwidth. Results of this experiment will be discussed in a later chapter.

Coherence versus Hamming Distance

Two arbitrary 2-FSK/S signals are produced to study the effects of the message on the coherence between a transmitted signal and its received counterpart. The mark frequency (f_m) for each signal is 3 kHz and the space frequency (f_s) is 1 kHz. During spectral processing f_m and f_s may fall across frequency bins so $\gamma^2(f_m)$ and $\gamma^2(f_s)$ are

estimated using linear interpolation between the coherence computed at adjacent frequency bins. To simulate bit errors in a received signal Y, the Hamming distance[28] between Y and the original transmitted signal[29], X, is varied and the coherence computed using methods suggested by Carter (1993). Here, X is generated with a uniform distribution of marks (1's) and spaces (0's). For each Hamming distance $Y = G(X, d)$ where $G(X, d)$ replaces $G(p_1, p_2, p_3, \ldots)$ in Figure 10.1 and where $G(X, d)$ returns X with a random selection of d bits (the Hamming distance) inverted. For example, when d is zero $G(X, d)$ returns X, and when d is ten, $G(X, d)$ returns X with 10 randomly chosen bits inverted.

To see the relationship between coherence and Hamming distance, the coherence between the two arbitrary and noise-free 2-FSK/S signals is plotted against Hamming distance. Specifically, the mean of the coherence of the mark and space frequencies, the peak coherence across the bandwidth of interest, and the CMD are all plotted against the Hamming distance. It is expected that as the Hamming distance increases, the coherence will decrease.

A last part of this experiment revisits the *S-curve*, through Eq. (10.4) and Eq. (10.5) and the two arbitrary 2-FSK/S signals (this time with Gaussian noise), to study the effect that Hamming distance has on the SNR. If the coherence decreases with increasing Hamming distance, the *S-curve* is expected to vary between zero and $+1 - \epsilon$, where ϵ corresponds to the decrease in coherence from unity.

Time Sensitivity of Coherence

Carter (1993) suggests that coherence is sensitive to skew (or time misalignment) between signals. This experiment studies the effect that skew has on coherence. Three steps are required to study the effect. The first is to observe the coherence between a synthetic signal and a real signal. The second, compares the coherence between two real signals acquired by the same receiver at different times. For this case the message of each signal has the same statistics (*i.e.* a 511-bit pseudo-random sequence) but as a result of the delay the portions of the message captured by the receiver are different.

[28] The Hamming distance is the number of bits different between two binary sequences.

[29] An alternative interpretation is that Y is an arbitrary 2-FSK/S signal being compared against an arbitrary 2-FSK/S reference signal X, with varying levels of correlation.

The third step compares the coherence between two identical real signals acquired by the same receiver platform at the same time. In this instance, the messages are the same but the two signals are acquired at different antennas[30].

To begin, the coherence is calculated for a 2-FSK/S signal is compared with an FSK Alt. Wide/R signal (see Table 10.1). The synthetic signal is constructed to have the same symbol frequencies and frequency deviation as the real signal, but the information bits are varied in a uniformly random way for each trial. That is, in each trial the dataset for the FSK Alt. Wide/R signal remains the same, but a new 2-FSK/S signal is generated (*i.e.* a different random message for each trial). This test is repeated for a Stanag 4285/S signal and a Stanag 4285/R signal. In each case, though the spectra of the two signals may be similar, it is expected that the coherence between the two will be small because the symbol timing of the signals differ.

The second part of the experiment observes the coherence of two groundwave FSK Alt. Wide/R signals acquired on the same antenna of the receiver platform but at different times and with different messages. It is then repeated for two Stanag 4285/R signals. If coherence is insensitive to the skew, one would expect to see a high coherence for these cases because in each case the two spectra are nearly identical.

The last part of the experiment studies the coherence of two groundwave real FSK Alt. Wide signals acquired at the same time but via different antennas of the receiver platform. It is then repeated for two Stanag 4285/R signals. In each case the signals carry the same message and their spectra should be nearly the same except for frequency-selective fading and variances between antenna characteristics. It is, therefore, reasonable to expect their coherence to be near unity.

10.3 Entropy as a Signal Feature

Efficient encoding of a message so that no information is lost in transmission depends on an understanding of entropy. Indeed, Shannon (1948) shows that the limit of efficient encoding is the entropy of the message sequence. His definition of entropy is a measure of the uncertainty of the occurrence of an event or the information that event

[30] Recall that the digital receivers are connected to an antenna array—see Chapter 11.

imparts when it occurs. Specifically, entropy is defined as

$$H(\mathbf{X}) = \sum_{i=1}^{N} P(x_i) I(x_i) \tag{10.14}$$

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

$$I(x) = -\log_N P(x). \tag{10.15}$$

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

$$H(\mathbf{X}) = -P(0)\log_2 P(0) - [1 - P(0)]\log_2 [1 - P(0)] \tag{10.16}$$

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

Shannon shows that entropy computed over the length of a message, provided successive symbols are statistically independent, Eq. (10.14) can be estimated by

$$\hat{H}(\mathbf{X}) = \frac{1}{L} \sum_{i=1}^{L} \hat{P}(x_i) \hat{I}(x_i), \tag{10.17}$$

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

$$\lim_{L \to \infty} \hat{H}(\mathbf{X}) = H(\mathbf{X}). \tag{10.18}$$

Benedetto *et al* (2002) base their definition of entropy on Shannon's formulation, but present a different method of computing it. They create a measure of the similarity or dissimilarity of two information sources, $\mathbb{A}$ and $\mathbb{B}$, as representative of the true entropy (see Figure 10.6). In particular, they ...

> "... define in a very general way a concept of remoteness (or similarity) between pairs of sequences based on their relative informational content. We devise, without loss of generality with respect to the nature of the strings of characters, a method to measure this *distance* based on data-compression techniques. The specific question we address is whether this informational *distance* between pairs of sequences is representative of the real semantic difference between the sequences." — (Benedetto *et al* 2002)

The method compresses a long sequence A from $\mathbb{A}$ and subtracts this compressed length from the length of the compressed sequence $A + b$, where $A + b$ is the concatenation of A and a small sequence b from $\mathbb{B}$. This is the entropy of $\mathbb{A}$ (designated by Δ_{Ab}). In a similar manner they compute the entropy of $\mathbb{B}$ as Δ_{Ba} and the self-entropies Δ_{Aa} and Δ_{Bb}. They then define the relative entropy between A and B as

$$S_{AB} = \frac{\Delta_{Ab} - \Delta_{Bb}}{|b|}, \tag{10.19}$$

where $|b|$ is the length of b, and the relative entropy between B and A as

$$S_{BA} = \frac{\Delta_{Ba} - \Delta_{Aa}}{|a|}, \tag{10.20}$$

where $|a|$ is the length of a. The total entropy, or entropic distance, between the two information sources is then by definition

$$S_T \equiv \frac{\Delta_{Ab} - \Delta_{Bb}}{\Delta_{Bb}} + \frac{\Delta_{Ba} - \Delta_{Aa}}{\Delta_{Aa}} = \frac{|b|\, S_{AB}}{\Delta_{Bb}} + \frac{|a|\, S_{BA}}{\Delta_{Aa}}. \tag{10.21}$$

As an example, consider two segments of 200 characters each from information sources $\mathbb{C}$ and $\mathbb{D}$. One sequence, C, consists of a hundred bytes[31] of \$00 followed by a hundred bytes of \$01. The second sequence, D, has alternating \$00 and \$01 bytes. A 12-bit Lempel-Ziv-Welch (LZW) algorithm (Welch 1984) compresses each sequence equally well with a compression ratio of 77.5% (*i.e.* the compressed length is 45 bytes). The reason that this occurs is that each sequence contains repeating patterns; the LZW algorithm is efficient at compressing repeating patterns. Efficiency decreases when

[31] Recall that a hexadecimal symbol is prefixed by \$. A hexadecimal zero is represented by \$00.

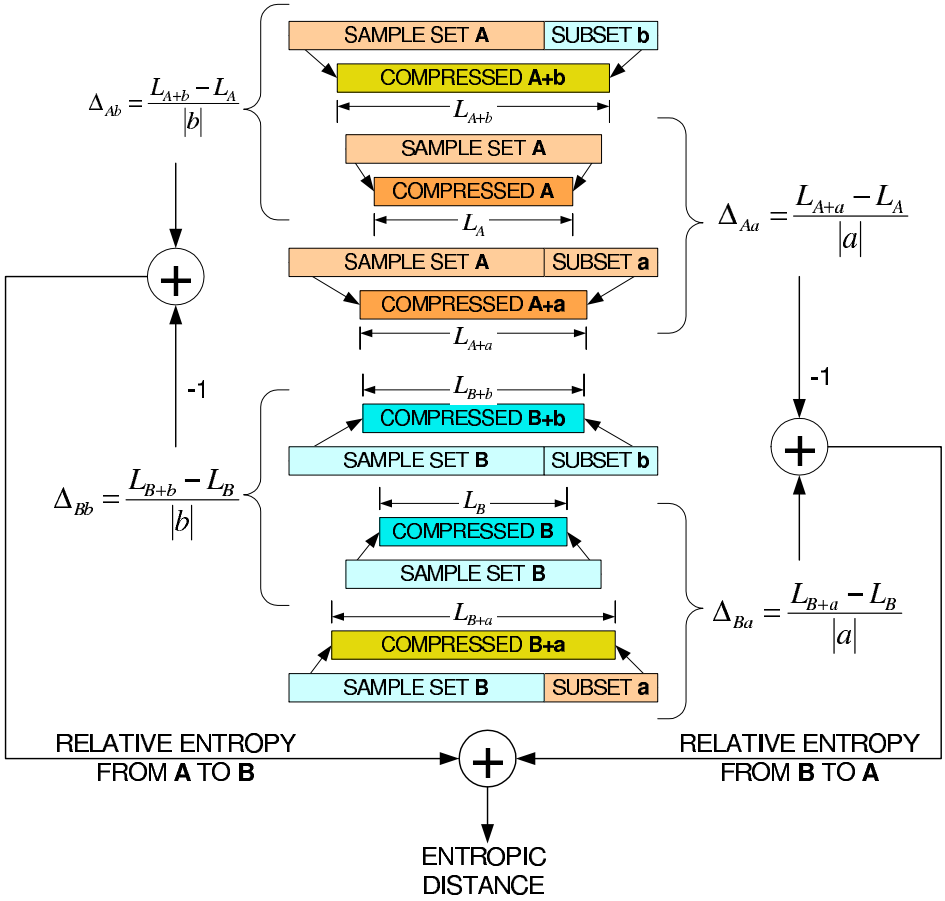

Figure 10.6. Benedetto's entropy calculation method. Benedetto's entropy calculation method involves compressing segments of two information sequences and measuring their lengths. The method compresses a long sequence A from $\mathbb{A}$ and subtracts this compressed length from the length of the compressed sequence $A + b$, where $A + b$ is the concatenation of A and a small sequence b from $\mathbb{B}$. This is the entropy of $\mathbb{A}$ (designated by Δ_{Ab}). In a similar manner the entropy of $\mathbb{B}$ is Δ_{Ba} and the self-entropies are Δ_{Aa} and Δ_{Bb}. In this diagram, Δ_{Ab} and Δ_{Bb} are normalized by the length of b; Δ_{Aa} and Δ_{Ba} are normalized by the length of a.

a small sequence of D (say 20 bytes) is appended to C. In this case the compression ratio drops to 71.5% (*i.e.* the compressed length is 57 bytes). Though a small change, the less efficient compression is a measure of the difficulty that the LZW algorithm has in forming codewords for the 20-byte sequence given the codewords formed on C. Performing a similar exercise for the sequence D, to which is appended a small 20-byte sample of itself, yields a compression ratio of 77% (*i.e.* the compressed length is 46 bytes). Applying Eq. (10.19) provides a relative entropy measure between $\mathbb{C}$ and $\mathbb{D}$ of $\frac{11}{20}$. The same process estimates the relative entropy from $\mathbb{D}$ to $\mathbb{C}$ at $\frac{3}{10}$. Application

of Eq. (10.21) shows that the entropic distance is 13. The whole process is analogous to a measure of the difficulty that an English-speaking person has in learning Chinese or the difficulty that a Chinese-speaking person has in learning English. Section 12.3 elaborates on reasons for determining entropy this way, instead of that suggested by Shannon (1948).

Benedetto *et al*. apply their methods to authorship identification, and language classification. But, imagine that a baseband HF signal, $\mathbb{Y}$, from Table 10.1, is quantized by a Q-level analog-to-digital converter (ADC). Assuming a uniform quantizer, the Q levels represent a dynamic range of $20\log_{10} Q$ decibels (relative to the smallest quantizable signal) and are also equivalent to an alphabet of Q symbols of $\log_2(Q)$ bits each. For example if Q is 256, the dynamic range is $\approx$ 48 dB and the alphabet consists of 8-bit symbols from hexadecimal values \$00 to \$FF. The quantization of $\mathbb{Y}$ corresponds to a certain arrangement of the Q-symbol alphabet. If now another signal, $\mathbb{X}$ drawn from a random selection of the Q-symbol alphabet, is chosen to be a reference then Benedetto's entropy function Eq. (10.21) can be used to measure the distance between the baseband signal and the reference. It is on this basis that the following experiments with entropy are presented.

Experiments with Entropy

For the measurement of entropy, the information source (or unknown received signal) consists of a digitized HF signal (subsequently downconverted to baseband in the digital domain); the reference signal is a continuously random selection of symbols from a uniformly distributed alphabet. The idea then, is to determine the entropic distance between each of the signals in Table 10.1 (see also Figure 10.2) and the uniformly distributed reference.

With this background, four experiments investigate the usefulness of entropic distance for modulation recognition. The first looks at the effects of compression algorithm and structure of the information sequence on the estimate of self-entropy. The second considers the effect of the quantizer resolution on relative entropy. The third addresses the issue of whether or not spectral domain data can be used with the entropy algorithm. The last experiment measures entropic distance between synthetic and real HF signals.

Effects of Compression Methods

Compression is necessary for the computation of Benedetto *et al*'s (2002) entropy. So how does Benedetto's entropy compare with Shannon's (1948) measure of entropy? And, what is the effect of the compression method on entropy? There are many compression algorithms, however, in this study only two are considered: 12- and 13-bit Lempel-Ziv-Welch (Welch 1984) and Zip 2.3 compression[32].

First, two information sources, $\mathbb{A}$ and $\mathbb{B}$, are contrived that have equal probabilities of a zero-bit. The probability of a zero-bit is varied and the self-entropy of the sequence $A + b$ is calculated with Benedetto's method (using 12-bit LZW compression). The self-entropy, Δ_{Ab}, is assigned the average of three trials for each variation of the probability. In each trial, the calculation of self-entropy uses a new A-sequence and B-sequence. The experiment is repeated three times and in each case the length of A and B is kept constant at 10,000 bits. However, the length of b is varied, starting with $|b| = 100$, then $|b| = 500$, and then finally $|b| = 1000$. The whole process is then repeated for cases where the length of b is held constant (either $|b| = 25$ or $|b| = 250$) and the length of A and B is varied such that $A, B \in \{10^2, 10^3, 10^4, 10^5\}$. The self-entropy, Δ_{Ab}, is compared against that determined by Shannon's method.

The second part of this experiment repeats the first but, whereas in the first part the probabilities are variable and the sequence lengths restricted to a finite set, in this next part the probability of a zero-bit is constrained and the length of A and B is allowed to vary. The length of b is also allowed to vary. Again information sources, $\mathbb{A}$ and $\mathbb{B}$, are used to generate sequences A and B. Two specific zero-bit probabilities are considered: 10% and 90%. The results will provide a surface that has three dimensions: sequence length, length of b, and self-entropy. The results are expected to show the degree to which the estimate of self-entropy (via Benedetto's method with 12-bit LZW compression) respresents the true self-entropy.

A last part of this experiment compares the efficiency of two compression algorithms, LZW and Zip 2.3, in relation to the calculation of the self-entropy of $\mathbb{A}$ via Benedetto's method. It also addresses the effect of information structure on entropy.

[32] Zip 2.3 is based on the search algorithm of Rabin and Karp—see Sedgewick (1988)—and the compression algorithm of Fiala & Greene (1989).

To compare compression efficiency, the LZW and Zip 2.3 algorithms are applied to sequences from $\mathbb{A}$. The probability of a zero-bit is varied and as well as the length of the sequence A. A surface-plot of the compressed sequence length as a function of probability and sequence length is expected to show that one of the compressions algorithms is more efficient than the other. The second part of the experiment is also repeated with Zip 2.3 compression instead of LZW compression.

Regarding information structure, three binary sequences are constructed and their entropies computed, again using Shannon's method and Benedetto's method for the two compression algorithms. One binary sequence consists of uniformly distributed ones and zeros. Another is skewed so that the first half of the sequence consists of binary zeros and the last half binary ones. A third sequence is the reverse of the second: binary ones followed by binary zeros. The entropy for each sequence is observed for changes in the lengths of the sequences. However, for this situation, the length of the appending sequence is fixed at 500 symbols.

Effect of Quantizer Resolution on Entropy

This second experiment with entropy determines the effect that a quantizer has on the estimate of relative entropy. The Q levels of the quantizer correspond to an alphabet of Q symbols with a theoretical dynamic range of $20\log_{10} Q$. For example, if $Q = 2^n$ and $n = 16$ bits then the dynamic range is 96 dB. In practice, the dynamic range of a quantizer is less than this due to quantization error, clock jitter, and thermal noise. Ignoring this for the present discussion is acceptable. Consequently, one expects that if the received signal level stays the same and if the size of the alphabet increases then the total entropy should also increase. Why? Chapter 12 answers this question. For now it is sufficient to explain the testing methodology.

Assume a received and unknown signal, $Y(t)$, and a reference signal, $X(t)$, corresponding to a random arrangement of a Q-symbol alphabet. In this test the effect of Q on the relative entropy—with respect to $X(t)$—of synthetic m-ary FSK, m-ary PSK, and Stanag 4285 signals is observed. Each synthetic signal is compared, via Eq. (10.19), to a sequence of randomly distributed symbols from the Q-symbol alphabet for various sequence lengths and various Q.

The Stanag 4285/S signal is used as a basis because of its broad spectral characteristics (*c.f.* PSK and FSK). This makes the signal appear to have some randomness, from the point of view of the compressor (Zip 2.3 in this case), and therefore it is similar to the reference signal, $X(t)$. Control of this comparison is provided by setting $Q = 2^n$ where $n \in \{8, 10, 12, 14, 16\}$ bits.

The experiment is repeated for 2-PSK/S and 2-FSK/S signals. However, for these trials $Q \in \{2^8, 2^{16}\}$ symbols and 12-bit LZW compression is used. Moreover, the length of the appended sequences (*i.e.* $|a|$ and $|b|$) is constant at 500 symbols. Shannon's entropy is also plotted as a reference point in each trial.

Entropic Distance for Real Signals

The next experiment consists of two parts. Both parts utilize Zip 2.3 compression and LZW compression with 12-bit or 13-bit codes.

In the first case, entropic distance between synthetic signals of Table 10.1 is observed with and without added Gaussian noise. For each trial, $\mathbb{Y}$ is one of the synthetic signals in Table 10.1 while the reference signal, $\mathbb{X}$, is a uniformly random arrangement of symbols from the Q-symbol alphabet. Entropic distance between each $\mathbb{Y}$ and $\mathbb{X}$ is computed for various Q and various segment lengths.

The second part of the experiment is similar, in that it investigates the usefulness of entropic distance to separate the real signals in Table 10.1. Data points for both parts of the experiment are averaged over ten trials and then interpolated with cubic splines. Note that in all trials the lengths of a and b are constant, at 500 symbols, while the lengths of A and B vary directly so that the abscissa (or segment length variable) represents the length of $A + a$, $A + b$, $B + a$, and $B + b$.

At this point, the explanation of the procedures for this experiment seem vague. The vagueness is deliberate. In this instance, a good explanation of the results is better served by keeping details of the procedure close to the discussion. Chapter 12 provides the necessary procedural detail in conjunction with the results.

Entropic Distance with Spectral Data

As described above, the calculation of entropy applies to time-domain data. Does it also apply to frequency-domain data? That is, can Benedetto's method be applied to signals defined in the frequency domain? If not, is there a frequency domain representation of Benedetto's method?

Let one information source, $\mathbb{A}$, be defined in the frequency domain with a bandwidth, BW_A, and centre frequency, f_A. And let the another source, $\mathbb{B}$, also be defined in the frequency domain as a bandwidth of, BW_B, and centre frequency, f_B. Considering the representation in Figure 10.7, one can surmise that Benedetto's entropy will vary with frequency.

If the sequences to be compressed are chosen from regions 1 and 2 the entropic distance will be low; the compression algorithm will compress the sequences equally well, because the samples in each of these regions are unorganized arrangements of

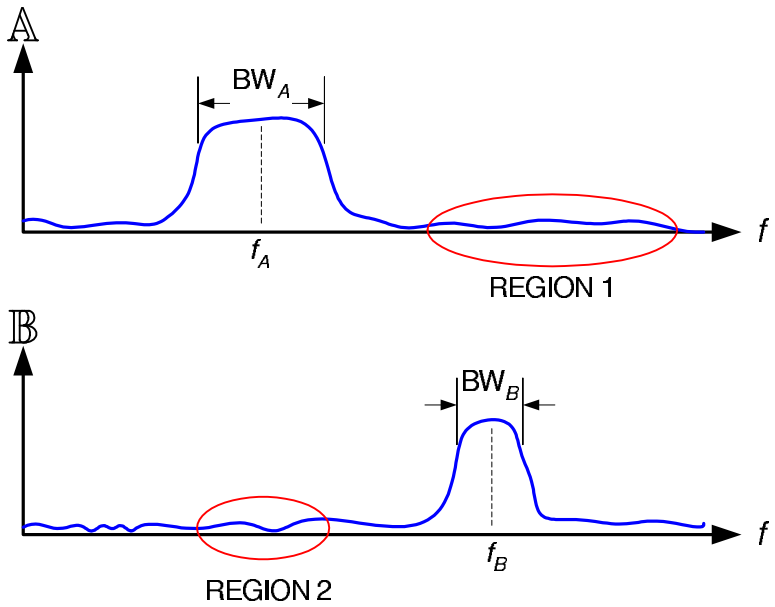

Figure 10.7. Arbitrary spectra for entropic distance calculations. Arbitrary spectra for entropic distance calculations pose a problem—entropy will vary with frequency. If a random segment is drawn from region 1 and a random appending sequence is drawn from region 2, the entropic distance will be low. It will be low, because the noisy areas of region 1 and 2 contain similar random arrangements of symbols from the Q-symbol alphabet. A similar low entropic distance should be achieved if segments are drawn from the passband regions (centred at f_A and f_B). A large entropic distance is expected when one segment is drawn from a passband region and another from a stopband region.

the Q-symbol alphabet . As a result the entropic distance nears zero. Similarly, if the sequences are chosen from the passband regions (centred at f_A and f_B) the entropic distance will also be near zero, because in this case the samples are organized and again the compression algorithm will compress the sequences equally well. The only situation of high entropic distance occurs if the sequences are chosen from either passband region and a stopband region, since the arrangements of the Q-symbol alphabet for each of the sequences are significantly different. That is, the sequence selected in the passband will be organized and of high magnitude compared to the unorganized and low magnitude sequence selected from the stopband. It is obvious that entropic distance cannot be used on spectral data and therefore no experiment is necessary to show this.

10.4 Signal-to-Noise Ratio as a Signal Feature

Modulation recognition algorithms often require an estimate of signal-to-noise ratio (SNR). Aisbett points out that

> "... noisy signals are more similar to each other, regardless of modulation type, than they are to strong signals of the same modulation type." — (Aisbett 1986)

What this means, in practical sense, is that a modulation recognition algorithm must account for the SNR in its decision making. In part, Aisbett's work is based on the analysis of narrowband signals with narrowband Gaussian noise provided by Whalen (1971) and later by McDonough & Whalen (1995).

Assume a narrowband signal with narrowband Gaussian noise. This noise has a zero mean and variance (or noise power), σ_n^2. The passband representation of this signal is

$$r(t) = A(t)\cos(\omega_c t + \theta) + n(t), \tag{10.22}$$

where $A(t)$ is the deterministic narrowband envelope function, $n(t)$ is the narrowband Gaussian noise, ω_c is the carrier frequency, θ is a random phase angle, and t is time.

Decomposing Eq. (10.22) into in-phase and quadrature-phase components yields

$$r(t) = i(t)\cos\omega_c t - q(t)\sin\omega_c t, \tag{10.23}$$

where

$$i(t) = A(t)\cos\theta + n_i(t), \text{ and} \tag{10.24}$$
$$q(t) = A(t)\sin\theta + n_q(t). \tag{10.25}$$

The functions $n_i(t)$ and $n_q(t)$ are in-phase and quadrature-phase components, respectively, of the narrowband Gaussian noise. The instantaneous amplitude, $z(t)$, of the envelope is simply

$$z(t) = \sqrt{i^2(t) + q^2(t)}, \tag{10.26}$$

and its instantaneous phase is

$$\phi(t) = \arctan\left(\frac{q(t)}{i(t)}\right). \tag{10.27}$$

Another representation of the instantaneous amplitude that is used by Aisbett (1986) is

$$z(t) = \sqrt{r^2(t) + \widehat{r}^2(t)}, \text{ and} \tag{10.28}$$
$$z(t) = B(t)\cos\phi(t) + n(t), \tag{10.29}$$

where $\widehat{r}(t)$ is the Hilbert transform of $r(t)$, and $B(t)$ is the amplitude of the envelope. This definition is equally valid here and shall be assumed forthwith. The instantaneous phase can also be represented in terms of the received signal and its Hilbert transform so that Eq. (10.27) becomes

$$\phi(t) = \arctan\left(\frac{\widehat{r}(t)}{r(t)}\right). \tag{10.30}$$

Whalen (1971) proceeds to show that the probability density function of $z(t)$ is a Gaussian density independent of θ and valid for all $\phi(t)$ whereby

$$p_z(z) = \frac{z(t)}{\sigma_n^2} e^{-\frac{z^2(t)+B^2(t)}{2\sigma_n^2}} I_0\left(B(t)\frac{z(t)}{\sigma_n^2}\right), \tag{10.31}$$

where $I_0(\cdot)$ is the Bessel function of the first kind. With this density function, the moments of $z(t)$ are shown to be

$$E\left\{z^n(t)\right\} = \left(2\sigma_n^2\right)^{\frac{n}{2}} \Gamma\left(\frac{n}{2}+1\right) {}_1F_1\left(-\frac{n}{2};1;-\frac{B^2(t)}{2\sigma_n^2}\right), \tag{10.32}$$

with $\Gamma(\cdot)$ as the gamma function, and where ${}_1F_1(a;b;c)$ is the confluent hypergeometric function. Note that $\frac{B^2(t)}{2\sigma_n^2}$ is the true SNR, where $\frac{B^2(t)}{2}$ is the average power of the envelope. The variance, σ_n^2, is the average power of the noise.

Having this in mind, the second moment of $z(t)$ is a measure of the average signal plus noise power and is

$$E\left\{z^2(t)\right\} = B^2(t) + 2\sigma_n^2, \tag{10.33}$$

and, for reasons soon to be clear, the fourth moment of $z(t)$ is

$$E\left\{z^4(t)\right\} = B^4(t) + 8\sigma_n^2 B^2(t) + 8\sigma_n^4. \tag{10.34}$$

Now Aisbett (1986) defines a function for computing signal power and for discriminating constant envelope signals from varying envelope signals. The *hash* function, so named here for convenience, is twice the product of the means of two signals less their covariance. The name of the function is derived from the way that Aisbett specifies the function; $\sharp(X,Y)$ is used in place of the more common $f(X,Y)$ representation of a function—the $\sharp$ symbol giving rise to the *hash* designation. Hereafter, Ψ shall be used in the place of the $\sharp$ operator.

Given two time domain signals, $X(t)$ and $Y(t)$, the *hash* function is

$$\Psi(X,Y) = \frac{2}{N^2}\sum_{m=1}^{N} X(mT)\sum_{m=1}^{N} Y(mT) - \frac{1}{N}\sum_{m=1}^{N} X(mT)Y(mT), \tag{10.35}$$

$$\Psi(X,Y) = 2E\left\{X\right\}E\left\{Y\right\} - E\left\{XY\right\}, \tag{10.36}$$

where N is the number of samples in each sequence, T is the sample period, and $E\left\{\cdot\right\}$ is the expectation operator. The *hash* function is an unbiased estimator over large numbers of observations provided the noise is stationary and the noise at different observations intervals is uncorrelated.

From Eq. (10.36), Aisbett shows that an estimate of signal power is simple to compute. To estimate signal power one computes

$$\Psi\left(z^2(t), z^2(t)\right) = 2E\left\{z^2(t)\right\} E\left\{z^2(t)\right\} - E\left\{z^2(t) \cdot z^2(t)\right\}. \tag{10.37}$$

Substituting for $E\left\{z^2(t)\right\}$ and $E\left\{z^2(t) \cdot z^2(t)\right\}$ from Eq. (10.33) and Eq. (10.34) confirms that $\Psi\left(z^2(t), z^2(t)\right)$ is proportional to the square of the signal power, $B^2(t)$.

Thus we see that $E\left\{z^2(t)\right\}$ is a measure of the signal plus noise power, and that $\Psi\left(z^2(t), z^2(t)\right) = B^4(t)$. It is therefore possible to estimate the SNR of a signal with the *hash* function such that

$$\mathrm{SNR_e} = \frac{\sqrt{\Psi\left(z^2(t), z^2(t)\right)}}{E\left\{z^2(t)\right\} - \sqrt{\Psi\left(z^2(t), z^2(t)\right)}}. \tag{10.38}$$

The numerator in this equation is an estimate of the peak envelope power (PEP). The average signal power is half the PEP. Another way of interpreting Eq. (10.38) is a ratio of the estimate of the PEP to the average noise power so that

$$\mathrm{SNR_e} = \frac{P_{\mathrm{PEP}}}{P_{\mathrm{PEP+n}} - P_{\mathrm{PEP}}}, \tag{10.39}$$

where P_{PEP} is the peak envelope power, and $P_{\mathrm{PEP+n}}$ is the PEP plus twice the average noise power. The denominator is equivalent to $2\sigma_{\mathrm{n}}^2$. Thus, knowing that the average signal power is half the PEP, Eq. (10.39) represents the ratio of the average signal power, $\frac{B^2(t)}{2}$, to the average noise power, σ_{n}^2.

To estimate the SNR, one must first compute the Hilbert transform of $r(t)$ and determine $z(t)$ from Eq. (10.28) before inserting the result into Eq. (10.38).

Experiments with SNR

The analysis of Eq. (10.38) proceeds as follows. Five signals are contrived with added Gaussian noise namely, 2-FSK/S, 4-FSK/S, 2-PSK/S, 4-PSK/S, and Stanag 4285/S (8-PSK). Though the ultimate goal is to apply parameters to *real* HF signals with non-Gaussian noise and co-channel interference, for the current discussion Gaussian noise is sufficient. The true SNR of each of the signals is varied and the effect on Eq. (10.38) is observed.

Next the SNR estimator is applied to real signals in Table 10.1. The accuracy of the estimate cannot be confirmed because the input SNR of the narrowband receiver for each real signal is unknown. What can be accomplished is a comparison of the estimate against another SNR estimate from the power spectrum of each signal. This comparison is made and the results tabulated.

10.5 Summary

This chapter discusses three features for modulation recognition of HF signals. The first feature is the coherence function, which measures the similarity of two signal spectra. A new function, coined the coherence-median-difference (CMD), is used in the analysis of coherence for FSK signals. The second feature is a measure of entropy between two signals. Entropy is determined via Benedetto *et al*'s (2002) method, which measures the difficulty that a compression method has in compressing a concatenation of data sequences. A procedure that uniquely applies Benedetto's method to HF signals is described. The last feature is signal-to-noise ratio (SNR). The SNR is estimated with the help of Aisbett's (1986) signal power estimator.

For each feature a number of experiments are discussed. To analyze the coherence function the experimental methods include analyses of the effects of

- segment length and overlap,
- signal-to-noise ratio (SNR),
- Hamming distance, and
- misalignments in time.

Experiments with entropy utilize Benedetto *et al*'s (2002) entropy method. Specifically, the experiments investigate

- the effects of compression algorithm,
- the effects of quantizer resolution on entropy estimates,

- entropy with time-domain data versus entropy with spectral data, and
- entropic distance between synthetic and real signals.

To understand the usefulness of Aisbett's signal power estimator, an experiment is described that relates an estimate of the SNR of a synthetic signal to its true SNR.

Subsequent chapters detail the test setups and experimental observations. Chapter 11 addresses equipment, while Chapter 12 makes observations.

Chapter 11

Experimental Setup

EXPERIMENTS described in the previous chapter utilize a forerunner of the broadband receiver described in Chapter 6. The receiver is similar to the system described in Section 6.3 at Swan Reach, South Australia. However, it has three key differences: the maximum receive bandwidth is limited to 24 kHz instead of the 153.6 kHz, the antenna array is L-shaped rather than circular, and the location is near an industrial estate in Adelaide, Australia rather than the remote location of Swan Reach. This receiver is that used to collect the HF signals listed in Table 10.1 on 2 December 2003, 18 December 2003, and 15 April 2005.

The architecture of the research platform, mentioned in Figures 1.1 and 6.4, is also described in this chapter. The platform consists of an integrated modular system of **MATLAB**® scripts as well as numerous other miscellaneous **MATLAB**® scripts, and is primarily used for the analysis of the signals in Table 10.1.

11.1 Receiver Chronology

An earlier discussion highlighted the important differences between the three HF receiver systems used in this thesis (see Chapter 6), as well as their chronological order of development. Figure 11.1 shows this order. Chapter 6 discusses the swept-narrowband receiver and the broadband receiver. This chapter describes the narrowband receiver of CTD Stage 2, and the research platform mentioned in Figures 1.1 and 6.4.

The narrowband receiver is an improvement upon the early-prototype receiver. It has more antennas, greater signal processing capabilities, and utilizes more digital receivers. And, like the swept-narrowband receiver, the antennas are arranged in an L-shape and it is located in an industrial estate of Adelaide. This receive site is subject to significant environmental HF noise. Two data collection sessions were conducted with this receiver in December 2003. Another session for collecting data occurred in April 2005. The first session in December 2003 targeted groundwave transmissions of common HF signals, under the control of Ebor Computing. The second session in December 2003 targeted high-angle skywave transmissions also under the control of Ebor Computing. The last data collection session focussed on signals (both groundwave and skywave) of opportunity out of the control of Ebor Computing. Since the receiver was located in an industrial estate, all the data sets are contaminated with significant environmental noise. Only the first data set is analyzed in this thesis. Analysis of the remaining data sets is still required and is an issue for further work.

Though data collected with the broadband receiver (the product of CTD Stage 3) is useful for HF noise measurements, the data set also includes known HF signals, transmitted from various distant sites around Australia with disparate azimuths and elevation angles-of-arrival. The different directions are a result of the geography of Australia, while the different elevations are a result of the distance between transmit and receive sites as well as the state of the ionosphere at the time of recording. Consequently, this data set can also be used by the research platform for the analysis of signal features suitable for modulation recognition. Monumental effort is required to analyze all the signals in this data set because of the large volume of data ($\approx$ 300 GB). Appendix B highlights the transmissions under the control of Ebor Computing during the data collection session. Chapter 13 discusses future analyses for this data set.

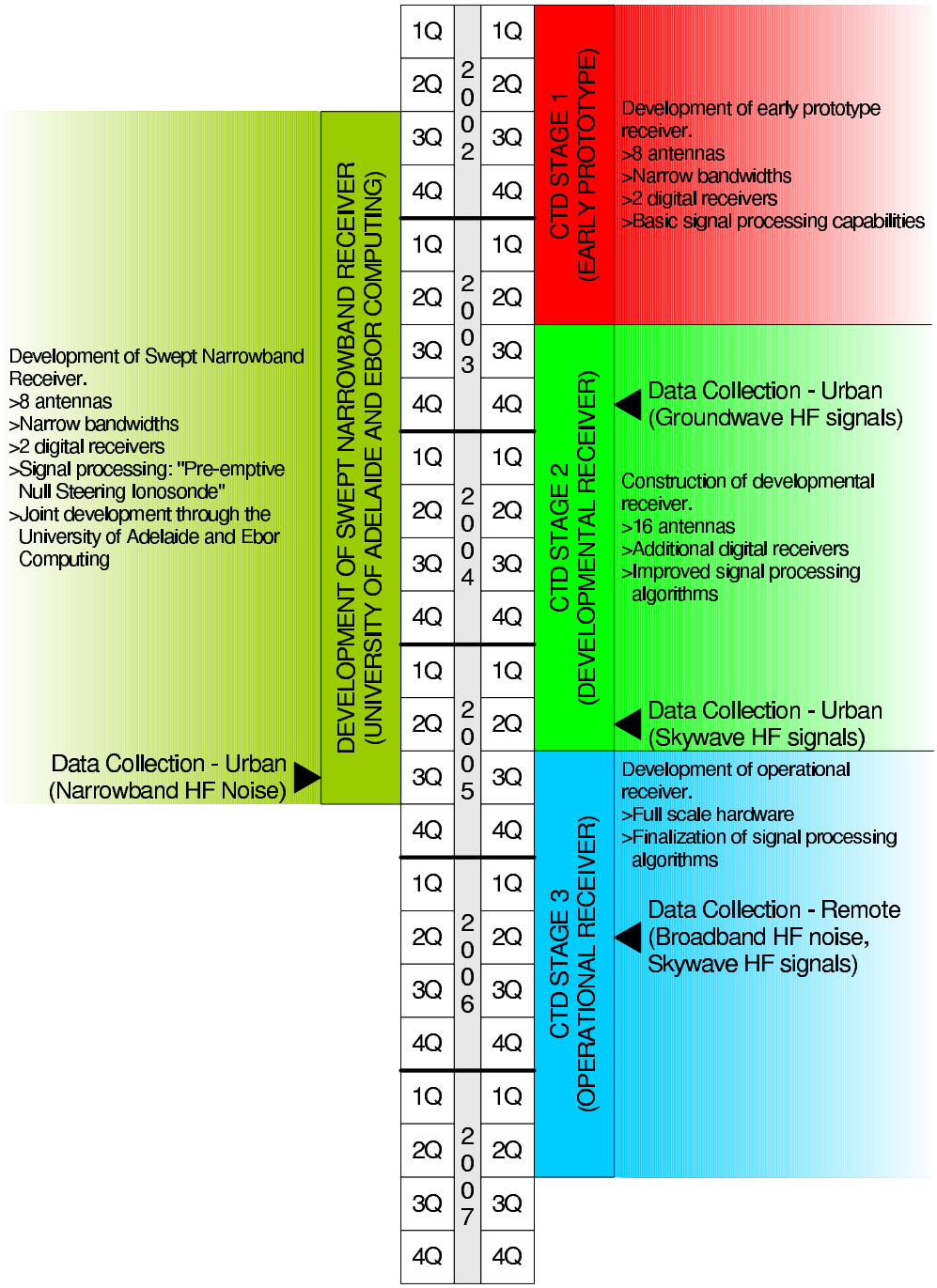

Figure 11.1. Chronology of receiver development (repeated). A chronology of the development of the broadband receiver and swept-narrowband receiver. Development of the broadband receiver occurred in three stages: an early prototype, a developmental receiver, and an operational receiver. The swept-narrowband receiver was developed in parallel by Brine *et al* (2002) through the University of Adelaide with assistance from Ebor Computing.

The research platform consists of an integrated modular system of **MATLAB**® scripts as well as numerous other miscellaneous **MATLAB**® scripts[33], to extract and analyze signal features useful for automatic modulation recognition. Though signal classification is not a topic directly addressed by this work, the research platform contains an interface to which a classification module (written for **MATLAB**®) can be added.

11.2 Narrowband Receiver

The narrowband receiver has elements similar to the swept narrowband receiver and the broadband receiver. Elements common to the swept narrowband receiver and the narrowband receiver are the antennas, antenna array, feeders, and analog-to-digital converter. Elements common to the broadband receiver and the narrowband receiver are the ICS554 digital receiver cards (the narrowband receiver has two), which encompass the ADCs and DDCs.

The antenna array is an L-shape and is oriented slightly off true-north (see Figure 11.2). It consists of eight Rohde & Schwarz HE011 active whip antennas (see Appendix E for more information), for which the antenna spacings are logarithmic (*e.g.* 5×2^0 m, 5×2^1 m, ..., 5×2^3 m). Antenna spacing and array shape is not important for this work, but is important for the CTD.

To record known groundwaves, an HF modem—Adaptive Radio Modem ARM-9401 by BAE Systems (2002)—is setup with an HF transmitter (1 W) approximately 300 m from the antenna array. At this distance, groundwaves from the transmitter arrive at the antennas relatively unattenuated and undistorted, but still containing noise and interfering signals. Various modulation schemes (see Table 10.1 and Appendix B) are implemented with the modem, which keys the transmitter with an audio baseband signal.

Signals received by the antennas are passed through bandpass filters (corner frequencies are 2.5 MHz and 32 MHz) before reaching the digital receivers. Figure 11.3 illustrates this. Bandpass filters are used in this receiver to provide an anti-aliasing function for the digital receivers. The narrowband receiver pre-dates the development of

[33]The scripts comprise about 35,000 lines of **MATLAB**® code.

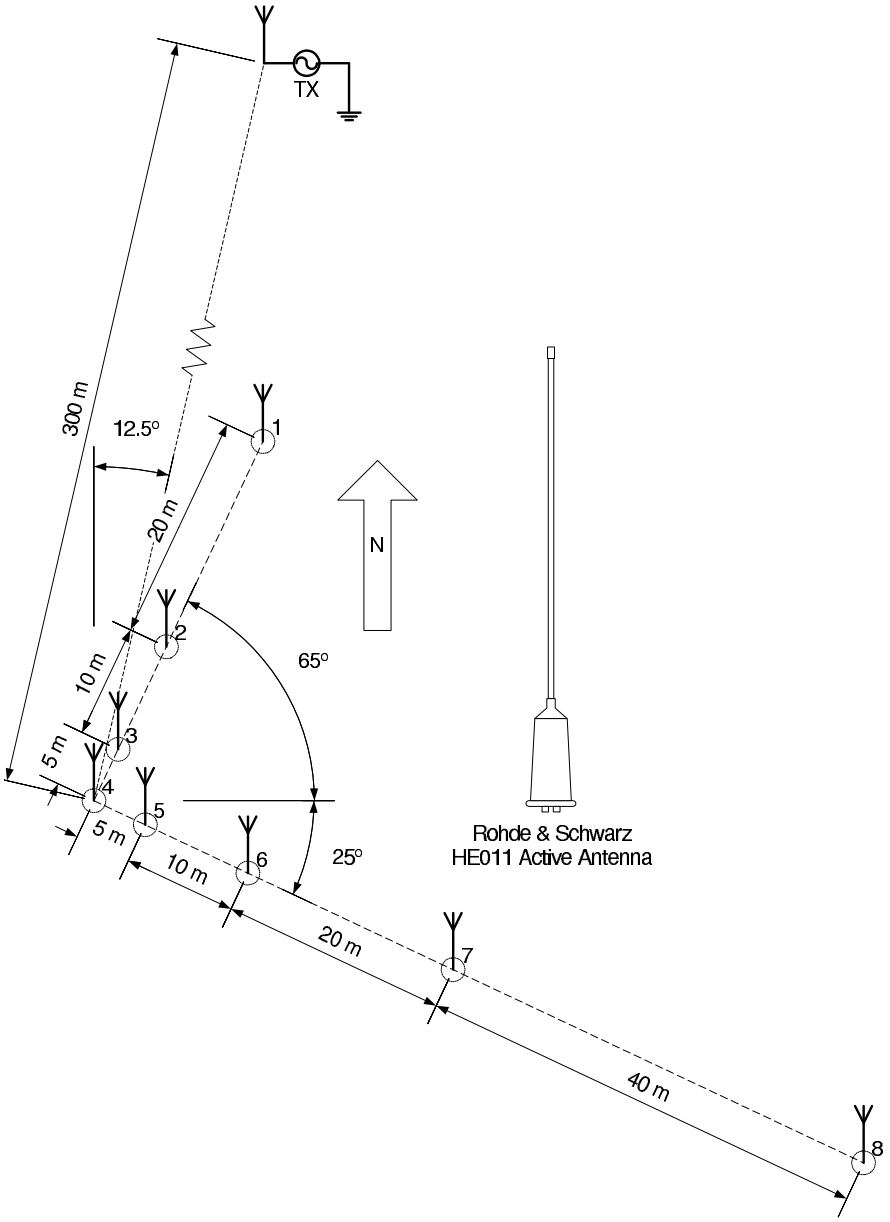

Figure 11.2. L shaped array for the narrowband receivers. The swept narrowband receiver and the narrowband receiver (CTD Stage 2) share the same antenna array. The antenna array has an L-shaped configuration and consists of eight Rohde & Schwarz HE011 active whip antennas (see Appendix E for more information). The antennas in each leg of the array are logarithmicly spaced to aid unambiguous direction finding. An HF modem (British Aerospace ARM-9401) feeds an audio baseband signal to an HF transmitter (1 W) approximately 300 m from the antenna array. At this distance, groundwaves from the transmitter arrive at the antennas relatively unattenuated and undistorted but still containing noise and interfering signals.

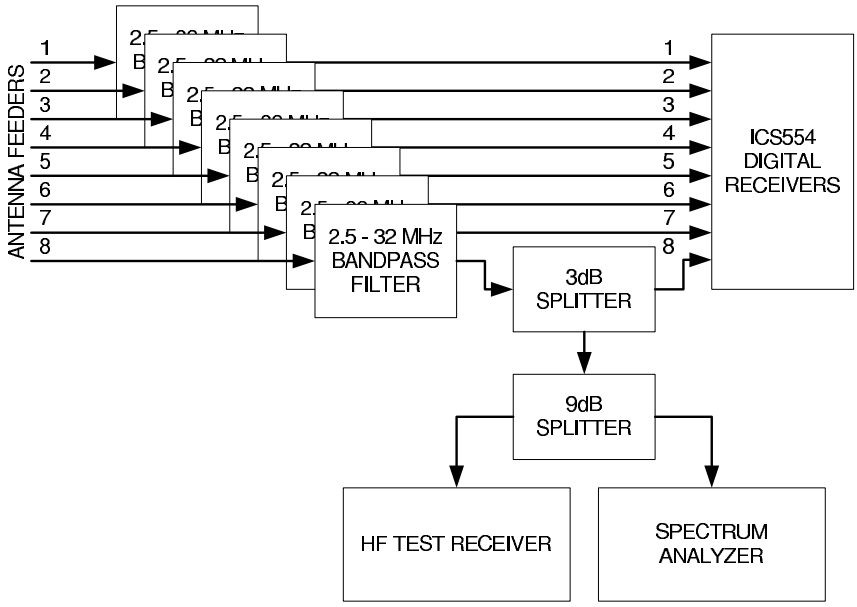

Figure 11.3. External components for the narrowband receiver. External components connected to the narrowband receiver include eight bandpass filters (2.5 MHz to 32 MHz), two RF splitters, and test gear. The bandpass filters provide an anti-aliasing function for the ADCs of the narrowband receiver. The 3 dB splitter feeds a signal, via the 9 dB splitter, to test gear that includes a Kenwood HF receiver and a Rohde & Schwarz spectrum analyzer.

the wideband gain control system, and therefore does not use gain control. For testing purposes, a 3 dB splitter is used on the eighth channel to send a signal, via a 9 dB splitter, to a Kenwood HF receiver, and a Rohde & Schwarz spectrum analyzer.

A dual-CPU rackmount computer houses two ICS554 cards[34], whereas in the simplified broadband receiver (see Section 6.3.) one dual-CPU rackmount computer is required for one ICS554. Since the maximum bandwidth of the narrowband receiver is only 24 kHz, one dual-CPU computer and its hard drive is sufficient to handle the data rate produced by two ICS554 cards. In the broadband receiver, with a bandwidth of 153.6 kHz, the data rate produced by two ICS554 cards easily overcomes its data recording facility. Therefore, the simplified broadband receiver requires two rackmount computers—one for each ICS554 card.

[34]Each ICS554 receiver card contains 16 independent digital receivers.

11.3 Matlab Test Platform

The research platform, developed entirely by the author of this work, consists of functions, written in the **MATLAB**® environment, for the investigation of parameters suitable for automatic modulation recognition of HF signals (real or synthetic). These functions form part of three broad categories: transmission, reception, and feature extraction. The transmit category includes all necessary functions to model baseband HF signals and then to upconvert the signals to the passband for transmission of the signals over a simulated HF channel. Functions in the receive category provide filtering, carrier estimation, and downconversion to baseband. The feature extraction category attempts to identify unique features of the signals.

Recall that modulation recognition is a process that determines the modulation of a signal with no prior knowledge of that signal. The process generally comprises three steps: parameter extraction, feature selection, and classification, though sometimes parameter extraction and feature selection are combined and called feature extraction.

Parameter extraction attempts to isolate unique characteristics of the signal so that the signal can be classified. Fundamental characteristics are frequency, phase, and amplitude, but statistical measures (*e.g.* standard deviation, n^{th}-order moments, nth-order cumulants) are not uncommon. Whatever the parameters, they are chosen to form an N-dimensional vector.

Feature selection then transforms the N-dimensional vector into an M-dimensional feature space, where $M \leq N$. With an optimal choice of parameters, the M-dimensional feature space consists of M orthogonal basis vectors. This may not be possible for a sub-optimal parameter list, but the more orthogonal the basis vectors, the easier a classifier can separate signal types.

The classification step identifies a modulation based on its components of the M-dimensional feature space. Common methods for grouping of features include decision theoretic methods, artificial neural networks (ANNs), pattern recognition algorithms, and statistics. There are numerous references for those desiring detailed information on classification for automatic modulation recognition (Cao *et al* 2003, Choi & Lee 2003, Choi & Kim 2000, Distante *et al* 2002, Fragoulis *et al* 2001, Kuo & Landgrebe 2004, Landgrebe 1997, Mitchell & Westerkamp 1999), as discussed in Chapter 9.

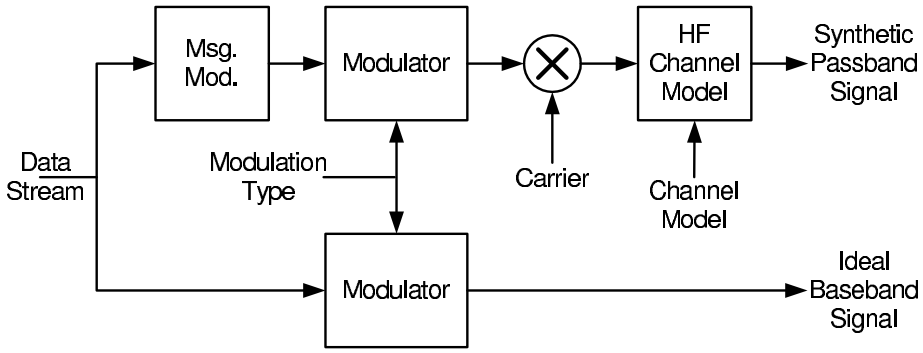

Figure 11.4. Transmit section of the Ⓢigns toolbox. The transmit section of the Ⓢ**igns** toolbox provides two outputs: a synthetic passband signal and an ideal baseband signal. A binary stream is modulated to form an ideal baseband signal. The signal is ideal because it is not affected by the transmission medium. The other branch models the effects of the transmission medium on the signal. The first box allows the user to modify the bit stream (*e.g.* introduce bit errors). The modulator encodes the bit stream in an analog waveform. Upshifting to the carrier frequency occurs next, followed by modification of the passband signal by the HF channel model.

The aforementioned **MATLAB**® functions are part of a custom **MATLAB**® toolbox called Ⓢ**igns** . The name is derived from the concept that modulation recognition involves analysis of signal identifying parameters (or signs) from a received signal. The Ⓢ**igns** toolbox is useful for observing the performance of a particular parameter against synthetic or real HF signals. Figures 11.4, 11.5, and 11.6 illustrate the system blocks in this toolset.

The transmit toolbox takes as an input, among other things, a binary data stream. This data stream is modulated with the chosen modulation scheme to produce an ideal baseband reference signal—X in the context of Figure 10.1. The other path generates a synthetic passband signal that ultimately, after downconversion, becomes Y in the context of Figure 10.1. In this path, the data stream can have its message modified to suit the purpose of the investigation (*e.g.* simulation of message variations, bit errors, addition of error-correcting codes, and so on). The modified message is then modulated with the chosen modulation type, upconverted, and then passed through an appropriate HF channel model. The modulator, in the transmit toolbox, is capable of generating a number of baseband signals such as m-ary FSK, m-ary PSK, Stanag 4285 (a military standard 8-PSK) signal, and Mil-Std-188-110A FSK signals. The HF channel model can be configured to follow a user-specified algorithm, the narrowband Watterson & Coon (1969) model, or other wideband models (Lemmon & Behm 1991, Lemmon

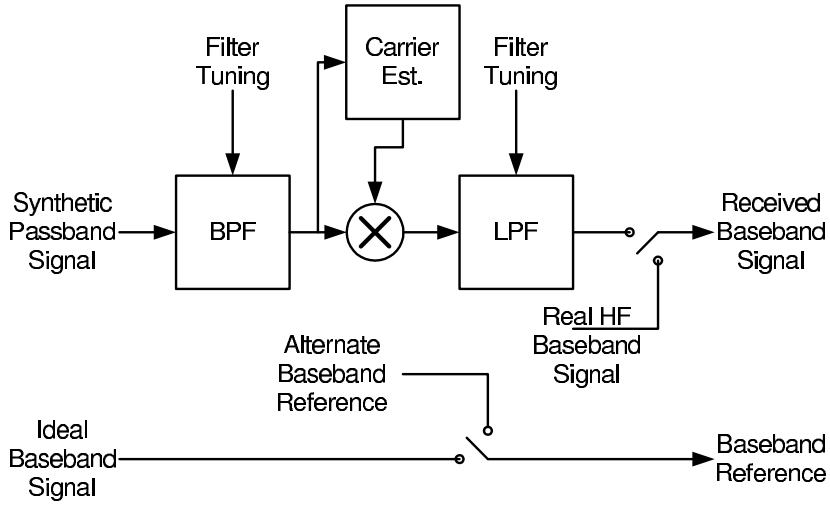

Figure 11.5. Receive section of the Ⓢigns toolbox. The receive section of the Ⓢ**igns** toolbox operates in a number of modes. In the simulation mode, the ideal baseband signal from the transmit section is allowed to pass through with no modification. The synthetic passband signal, however, is filtered and downconverted to baseband. Alternatively, real HF signals can be switched in to replace the synthetic signals in either the baseband reference path, or the path for the received baseband signal.

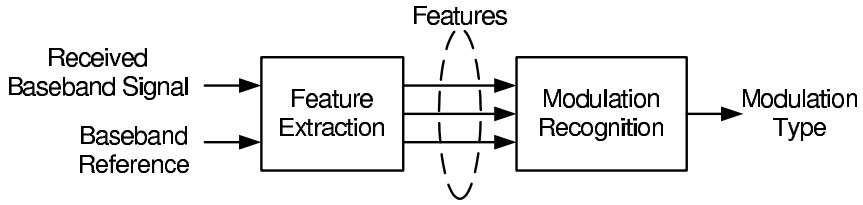

Figure 11.6. Modulation recognition section of the Ⓢigns toolbox. The modulation recognition section of the Ⓢ**igns** toolbox consists of two modules: a feature extraction module and a classification module. Both modules can be customized. Currently, the feature extraction module can generate twelve different signal features. Classification can be implemented any way the user chooses.

& Behm 1993). Whatever the model, it is important that the model account for Doppler shifts, deep fades, impulsive noise (Giesbrecht *et al* 2006, Johnson *et al* 1997) (as opposed to Gaussian noise), and multi-modal signals (*i.e.* signals propagating by more than one ionospheric layer).

The receive toolbox is designed to reverse the processing applied by the transmit toolbox. However, the receive toolbox also incorporates switches that allow for the synthetic signals to be ignored and for alternate signals (real or synthetic) to be output to the feature extraction toolbox. For example, one may develop a feature extraction tool, test it on synthetic signals and then apply a recorded real signal to test the robustness

of the new tool. The downconversion path provides filtering and carrier estimation. The filters are tuneable to suit the particular HF signal. A carrier extraction function can be programmed with the true carrier frequency, or it can estimate the carrier based on the received signal. Other modules can be added if necessary to support analysis of technically complex signals.

Features can be extracted from simulated or real HF signals with the feature extraction toolset. The feature extraction block includes a set of algorithms for twelve parameters: centre frequency, bandwidth, SNR estimators (Aisbett 1986), signal envelope, symbol frequencies, cross-Margenau-Hill distribution (Ketterer *et al* 1999), kurtosis (Akmouche 1999), signal constellation, auto- and cross- PSD, modulation level, auto-regressive covariance (Ketterer *et al* 1999), and entropic distance (Benedetto *et al* 2002). The primary focus of the research platform is the investigation of signal identifying features.

The Ⓢ**igns** toolbox, though sufficient for the work in this thesis, is not complete. The HF Channel Model provides an interface for Watterson's (1969, 1970) narrowband model, Vogler's (1988, 1990, 1992) wideband model, and Lemmon's (1991, 1993) wideband model. None of these models are implemented. The current version of the Ⓢ**igns** toolbox implements a simple additive white-Gaussian noise model for the HF channel. Moreover, the carrier estimate block depicted in Figure 11.5 requires implementation. Additionally, only entropic distance, coherence, and the SNR estimator have are contained in the feature extraction module. Interfaces for the other parameters are provided in the module. Finally, an interface is provided by Ⓢ**igns** that enables a classification module to be added to the toolbox. The current Ⓢ**igns** toolbox does not include a classifier. A simple classification algorithm could include a majority logic algorithm and a thresholding algorithm.

A user guide for the Ⓢ**igns** toolbox is included in Appendix G.

11.4 Summary

This chapter describes the narrowband receiver in relation to the swept narrowband receiver and the broadband receiver. The narrowband receiver has components in common with the swept narrowband receiver and the broadband receiver. It was developed in parallel with the swept narrowband receiver and prior to the broadband receiver. The narrowband receiver is used to record certain of the HF signals highlighted in Appendix B.

The chapter also summarizes the **MATLAB**® research platform, which consists of numerous functions and the Ⓢ**igns** toolbox. The Ⓢ**igns** toolbox, described in detail in Appendix G, is a modular system of **MATLAB**® scripts and functions that can be used to extract features from signals and to automatically recognize them.

Having described the experimental setup, the discussion now focuses on the results of the experiments in Chapter 12.

Chapter 12

Feature Parameters of the Signal Set

EXTRACTION of useful signal-identifying parameters is an important part of automatic modulation recognition. The question is: which parameters are useful? Chapter 10 describes three parameters that may aid automatic modulation recognition: coherence, entropic distance, and signal-to-noise ratio. This chapter analyzes the usefulness of these parameters for identifying a set of HF signals (real and synthetic).

It is shown that coherence is affected by the message of the signal as well as the signal-to-noise ratio (SNR). Entropic distance is affected by the compression method, is sensitive to the message structure, and is able to separate some signals. A modification of Aisbett's (1986) hash function provides an estimate of SNR that is linear over a moderate range, but is more useful for constant envelope signals than it is for signals with a varying envelope.

12.1 Introduction to Results

The topic of modulation recognition is not new. It has been a field of research interest for at least the last 40 years. And, it continues to progress with technological developments. Yet for all the previous research, a truly robust and universal modulation recognition technique is still elusive. Nevertheless, this chapter investigates three parameters that could contribute to such a modulation recognition algorithm.

Chapter 10 discusses methods and experiments to study the coherence function, entropic distance, and signal-to-noise ratio. The results of these experiments are discussed in the following three sections. Coherence is shown to be a parameter that provides a *yes/no* result. Entropic distance can be used to separate HF signals based on their modulations. Finally, an estimator is found to provide a reasonable estimate of signal-to-noise ration (SNR) over a moderate range.

12.2 Coherence Results

Chapter 10 describes the coherence function and its relationship with signal-to-noise ratio (SNR). It also describes a parameter, the coherence-median-difference (CMD), for m-ary FSK signals that is useful for measuring the dominance of the coherence at the symbol frequencies to the coherence at all other frequencies in the bandwidth. Typically, coherence must be estimated as the function itself can be intractable. Previous work (Carter 1993) describes a segment overlapping technique using Welch's (1984) periodigram method. Carter's work also stresses the time-sensitivity of the coherence function. Consequently, there are four experiments designed 1) to investigate the effects of the number of segments and percent overlap in the overlapping technique, 2) to study the relationship between coherence and SNR, 3) to reveal the dependency of the coherence function on the message of digital signals, and finally 4) to determine with real signals, the sensitivity of coherence to timing.

Coherence versus Number of Segments and Overlap

Coherence, as seen previously, can be difficult to calculate especially if the calculations of the power spectra of the signals in question are intractable. The WOSA technique

(see Chapter 10) can be used in the estimation of coherence. Recall that WOSA breaks a time-domain data sequence into windowed segments that are overlapped. The Fourier transform is then applied to each of the overlapped segments, the results of which are used to generate the auto- and cross- power spectra. Coherence is then calculated with the estimates of the power-spectra. However, the quality of the coherence measure is affected by the number of segments and the amount of overlap. This experiment measures these affects.

Nuttall (1958) shows that two random processes are separable if their cross-correlation is directly proportional to the auto-correlation of one of the processes. Chapter 10 describes a separable sinusoidal process that is used here to control the coherence and thereby observe the effects of the number of segments and amount of overlap.

Two arbitrary tones, X and Y, are simulated to create a separable process. The signal $X(t) = \cos(\omega_x t)$ and $Y(t) = \cos(m\omega_x t)$. It is known (see Appendix A) that in this scenario the coherence follows a $\text{sinc}^2(m)$ function defined by Eq. (10.12), and repeated here with $n = 1$

$$\gamma^2(\omega) = \frac{\{\text{sinc}(\pi[m+1]n) + \text{sinc}(\pi[m-1]n)\}^2}{1 + \text{sinc}(2\pi mn)}, \tag{12.1}$$

where Figure 10.4 illustrates the nature of the coherence function. In this instance $\omega_x = 1$ rad/s and therefore the frequency of $Y(t)$ depends only on the normalized frequency m. The sampling rate is chosen to be 7958 Hz so that 50,000 samples are contained within the period of $X(t)$. A large number of samples are required so that a reasonable number of samples are present in each overlapped segment. For example, at an overlap of 20% and 128 segments, the number of samples in each segment is 484. At 70% overlap this increases to 1278. Furthermore, each segment is windowed with a Hann window of a length equal to the segment length. Carter (1993) emphasizes the need for resolvability of spectra in the coherence estimate; the Hann window provides fine frequency resolution and a minimum attenuation of 13 dB for sidelobes (Harris 1978).

Table 12.1. Theoretical coherence values of two arbitrary sinusoids. The true coherence of two arbitrary sinusoids for various m. In this case, the curve is defined by Eq. (10.12) and displayed in Figure 10.4. For integer values of m, the coherence is zero except for the case of $m = 1$, which says that the frequency of $Y(t)$ is identical to that of $X(t)$.

m	Exact Coherence	Decimal Equivalent
$\frac{2}{3}$	$\frac{864}{25\pi(8\pi-3\sqrt{3})}$	0.5518
1	1	1.000
$\frac{5}{4}$	$\frac{4000}{81\pi(5\pi+2)}$	0.8877
2	0	0.0000
$\frac{5}{2}$	$\frac{400}{441\pi^2}$	0.0919

Estimation of the coherence[35] of $X(t)$ and $Y(t)$ is repeated for various normalized frequencies, number of segments, and overlap. For this experiment the normalized frequency, m, takes on values from Table 12.1. For each value of m, the number of segments is varied from 4 to 128 (in increments of 1) at an overlap of 20%. The coherence estimation is then repeated for overlaps of 50% and 70%. Thus, for each value of m there are 375 coherence calculations.

Since the coherence at any m is independent of frequency (*i.e.* the coherence is constant), the coherence estimate should also be constant. Consider the case where $m = \frac{2}{3}$ with the corresponding coherence in Table 12.1. What is the coherence estimate? Figure 12.1 shows the estimated coherence at 20% overlap and for 16, 32, 64, and 128 segments. Note that the support for each plot is very near the true coherence, but that numerous peaks appear. These excursions from the true coherence are related to the overlap.

From Eq. (A.23), there are 3846 samples in the overlap region for 16 segments. At a sampling rate of 7958 Hz this corresponds to a time period of approximately 483 ms and a bandwidth of 2.06 Hz. Close examination of the peaks reveals that they are indeed about 2 Hz apart. Repeating the analysis for the other cases shows that for 32 segments the bandwidth between peaks is 4.11 Hz, for 64 segments the bandwidth is 8.18 Hz, and for 128 segments the bandwidth is 16.32 Hz. Similar results are observed for 50% overlap and 70% overlap.

[35] This estimation of coherence uses the **MATLAB**® *cohere* function.

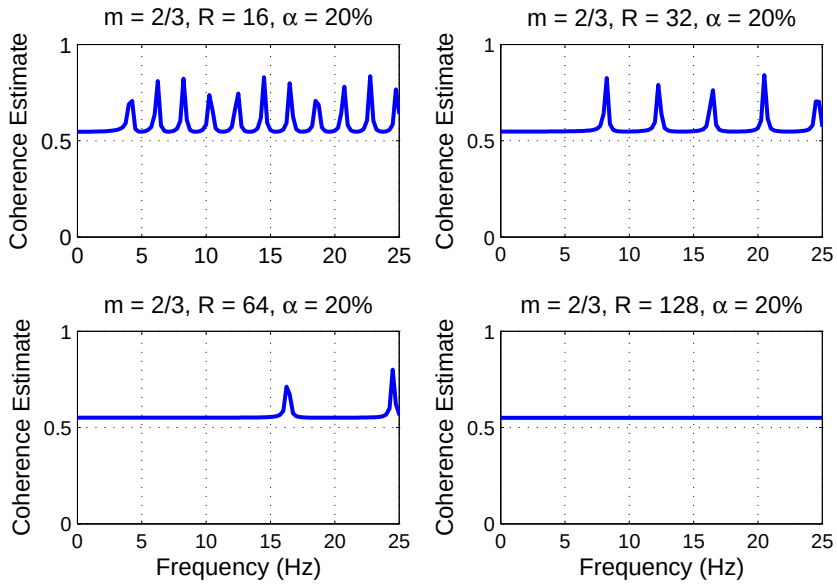

Figure 12.1. Coherence estimate at 20% overlap ($m = \frac{2}{3}$). The estimate of coherence improves with an increase in the number of overlapped segments. In this case the normalized frequency, $m = \frac{2}{3}$ and the true coherence is approximately 0.5518. The support in each plot is near the true coherence. The excursions from the true coherence are related to the overlap and normalized frequency, m. As the overlap increases, the separation bandwidth between the peaks increases. The normalized frequency controls the number of full cycles of each waveform in the coherence calculation. The constants R and α are the number segments and overlap percentage.

Note also, that the first peak in the train of peaks occurs at double the separation bandwidth. For the 16 segment case, the first peak appears at $\approx$ 4 Hz. At 32, 64, and 128 segments the first peaks appear at $\approx$ 8 Hz, $\approx$ 16 Hz, and $\approx$ 32 Hz respectively. For the latter, the first peak appears outside the plotting window, hence a horizontal line is observed from DC to 25 Hz. Clearly the position of the peaks are controlled by the amount of overlap and number of segments.

This behaviour is a characteristic of the WOSA method. Each segment to be analyzed contains some data from the previous segment (this is true of all segments except the first) and data unique to the current segment. Therefore, the auto- and cross-power spectra of the segment contain components common to two overlapped segments. The WOSA method then averages all the auto- and cross-power spectra from each segment and, depending on the nature of the signals being correlated, the spectral components

resulting from the overlapped data can add constructively or destructively. The coherence can then have peaks or dips at frequencies corresponding to the size of the overlap. As a result, the first peak is missing from the coherence estimate, which makes the second appear as though it is the first at double the separation bandwidth. This is because of the R segments used in the coherence estimate, only the last $R-1$ segments are overlapped with a preceding segment. The first segment does not have and the common constructive/destructive spectral components that the other segments inherit.

The summary of all of this is that as the number of segments increase the coherence estimate converges on the true coherence, and the deviation of the coherence estimate from the true coherence decreases. This is just that which Carter (1993) concludes. It is, therefore, no surprise that if the mean coherence across all frequencies is used as an estimate of the true coherence, that Figures 12.1 to 12.5 show a mean coherence estimate converging on the true coherence with increasing number of segments[36]. Moreover as the overlap percentage increases, the variance of the coherence estimate across all frequencies decreases. However there is a point where further increase in the number of segments provides only marginal improvement in the coherence estimate. This point is about 64 segments. The point of diminishing returns also applies to the percentage overlap. Increasing the overlap above 50% provides little improvement in the variance.

One may question the reason for using the mean coherence across all frequencies rather than the minimum coherence as an estimator of the true coherence. After all, the support displayed in Figure 12.1 suggests the minimum coherence as the best estimator. If the coherence is estimated for numerous values of m and the maximum, minimum, and mean values of the coherence estimate across all frequencies are plotted, a series of curves result that show the benefits of using the mean as the best estimator.

As an estimator of true coherence, the maximum of the coherence estimate across all frequencies overestimates the true coherence for $m < 1$. This is obvious from Figure 12.1 and is reflected in Figure 12.6. However, for $m \geq 1$, the maximum is a good estimator of the true coherence. Why is this so? The same logic that explains the peaks in Figure 12.1 explains the dips in Figure 12.7. For the scenario described by Eq. (10.12),

[36] For $m = 1$ the mean coherence estimate is unity. The variance of the estimate is negligible (on the order of 10^{-32}). No curves are shown for this scenario.

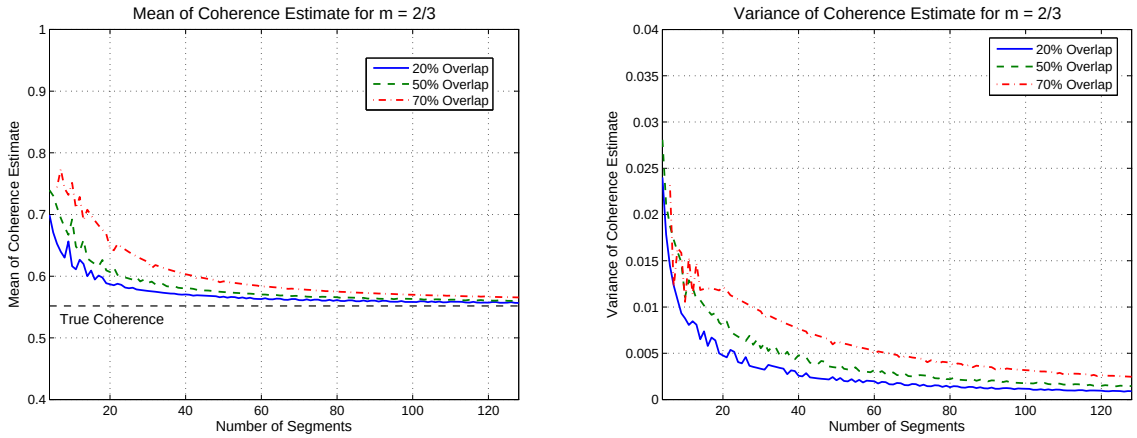

Figure 12.2. Mean & variance of coherence estimate at 20%, 50%, & 70% overlap ($m = \frac{2}{3}$). The mean estimate of coherence (*left*) converges on the true coherence with increasing number of overlapped segments. As well, an increasing number of segments reduces the variance of the coherence estimate (*right*). Increasing the number of segments beyond ≈ 64 provides only marginal improvement in the coherence estimate. The point of diminishing returns also applies to the percentage overlap. Increasing the overlap above $\approx 50\%$ provides little improvement in the variance. In this case the normalized frequency, $m = \frac{2}{3}$ and the true coherence is approximately 0.5518.

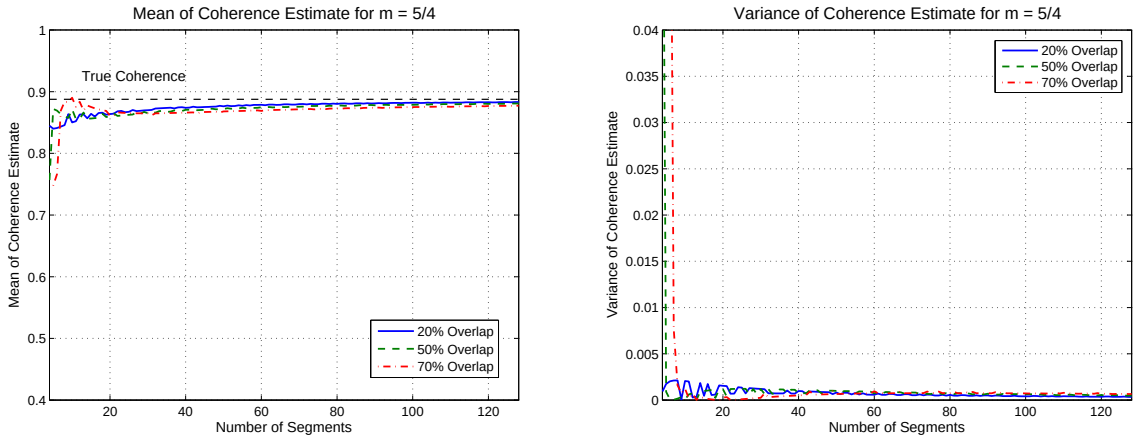

Figure 12.3. Mean & variance of coherence estimate at 20%, 50%, & 70% overlap ($m = \frac{5}{4}$). The mean estimate of coherence (*left*) converges on the true coherence with increasing number of overlapped segments. As well, an increasing number of segments reduces the variance of the coherence estimate (*right*). Increasing the number of segments beyond ≈ 64 provides only marginal improvement in the coherence estimate. The point of diminishing returns also applies to the percentage overlap. Increasing the overlap above $\approx 50\%$ provides little improvement in the variance. In this case the normalized frequency, $m = \frac{5}{4}$ and the true coherence is approximately 0.8877.

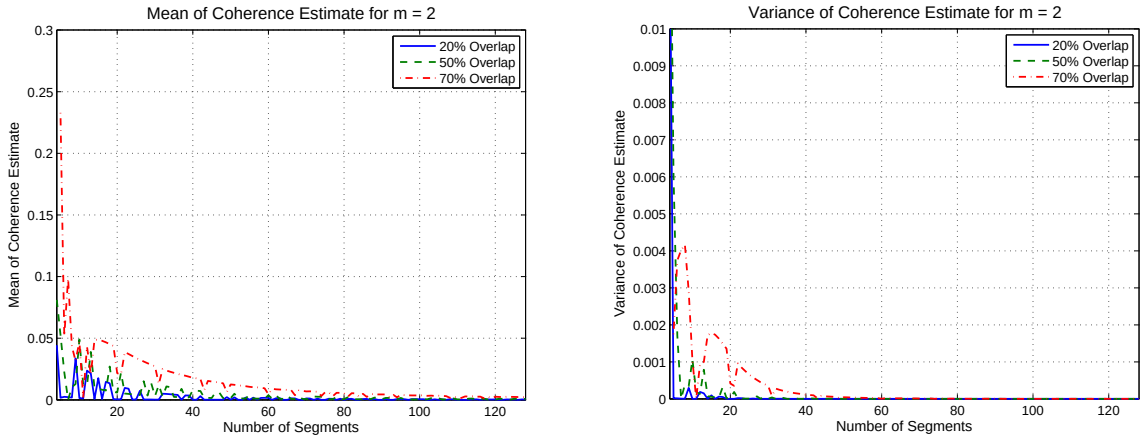

Figure 12.4. Mean & variance of coherence estimate at 20%, 50%, & 70% overlap ($m = 2$). The mean estimate of coherence (*left*) converges on the true coherence with increasing number of overlapped segments. As well, an increasing number of segments reduces the variance of the coherence estimate (*right*). Increasing the number of segments beyond ≈ 64 provides only marginal improvement in the coherence estimate. The point of diminishing returns also applies to the percentage overlap. Increasing the overlap above $\approx 50\%$ provides little improvement in the variance. In this case the normalized frequency, $m = 2$ and the true coherence is exactly zero.

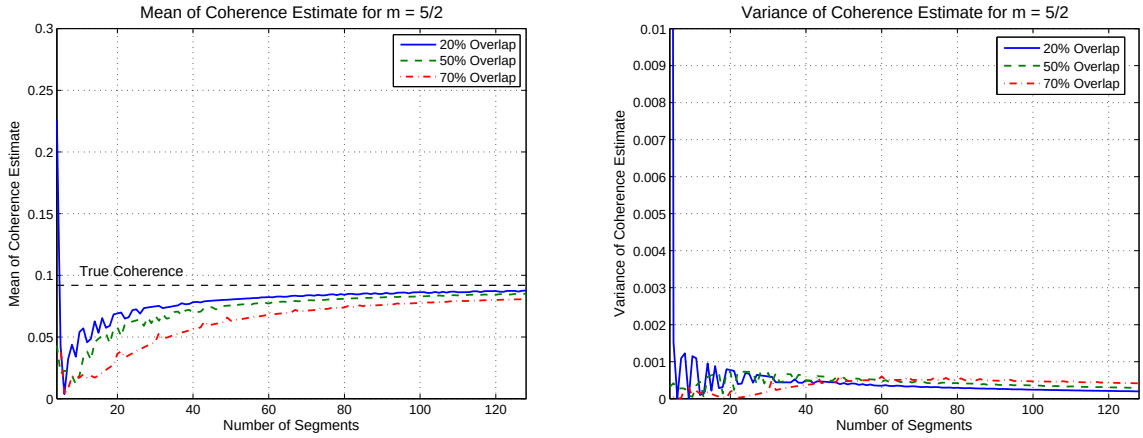

Figure 12.5. Mean & variance of coherence estimate at 20%, 50%, & 70% overlap ($m = \frac{5}{2}$). The mean estimate of coherence (*left*) converges on the true coherence with increasing number of overlapped segments. As well, an increasing number of segments reduces the variance of the coherence estimate (*right*). Increasing the number of segments beyond ≈ 64 provides only marginal improvement in the coherence estimate. The point of diminishing returns also applies to the percentage overlap. Increasing the overlap above $\approx 50\%$ provides little improvement in the variance. In this case the normalized frequency, $m = \frac{5}{2}$ and the true coherence is approximately 0.0919.

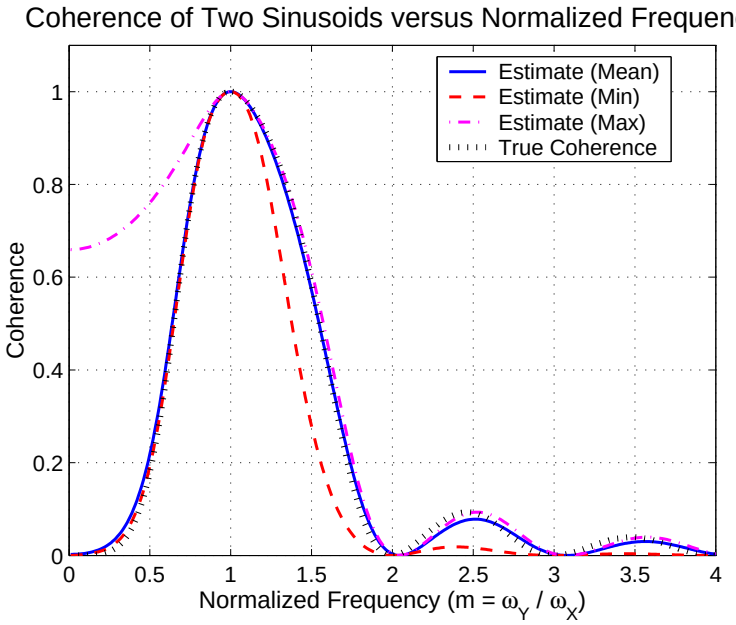

Figure 12.6. Mean, minimum, & maximum as estimators of true coherence. The mean of the coherence estimate across all frequencies is the best estimator for all values of m. As an estimator of true coherence, the maximum of the coherence estimate across all frequencies overestimates the true coherence for $m < 1$, but provides an excellent estimator for $m \geq 1$. The minimum is a good estimator for $m < 1$ but, it underestimates the true coherence for $m \geq 1$.

peaks appear in the coherence estimate when there is less than one full cycle of one or both of the waveforms involved in the coherence calculation; dips appear in the coherence estimate when there are more than one full cycle of one or both of the waveforms involved in the coherence calculation. For $m = \frac{2}{3}$, $Y(t)$ has less than one full cycle in the period of $X(t)$. However, if $m \geq 1$ then $Y(t)$ contains at least one full cycle in the period of $X(t)$. Recall that $X(t)$ and $Y(t)$ have infinite energy and therefore no Fourier transform. A Fourier transform does exist, however, if the transform is computed over a finite time interval (typically one period of the waveform). Thus the maximum of the coherence estimate is a good estimator of the true coherence when $m \geq 1$ and it is a poor estimator when $m < 1$.

Now, what of the minimum of the coherence estimate across all frequencies as an estimator of the true coherence? In fact, Figure 12.6 shows that the minimum is a good estimator for $m < 1$ and that it underestimates the true coherence for $m \geq 1$. This makes sense, because Figure 12.7 shows that for $m = \frac{5}{4}$ dips appear in the coherence

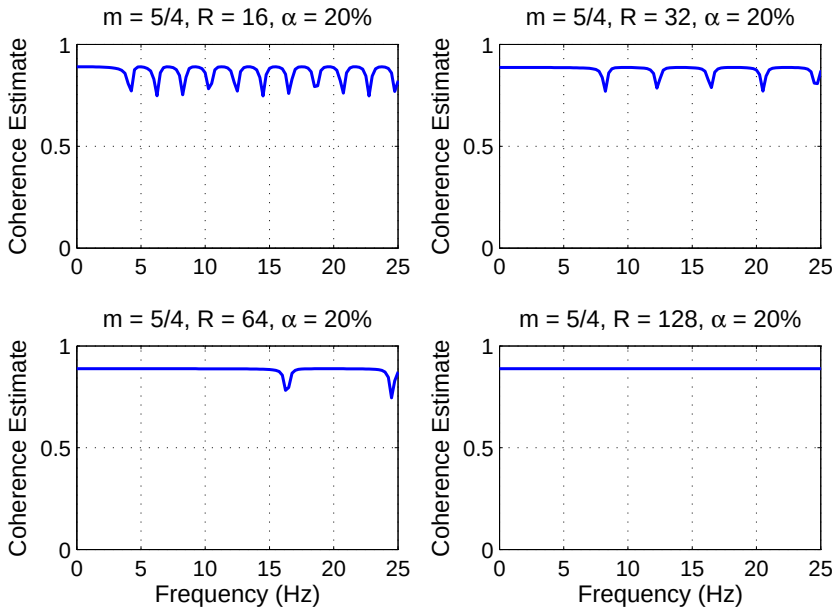

Figure 12.7. Coherence estimate at 20% overlap ($m = \frac{5}{4}$). The estimate of coherence improves with an increase in the number of overlapped segments. In this case the normalized frequency, $m = \frac{5}{4}$ and the true coherence is approximately 0.8877. The support in each plot is near the true coherence. The excursions from the true coherence are related to the overlap and normalized frequency, m. As the overlap increases, the separation bandwidth between the dips increases. The normalized frequency controls the number of full cycles of each waveform in the coherence calculation. The constants R and α are the number segments and overlap percentage.

estimate across all frequencies. More generally, Figure 12.6 says that for $m \geq 1$ the minimum of the coherence consistently underestimates the true coherence.

This finally leaves the mean of the coherence estimate across all frequencies as the best estimator of the true coherence for the entire range of m but, for $m \geq 1$ the maximum is a better estimator, and for $m < 1$ the minimum is a better estimator. The mean of the coherence estimate is used extensively in the subsequent studies of coherence.

Of course, the discussion on effects of m are particular to this experiment. Nevertheless, the effects of the number of segments and amount of overlap apply to any process.

Coherence versus SNR

Chapter 10 shows that the coherence is related to the signal-to-noise ratio (SNR) of a signal. In fact, an S-shaped curve is illustrative of this relationship (see Figure 10.5).

The question is: how does this *S-curve* behave inside and outside a particular bandwidth.

Assume a bandlimited sinc function with a centre frequency of 1 kHz. Add to this sinc function white Gaussian noise confined to a 600 Hz bandwidth about the centre frequency. One finds that the *S-curves* in the 600 Hz band fall on top of each other, while the *S-curves* outside the 600 Hz band are shifted away from the 0 dB SNR position (see Figure 12.8). Coherence curves outside the 600 Hz band (*i.e.* $\gamma^2(f)$ 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.

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 a *brute-force* method, 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 2-FSK/S signal with mark frequency, $f_m = f_1$, and space frequency, $f_s = f_2$, and the received signal is a 2-FSK/S signal with $f_m = f_3$ and $f_s = f_4$ the coherence will only indicate that the two signals are not the same. Inferring that the received signal is a 2-FSK signal, with different mark and space frequencies, from knowledge of the reference signal and the calculated coherence is unlikely.

Coherence vs. Hamming Distance

Two arbitrary 2-FSK/S signals are simulated to study the effects of the message on the coherence between a transmitted signal and its noisy received counterpart. The mark frequency (f_m) for each signal is 3 kHz and the space frequency (f_s) is 1 kHz. One signal is assigned to the baseband reference, $X(t)$ and the other is assumed to be the noisy received signal, $Y(t)$. The Hamming distance between $Y(t)$ and $X(t)$ is adjusted

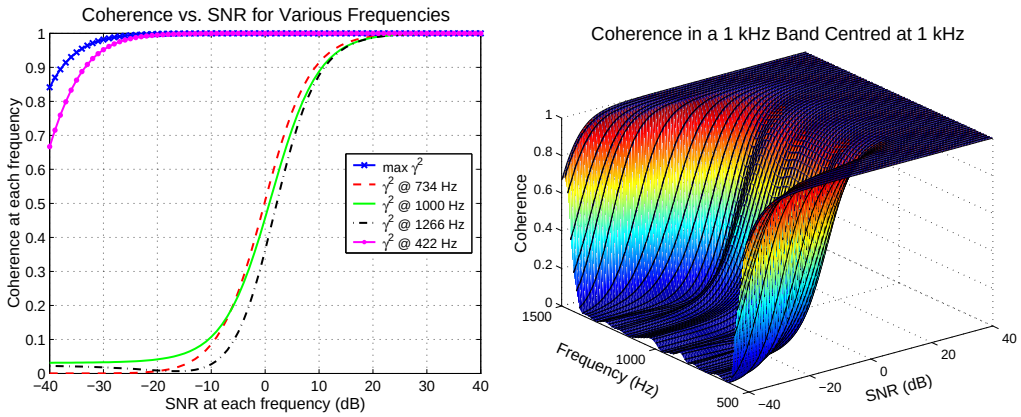

Figure 12.8. Coherence versus SNR. Coherence versus SNR and frequency about a bandlimited sinc function (*right*). The bandwidth of the sinc function is 1 kHz and the bandwidth of the AWGN is 600 Hz. The projection (*left*) shows that the *S-curves* in the 600 Hz band are nearly identical. Curves outside the band are shifted left.

to simulate bit errors and the coherence is computed using the WOSA method. The baseband reference has a uniform distribution of marks (1's) and spaces (0's).

Figure 12.9 illustrates the effects of Hamming distance on coherence between $X(t)$ and $Y(t)$. One immediately observes 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 coherence-median-difference (CMD) indicates that the peak coherence is generally dominated by the coherence at the mark and space frequencies. This suggests that for long finite-length messages the coherence function can be used to identify synthetic 2-FSK signals with a thresholding technique.

The peak coherence also follows closely the coherence at f_s. The fact that the peak coherence is larger than the coherence at f_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.

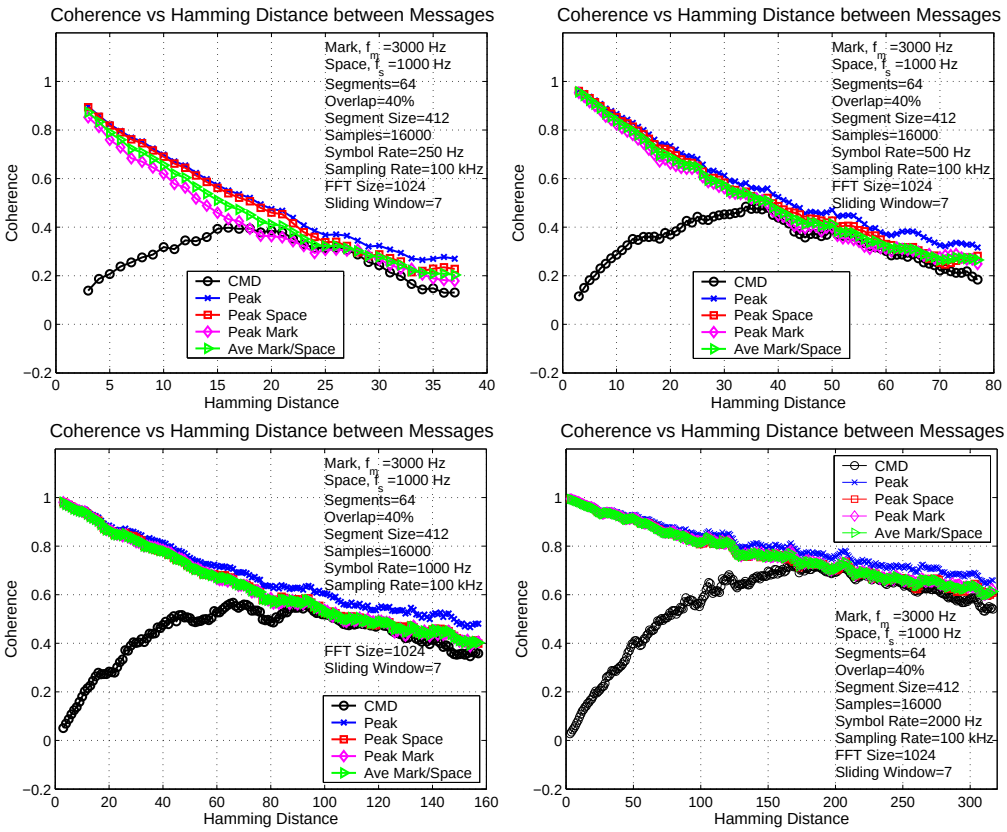

Figure 12.9. Coherence & CMD of 2-FSK/S signals versus Hamming distance. Coherence and CMD of two 2-FSK/S signals as a function of Hamming distance between the signals. The figures show coherence and CMD for two 2-FSK/S signals each having 40 symbols (*upper-left*), 80 symbols (*upper-right*), 160 symbols (*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.

Now, why does the coherence vary with the message? Consider the power spectra of two 2-FSK/S signals with no noise or interference. If each 2-FSK/S signal consists of the first half being marks and the second half spaces there would be two relatively "clean" tones in the respective power spectra. The coherence between these two 2-FSK signals will be near unity. However, as the distribution of marks and spaces of the second 2-FSK/S signal becomes more uniform the resulting numerous symbol transitions spreads the signal energy over many more frequencies. So even though the second 2-FSK signal contains the same tones as the "clean" signal, the overall coherence is affected because the signal energy is spread.

In all cases as the Hamming distance increases the coherence decreases. Moreover, as the Hamming distance increases the power spectra of the two signals become more dissimilar. Eventually a Hamming distance is reached that makes the two signals very different and hence the coherence is low. The implication of these results is that no matter what the length of the message, increasing Hamming distance (or for that matter bit errors) will eventually reduce the coherence to zero. In addition, the length of the message affects the rate at which the coherence tends to zero. Coherence of signals with short messages tends to zero more quickly than the coherence of signals with long messages. An alternative interpretation is this. The shorter the sequence of samples used in the coherence estimation, the more likely the resultant coherence will be low. The longer the sequence of samples used in the estimation, the more likely the coherence estimation will yield a high coherence. Therefore, in a simulation environment, coherence could be a useful detector of 2-FSK signals provided the lengths of the sequences used in the coherence estimation are large, but not so large as to allow the coherence to drop unacceptably low.

There may be merit in investigating the effects of Hamming distance on the coherence of two synthetic 2-PSK signals but, the comments on coherence with 2-FSK/S signals appear to apply to synthetic 2-PSK signals as well. A brief application of the coherence function to two such signals, each conveying a 4096-bit message with Hamming distance of 2053, yields a low coherence. The usefulness of coherence does not look promising. Later results show that the coherence of real signals is much more sensitive to timing, bit errors, and the message.

Let us revisit the *S-curve* and see how it relates to changes in the Hamming distance. If *S-curves* are plotted for various Hamming[37] distances between two 2-FSK/S signals, a series of curves result (see Figure 12.10). 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 2-FSK/S signals become less correlated. For positive SNR the average coherence of the mark and space frequencies is primarily affected by the Hamming distance. As Hamming distance increases the coherence decreases (particularly noticeable at high SNR)—eventually to zero. Consequently, low correlation between the messages carried by the two signals has the same effect as high noise levels.

If one considers the coherence-median-difference (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 2-FSK/S 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 of the mark and space frequencies.

The high level of coherence 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 in the sense that it reliably identifies whether or not a received signal is identical to a reference signal.

Time Sensitivity of Coherence

It is known (Carter 1993) that coherence is sensitive to timing misalignment. To study this effect, three experiments are conducted.

In the first experiment the research platform generates an 2-FSK/S reference signal with the same mark and space frequencies as a FSK Alt. Wide/R signal (*i.e.* 1915 Hz and 2085 Hz respectively). The HF modem generates the FSK Alt. Wide/R signal (see Table 10.1), which is transmitted via groundwave to the receiver. Both signals carry messages consisting of random bits but, the message on the FSK Alt. Wide/R signal

[37] Recall that Hamming distance is the bit-wise difference between binary sequences.

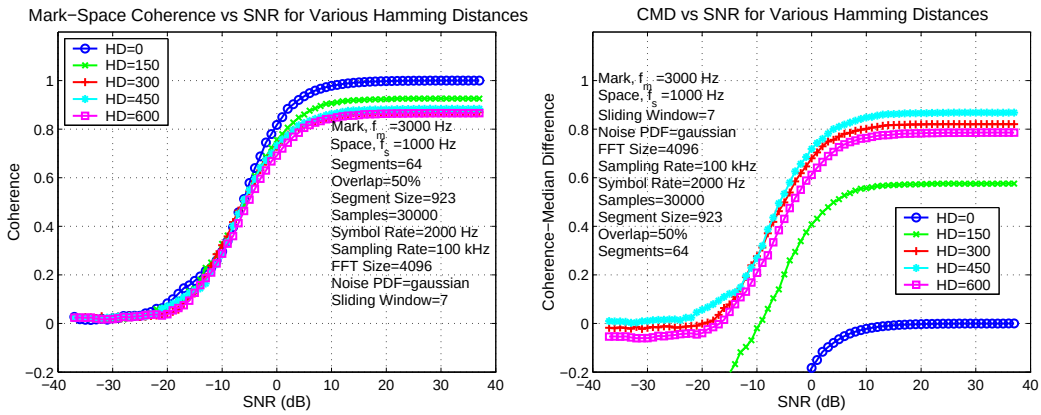

Figure 12.10. Coherence versus SNR at various Hamming distances. Average coherence $(\gamma^2(f_m)/2 + \gamma^2(f_s)/2)$ at the mark and space frequencies versus SNR (*left*) and CMD at the mark and space frequencies versus SNR (*right*) for various Hamming distances. Gaussian noise is bandlimited to 3 kHz around $f_c = f_m/2 + f_s/2$. As Hamming distance increases coherence decreases at positive SNR. At negative SNR coherence decreases to zero because the 2-FSK/S 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 tends to zero because the average mark-space coherence is the same as the median coherence for high SNR. A 7-element sliding boxcar window is used to smooth the results.

has structure defined by Mil-Std-188-110A (U.S. Dept. of Defense 1991) whereas the reference signal does not have any structure. Numerous trials are conducted whereby in each trial the random message of the 2-FSK/S signal changes and the coherence is estimated between the 2-FSK/S signal and the FSK Alt. Wide/R signal.

As can be seen in Figures 12.11 and 12.12, the power-spectral-densities of the reference signal and the real signal are similar but not identical. Therefore, it is no surprise that the coherence in Figures 12.13 and 12.14 show no significant peaks at the mark and space frequencies. It so happens that Trial 36 in Figure 12.13 yields the highest coherence. Clearly in this trial the synthesized FSK signal more closely matches the real signal than do the other trials. And, even though there is a wall of peaks at about 2 kHz, the generally low coherence is less than ideal. The coherence needs to be significantly higher (*i.e.* much closer to unity) in order to be definitive about the classification of a signal based on the coherence.

Consequently, this experiment verifies statements made earlier, that the coherence will only reliably yield a *yes/no* answer as to whether or not the received signal is identical to

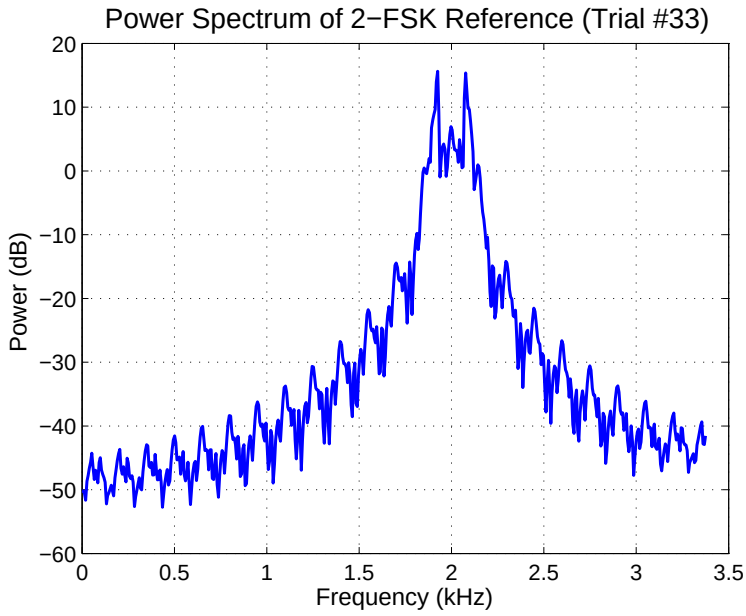

Figure 12.11. A synthetic 2-FSK signal. The synthetic 2-FSK signal is used as a baseband reference with symbol frequencies identical to those in Figure 12.12. The message carried by the 2-FSK/S signal consists of random bits selected from a uniform distribution. The peakedness in the spectrum is an artifact of the random message.

a reference signal. If the reference signal is changed to an ideal FSK Alt. Wide/S signal, then perhaps the coherence will be greater. This however does raise the question that perhaps the coherence would be high if a receiver were to maintain a local copy of a such a signal instead of a synthetic 2-FSK signal.

So, what if the signal reference is a FSK Alt. Wide/R signal instead of a synthetic 2-FSK signal? Assume the signal in Figure 12.12 is the baseband reference signal and the signal in Figure 12.15 is the noisy received signal. The same HF modem produces both FSK Alt. Wide/R groundwave signals but at different times, so that they can be sensed by the same antenna of the L-shaped antenna array. Furthermore, each signal carries a message from a common 511-bit pseudo-random sequence. However, the receiver will see two different messages because the signals are received at different time instants (*i.e.* different points in the 511-bit sequence). Since the messages originate in the same sequence, the power spectra of the two signals are nearly the same and consequently one would expect the coherence to be high. But, contrary to what we might naively expect, the coherence in Figure 12.16 is very low.

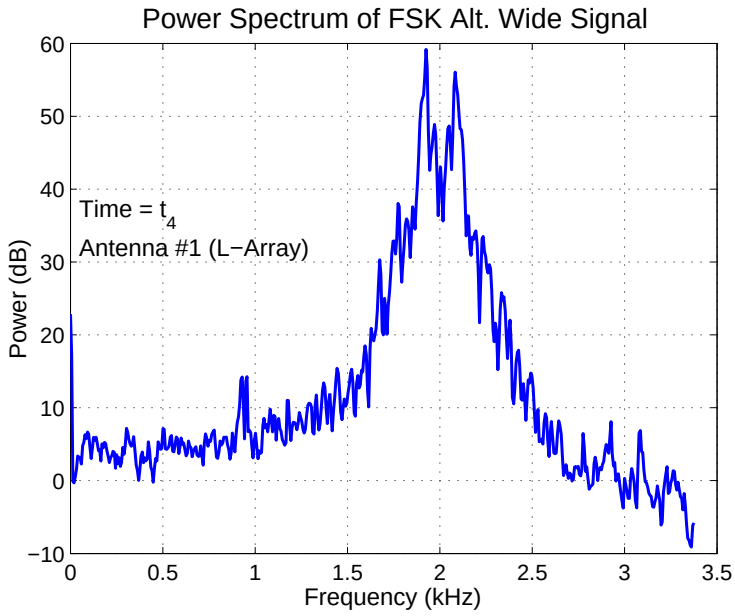

Figure 12.12. Power spectrum of an FSK Alt. Wide/R signal. The FSK Alt. Wide/R groundwave signal carries a message from a 511-bit pseudo-random sequence that is structured according to Mil-Std-188-110A (U.S. Dept. of Defense 1991). Antenna #1 of the L-shaped array is used to acquire this signal. The subscript on the time variable denotes the n^{th} recording in a sequence of m recordings for the particular experiment. Note the mixer products at about 1 kHz and 3 kHz that suggest inadequate suppression of side-lobes in the transmitter or receiver—neither of these issues are important in the current discussion. The measure of power is uncalibrated in this figure.

Near the mark and space frequencies (1915 Hz and 2085 Hz respectively) the coherence is lower than in Figure 12.14! However, careful inspection of Figures 12.13 and 12.14 shows that in general the synthesized reference provides no better a result. Therefore, the message plays an important role in the coherence, despite the similarities of the spectra. The greater the correlation between the messages of the two signals the larger the coherence. But, since the HF modem at the transmitter generates different random messages at different times the resulting coherence is low.

Now consider the case where the signal reference and the received signals are captured at the same time but on different antennas (see Figures 12.12 and 12.17). Because the signals carry the same message and timing the coherence in Figure 12.18 is unity in the passband of the signal. The coherence rolls off to something much less than one at frequencies outside the passband. This makes intuitive sense because there are no signals

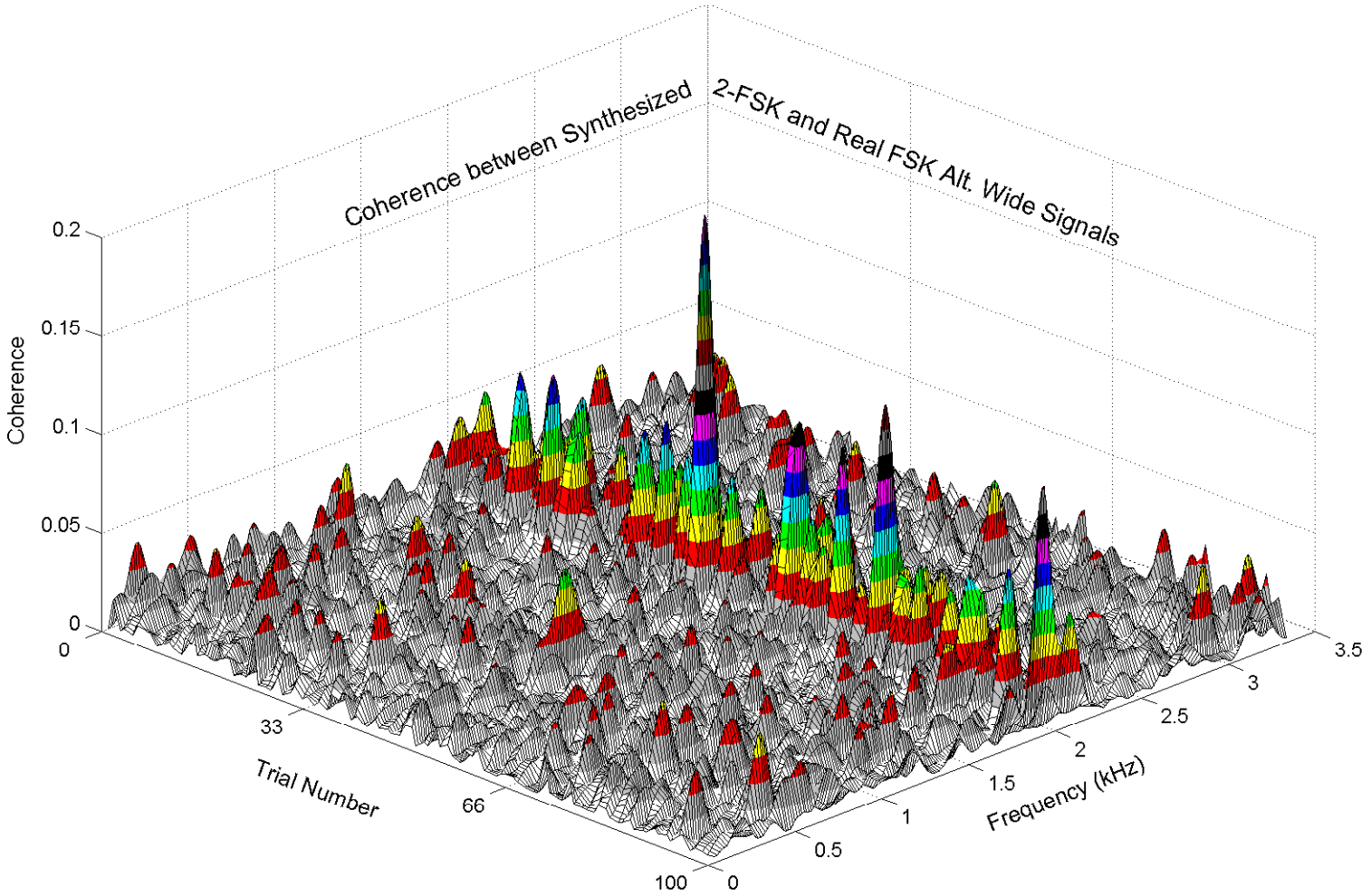

Figure 12.13. Coherence of 2-FSK/S & FSK Alt. Wide/R signals. The maximum coherence between 2-FSK/S and FSK Alt. Wide/R occurs about Trial 36. Note the valleys between the peaks along the wall at 2 kHz. For these trials the messages in the synthetic 2-FSK signal do not match the message in the real FSK Alt. Wide signal particularly well. The trials yielding relatively high peaks indicate better correlation of the messages.

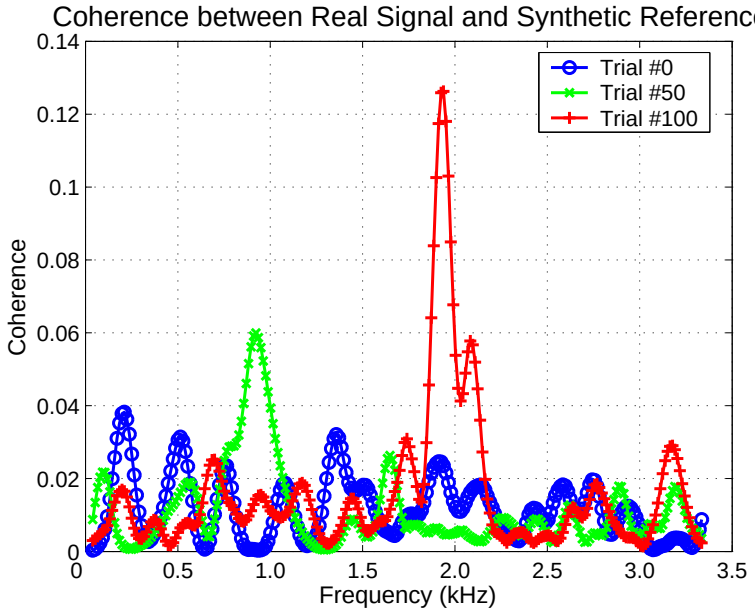

Figure 12.14. Coherence of 2-FSK/S & FSK Alt. Wide/R signals for various trials. Coherence between the FSK Alt. Wide/R signal and the synthetic 2-FSK reference for various trials. Trial 100 appears to match the real signal at the mark and space frequencies (1915 Hz and 2085 Hz respectively) more closely than Trials 0 and 50. Compare these results with Figure 12.13.

outside the passband or, there are incoherent near-field signals outside the passband, which allows uncorrelated noise to dominate.

In the first experiment a 2-FSK/S signal is compared against a FSK Alt. Wide/R signal in the coherence calculation, which results in a low coherence. One would expect that comparing two FSK Alt. Wide/R signals would produce a better coherence result. However as shown above, this is not necessarily the case. Numerous trials are necessary to discover an appropriate arrangement of bits in the FSK Alt. Wide/R reference signal that will yield the highest coherence with the noisy received signal. This trial-and-error method is processing intensive, especially when trying to try to answer the question: how many trials are needed before we can be satisfied with the coherence results? For real-time software radio systems such a method is unsatisfactory. At best, using coherence to identify an unknown signal is a *brute-force* method suitable only for non-real-time analysis. Consequently, the use of coherence to identify an unknown real signal is not practical, at least, not in isolation of other parameters.

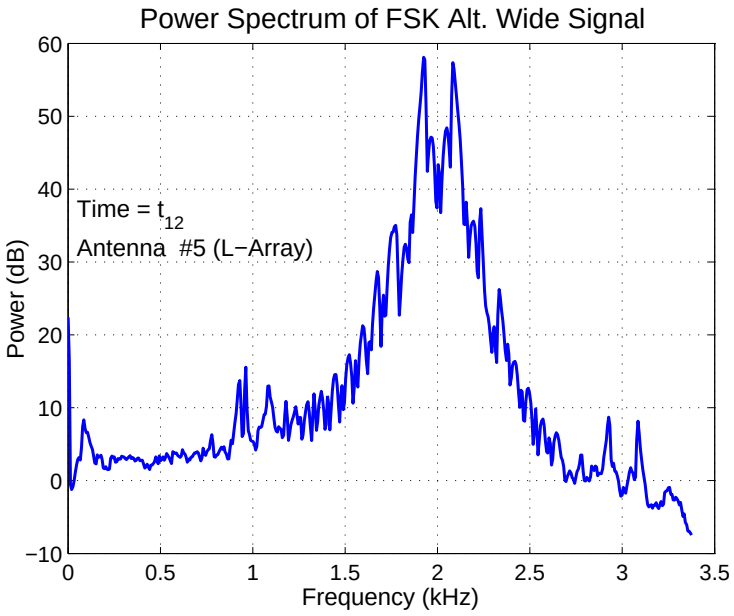

Figure 12.15. Power spectrum of another FSK Alt. Wide/R signal. Shown is the power spectrum of an FSK Alt. Wide/R signal. Comparing this spectrum with that of Figure 12.12 discloses only minor variations. In fact, the signal above is sensed by the same antenna but at a different time than the signal in Figure 12.12. The message in this signal and the message in the signal of Figure 12.12 are from the same 511-bit pseudo-random sequence. At the receiver each signal appears to have a different message because the receiver detects each signal at a different time. Antenna #5 of the L-shaped array is used to acquire this signal. The subscript on the time variable denotes the n^{th} recording in a sequence of m recordings for the particular experiment. The measure of power is uncalibrated in this figure.

The preceding analysis deals only with FSK type signals. One may argue that the findings of the analysis may improve if a different modulation is chosen. Indeed, that may be the case. Therefore replace the FSK signals with Stanag 4285 (*i.e.* an 8-PSK modulation) signals and repeat the tests to see if modulation plays a role in the timing sensitivity of the coherence measure.

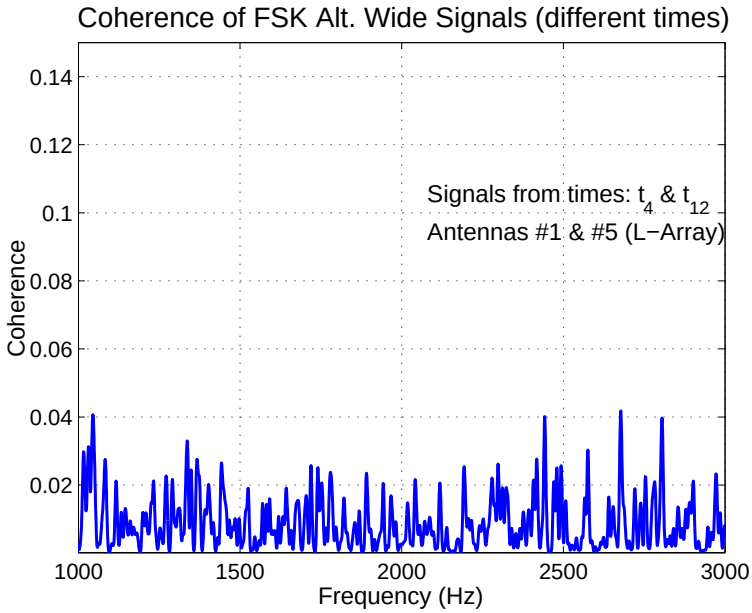

Figure 12.16. Coherence of two FSK Alt. Wide/R signals received at different times (same antenna). The coherence between the signals of Figures 12.12 and 12.15 is very low at the mark and space frequencies. The low coherence is primarily a result of the uncorrelated messages carried by each signal. Other factors contributing to the low coherence is independent noise and interference. Antennas #1 and #5 of the L-shaped array are used to acquire the signals for the coherence calculation. The subscripts on the time variables denote the n^{th} recordings in a sequence of m recordings for the particular experiment.

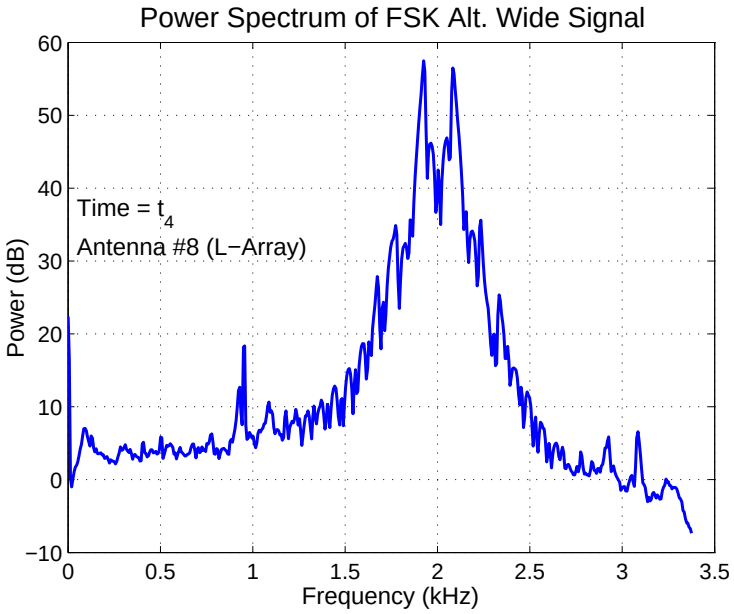

Figure 12.17. Power spectrum of another FSK Alt. Wide/R signal (different antenna). Shown is the power spectrum of a FSK Alt. Wide/R signal received on antenna different from the antenna used to acquire the signal in Figure 12.12. In this instance, the signal above and that in Figure 12.12 are acquired at the same time and carry the same message. Careful inspection of the two signals reveals only slight differences. These differences are due to time delay (hence phase difference) between the two signals as well as noise and interference from common and independent sources. Antenna #8 of the L-shaped array is used to acquire the signal. The subscript on the time variable denotes the n^{th} recordings in a sequence of m recordings for the particular experiment. The measure of power is uncalibrated in this figure.

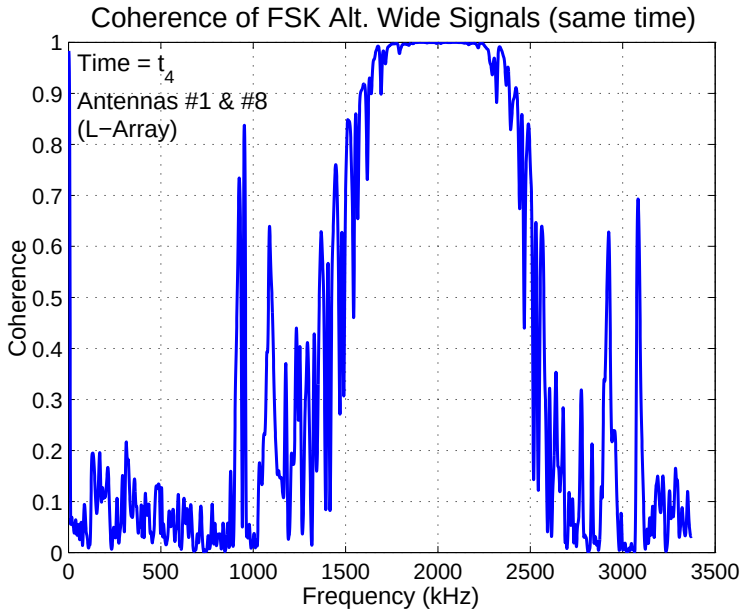

Figure 12.18. Coherence of two FSK Alt. Wide/R signals received at the same time (different antennas). The coherence of two FSK Alt. Wide/R signals (see Figures 12.12 and 12.17) received on two separate antennas but at the same time. Note the unity coherence in the passband (approx. 1900 Hz to 2100 Hz). This high coherence reflects the high correlation between the two signals. Outside the passband the coherence drops rapidly to near zero. The low coherence outside the passband is due to the independent components of the two signals at these frequencies. Antennas #1 and #8 of the L-shaped array are used to acquire the signals for the coherence calculation. The subscript on the time variable denotes the n^{th} recording in a sequence of m recordings for the particular experiment.

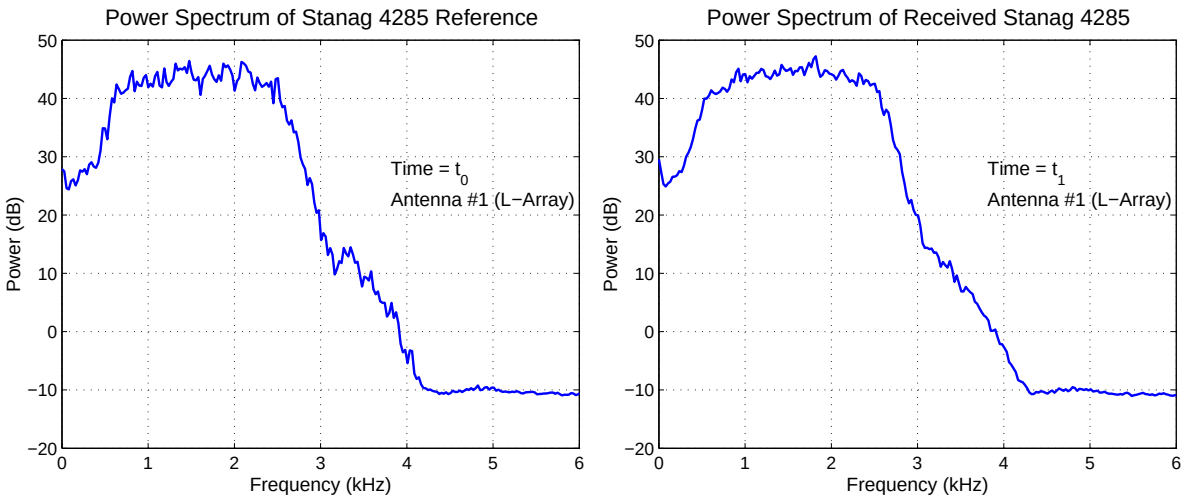

Figure 12.19. Reference and received Stanag 4285 signals (same antenna). The reference Stanag 4285/R signal (*left*) corresponds to $X(t)$ in Figure 10.1. The Stanag 4285/R signal at *right* is the received signal, $Y(t)$, in Figure 10.1. The spectra are similar and one would expect the coherence to be high. Both signals are sensed by the same antenna but at different time instants. The message in each signal is from the same 511-bit pseudo-random sequence encoded at 75 baud. At the receiver each signal appears to have a different message because the receiver detects each signal at a different time. Antenna #1 of the L-shaped array is used to acquire the signals for the coherence calculation. The subscripts on the time variables are arbitrary and serve only to indicate relative time instants for the particular experiment. Measures of power in both spectra are uncalibrated with respect to the field strengths at the antenna.

As above, consider the coherence between 1) two real Stanag 4285 signals acquired at different times via the same antenna; 2) two real Stanag 4285 signals received at the same time but through different antennas. In both cases the signals are transmitted from the HF modem with the same parameters—75 baud, long interleaving, and a 511-bit pseudo-random sequence . The similarities of these two signals (see Figure 12.19) suggest a high coherence.

And again, Figure 12.20 shows that the coherence is very low. From the point of view of the receiver, this is because both signals carry a different random message. Both signals are from the same 511-bit pseudo-random sequence, but sampled at different times. The effect is that the receiver sees two different messages. One consolation is the relatively high coherence at DC (see circled point in Figure 12.20). However, this is not unexpected as no message information is carried by the DC component and therefore the effect of timing misalignment is not as significant as at other frequencies.

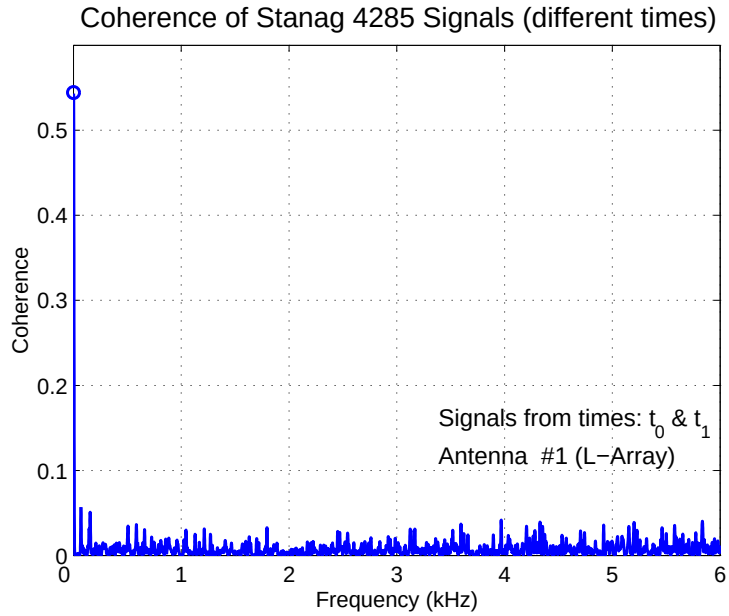

Figure 12.20. Coherence of two Stanag 4285 signals (same antenna). The coherence of similar Stanag 4285 signals is low because the signals are not synchronized. That is, the messages are not aligned in a bit-wise fashion. There is a relatively high coherence at DC (*circled*). This is not unexpected because no information (other than the mean signal level) is carried at DC and therefore the effect of timing misalignment is not as significant as at other frequencies. Antenna #1 of the L-shaped array is used to acquire the signals for the coherence calculation. The subscripts on the time variables are arbitrary and serve only to indicate relative time instants for the particular experiment.

Now let the reference be a Stanag 4285 sensed at one antenna and the noisy received signal the same Stanag 4285 signal sensed at another antenna of the same L-shaped array. And, if both signals are acquired at the same time, as shown in Figure 12.21, then the coherence of two such signals should be similar to that in Figure 12.18. Indeed, the coherence does similarly behave. Figure 12.22 shows that the coherence is unity in the passband of the signals and is less than one outside the passband. The high coherence reflects the high correlation between the messages of the two signals. The low coherence outside the passband is due to the independent and uncorrelated components of the two signals at these frequencies.

So, it is seen that the move to a different digital modulation scheme—in this case moving from FSK Alt. Wide to Stanag 4285 (an 8-PSK signal)—does not change the conclusion that coherence provides a *yes/no* answer to the question: is signal $Y(t)$ identical to signal $X(t)$? The ultimate goal is the development of a robust modulation recognition

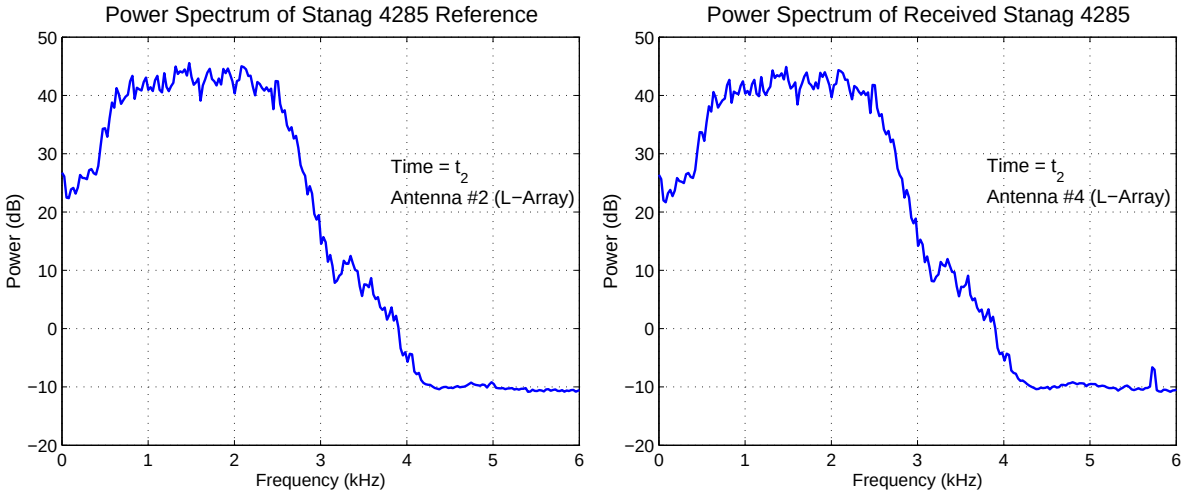

Figure 12.21. Reference and received Stanag 4285 signals (different antennas). The reference Stanag 4285/R signal (*left*) corresponds to $X(t)$ in Figure 10.1. The Stanag 4285/R signal at *right* is the received signal, $Y(t)$, in Figure 10.1. The spectra are similar and one would expect the coherence to be high. Both signals are sensed by by different antennas at the same time. The message is the same in each signal and is from a 511-bit pseudo-random sequence encoded at 75 baud. Antennas #2 and #4 of the L-shaped array are used to acquire the signals for the coherence calculation. The subscripts on the time variables are arbitrary and serve only to indicate relative time instants for the particular experiment. Measures of power in both spectra are uncalibrated with respect to the field strengths at the antenna.

method that, though unlikely, can identify numerous common modulation types. For HF communications, m-ary FSK and m-ary PSK modulations are extremely popular. Therefore, if coherence cannot be successfully used to recognize even the most basic of modulation types it is of little use.

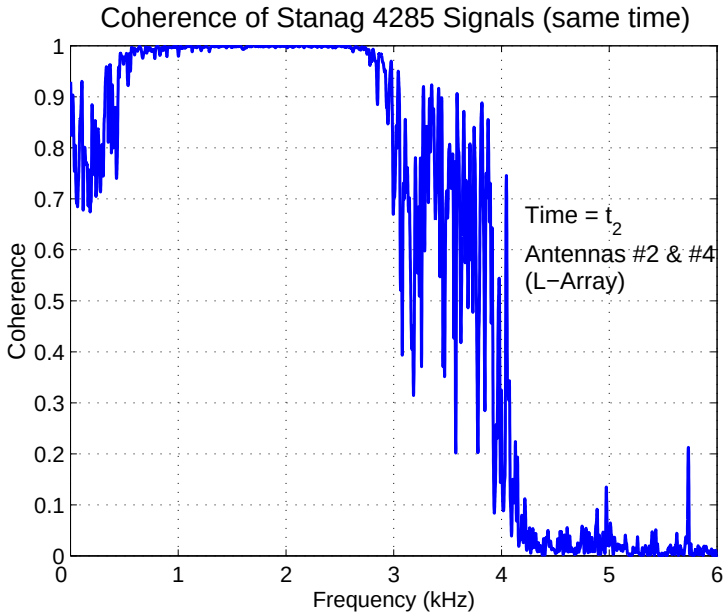

Figure 12.22. Coherence of two Stanag 4285 signals (different antennas). The coherence of two nearly identical Stanag 4285 signals is high in the passband of the signals and low elsewhere. This high coherence reflects the high correlation between the two signals. Outside the passband the coherence drops rapidly to near zero. The low coherence outside the passband is due to the independent and uncorrelated components of the two signals at these frequencies. Antennas #2 and #4 of the L-shaped array are used to acquire the signals for the coherence calculation. The subscripts on the time variables are arbitrary and serve only to indicate relative time instants for the particular experiment.

12.3 Entropy Results

Chapter 10 describes entropy of an information source. It also presents two ways of calculating entropy. However, Chapter 10 does not explain how this signal feature is useful. A calculation of entropy is typically applied to symbols (*e.g.* characters or bits) to determine a measure of the information content of those symbols. Shannon (1948) applies entropy to the channel capacity problem, and Benedetto *et al* (2002) applies entropy to written characters and language. There is no reason why this same entropy measure cannot be applied to samples from a digital receiver. Indeed, that is the reason for investigating entropy as a signal feature.

In the ensuing discussion of entropy, the digital receiver of Chapter 11 is regarded as an information source providing an arrangement of symbols from a particular alphabet. The arrangement of symbols is directly related to the perturbations of the electric field present at the inputs to the digital receiver, that is the antennas. The size of the alphabet depends on the dynamic range of the digital receiver. Typically, this dynamic range is evident by the number of bits in each digital sample output by the receiver; the number of bits being dependent on the resolution of the analog-to-digital (ADC) converter and subsequent digital downconverter (DDC). Each symbol in the alphabet is then determined by combinations of these bits in each fixed size sample.

In the context of Figure 10.1, the digital receiver therefore provides two information streams (both in a digital form): a downconverted signal, $Y(t)$, and a reference signal or alphabet, $X(t)$.

Effects of Compression Methods on Entropy

Two binary sequences A and B, with equal probability of a zero-bit, are created to compare the self-entropy as defined by Eq. (10.14) with that of Eq. (10.19). It is important to remember that in this experiment it is the self-entropy, Δ_{Ab} that is of interest and not the relative entropy, S_{AB}. The compression algorithm used for Eq. (10.19) is the LZW algorithm with 12-bit codes. The probability of a zero-bit is varied and plotted in Figure 12.23. In this case, the lengths of A and B are constant, while the length of the appended sequences vary. As can be seen, the length of b does not greatly affect Δ_{Ab}.

In fact, Δ_{Ab} greatly underestimates $H(\mathbf{X})$. At a probability of 0.5, Δ_{Ab} underestimates $H(\mathbf{X})$ by 70%!

Now if the length of b (hereafter designated by $|b|$ and similarly for a its length is $|a|$) is held constant and the length of the sequence (*i.e.* A and B) is varied, as in Figure 12.24, Δ_{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 sequence lengths of 100, $|b| = 25$, and $|a| = 25$ the entropy Δ_{Ab} approaches Shannon's entropy curve and at times surpasses it.

If Δ_{Ab} is plotted (see Figure 12.25) as a function of the length of the message and $|b|$—or $|a|$ and similarly throughout the discussion where the context dictates—it becomes clear that entropy is relatively flat for long messages and large $|b|$. Only when the

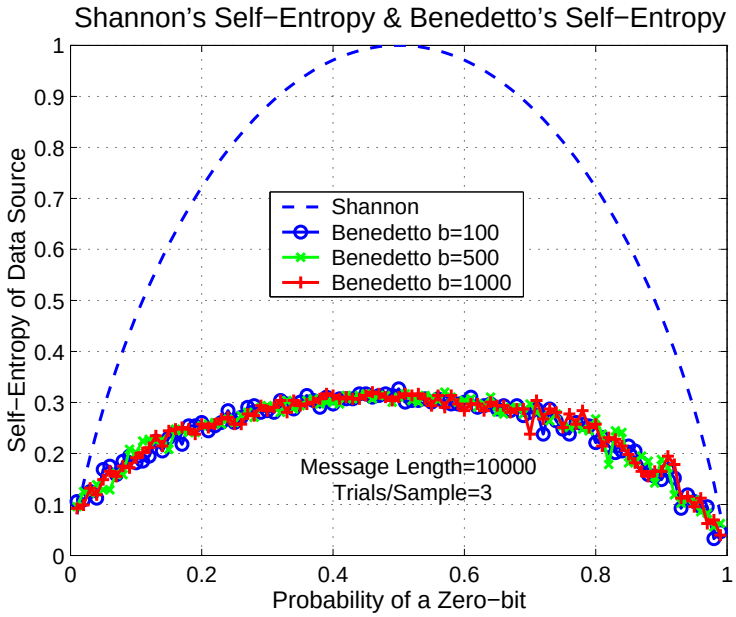

Figure 12.23. Shannon's entropy vs. Benedetto's entropy (12-bit LZW). Benedetto's method of calculating self-entropy, Δ_{Ab}, underestimates Shannon's entropy computed with Eq. (10.14) for various zero-bit probabilities and lengths of appended sequences. The message length is 10,000 bits with the probability 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.

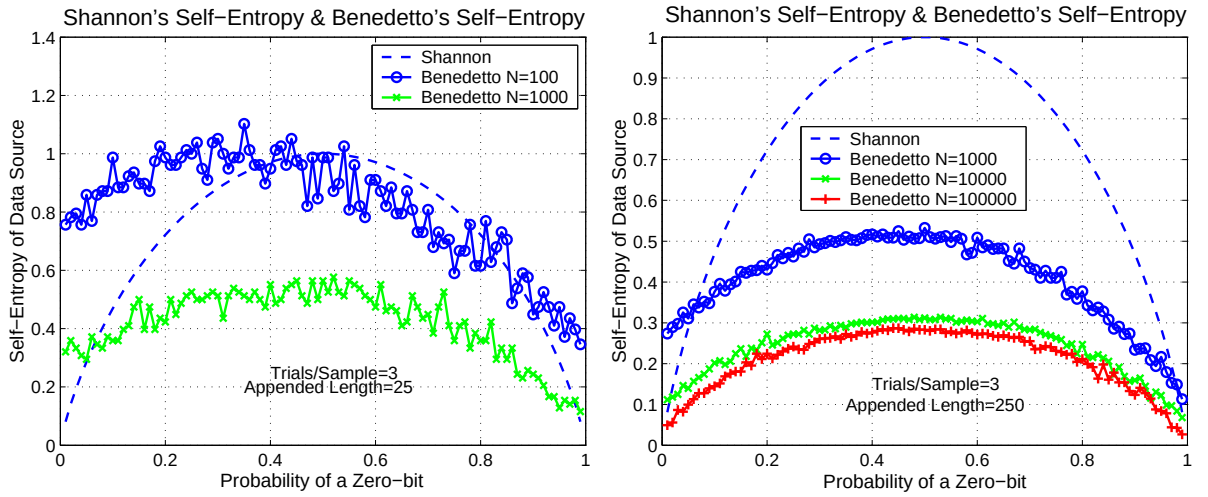

Figure 12.24. The effect of message length on Benedetto's self-entropy. Self-entropy, Δ_{Ab}, for various message lengths (N) and $|b| \in \{25, 250\}$. Self-entropy is computed in the same fashion as Figure 12.23 with the exception that the appended sequence is constant for varying message lengths, N. As the message length decreases the self-entropy curve via Benedetto's method approaches Shannon's entropy curve. The "roughness" of the curve increases as the appended length decreases (*c.f. left* against *right*). Skewing is due to the LZW compression algorithm.

message length decreases and $|b|$ decreases is there a "hill" in the entropy at the origin. In addition, the entropy appears slightly more sensitive to changes in $|b|$ as the message length decreases than for changes in message length as $|b|$ decreases.

In part, this behaviour 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 short $A + b$ sequences 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 a result of inefficiencies in the compression algorithm. If one considers the performance of the LZW algorithm as compared to another

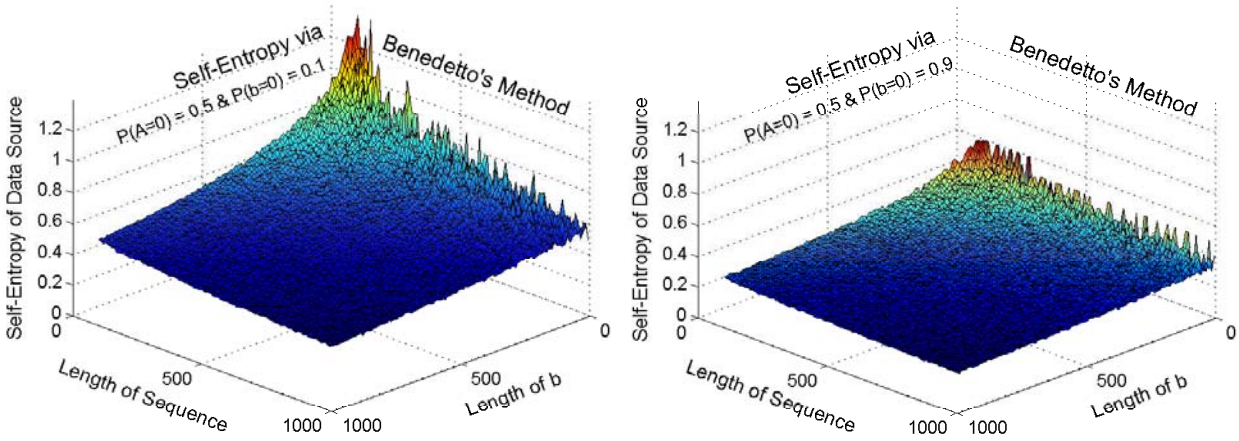

Figure 12.25. Effect of probability and message length on Benedetto's self-entropy. Surface plot of Benedetto's self-entropy, Δ_{Ab}, for various message lengths (N) and various $|b|$. In both diagrams the probability of a zero-bit in the sequence is $P(A = 0) = 0.5$. The probability of a zero-bit in the appended sequence is $P(b = 0) = 0.1$ (*left*) and $P(b = 0) = 0.9$ (*right*). The rise in the surface at the origin is due to the LZW compression algorithm.

compression algorithm, say the Zip 2.3 algorithm[38], it is apparent why the entropy rises at the origin in Figure 12.25. Figure 12.26 shows that the 12-bit LZW algorithm is generally less efficient than the Zip 2.3 algorithm and it is also less efficient[39] at compressing 1's than it is at compressing 0's. It is this skewing effect that yields the skewing in entropy seen previously. In fact, it is well known that the basis for the LZW algorithm is the LZ77 algorithm (Lempel & Ziv 1977). The LZ77 algorithm does not optimally encode a file, but the compression improves as the length of the file increases. The Zip 2.3 algorithm, when applied to Eq. (10.19), provides much more consistent measures of entropy. Figures 12.27 and 12.28 illustrate this.

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 Eq. (10.17) will compute identical entropies for S_1

[38] Zip 2.3 is based on the search algorithm of Rabin and Karp (see Sedgewick (1988)) and the compression algorithm of Fiala & Greene (1989).

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

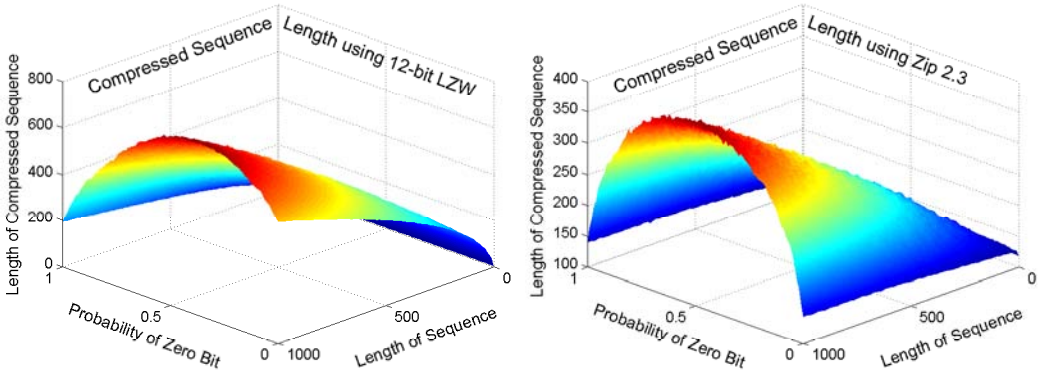

Figure 12.26. A comparison of 12-bit LZW and Zip 2.3 compression algorithms. Surface plots of 12-bit LZW (*left*) and Zip 2.3 (*right*) compression algorithms for various message lengths and zero-bit probabilities show that the LZW algorithm is not as efficient as the Zip 2.3 algorithm. In particular, the LZW algorithm is less efficient at compressing 1's that it is a compressing 0's.

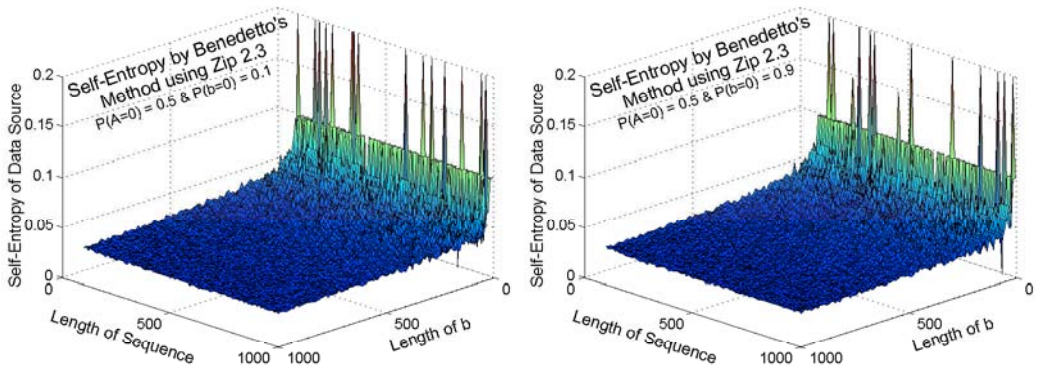

Figure 12.27. Benedetto's entropy with Zip 2.3 at various message lengths. Surface plot of self-entropy, Δ_{Ab}, using the Zip 2.3 compression algorithm, for various message lengths, N, and various $|b|$ or $|a|$. In both diagrams the probability of a zero-bit in the message is $P(A \cup B = 0) = 0.5$. The probability of a zero-bit in the appended sequence is $P(a \cup b = 0) \in \{0.1, 0.9\}$. The rise at low $|b|$ is consistent with the observation in Figure 12.24; the "spikes" are attributed to the randomness of the message.

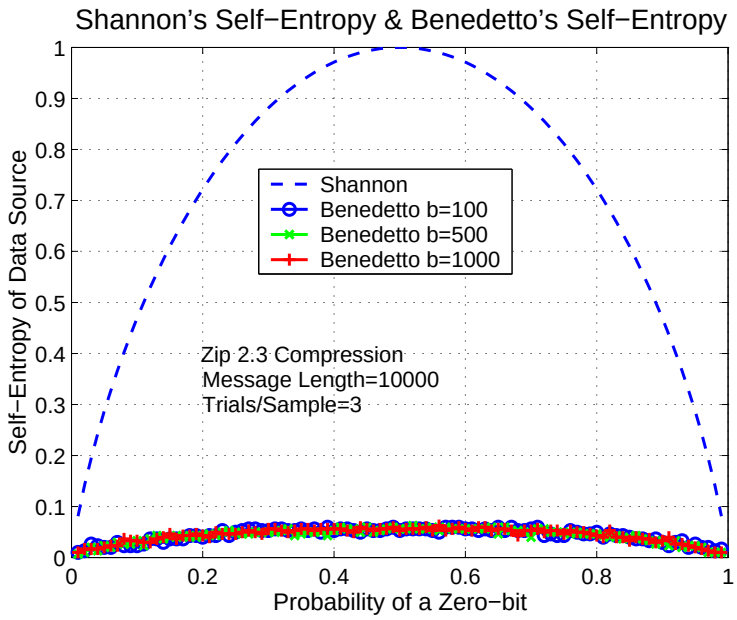

Figure 12.28. Shannon's entropy vs. Benedetto's entropy (Zip 2.3). Comparison of self-entropy computed with Benedetto's method (with Zip 2.3 compression) and Shannon's entropy formula. Compare this with that of Figure 12.23. Though the overall entropy is lower, the skewing present in Figure 12.23 is not present here.

and S_2 while Benedetto's method in Eq. (10.19) will not. Figure 12.29 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 Eq. (10.19) 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.

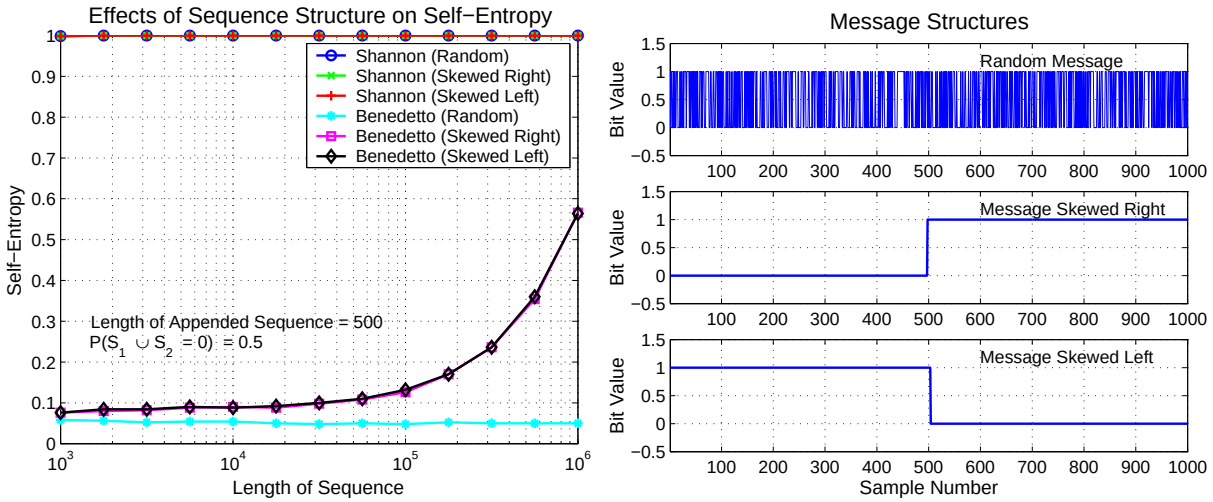

Figure 12.29. Effect of message structure on self-entropy. Self-entropy (*left*) as computed by Shannon's method and Benedetto's method for a message with a random arrangement of bits and a message with a structured arrangement of bits (*right*). Zip 2.3 compression is used for the calculation of entropy by Benedetto's method. Note that Shannon's method is not affected by the structure of the message, whereas for skewed messages Benedetto's method diverges from a constant. For a random arrangement of symbols, Benedetto's method yields a constant self-entropy, albeit underestimating the true self-entropy.

Effects of Quantizer Resolution on Entropy

The Q levels of the quantizer correspond to an alphabet of Q symbols[40] with a theoretical dynamic range of $20\log_{10} Q$. For example, if $Q = 2^n$ and $n = 16$ bits then the dynamic range is 96 dB. In practice, the dynamic range of a quantizer is less than this due to quantization error, clock jitter, and thermal noise. Ignoring this for the present discussion is acceptable. Consequently, one expects that if the received signal level stays the same and if the size of the alphabet increases then the total entropy should also increase. Why?

Assume that two information sources are available being, $\mathbb{A}$ and $\mathbb{B}$. Now recall that entropic distance Eq. (10.21) requires the compression of segments $A + a$, $A + b$, $B + a$, $B + b$, A, and B where a and b are small segments of $\mathbb{A}$ and $\mathbb{B}$ respectively. The sequence A is drawn from the information source $\mathbb{A}$ that corresponds to, $Y(t) = G[X(t)]$, the noisy and distorted received signal in Figure 10.1. The sequence B is drawn from the information source $\mathbb{B}$, corresponding to $X(t)$ in Figure 10.1, that represents a random

[40] Recall that alphabet symbols correspond to possible sample values output from the digital receiver.

arrangement of symbols from the Q-symbol alphabet. In this case, at least conceptually, $G(\cdot)$ provides a signal structure for $Y(t)$ given the random nature of $X(t)$.

As a typical compressor (*e.g.* LZW) sequentially processes $A + b$ it begins by creating a dictionary of symbols present in A. A majority of the symbols of A are present in B, because B is drawn from the entire Q-symbol alphabet. However, the symbols of A do not necessarily span the whole range of the alphabet. For example, a signal whose maximum amplitude is half the quantizer's full-scale range would only span half the symbols in the Q-symbol alphabet. Eventually, the compressor reaches the end of A in the $A + b$ sequence and begins compressing b. The data segment, b, may well span the entire Q-symbol alphabet since b came from $\mathbb{B}$. As a result, the compressor sees new symbols that are not already present in its dictionary formed on A and therefore creates new dictionary entries. This causes the efficiency of the compressor to decrease and a corresponding increase in the compressed length of $A + b$. A similar argument goes for $B + a$ but does not hold for $A + a$, $B + b$, A, or B because for these sequences the compressor dictionary consists of symbols entirely from $\mathbb{A}$ or $\mathbb{B}$. However, for $B + a$ the reduction in efficiency is marginal because the symbols in a are very likely already in B. So, if the received signal level stays the same and Q increases, the entropy should increase because B is drawn from a larger alphabet, which increases the likelihood that symbols exist in B that are not already in A.

Figure 12.30 does indeed reflect this logic by showing that for a synthetic Stanag 4285 signal the relative entropy measure, as computed by Eq. (10.19), increases as Q increases. To create this figure, a the research platform generates a Stanag 4285 signal with a symbol rate of 1200 bit/s and a sampling rate of 8192 Hz. The length of the appended sequences (*i.e.* a and b) is fixed at 500 symbols. Zip 2.3 compression is also used in Eq. (10.19).

As the quantizer resolution increases, the dynamic range of the digital receiver increases, and consequently so does the size of the alphabet. The dynamic range of the Stanag 4285/S signal, however, is constant. Therefore the compressor has more difficulty compressing the $A + b$ sequences since the range of values in A are only a subset of the range of values in b. Therefore it is no surprise that the relative entropy of the Stanag 4285/S signal (with a constant dynamic range) increases with Q.

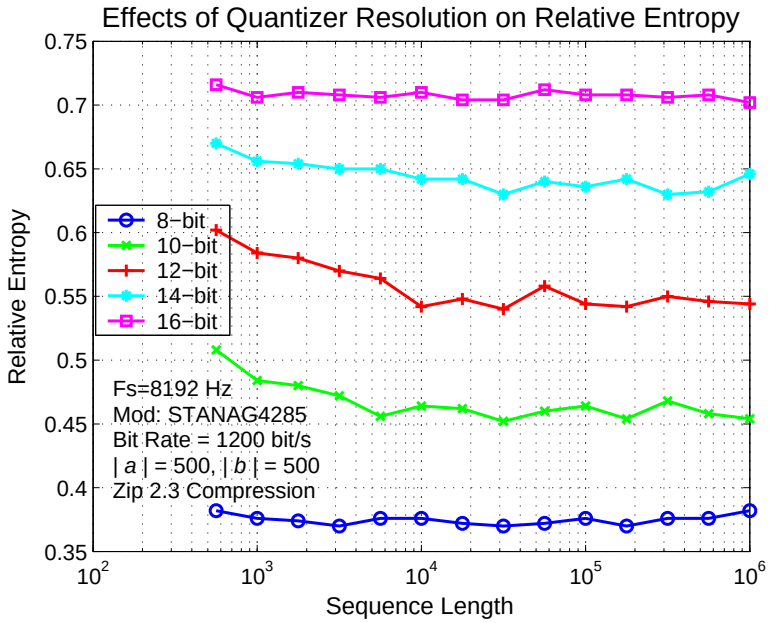

Figure 12.30. Effects of quantizer resolution on relative entropy. A change in the resolution of the quantizer (measured by Q) affects relative entropy. An increase in Q, while the dynamic range of the signal is constant, increases the relative entropy. The relative entropy increases because the compressor has more difficulty compressing $A + b$ sequences, since the range of values in A are only a subset of the range of values in b. In this instance A is a synthetic Stanag 4285 signal (*i.e.* an 8-PSK signal) carrying a random message at 1200 bit/s and B is a random selection of symbols from the Q-symbol alphabet. The lengths of a and b are constant at 500 symbols. Here A is digitized with a Q-level quantizer such that $Q = 2^n$ where $n \in \{8, 10, 12, 14, 16\}$ bits. Zip 2.3 compression is used.

Next, A is allowed to take on any of the synthetic signals in Table 10.1 and the effect of quantizer resolution (8-bits and 16-bits) on entropic distance is observed. In each case the reference signal, B, is a random selection of symbols from the Q-symbol alphabet. Figure 12.31 shows that Eq. (10.21) provides good separation of signals for sequence lengths greater than about 4,000 samples. The separation between modulations also increases with Q. For an 8-bit quantizer the alphabet is quite coarse compared to the 16-bit quantizer. Note that the entropic distance for 2-FSK/S briefly exceeds Shannon's entropy for the case where the quantizer has 16-bits. The other curves also have a peak at this point, and it shows that Benedetto's method of estimating entropy is not bounded by the limits set by Shannon. In the current discussion, the reasons for this

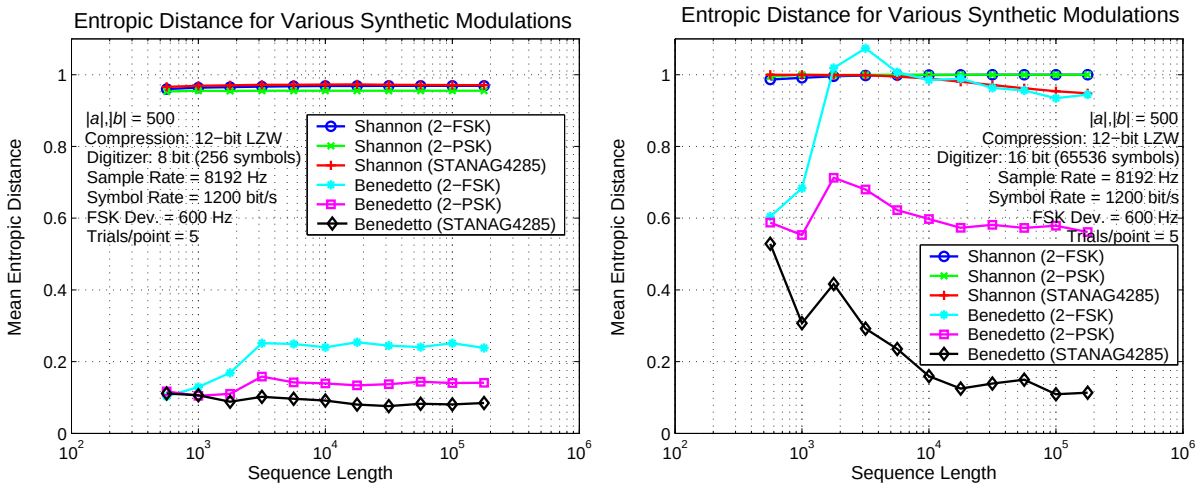

Figure 12.31. Effects of quantizer resolution on entropic distance. A change in the resolution of the quantizer (measured by Q) affects entropic distance. An increase in Q, while the dynamic range of the signal is constant, increases the entropic distance. The entropic distance increases because the compressor has more difficulty compressing $A+b$ sequences, since the range of values in A are only a subset of the range of values in b. Shown are the entropic distance of three synthetic signals represented by A with respect to B, a random selection of symbols from the Q-symbol alphabet. The synthetic signals represented by A are 2-FSK, 2-PSK, and Stanag 4285 (*i.e.* an 8-PSK signal) all of which carry a random message at 1200 bit/s. The length of the appended sequences (a and b) is constant at 500 samples. Here A is digitized with a Q-level quantizer such that $Q = 2^8$ (*left*) and $Q = 2^{16}$ (*right*). The compression algorithm is 12-bit LZW. Entropy as computed by Shannon's method is also included for reference; it provides no separation of signals yet, entropic distance does.

behaviour is largely irrelevant if such excursions provide greater separation between modulations.

An intuitive explanation for this peak is that just prior to the 4,000 sample mark, the entropic distance for both the 8-bit and 16-bit cases undergo a rapid increase in an environment where the number of samples in A is comparable to the number in b ($|b| = 500$); as a result Δ_{Ab} is greatly affected by the samples in b. On the other hand, Δ_{Ba} is relatively unaffected since symbols in a are, for the most part, included in B. Consequently Eq. (10.21), which is repeated here for convenience,

$$S_T \equiv \frac{\Delta_{Ab} - \Delta_{Bb}}{\Delta_{Bb}} + \frac{\Delta_{Ba} - \Delta_{Aa}}{\Delta_{Aa}}, \tag{12.2}$$

has a first term that is small and sensitive to slight changes in Δ_{Bb}. The second term is even smaller since the samples of a are likely entirely contained within B. However, as the length of A increases (the length of B identically increasing) a point is eventually reached whereby Δ_{Ab} becomes significant and as a result the first term in Eq. (12.2) grows rapidly. The second term in the equation will also grow, but not as quickly because a is almost certainly entirely included in B and Δ_{Aa} will remain relatively constant. This is demonstrated by a plot (Figure 12.32) of the self-entropies for the 2-FSK/S case.

In general for the 8-bit quantizer, $\Delta_{Bb} \approx \Delta_{Ba}$ since the symbols of a and b are already in B and, though not randomly arranged, the symbols in a have a decreasing effect on Δ_{Ba} as the length of B increases. However, $\Delta_{Ba} > \Delta_{Aa}$ since $B + a$ is based on the whole Q-symbol alphabet and the symbols are mostly unstructured (*i.e.* random), which makes the job of the compressor more difficult. Conversely $A + a$, possibly not drawn from

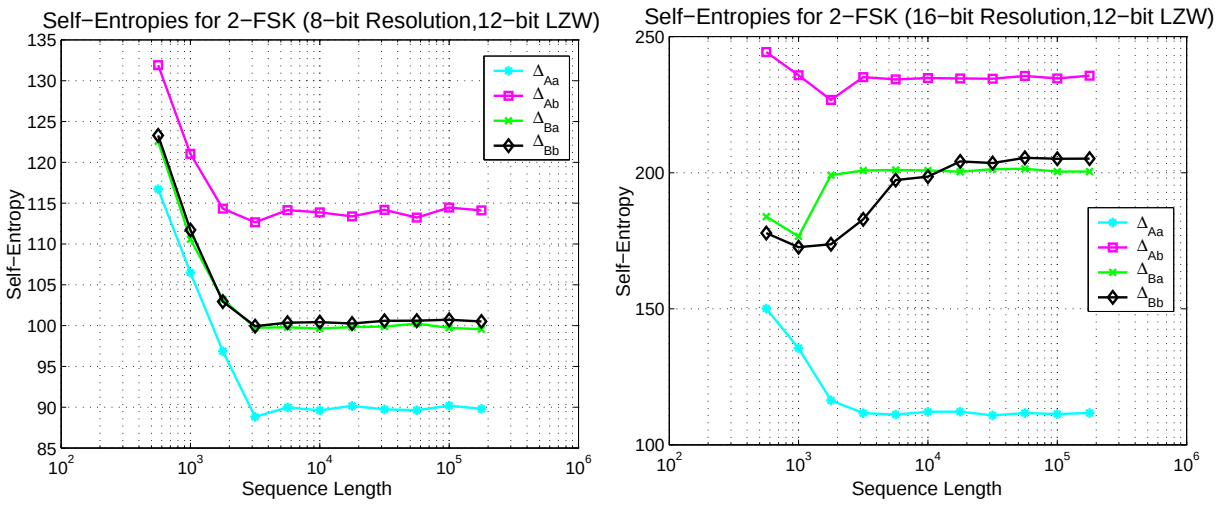

Figure 12.32. Self-entropies for 2-FSK/S with LZW compression. With 8-bit symbols (*left*), the self-entropies confirm that the first term in Eq. (12.2) is larger than the second term. The implication is that as the length of A increases (the length of B identically increasing) a point is eventually reached whereby Δ_{Ab} becomes significant and as a result the first term grows more rapidly then the second term. The *stagnation point* where the self-entropies transition from a non-zero slope to a constant level is related to the size of the codebook for the 12-bit LZW compression method. For a quantizer of 16-bits (*right*) the self-entropies have almost the same general order as the 8-bit case. The unusual upward trend of Δ_{Bb} and Δ_{Ba} is related to the size of the codewords versus the size of each symbol to be compressed; a topic to be discussed later.

the entire Q-symbol alphabet, is structured (*i.e.* A has a specific arrangement of symbols) and therefore the compressor is able to compress more efficiently as a result of the structure. Now $\Delta_{Ab} > \Delta_{Bb}$ since $A + b$ consists of a somewhat structured arrangement of symbols (possibly from a smaller range than B) to which is appended a sequence of random symbols from the entire range of B, which decreases the efficiency of the compressor to an extent that makes the self-entropy Δ_{Ab} greater than that of Δ_{Bb}. Of course Δ_{Bb} will be smaller than Δ_{Ab} since both B and b are drawn from the same set of symbols, yet the lack of structure in B and b does decrease compressor efficiency slightly more than the decrease in efficiency observed when compressing $A + a$.

For the 16-bit quantizer the self-entropies have almost the same general order as the 8-bit case but, there is an unusual upward trend in Δ_{Bb} and Δ_{Ba}. The aberrations of this scenario are related to the size of the compressor codewords, the size of the codeword dictionary, and the size of each symbol. Once the dictionary is exhausted, the entropic distance stagnates because the compressor has to compress new symbol strings with codewords from its limited dictionary. Later, it will be shown that the *stagnation point* for LZW compression moves with the size of the dictionary. This is not necessarily the case for other compression methods. More importantly, though, the number of bits in each sample is greater than the number of bits in each codeword. This too will be discussed later; the findings demonstrating that the codeword size must be larger than the symbol being compressed. For the moment ignore the 16-bit case. Then the summary of all of this is the following inequalities:

$$\Delta_{Ab} > \Delta_{Bb} > \Delta_{Aa} \text{ and } \Delta_{Bb} \approx \Delta_{Ba}, \tag{12.3}$$

$$\frac{\Delta_{Ab}}{\Delta_{Bb}} > \frac{\Delta_{Ba}}{\Delta_{Aa}}. \tag{12.4}$$

Equation (12.4) is easily validated with values from the 8-bit case in Figure 12.32. Therefore the second term in Eq. (12.2) is not insignificant but nonetheless is smaller than the first term; a result that confirms the logic above. However, the results of the 16-bit scenario in Figure 12.32 are not as conciliatory.

So it can be seen that the resolution of the quantizer, or alternatively the size of the Q-symbol alphabet, affects the absolute value of the entropic distance. The larger Q

the larger the entropic distance for a signal with a constant dynamic range. Moreover, it is apparent that entropic distance can be used as a separator of modulations.

Figure 12.31 suggests that noiseless 2-FSK/S, 2-PSK/S, and Stanag 4285/S signals can be separated with the entropic distance measure. There is clear separation of signals when the quantizer resolution is 8 bits and even greater separation when the resolution increases to 16 bits. Again, the separation is a direct effect of the increase in the size of the Q-symbol alphabet while the dynamic range of the signals remain the same. Shannon's entropy is also shown in the figure and it is quite obvious that measure provides no separation at all. The next section addresses signal separation in more detail.

Entropic Distance for Real Signals

It can now be said that at least two factors are important for estimating the entropy of signals in an effort to separate them: the compression method, and the size of the alphabet from which samples of the signals are drawn. The compression should be lossless and must be well-understood for a proper interpretation of the entropy results. For most entropic distance calculations in this thesis LZW compression receives preferential treatment at the expense of Zip 2.3 compression. Occasionally, Zip 2.3 compression is useful for comparison purposes. Lastly, the size of the alphabet must be large enough to provide separation of signals, yet not so large that the compressor has difficulty encoding symbols of the alphabet.

The investigation now turns to the application of the entropic distance measure to real HF groundwaves; the aim being to determine whether or not entropic distance is a useful feature for separating modulations. The real signals in Table 10.1 are Stanag 4285, FSK Alt. Wide, FSK Wide, and FSK Narrow. The Stanag 4285 signal has an 8-PSK modulation with a frame structure and specialized encoding. The FSK signals are based on 2-FSK modulation and also have a frame structure and specialized encoding.

In a manner similar to the previous section, set $\mathbb{A}$ to each of the real HF signals and set $\mathbb{B}$ to a random arrangement of symbols from the Q-symbol alphabet. Then compute the entropic distance with 12-bit LZW compression and Zip 2.3 compression. The results are characteristically similar to those of Figure 12.31 but, Figure 12.33 shows

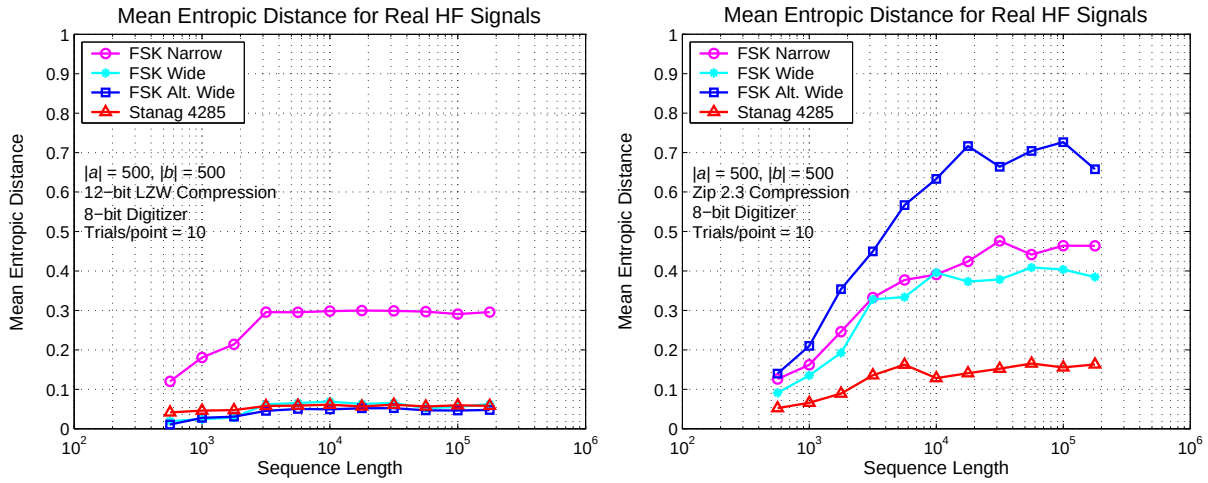

Figure 12.33. Entropic distance between the real signals of Table 10.1. Entropic distance of Stanag 4285/R, FSK Alt. Wide/R, FSK Wide/R, and FSK Narrow/R signals with respect to random sequences of the Q-symbol ($Q = 256$) alphabet for varying lengths of A and B. The length of a and b is constant at 500 symbols. Here $\mathbb{B}$ is a random selection of symbols from the Q-symbol alphabet and $\mathbb{A}$ is a real HF groundwave signal digitized with the equivalent of an $\approx$ 8.5-bit quantizer. Each point is the mean of ten (10) trials. In all cases the real signals carry a random message. Note that the entropic distance with LZW compression and a 256-symbol alphabet provides little separation (*left*), yet entropic distance with Zip 2.3 compression provides some useful separation (*right*). The standard deviation at each point is negligible for both plots.

that entropy computed using 12-bit LZW compression and an 8-bit alphabet provides little discernible separation of the signals (except for the FSK Narrow modulation). Entropy computed using Zip 2.3 compression provides better separation of the signals. One might suggest using a larger Q-symbol alphabet with 12-bit LZW compression but this does not necessarily improve the separation of signals.

Why does the entropic distance with LZW compression perform so poorly against real signals? The previous section shows that synthetic signals can be separated using the entropic distance measure with LZW compression. Actually the performance is not as bad as one might think. Consider Figure 12.31 again in the light of Figure 12.33. For synthetic Stanag 4285 and real Stanag 4285 the entropic distance is low. Only entropic distances of FSK Wide/R and FSK Alt. Wide/R signals have no match in Figure 12.31, whereas the entropic distance of the FSK Narrow/R signal seems to match that of the 2-FSK/S signal in Figure 12.31.

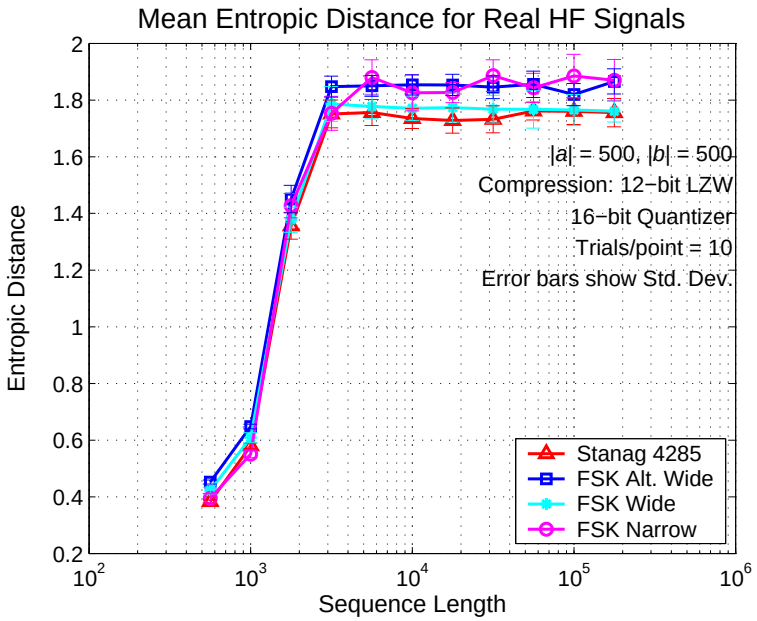

Figure 12.34. Entropic distance of real HF signals (16-bit quantizer). Increasing the quantizer resolution to 16 bits does not improve the separability of the modulation types. The samples of the real signals consist of 8-bits; but the large alphabet, from which the reference signal is drawn, contains 256 times as many symbols. Therefore, the differences between entropic distance measures are insignificant.

What if the quantizer resolution is increased? Will that improve the separation offered by the entropic distance with LZW compression? The previous section suggests that it will. Figure 12.34 contains the results of setting the quantizer resolution to 16 bits and recomputing the entropic distance of the real HF signals (again with a 12-bit LZW compressor).

Clearly there is no separation of signals. Indeed, the standard deviations at each group of points overlap. It must be remembered that in this experiment the real signals consist of 8-bit samples, whereas the reference signal is a random arrangement of 16-bit symbols from the entire alphabet. Therefore, since the alphabet is 256 times larger than the range of the samples offered by the signals, the differences between entropic distance measures are insignificant. Increasing the quantizer resolution for pure (*i.e.* noise-free and distortion-free) synthetic signals does provide greater separation. Here, however, the real signals are not pure. To the compressor these impure real signals appear closer to a random arrangement of symbols than do the pure synthetic

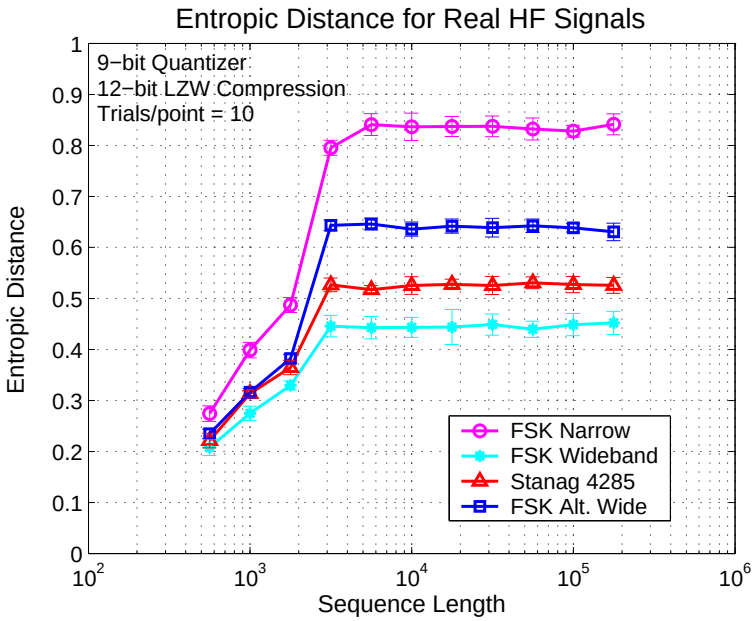

Figure 12.35. Entropic distance of real HF signals (9-bit quantizer). Increasing the quantizer resolution to 9 bits does improve the separability of the modulation types—the standard deviations do not overlap! In fact, the samples of the real signals consist of approximately 8.5-bits. The 9-bit alphabet is close to the sample size so the difference between the entropic distances of the signals are greatly amplified. It is now clear, that another condition for successful application of Benedetto's entropy to modulation recognition is that the alphabet size of the reference signal must approximate the range of the samples from the real signals.

signals. Therefore the increase in quantizer resolution does not, at least in this scenario, provide the expected improvement in separation.

Close inspection of the samples from the real signal dataset shows that, in general, they consist of more than 8 bits but less than 9 bits. This raises a question—what happens to the separation of the signals if the entropic distance measure works on 9-bit symbols?

Well, increasing the quantizer resolution to 9-bits definitely improves the separability of the signals. Note that the standard deviations do not overlap. The 9-bit alphabet, from which the reference is drawn, is comparable to the range of samples provided by the real signals. Consequently, the differences between the entropic distances are greatly amplified. It is now clear, that another condition for successful application of Benedetto's entropy to modulation recognition is that the alphabet size of the reference signal must be carefully chosen. The reason that the FSK Narrow/R signal separates

from the other real signals, for the case of the 8-bit quantizer, is that the samples from the signal's dataset do not span the entire 256-symbol alphabet.

We are now confronted with another question. For a given sample size (in bits) of a real signal, what is the optimal quantizer resolution for the reference signal (*i.e.* the random arrangement of symbols from the Q-symbol alphabet)? A plot of a "separation coefficient" versus resolution of the quantizer versus the resolution of the samples in the real signal would be useful. Perhaps the "separation-coefficient" might be as simple as the mean difference between entropic distances for all signals of interest normalized by the number of signal. Or, alternatively, the standard deviation of separation distances between all pairs of signals.

To determine an appropriate alphabet size, the mean separation distance (MSD) is defined as a measure of the "empty space" between curves of entropic distance. A large positive MSD indicates that the curves are sufficiently separable. A low or negative MSD indicates that the curves generally overlap and are not easily separable. Let S_{ij} be the random variable representing all trials at the j^{th} plotting point for the i^{th} entropy curve, and let $\vec{S}_i$ represent a vector of averages for the i^{th} curve such that $\vec{S}_i = [\mathrm{E}(S_{i1})\ \mathrm{E}(S_{i2})\ \mathrm{E}(S_{i3})\ \cdots\ \mathrm{E}(S_{ij})]$, where $\mathrm{E}(\cdot)$ is the expectation operator. Similarly, the standard deviation vector is, $\vec{\sigma}_i = [\sigma(S_{i1})\ \sigma(S_{i2})\ \sigma(S_{i3})\ \cdots\ \sigma(S_{ij})]$, where $\sigma(\cdot)$ is the standard deviation function. Then the mean separation distance is

$$\mathrm{MSD} = \mathrm{E}\left(\frac{2!(N-2)!}{N!}\sum_{i=0}^{N-2}\sum_{j=i+1}^{N-1}\|\vec{S}_i - \vec{S}_j\| - \|\vec{\sigma}_i - \vec{\sigma}_j\|\right), \tag{12.5}$$

where $\|\cdot\|$ is the Euclidean distance operator and N is the number of entropy curves being compared. The $\|\vec{S}_i - \vec{S}_j\|$ term measures the distance between the curves at specific sequence lengths. The term $\|\vec{\sigma}_i - \vec{\sigma}_j\|$ measures the total spread of each pair of curves at specific sequence lengths. Subtracting the two terms characterizes the "empty-space" between pairs of curves where individual trials are unlikely to overlap.

Equation Eq. (12.5) is plotted in Figure 12.36 and shows that a 512-symbol (9-bit quantizer) alphabet is near optimal for the real signal set in Table 10.1. To generate this plot, the entropic distance curves are recalculated for quantizer resolutions of 10, 11, 12, 13, 14, and 15 bits and the MSD is applied to each set of curves. The results in the figure show that larger or smaller alphabet sizes for $\mathbb{B}$ actually decrease the mean separation

distance between signals. Therefore, the most appropriate quantizer to use with the LZW algorithm is one that has a resolution similar to or slightly larger than the resolution of the real signal. This is necessary so that the dictionary in the LZW algorithm does not become so large that the effect of b goes unnoticed. However, note that the 8-bit alphabet also shows a relatively high MSD. This is a result of the large separation between the FSK Narrow/R signal (see Figure 12.33) and the other real signals, for which there is little separation. The negative values of the MSD infer that the mean distances between the entropic curves is less than the variation of the trials at each sequence length. In other words, negative MSD identifies a situation when the entropic distance curves overlap and where the signals cannot be separated. Negative values for the MSD are possible because the MSD is the difference of two squared values. The variance of the entropic distance estimate can be larger than the entropic distance itself. Clearly, the MSD cannot be used exclusively to choose an alphabet size but must be used in conjunction with an entropic distance plot. So in that regard the MSD cannot be used as a predictor. Instead it must be used as an indicator of suitability of fit of a particular resolution for the quantizer for a given data set. Hence, to optimally separate signals with the entropic distance feature the size of the alphabet must be the same or slightly larger than the span of the samples of the signals.

So far, entropic distance appears promising as a feature for separating modulation types. There is a slight problem with the preceding interpretation, which is found in the representation of the samples (or symbols) of the signals (synthetic and real). Each sample is a whole number, not an integer, in a range determined by the number of bits of resolution. For example, when it is said that the resolution of a real signal is 8.5 bits (see Figures 12.35 and 12.36) it really means that each whole numbered sample is confined to the range $\{-182.000\ldots, -181.000\ldots, -180.000\ldots, \ldots, +181.000\ldots\}$. Similarly, when a 9-bit resolution is present it implies that each whole numbered sample is from the finite range of $\{-256.000\ldots, -255.000\ldots, -254.000\ldots, \ldots, +255.000\ldots\}$. The key here is not the range, but the representation of the sample. Each sample or symbol is represented by an 8-byte floating point number. The difficulty with this is that each of the compression algorithms LZW and Zip 2.3 are word-based compressors. That is to say, the compressors work on m-bit words. Until now, the word size

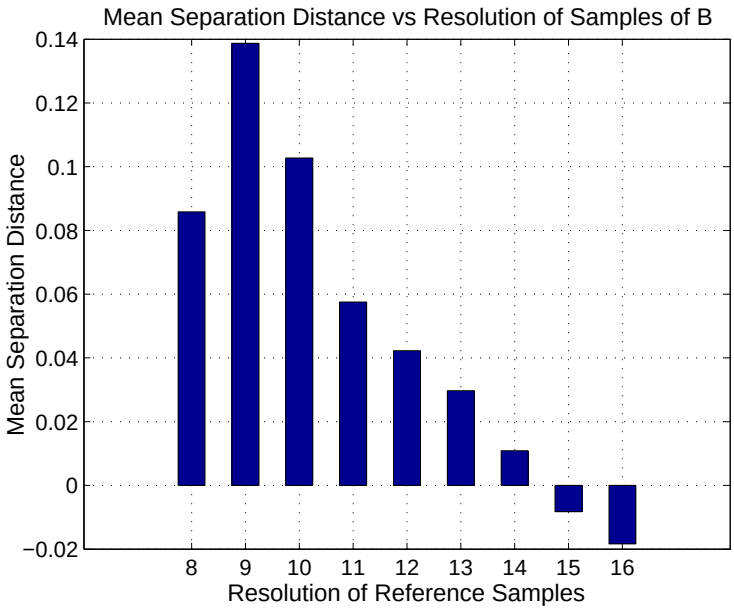

Figure 12.36. The MSD measures for various quantizer resolutions. Compared with Figure 12.33, entropic distance between the real HF signals (with $\approx$ 8.5-bit resolution) and random sequences with a 512-symbol alphabet ($Q = 512$) provide the optimum separation of signals. Clearly, increasingly large alphabets decrease the mean separation distance. Alphabets that are too small also decrease the MSD. Negative MSD values indicate that the entropic distance curves overlap and therefore separation of signals is not possible. Remember, that MSD is not mean-squared difference, but the mean separation distance. The mean separation distance is the difference of two squared values. Consequently, the MSD is allowed to be negative.

is 8 bits. Therefore for the self-entropy measure to truly represent the entropy of a sequence, each sample should fit within the word size of the compressor.

Yet, one may ask why the results show separation of signals. The answer is that in general the 8-byte floating point numbers consist mostly of zeros for the data sets. For example see Figure 12.37, where $-155.000\ldots$ in the 8-byte floating point representation of **MATLAB**® is hexadecimal \$C0 63 60 00 00 00 00 00, and $+120.000\ldots$ is hexadecimal \$40 5E 00 00 00 00 00 00. The strings of zero-bytes compress easily and the structure of each signal is maintained, at least approximately, in the most-significant bytes—typically bytes 6 to 8—of each double-precision sample. Separation of signals is, therefore, still observable. Nevertheless, the fact that each sample is spread across more than one word of the compressor does make the previous results somewhat dubious, and so the next section addresses this issue.

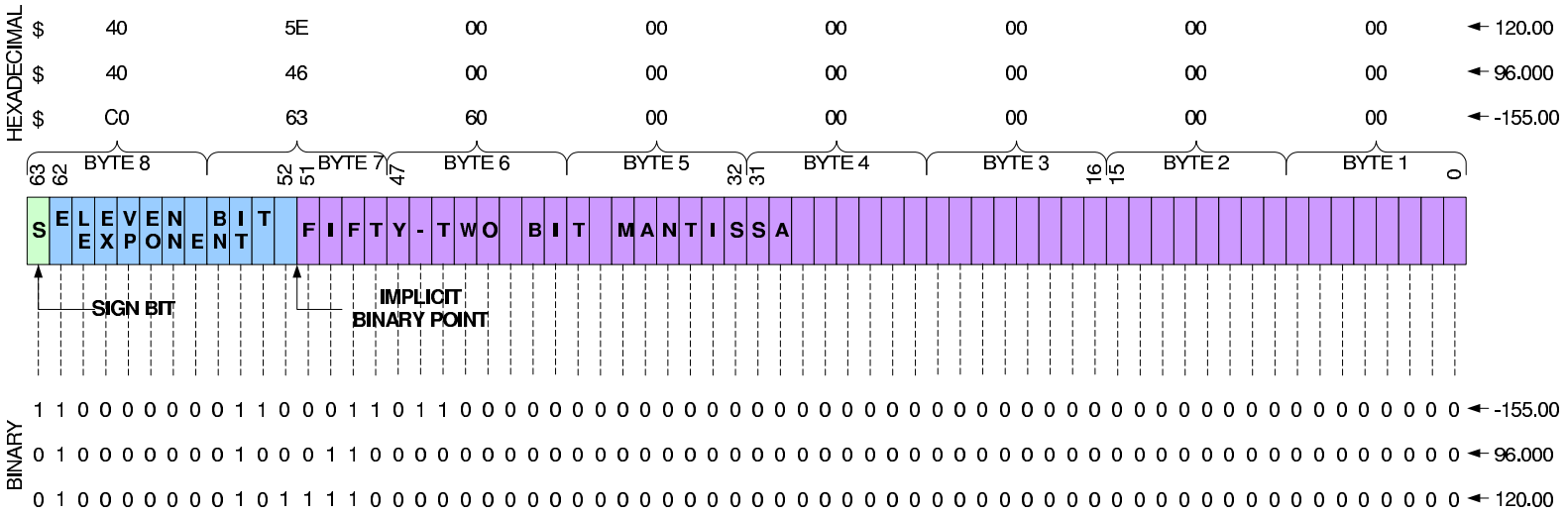

Figure 12.37. IEEE 754 double-precision floating-point representation. The samples of the synthetic and real signals are confined to whole numbers for the entropy calculations. The **MATLAB**® environment uses an IEEE 754 double-precision floating-point representation for numbers. Therefore, each sample consists mostly of zero-bytes. The zero-bytes compress easily, and so the entropy measures are affected more by the most-significant bytes (typically bytes 6, 7, and 8) of the floating point number than by least-significant bytes (typically bytes 1 to 5). The structure of each signal is maintained, at least approximately, in these most-significant bytes and therefore separation is still observed in earlier figures. Three examples are shown for decimal numbers 120.00, 96.000, and –155.00.

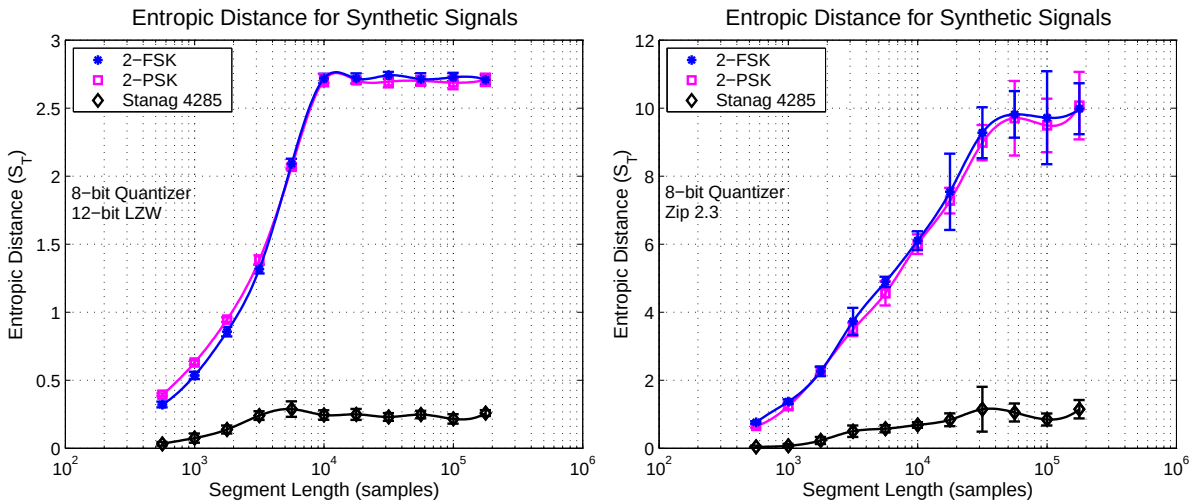

Figure 12.38. Entropic distances for synthetic HF signals (again). Entropic distance of various synthetic signals from Table 10.1 for varying segment lengths. Entropic distance is computed with 12-bit LZW compression (*left*) and Zip 2.3 compression (*right*). Error bars signify the standard deviation of the ten trials at each averaged data point. Both sets of curves are interpolated with a cubic spline. Neither show separation of 2-FSK/S and 2-PSK/S signals, yet the Stanag 4285/S signal is distinct. The Stanag 4285/S signal is a noise-like 8-PSK signal and is therefore closer, in an entropic distance sense, to the random arrangement of samples of $\mathbb{B}$.

Entropic Distance of Real Signals (Reprised)

This time each sample (or symbol) is converted to an m-bit word (*i.e.* an integer) in line with the word-size of each compressor. From this point forward all samples (or symbols) are m-bit integers; no longer is the 8-byte floating point representation used.

With this in mind, $\mathbb{A}$ is set to each of the synthetic signals, in turn, from Table 10.1 and the effect on entropic distance is observed. Figure 12.38 shows that Eq. (10.21), as computed with 12-bit LZW compression, does not discriminate 2-FSK/S and 2-PSK/S signals but does separate a Stanag 4285/S signal. Similar results are achieved when Zip 2.3 compression is used. These curious results can be explained in the following way.

The nature of the time-series waveforms of 2-FSK/S and 2-PSK/S signals include large swings from positive to negative (*i.e.* a high slew-rate) across the Q-symbol alphabet (see Figure 12.39). A histogram of the time-series data shows that a large proportion of the samples are at the positive and negative extremes of the signal. A Stanag 4285/S signal, on the other hand, appears more random in nature with smaller swings and an

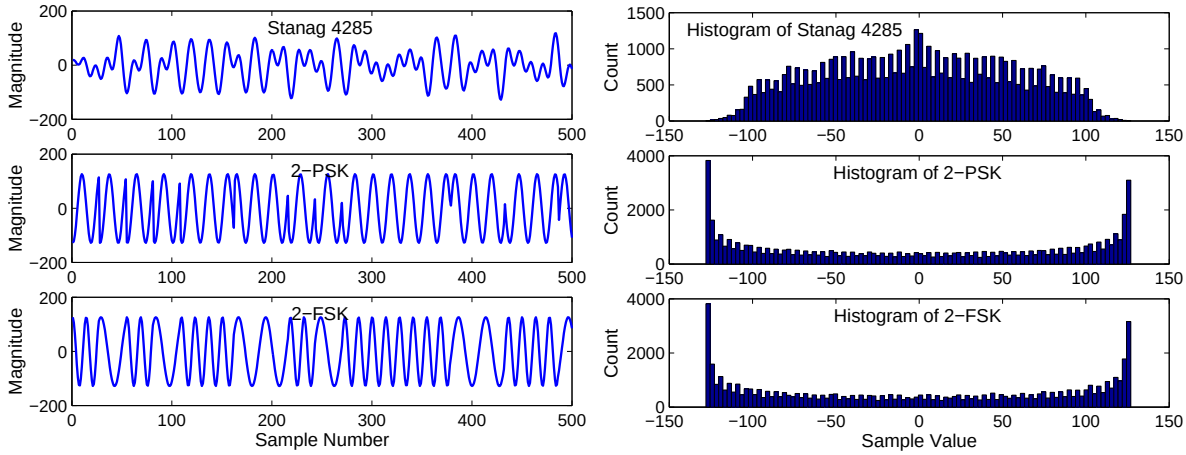

Figure 12.39. Histogram of time series data. The synthetic signals (*left*) consist of Stanag 4285, 2-PSK, and 2-FSK. Their histograms (*right*) reveal that Stanag 4285 has an approximately uniform distribution of samples, whereas there is high probability that samples will be at the extremes for 2-PSK and 2-FSK. There is no wonder, therefore, that Stanag 4285 is more like the random arrangement of symbols in the reference signal, $\mathbb{B}$, than the other signals. In this example the synthetic signals do not have added noise.

approximately uniform distribution of the samples across the range of the Q-symbol alphabet. The reference signal, $\mathbb{B}$, is a uniformly random arrangement of symbols from the alphabet. The entropic distance for the Stanag 4285/S signal is thus small because it is more closely related to the random arrangement of symbols in $\mathbb{B}$ than are the 2-FSK/S and 2-PSK/S signals. The 2-FSK/S and 2-PSK/S signals are dissimilar to $\mathbb{B}$ and hence the entropic distance is large.

Unlike the conclusion of the previous section, it now appears that entropic distance is not that useful. But, what happens if Gaussian noise[41] is added to each signal? Doing so, actually improves the separation of the signals (see Figure 12.40). Only a small noise perturbation (such that the SNR of the synthetic signals is 40 dB) causes significant separation of the signals for the case where 12-bit LZW compression is used. The improvement in separation is marginal for the Zip 2.3 case. Regardless, the resulting change in entropic distance is due to the noise immunity of the signals. It is well known that a 2-FSK signal is more susceptible to the detrimental effects of noise than is 2-PSK. In fact Couch II (1990) shows that the bit-error-rate (BER) performance of 2-FSK is 3 dB poorer than 2-PSK. Consequently, with added noise the 2-FSK/S signal tends

[41] Though not always an accurate model for HF noise, additive white-Gaussian noise is suitable for the present discussion.

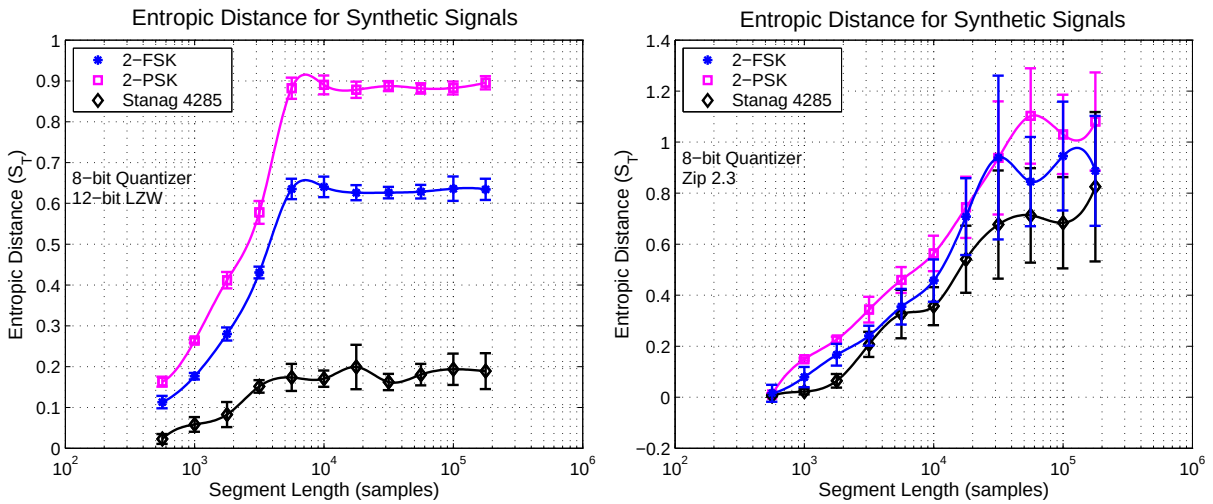

Figure 12.40. Entropic distances for synthetic HF signals with Gaussian noise. Entropic distance of various synthetic signals (with Gaussian noise) from Table 10.1 for varying segment lengths and an SNR of 40 dB. Entropic distance is computed with 12-bit LZW compression (*left*) and Zip 2.3 compression (*right*). Error bars signify the standard deviation of the ten trials at each averaged data point. In this instance, the entropic distance curves at *left* separate, but not particularly well at the *right*.

towards a random arrangement of symbols more quickly than does the 2-PSK/S signal and therefore the separation increases between the 2-FSK/S and 2-PSK/S signals. The entropic distance of the Stanag 4285/S signal does not change significantly, because it is already approximately random, and its noise immunity is even greater than that of 2-PSK (*c.f.* entropic distance of Stanag 4285 in Figures 12.38 and 12.40). The improvement in separation due to noise may, in practice, be useless if the SNR is unknown. To circumvent this problem the SNR can be estimated, but this is a discussion to be held later. For now assume that the SNR is known.

Now let $\mathbb{A}$ take on each of the real signals of Table 10.1 and let $\mathbb{B}$ be a uniformly random arrangement of symbols from the Q-symbol alphabet. Separation of signals is possible using entropic distance with 12-bit LZW compression (see Figure 12.41). Application of the entropic distance measurement method to real HF groundwaves results in characteristics similar to those of Figure 12.40. Figure 12.41 shows that entropy computed using 12-bit LZW compression and an 8-bit alphabet provides good separation between signals whereas entropy calculated with Zip 2.3 compression provides marginal separation.

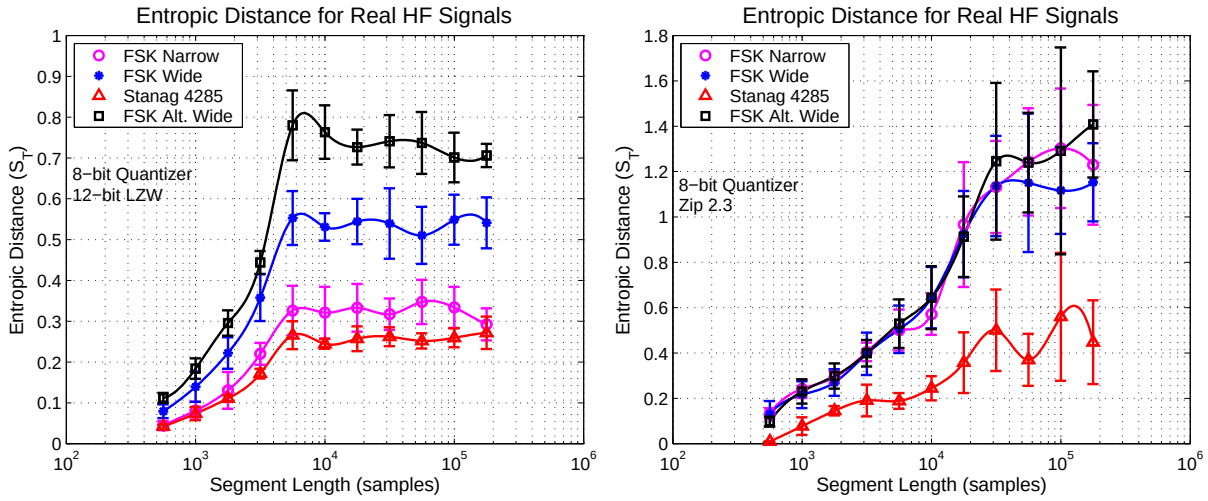

Figure 12.41. Entropic distances between real HF signals (reprised). Entropic distance between various real HF signals and random sequences of the Q-symbol ($Q = 256$) alphabet for varying segment lengths. Entropic distance is computed with 12-bit LZW compression (*left*) and Zip 2.3 compression (*right*). With LZW compression the entropic distance can separate signals more easily than is the case for Zip 2.3 compression. Error bars signify the standard deviation of the ten trials at each averaged data point. A cubic spline is used to interpolate the curves.

Note that entropic distance computed with 12-bit LZW compression saturates at approximately 5,000 samples, whereas entropic distance calculated with Zip 2.3 compression saturates at about 30,000 samples. This *stagnation point* is related to the size of the dictionary used for the respective compression algorithms. The codebook size is 5,021 for the 12-bit LZW algorithm here; the codebook size for the Zip 2.3 algorithm, though interesting, is not known by the author other than the suggestion that it is about 30,000 codes. If, for the LZW algorithm, the size of the codes are increased from 12 bits to 13 bits the saturation point moves to approximately double that of the 12-bit case with a corresponding rise in the entropic distance (see Figure 12.42). Saturation occurs when the compressor exhausts its supply of unique codewords to represent the sequence being compressed. When this happens, the compressor is forced to use codewords that are not optimal for all symbol strings in the sequence and consequently the compressor efficiency stagnates.

The preceding analysis deals with samples contained within an 8-bit byte. Though the sequence of samples from the HF receiver have 20-bit resolution, the loss of significance in going to 8-bit samples is minimal because the amplitudes of the received

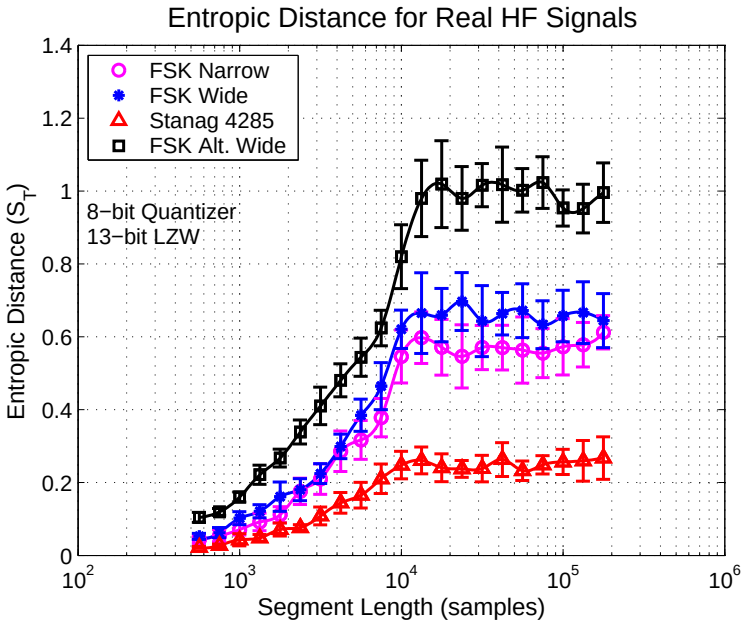

Figure 12.42. Entropic distances between real HF signals—13-bit LZW compression. Entropic distance of various real HF signals using 13-bit LZW compression. The *stagnation point* is approximately double that observed with 12-bit LZW compression; a result of increasing the range of codes by a factor of 2 and increasing the codebook size. Error bars signify the standard deviation of the ten trials at each averaged data point. A cubic spline is used to interpolate the curves.

signals are small. On this basis, the former work (see previous section) utilizes samples mapped to whole numbers in the range -128 to +127 but stored as 8-byte floating point numbers. Results similar to those presented here are achieved and the explanation for the similarities is simple. With whole numbers in the specified range, an 8-byte floating point number contains mostly zero-bytes in a long string. Both 12-bit LZW compression and Zip 2.3 compression are able to efficiently compress strings of zeros and therefore the additional zeros contribute little to the entropic distance equation. The remaining non-zero bytes compress slightly less efficiently because the sample is now spread across more than one byte. Nevertheless, the results of the previous work show that even whole numbered samples stored in floating point format can be used in the entropy method described above to separate HF signals. When entropic distance is computed on floating point samples, the mean-separation-distance (MSD) is a useful tool for choosing an appropriate scaling factor. When entropic distance is calculated

on integer samples, the MSD is useful for choosing an optimal resolution for output samples of the digital receiver.

Another point of discussion is that of entropic distance between signals. The experiments above show the entropic distance between a signal and a uniformly random arrangement of symbols from the Q-symbol alphabet. But consider the random arrangement of symbols as a reference point, and then the entropic distance between signals can be calculated simply as the difference of entropic distances. For example, in the saturation region of Figure 12.41 the entropic distance of the FSK Narrow/R signal is about 0.32 while the entropic distance of the FSK Wide/R signal is about 0.55. The entropic distance between the two signals could be computed as $0.55 - 0.32 = 0.23$.

Thus it is clear that modulation types of real HF signals can be separated using the entropic distance measure provided three conditions are met:

1. an appropriate lossless compression algorithm is used (*e.g.* LZW);
2. the number of bits representing each symbol in the reference alphabet must be equal to or slightly greater than the resolution of the signals of interest; and
3. the reference signal should be a random arrangement of symbols from the reference alphabet.

A Note on Shannon's Entropy

Shannon (1948) shows that the computation of true entropy relies on the independence of successive symbols. That is, if successive symbols (in this case quantized samples) are not independent then conditional entropies must be used to determine the true entropy. For the synthetic signals in Table 10.1, successive samples are assumed independent even though, strictly speaking, they are not independent. They are not independent because successive samples can be predicted with some certainty based on foreknowledge of the signal waveform. For example, in any given bit period of a synthetic 2-FSK signal (without added noise) and given any sample of the waveform, the next sample is predictable knowing that the signal is a sinusoid. This is the case provided the next sample is not at a bit boundary. The addition of noise reduces the

certainty of the next sample value, but does not reduce that certainty to zero. Therefore successive samples of the synthetic signals in Table 10.1 are not strictly independent. However, successive samples from real HF signals have a larger degree of independence because of the nature of HF communications, which encompasses impulsive HF noise, fading, multipath, and Doppler spreading; and most importantly, the receiver is not necessarily aware of the type of incoming signal.

One may ask why Benedetto *et al*'s (2002) method is used in place of Shannon's method for calculating entropy. The answer lies in Shannon's definition of entropy. His calculation provides an entropy measure for all symbols in the information sequence and does not account for patterns or groupings of symbols in the information stream. Though not shown in Figures 12.38 to 12.42, Shannon's entropy yields a relatively constant value (near one) for all signal types across all segment lengths and is therefore of little use for signal identification.

12.4 Signal-to-Noise Ratio

The analysis of the signal-to-noise (SNR) estimate of Eq. (10.38) begins by generating five synthetic digital signals with added Gaussian noise. Each signal carries an identical random binary message from a Bernoulli distribution. These signals include the 2-FKS/S, 2-PSK/S, and Stanag 4285/S signals from Table 10.1 as well as 4-FSK/S and 4-PSK/S signals. Moreover, the m-ary FSK and m-ary PSK signals have a constant envelope of one. The Stanag 4285/S signal does not have a constant envelope; it varies between –1 and +1. Though the ultimate goal is to apply parameters to real HF signals with non-Gaussian noise and co-channel interference, for the current discussion Gaussian noise is sufficient. Next, Eq. (10.38) is applied to each synthetic signal at various SNRs with results displayed in Figure 12.43.

Equation (10.38) does appear to provide a useful measure of SNR for each signal. For the most part, the estimate follows a log-linear relationship with the true SNR (*i.e.* the theoretically expected SNR). For 2-FSK the linearity is about 25 dB along the abscissa. For Stanag 4285 the useful region is only about 7 dB, though Eq. (10.38) does underestimate the true SNR by about 2 dB for this signal. The linear range for all the other signals is about 15 dB. For weak signals with an SNR above –5 dB the relationship is

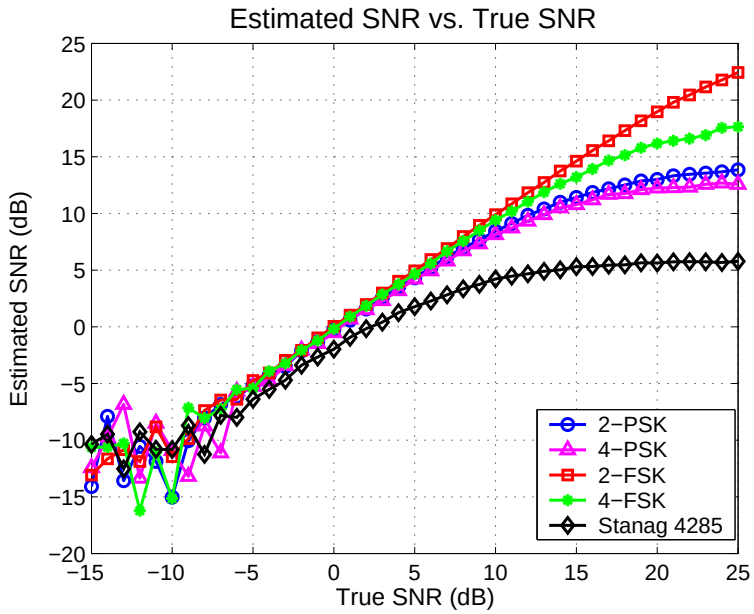

Figure 12.43. Estimates of SNR for various synthetic digital signals. The SNR estimator in Eq. (10.38) is based on Aisbett's *hash* function. Estimates of the SNR are reasonable for a fairly wide range. The estimate for the FSK signals is more log-linear than for the PSK signals. It appears that the greater the number of modulation levels, the further the estimate deviates from a log-linear one. For weak signals the relationship is log-linear because the noisy signals are similar. However for strong signals, the effects of the modulation become more pronounced. For the constant envelope signals (m-ary FSK and m-ary PSK) the log-linear relationship is obeyed for a greater range of SNR than for the Stanag 4285 signal, which does not have a constant envelope. For an SNR less than about –5 dB, the signals are too noisy and the estimate of SNR becomes erratic.

log-linear because the noisy signals are similar. However for strong signals, the estimator diverges from the log-linear trend. For the constant envelope signals (m-ary FSK and m-ary PSK) the log-linear relationship is obeyed for a greater range of SNR than for the Stanag 4285 signal, which does not have a constant envelope. When the true SNR is below –5 dB, the estimator is erratic; the signals are too weak and the noise dominates. In other words, from the point of view of the estimator, all the signals appear the same because of the noise. Therefore, within a limited range and with suitable scaling and translation (if necessary), Eq. (10.38) can provide a reasonable estimate of SNR.

What is also interesting is that direct application of Eq. (10.38) appears to separate the signals when the SNR is high. Is this a result of the modulation or is it caused by

something else? The envelope of each signal is constant and the same for each value of true SNR (this does not apply to the Stanag 4285/S signal as it does not have a constant envelope). Identical envelopes are necessary to ensure that the estimate of SNR does not vary from signal to signal as a result of different amplitudes. An alternate view of this is that with a constant envelope Eq. (10.38) should depend only on the biases in the estimators and on the average noise power; not on the signal envelope!

For example, suppose that the estimate of signal power from Eq. (10.37) yields

$$\sqrt{\Psi\left(z^2(t), z^2(t)\right)} = \alpha B^2(t), \tag{12.6}$$

instead of $B^2(t)$, where $0 \leq \alpha \leq 1$. And suppose, that the estimate of signal plus noise power provided by Eq. (10.33) yields

$$E\left\{z^2(t)\right\} = \beta B^2(t) + 2\sigma_n^2, \tag{12.7}$$

instead of $B^2(t) + 2\sigma_n^2$, where $0 \leq \beta \leq 1$. Inserting Eq. (12.6) and Eq. (12.7) into the SNR estimator of Eq. (10.38) gives

$$\mathrm{SNR_e} = \frac{\alpha B^2(t)}{(\beta - \alpha)B^2(t) + 2\sigma_n^2}. \tag{12.8}$$

As the noise power tends to zero (or the SNR increases to infinity) the SNR estimate becomes dependent only on α and β such that

$$\mathrm{SNR_e} = \frac{\alpha}{\beta - \alpha}. \tag{12.9}$$

This implies that if the second moment of $z(t)$ has a constant bias with regard to the peak signal power, $B^2(t)$, or if the signal power estimator of Eq. (10.37) has a constant bias with respect to $B^2(t)$, then the SNR estimate becomes constant; it is no longer affected by the true SNR but is dependent on the bias of the estimators. As the overall bias decreases (*i.e.* $\frac{\beta}{\alpha} \rightarrow 1$) the SNR estimate becomes infinite. Figure 12.43 does indeed show that the SNR estimate is independent of the true SNR for high values of the true SNR. Actually, one can determine α from plots of the estimated signal power in Figure 12.44.

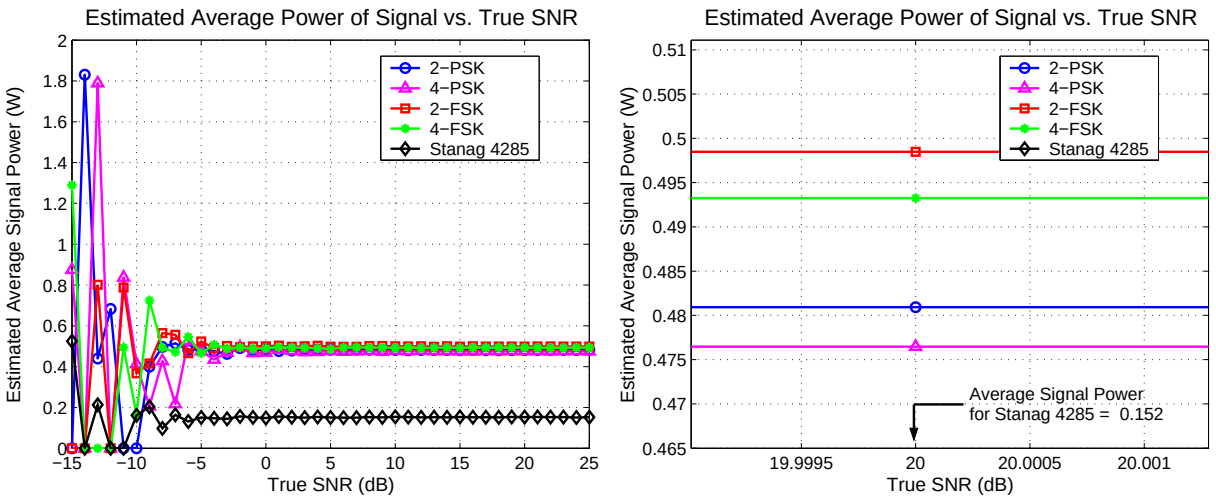

Figure 12.44. Estimates of signal power of various digital signals. The signal power estimator of Eq. (10.37) produces a value that is proportional to the square of the peak signal power. The average signal power estimate (*left*) is half the square-root of the output of Eq. (10.37). In fact, the result is very near the theoretically expected power of 0.5 for values of SNR greater than about –5 dB. Below this threshold, the signals are too noisy and the average power estimator produces unstable estimates. The power estimator underestimates the true signal power slightly (*right*). The theoretical signal power is 0.5.

Take for example the estimates of signal power at an SNR of 20 dB (see Figure 12.44 and Table 12.2). The ratio of the estimated average signal power to the theoretically expected signal power (see Figure 12.45) yields the proportionality constant, α. For large SNR, the noise power is weak and therefore the SNR estimator is affected more by the bias in the power estimators than by the noise. A consistent under-estimation of the signal power causes a constant SNR estimate. As a consequence, the SNR curves in Figure 12.43 saturate at high SNR. The last column in Table 12.2 agrees, asymptotically, with the estimates of SNR in Figure 12.43 at a theoretically true SNR of 20 dB.

Thus the separation of curves in Figure 12.43 is a due to biases in the signal power estimators at high SNR. To maximize the log-linear range over which the SNR estimator is useful, the magnitude of $B(t)$ in the received signal must be maintained as constant as possible. Automatic gain control (AGC) could be applied to the baseband signal prior to application of Eq. (10.38) but, in practice, it will be difficult to ensure that $B(t)$ is constant at low SNR. Moreover, the separation of curves is related to the bias of the SNR estimator and therefore Eq. (10.38), on its own, cannot be used as a signal feature to

Table 12.2. Estimating the bias of the signal power estimator. The bias of the signal power estimator is clear when the expected power level (see Figure 12.45) is compared with the experimental signal level for the various constant envelope signals. The ratio of these two values is the coefficient, α, of Eq. (12.9). The value of β in the equation is so close to unity, that it can be assumed that $\beta = 1$. When the expected SNR is high the effect of noise is negligible and the bias of the estimators dominate.

Modulation	Expected Signal Power	Estimated Signal Power	α $\frac{\text{Est.}}{\text{Exp.}}$	SNR $\frac{\alpha}{1-\alpha}$	SNR (dB)
2-FSK	.500	.498	.996	249	23.9
4-FSK	.500	.493	.986	70.4	18.5
2-PSK	.500	.481	.962	25.3	14.0
4-PSK	.500	.477	.954	20.7	13.2
Stanag 4285	.192	.152	.791	3.78	5.78

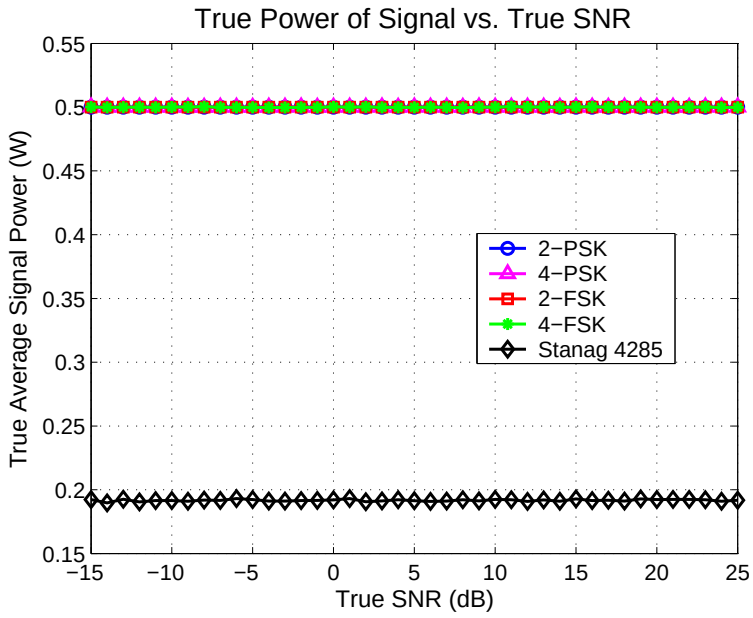

Figure 12.45. Theoretical average power for various digital signals. The expected average signal power is half the peak envelope power (PEP). For the constant amplitude signals (*i.e.* m-ary FSK and m-ary PSK) the average power is 0.5, but for the Stanag 4285 signal, the average power is 0.192.

Table 12.3. Comparison of the SNR estimator & SNR from the power spectrum. The comparison of the SNR estimates from the power spectra and the SNR estimates from Eq. (10.38) reveals that, though inconclusive, the measures provided by Eq. (10.38) are reasonable.

Modulation	Bandwidth (Hz)	SNR (dB) via Spectrum	SNR (dB) via Eq. (10.38)	Difference (dB)
FSK Narrow	175	13	8.4	4.6
FSK Wide	940	13	15	2.0
FSK Alt. Wide	260	16	26	10.0
Stanag 4285	2500	7.0	2.7	4.3

unambiguously separate constant envelope signals. It can only be used in conjunction with other signal features as part of a feature vector.

Now let us consider the estimation of SNR for the real signals in Table 10.1. The signals being considered are FSK Narrow/R, FSK Wide/R, FSK Alt. Wide/R, and Stanag 4285/R (see Figure 12.46). Estimation of the SNR from the spectra is crude and must be so since the actual input SNR at the narrowband receiver for each signal is unknown. The method involves determining the channel bandwidth (see Table 10.1) and then estimating the power of the highest sidelobe near the band edges and subtracting this from the apparent passband power level. Of course, the comparison method is Eq. (10.38). This equation is applied to each of the reals signals. Table 12.3 summarizes the results of both methods.

Though the results of this comparison are inconclusive, they do suggest that the SNR estimator of Eq. (10.38) provides reasonable measures of the SNR of a digital signal. Future work should address this issue in more detail by conducting an investigation that compares the results of Eq. (10.38) with known SNR for real signals.

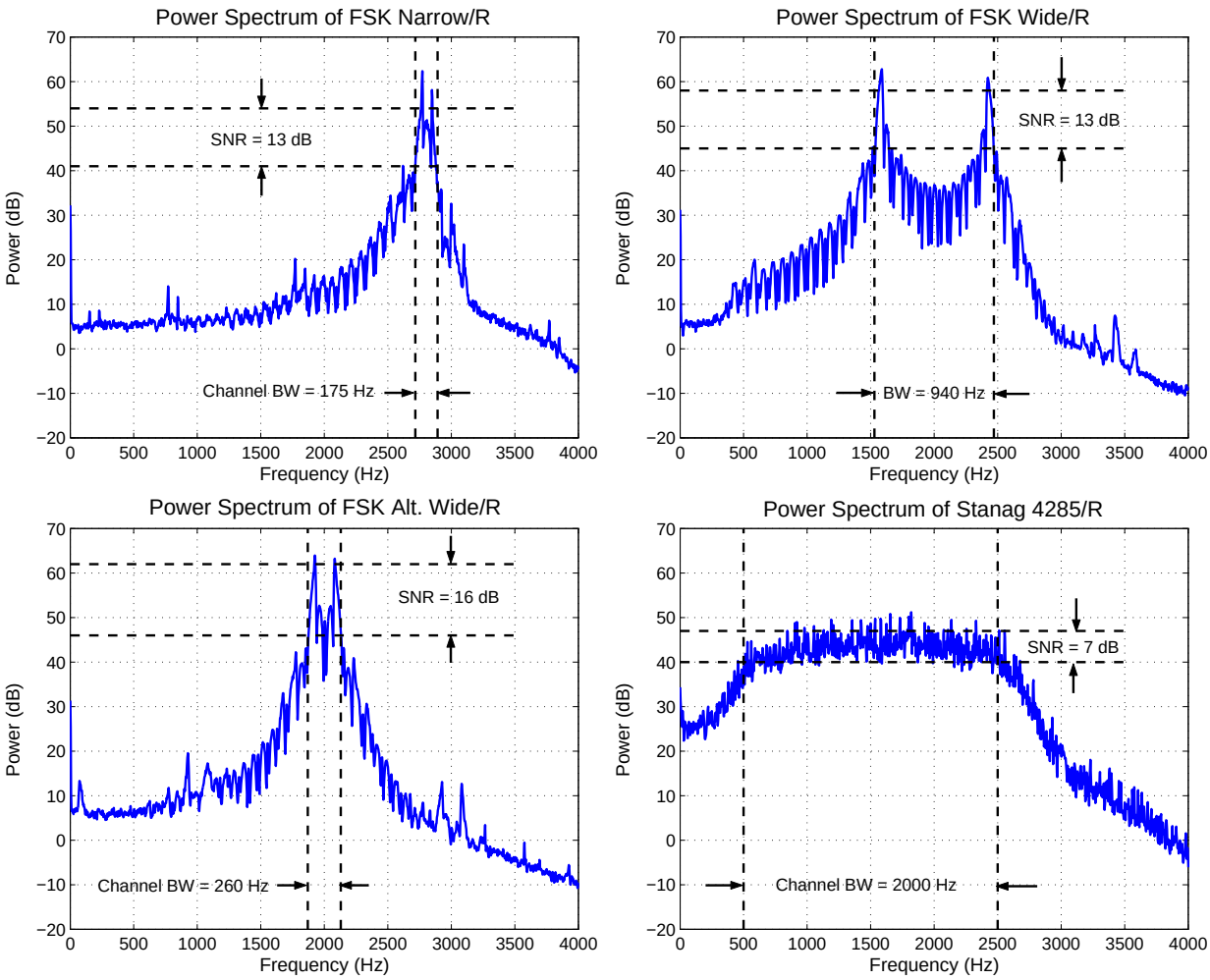

Figure 12.46. SNR estimation from power spectra of real digital signals. Estimation of the SNR from the spectra is crude; the method involves determining the channel bandwidth (see Table 10.1) and then estimating the power of the highest sidelobe and subtracting this from the apparent passband power level. For FSK Narrow/R (*top-left*) the SNR is approximately 13 dB. For FSK Wide/R (*top-right*) the SNR is also about 13 dB. The SNR for FSK Alt. Wide/R (*bottom-left*) is near 16 dB and for Stanag 4285/R (*bottom-right*) it is about 7 dB. The actual input SNR at the narrowband receiver for each real signal is unknown.

12.5 Summary

Coherence is shown to provide a simple *yes/no* answer to the question: is the received signal highly correlated with a known reference signal? It is therefore not practical for discerning a modulation directly from an unknown received signal. Moreover it is dependent on the message carried by the signal and is time sensitive.

Entropic distance on the other hand, does separate contrived and real signals. Various real and synthetic HF signals are shown to be separable using the parameter. This parameter is based on the efficiency of an algorithm at compressing real and synthetic HF signals. Choice of a compression algorithm plays an important role in the usefulness of the parameter. The Lempel-Ziv-Welch algorithm separates the signals well, whereas Zip 2.3 provides marginal separation. Other compression algorithms may also provide good separation. Entropic distance does have a dependency on signal-to-noise ratio (SNR). Knowledge of the SNR will improve the reliability of the entropic distance.

A reasonable estimation of signal-to-noise ratio (SNR) is possible with an estimator based on a function proposed by Aisbett (1986). The estimator works particularly well for constant envelope signals (m-ary FSK and m-ary PSK) with an SNR of approximately –5 dB to about +15 dB. Application of the estimator to a signal with a varying envelope (Stanag 4285) yields an inferior result. That is, the range over which the estimator is log-linear is much shorter, and the SNR estimate is biased. Suitable scaling and translation, based on signal type, should provide improved estimates for digital signals regardless of the type of envelope.

Much is said about coherence, entropic distance, and SNR in this chapter. Indeed, some of the results indicate that there is potential for the parameters. So, what can be concluded from the observations? This is the topic of the next chapter.

Chapter 13

Conclusions & Further Work for Part III

WITHOUT doubt the explorations of the previous chapters suggest that modulation recognition is a complex topic of research. Some questions are answered and yet others require more work. It is clear that there is much work to complete before robust and universal modulation recognition methods are available. This chapter summarizes and make conclusions about the work in Part III of the thesis. It also establishes a basis for further work.

Part III begins with a question: what signal features are useful for automatic modulation recognition? Chapters 9 to 12 describe current research, coherence and entropy and SNR parameters, experiments, test setups, and the results of applying the parameters to real and synthetic data sets. The actual question answered by these chapters is: are the coherence function, entropic distance, and SNR useful for modulation recognition? The answer to this is "unlikely", "for some signals", and "not by itself". These are typical answers that one would find in most papers on automatic modulation recognition, and which elicit a perceived conclusion that no feature extraction method will be applicable to all signals, and importantly, the general approach to research in this field is *ad hoc* . That is, brute-force and trial-and-error techniques are used to exhaust all possible signal identifying parameters. This is akin to using a shot-gun to kill a fly when all is needed is a local mesh (*i.e.* a fly swatter) in the vicinity of the target. Surely, there is a more efficient all-encompassing way to automatically recognize signals! A general conclusion is this: the common approach to automatic modulation recognition is inefficient and, except for cases where specific signals are targeted, will likely never converge on a sufficiently elegant solution. The outlook is not as dire as it appears. At least this *ad hoc* method reveals some useful information that can aid automatic recognition for classes of signals. Comments on this soon follow. For now, recall the journey from Chapter 9 to this point.

Chapter 9 describes the need for modulation recognition: the key need being the ability to have agile software radios that can switch from one modulation type to another automatically. Such a radio opens up the possibility of one device that can handle numerous signal types, which is especially useful to government agencies (*e.g.* Defence, spectrum management, emergency services). The chapter goes on to discuss the fundamental premise of modulation recognition. This premise is that with no foreknowledge of the signal, a variety of signal features can be constructed that form a unique feature vector orthogonal to other feature vectors. It is this orthogonality that allows a classifier to unambiguously determine the modulation type. Of course, as subsequent research points out, the choice of a feature vector is difficult. The search for useful signal features is well documented. Numerous researchers propose features such as m^{th} order moments, frequency, phase, amplitude, kurtosis, zero-crossings, carrier-to-noise (CNR) ratio, and others. Moreover, they suggest methods for signal analysis (*i.e.* either

extracting features or classification) using neural networks, decision theoretic methods, trees, pattern recognition, and entropy. Some of these parameters and methods are useful in some instances, but not in all cases. And, importantly, few researchers apply their methods to real signals.

Of the many possible parameters, three are discussed in detail in Chapter 10: coherence, entropic distance, and signal-to-noise ratio (SNR). Coherence is analogous to correlation in the spectral domain and entropic distance is a measure of a signal's information content based on Shannon's (1948) information entropy. The SNR parameter is not a new concept, but rather the discussion follows a method of estimating SNR from a power estimate proposed by Aisbett (1986).

Coherence, as suggested by Carter (1993), must be estimated for practical purposes because often the coherence function can be intractable (see Appendix A). A popular method for estimating coherence uses weighted overlapped segments. But, even with a good estimation method, the coherence function can be temperamental. Visual similarities between spectra do not necessarily imply high coherence. Moreover, the coherence is intimately related to SNR, and is greatly affected by signal timing. To study the coherence, Chapter 10 lays out four experiments that 1) show the importance of choosing the correct number of segments and overlap for the coherence estimator; 2) demonstrate the need to specify the bandwidth for the coherence estimator; 3) reveal the relationship between coherence and Hamming distance (a measure of the bit-wise difference between digital messages carried by the signals); and 4) divulge the sensitivity of the coherence function to signal synchronization. Further discussion of coherence, as it applies to m-ary FSK signals, leads to the definition of a new parameter called the coherence-median-difference (CMD). The CMD provides a a confidence measure that the coherence at the known symbol frequencies for m-ary FSK dominates coherence over the bandwidth of interest.

Chapter 10 continues by delving into a discussion of entropy and entropic distance. The entropic distance is a measure of entropy that utilizes compression algorithms. The method, proposed by Benedetto *et al* (2002), is primarily for language and author identification, but with an appropriate interpretation, is equally valid for signals. That being said, four experiments are defined that demonstrate a unique application

of the entropic distance parameter. The key is to understanding the results of the experiments is to recognize that digital samples from a signal form an arrangement of symbols (or characters) from a specific alphabet constrained by the dynamic range of the analog-to-digital converter (ADC). The entropic distance is calculated by comparing the entropy of the sequence of samples with the entropy of a random arrangement of samples drawn from the entire alphabet. Specifically, the experiments 1) observe the effects of the compression algorithm and structure of the signal on entropic distance (with the knowledge that the choice of compression algorithm is important); 2) consider the effects of the ADC resolution on entropic distance; 3) investigate whether or not entropic distance is a only a temporal measure or also a spectral measure; and 4) measure the entropic distance of real HF signals and synthetic signals in an attempt to separate signals. Finally, a logical discussion reveals that entropic distance (as defined) requires time-series data; spectral data cannot be used.

The last parameter presented in Chapter 10 is an SNR estimator based on Aisbett's (1986) *hash* function, which is the product of the means of two signals less their covariance. The *hash* function produces an estimate of signal power. With this power estimate, a particularly useful SNR estimator is derived. Two experiments estimate the SNR of synthetic signals and real HF signals, and it is shown that the estimates are comparable to spectrally-based estimates (see Table 12.3).

Though the true SNR of the real signals is unknown it is shown that their SNR estimates are reasonable.

Following the discussion on modulation recognition, Chapter 11 reviews the development of the narrowband and wideband receivers, and discusses the configuration of the narrowband receiver for the experiments of the previous chapter. An analysis package, developed in the **MATLAB**® environment, is also described. This toolbox models a transmitter, transmission medium, receiver, and feature extraction module. It is especially useful for comparing synthetic signals and real signals in the context of feature vectors.

Finally we arrive at Chapter 12. This chapter discusses the results of all the aforementioned experiments. It is shown that coherence is affected by the message of the signal as well as the signal-to-noise ratio (SNR). Entropic distance is affected by the compression method and is sensitive to the message structure but, nonetheless, is able

to separate some signals. A modification of Aisbett's (1986) hash function provides an estimate of SNR that is linear over a moderate range, but is more useful for constant envelope signals than it is for signals with a varying envelope.

There are three requirements to achieve an accurate estimate of coherence using the *WOSA* method: 1) an appropriate number of segments, 2) an appropriate weighting function; 3) and a suitable amount of overlap. As the number of segments increase the coherence estimate converges on the true coherence, and the deviation of the coherence estimate from the true coherence decreases. Moreover as the overlap percentage increases, the variance of the coherence estimate decreases. However there is a point where further increase in the number of segments or overlap provides only marginal improvement in the coherence estimate. This point is about 64 segments. The optimal amount of overlap is about 50% for a Hann window but will change for other weighting functions.

Coherence is intimately related to SNR, and is affected by the noise bandwidth, signal bandwidth, and frequency. For a strong signal and weak noise the coherence-SNR relationship shifts (see Figure 12.8) left implying that a high coherence is possible for lower overall SNR. For a weak signal and strong noise the coherence-SNR curve shifts right implying that a high coherence is possible only for higher overall SNR.

It has been said that coherence is sensitive to signal timing (Carter 1993). Indeed, by considering two signals having the same digital modulation carrying similar messages, one sees that as the Hamming distance between the messages increases, the coherence decreases. In simple terms, as the signals become more and more uncorrelated, the coherence tends to zero. From another viewpoint, increasing Hamming distance decreases the spectral similarities between the two signals. Moreover, the coherence is low between a real signal and a synthetic one of the same modulation. In fact, even the coherence between two real signals of the same modulation (and having the same message) is low if the two signals are misaligned in time. However, if the two real signals are synchronized the coherence is high (near unity). The conclusion of all of this is that coherence is robust in its ability to proclaim whether or not two signals are identical in the temporal and spectral domains. Coherence is not a parameter that can be used on its own to separate signals of different modulations.

The study of entropic distance shows that the choice of compression algorithm can greatly affect the entropy. Two compression algorithms are studied: LZW and Zip 2.3. The former compression method is better at compressing binary zeros than it is at compressing binary ones. The latter method, on the other hand, is equally efficient at compressing both binary zeros and binary ones. Thus the compression algorithm plays a critical role in the entropy measure. If an inappropriate compressor is used, the entropy measure can vary not only with the message but the compressor as well.

Next a unique application of the entropic distance is described whereby the samples of a signal are interpreted as symbols from a specific alphabet defined by the analog-to-digital converter (ADC) producing the samples. As the resolution of the ADC increases the entropic distance measure increases for a constant power signal. Two factors are therefore important for measuring entropic distance: 1) the compression algorithm and 2) the size of the alphabet from which the samples of the signals are drawn. The compression algorithm should be lossless and well-understood for a proper interpretation of the results; the alphabet size must be carefully chosen to match the dynamic range of the signals being analyzed. A new parameter, called the mean-separation-distance (MSD), aids this last issue. The MSD measures the separation between entropic distance curves. Maximizing the MSD can be achieved in two ways, either by adjusting the signal level so that it is contained within the dynamic range of the ADC, or by adjusting the dynamic range of the ADC to fit the signal.

Application of the entropic distance measure to signals (synthetic or real) does produce a separation dependent on the modulation types but, the correct interpretation of this separation requires knowledge of the SNR. At infinite SNR, the entropic distance provides no separation for constant envelope signals. Such signals span the dynamic range of the ADC in a similar manner and as a result their entropic distances with respect to the ADC alphabet are equivalent. Signals with varying envelopes tend to yield low entropic distance measures because the distribution of samples representing the envelopes are more noise-like than constant envelope signals. The entropy method compares these samples to a random arrangement of samples from the ADC alphabet in order to arrive at the entropic distance. Nevertheless, it is apparent that the entropic distance can be used to separate digital HF signals based on their modulation types. The entropic distance may well be a feature that, by itself, identifies signal types.

Signal-to-noise ratio can be calculated, estimated, and measured in many ways. Aisbett's *hash* function provides a useful estimate of peak-envelope-power (PEP). Aisbett also shows that the expectation of the square of a signal is an estimator of average signal power plus average noise power. The ratio of the PEP to the signal+noise power is a measure of SNR and this estimator follows a log-linear relationship with theoretical SNR. That is, given a signal with a known SNR, the SNR estimator generates an estimate of SNR that is directly proportional (on a log scale) to the actual SNR. It is particularly useful for constant envelope signals but less so for signals with varying envelopes. For constant envelope signals (m-ary FSK and m-ary PSK) the SNR estimator is accurate from about -5 dB to +15 dB. For varying envelope signals the range is smaller. For example, with Stanag 4285 this range is only -5 dB to about +2 dB. There is a lower limit to the practicality of the estimator and an upper limit to its usefulness. Below an SNR of -5 dB the noise dominates the estimator and it is unable to extract the signal from the noise. Beyond the upper limits the estimator is sensitive to the degree of consistency in the signal envelope. At high SNR, small variations in the level of the constant envelope affect the SNR estimate to the point that the estimate is no longer proportional to the actual SNR. The reason that these variations so confound the estimator is that the estimator is biased; at high SNR it consistently underestimates the true SNR. Practically this means that automatic-gain-control (AGC) is necessary to maintain the signal level but, at low SNR it will be difficult to maintain the level. Therefore within a limited range, and with suitable translation and scaling, the SNR estimator can provide a good estimate. As a signal feature, SNR cannot indicate the modulation of a signal; it can only support decisions based on other identifying features (*e.g.* entropic distance).

This part of the thesis began with questions and for all the questions that have been answered, more have been raised. For example, how well does the SNR estimator work on real signals? The SNR investigation in this part of the thesis does apply the SNR estimator to real signals but, the actual SNR of each of these signals is unknown. Further study of this question requires a test whereby signals are received with known SNR to which we compare the SNR estimate.

Another question is: can the entropic distance measure be used by a classifier to identify a signal? Or, will it actually need a complementary measure of SNR for the classifier to group signals? Answering these questions will require entropic distance measures for many signal types; many more than discussed in this work. Indeed, Benedetto *et al* (2002) consider 50 different languages, 90 different texts, and 11 different authors. To truly understand the usefulness of entropic distance in the context of signal processing, perhaps 50 to 100 different modulation types should be investigated.

If entropic distance cannot be used as a sole identifier, future work can continue to focus on creating orthogonal feature vectors for an unknown signals. These vector could consist of the following parameters: center frequency, bandwidth, signal-to-noise ratio, signal envelope, symbol frequencies, cross-Margenau-Hill distribution, kurtosis, signal constellation, signal power-spectral-density, modulation level, auto-regressive covariance, and entropic distance. These parameters can be obtained using the methods suggested by their proponents in the literature or by methods discussed in this thesis.

Other work should implement an HF channel model for the Ⓢ**igns** toolbox. Wideband models such as those proposed by Vogler and Hoffmeyer (1988, 1990, 1992) or Lemmon and Behm (1991, 1993) are now more acceptable than the narrowband model of Watterson (1969, 1970). The noise model from Part II should also be added to the HF channel model. The toolbox also requires a carrier estimation module, additional signal features (*e.g.* kurtosis, zero-crossings) found in the literature, and a classification algorithm.

An interesting fact about the narrowband receiver (the forerunner of the broadband receiver) is that it is a multiple-input multiple-output (MIMO) system because it has numerous receive elements (antennas). Others (Foschini 1996, Foschini & Gans 1998, Wolniansky, Foschini, Golden & Valenzuela 1998) show that a communication system consisting of multiple transmit and receive elements has numerous advantages over conventional communication systems with single transmit and receive elements. It may be that modulation recognition algorithms perform better in MIMO systems. Transmitting one or more test signals to a multi-element receiver and then analyzing the received signals would test this theory. The signal from each element of the receiver would be a delayed replica of the signals received by all the other elements. Signal

separation and signal enhancement techniques would be ideal for extracting information and noise components from this ensemble. By subtracting the noise from a linear combination of the received signals it may be possible to improve the overall signal-to-noise ratio (SNR) and thereby improve the accuracy of the modulation recognition schemes (Fabrizio, Abramovich, Anderson, Gray & Turley 1998). There is, therefore, significant work in analyzing and recognizing signals in the entire wideband data set. The data set (about 300 GB), used in Part II, contains many man-made signals acquired through four antennas of the wideband receiver. It is an ideal candidate data source to observe the performance improvement, if any, of modulation recognition algorithms being fed data from a MIMO receiver.

The additional work in the field of modulation recognition seems endless. This chapter summarizes the results of the investigations in this thesis and highlights many other areas of endeavour, in order to motivate an eventual robust and universal modulation recognition algorithm.

PART IV

Additional Information