Towards a General Model for Secure Speech Communications

by

William Robert Anderson

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical Engineering

Waterloo, Ontario, Canada, 1996

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

It is clear that the activities of an increasingly information-centered societal infrastructure are driving the rapid expansion of digital communications technology. The demand for services integrating voice, data, and video traffic is increasing with the technical sophistication of the consumer, and has already spawned a host of alternatives including ISDN, cellular, cable modem, personal communication systems, and Internet technologies. These technologies all represent steps in the evolution towards a global digital communications strategy. We will examine issues surrounding the implementation of speech services in a secure digital communications environment.

The reasons for the emergence of digital speech communications systems lie in their advantages over the older analog systems. These include improved signal quality, error recovery capabilities, and the ability to multiplex a signal with other conversations or with an entirely different form of digital service. Digital service also allows a significant improvement in the ability to protect a conversation from unwelcome scrutiny. This is fast becoming an important issue in the design of information systems. In the information age we have come to recognize that there is intrinsic value in most forms of communication, and that ensuring privacy can be tantamount to protecting material assets.

The addition of a cryptographic component into a speech communication system adds complications that are not encountered in a conventional system. The characteristics of an encryption coder increase the challenge of secure system design. The system designer must balance the requirements for high security and subjective speech quality with the conflicting desires for a low channel bit rate and limited system complexity. The optimal choice of source, encryption, and channel coder components to meet these objectives requires a thorough understanding of a large set of interrelated parameters.

In this research we will develop models of objective speech quality, bit rate, security, and complexity that are relevant to the design of an integrated secure speech communication

system. We will present experimental and theoretical evidence leading to an understanding of the interrelationships among the parameters of the secure system. The parameter models will then be integrated to define a multidimensional constrained optimization problem as a general model for the secure speech communication system.

Acknowledgements

I would like to express my appreciation to my supervisor, Dr. Gord Agnew for his advice and support throughout my graduate studies.

I would like to thank my friends: Claude Bergeron, Michael Cheung, Michael Farebrother, B.J. Lee, Robert Lehr, Biswajit Nandy, Loui Polic, Dave Simser, Gord Strachan, Vincent Wong, and Atsushi Yamada. Each has made a notable contribution to my education and the production of this thesis. I would also like to thank all of my ball hockey and broomball teammates and the UW Aiki Ju Jitsu club for some entertaining diversions.

I owe the greatest thanks to my dear wife Jen, who has made all of this worthwhile. Her support and encouragement has carried me through. We are now looking forward to the next challenge.

This thesis is dedicated to the memory of Jim and Diane Ohi.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

It may have surprised Alexander Graham Bell to learn that his curious invention of 1876 would be followed by over a century of intense interest and development. Verbal communication has an appeal so fundamental that it seems a device allowing natural speech communication over great distances was destined for success.

From its limited beginnings the global telecommunications infrastructure has developed steadily, to the point where it is now possible to reach most of the world by means of the analog circuit-switched telephone network. Despite the magnitude of this achievement, an increasingly sophisticated and information-centered society demands more service than the analog telephone network can provide. Both the cause and the cure to this problem lie in the rapid expansion of digital communications technology. There is an increasing demand among consumers for services integrating voice, data, and video traffic, and this has spawned a host of alternatives including ISDN, cellular, cable modem, personal communication systems, and Internet technologies. These technologies all represent steps in the evolution toward a global communications strategy. In this thesis we will focus on issues surrounding the implementation of speech services in a secure digital communications environment.

The reasons for the emergence of digital communications technology lie in its advantages

over analog transmission. These include improved signal quality, error recovery capabilities, and the ability to multiplex a signal with other conversations or with an entirely different form of digital service. Digital speech enjoys one other advantage in that it allows a significant improvement in our ability to protect a conversation from unwelcome scrutiny. This is fast becoming an important issue in the design of information systems. In the information age we have come to recognize that there is intrinsic value in most forms of communication, and that ensuring privacy can be tantamount to protecting material assets.

In progressing forward to a purely digital domain we must remember to look behind at the enormous investment in the existing telecommunications infrastructure. It is a practical consideration that demands the contemporary communications system be designed with a mind to the limitations of the vestigial analog system.

1.1 The Conventional Communication System

A conventional digital speech communication system consists of two components: a source coder and a channel coder. The source coder is responsible for transforming the analog speech signal to an approximate digital representation that is convenient for transmission. Toll quality speech in digital form can require significant bandwidth for transmission, often a rate in excess of the capacity of the analog telephone network. Digital communication systems must therefore compress the signal before transmission over analog media. The means of performing this compression are varied, but all are based on the potential to reduce the high level of redundancy in the natural speech signal.

The output of the source coder is processed by a channel coder responsible for providing reliable transmission on a potentially noisy medium as shown in figure 1.1. This operation expands the coded signal, adding the redundancy necessary to reconstruct the signal in the presence of errors.

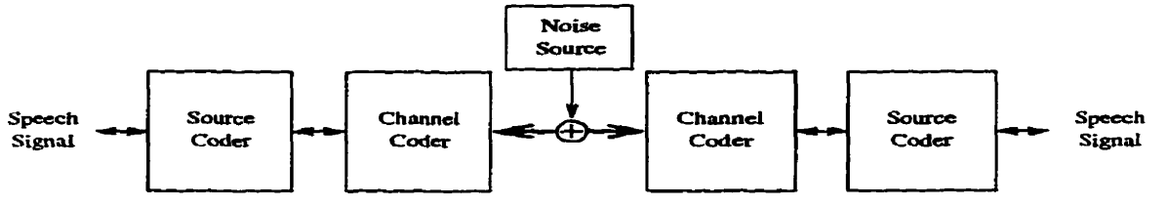


Figure 1.1: The Conventional Speech Communication System

1.2 The Secure Communication System

The secure communication system adds an encryption coder intended to protect the message from unauthorized access. The encryption coder is typically placed between the source and channel coders, as shown in figure 1.2. This arrangement sees the encryption coder operate on a compressed speech signal, and the channel coder operate on a randomized information stream. The result is a compressed and error-tolerant signal that is unintelligible to all but the intended recipient. In chapter 4 we will examine the effects of encryption coding in some detail.

It is most important to understand the goals and capabilities of the attacker. The goal of the attacker will typically be to determine the content of the encrypted message, but he may also be motivated to interfere with the transmission. Our treatment will focus on evaluations of the secrecy of the transmission in the presence of purely passive analysis, but we will address some of the more active efforts the cryptanalyst may make to gain advantage.

It is also important to recognize that the attacker has at his disposal the significant signal processing power of the human auditory system in addition to conventional computing resources. Human perceptual abilities are important in the context of speech encryption because of their vast processing power and adaptability. If the encryption mechanism is inadequate and exposes even some small part of the speech signal, then it may be possible for a human listener to form an understanding of the residual intelligible information.

Traditionally, each coder element in figure 1.2 has been treated independently in de-

signing an integrated secure system. This approach can lead to sub-optimal system designs when the encryption device manipulates the signal in a manner not anticipated by the conventional model. It will be the focus of our research to determine how the addition of an encryption device affects the performance of the integrated communication system. An integrated model of all relevant system characteristics will allow selection of more efficient system solutions.

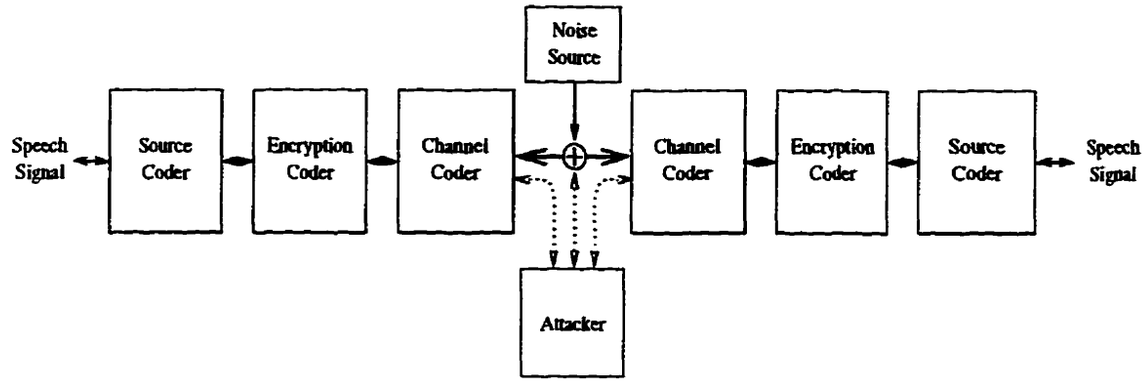


Figure 1.2: The Secure Speech Communication System

Some characteristics of the secure communication system are immediately obvious and suggest a complex relationship among the system parameters. For example, the main function of the source coder is to remove redundancy and thereby reduce the bit rate. Source coding can improve the security of a cryptosystem by removing redundant message components that may aid an attacker in deciphering the message. Some encryption systems exact a high price for channel errors, however, significantly reducing subjective quality evaluations on noisy channels. It is necessary to balance the advantage to security and bit rate that may be gained by aggressive source coding with the additional complexity that may be required in the channel coder to compensate for the effect of channel errors.

In general the design of secure speech communication system entails a challenging combination of objectives. The system designer is concerned with providing high security and subjective speech quality while requiring a low channel bit rate and overall system complex-

ity. The optimal choice of source, encryption, and channel coder components to meet these objectives requires a thorough understanding of a large set of interrelated parameters.

1.3 Scope of the Work

It is our intention to develop models of objective speech quality, bit rate, security, and complexity that are relevant to the design of an integrated secure speech communication system. Through experimental and theoretical work we plan to develop an understanding of the interrelationships among the primary parameters of the secure system. The parameter models will then be integrated to define a multidimensional constrained optimization problem as a general model for the secure speech communication system.

We will begin in chapter 2 with the development of a new technique for the estimation of high order conditional entropy. This technique allows the calculation of conditional entropy measures that complexity renders infeasible to the conventional direct calculation method. We developed this technique in order to compute an estimate of the entropy rate of the speech process. The entropy rate was found to be an essential measure that was central to many of the parameters computed for the general model. In chapter 3 we will present measures of the first to sixth order conditional entropy and obtain an estimate of the entropy rate of the speech process by applying a non-linear regression analysis to the experimental results.

The entropy estimation technique developed for chapter 2 will be shown to be more widely applicable to estimation problems involving probabilistic distributions of high model order. We will identify modelling activities where the new technique can be applied to obtain more accurate measures than were previously feasible.

The modelling process will begin in chapter 4 where we will present an argument for the development of theoretic measures of security and propose two unique indices for use in the general model of the secure speech system.

In chapter 5 we will develop models of objective quality and bit rate as a function of parameters under the control of a secure system designer. The segmental-SNR performance of a class of representative source coders will be examined under a variety of channel conditions, and measures of the informational divergence of the coded speech process will be computed to determine an analytic model relating bit rate to the level of redundancy in the source coded signal.

In chapter 6 we will perform experiments to evaluate the proposed theoretic security indices for the general model. This work will combine the results of our entropy estimation work and redundancy modelling to produce experimental models of cryptographic security.

We will then define rudimentary models for the complexity of source, encryption, and channel coders to add an essential dimension to the analysis of secure communications systems.

Finally, in chapter 7 we will combine the models of objective speech quality, bit rate, security, and complexity to define a general model of the secure communication system. We will outline the sets of objective, intermediate, and controlling parameters in the system and indicate the general interrelationships between them. Using these relationships we will formulate an optimization problem for secure system evaluation based on a general constrained objective function over the parameters of redundancy, encryption, and bit error rate.

We should note that while the modelling methodology and entropy estimation algorithms presented in this research are applied specifically to the case of the secure speech communication system, they may also be applied to other data sources that possess high levels of redundancy, such as video and facsimile.

Chapter 2

Entropy Estimation

2.1 Introduction

An accurate estimate of the information content of a digitized and band-limited signal is important to a broad class of activities including the development of compression algorithms, quality of service, and security measurements. As we will discuss in this chapter, this can be a very difficult measurement to perform for many classes of typical source processes.

Many of the activities necessary for the development of a general model of the secure speech communication system will require such a measurement to be performed on the conversational speech process. Our focus in this chapter will be to develop a general algorithm for entropy computation that can be applied to a general source process. In chapter 3 we will then apply this technique to the specific case necessary to our research.

Generally, the real sources of interest to designers of practical systems may be described as stochastic processes. To measure the entropy of a stochastic process it is necessary to determine an exact probabilistic model of the source process and compute the entropy rate directly from the model. There are two main considerations that determine the feasibility of an exact entropy calculation,

1. **State-Space Size.** The characteristic of memory in a process can lead to a very large state space for the complete probabilistic model.

2. **Computational Complexity.** The number of operations required for the exact calculation is a function of the extent of memory in the process. Full computation can require an infeasible number of operations.

For a general nth order entropy calculation, $O(|\mathcal{X}^n|)$ operations are required on a state-space of $|\mathcal{X}^n|$ elements, where $|\mathcal{X}|$ is the cardinality of the signal alphabet. These requirements can become prohibitive for measures beyond a small order. Presented in figure 2.1 is a graphical representation of the size of an exact nth order entropy calculation on an alphabet of size $|\mathcal{X}| = 256$. Noted in the figure are the nominal state-space sizes for calculations of order 4, 5, and 6. The shaded region indicates the maximum addressable memory size of a 32-bit architecture computing platform. Clearly, the state-space requirements of higher-order entropy calculations test the limits of current computational abilities. 64-bit architecture processors extend the addressable range to allow a theoretical 8th order state-space, but real memory limitations still constitute a significant obstacle.

In this chapter we will present a method of entropy estimation developed to allow high order conditional entropy measures with modest computing resources. In sections 2.2, 2.3, and 2.4 we will present the mathematical development for the new approach. This technique approximates an entropy measure by selecting a subset of the complete calculation space for exact computation and estimating the values not contained in the subset. The subset of computed measures can be chosen to adequately represent the conditional entropy measure by a sequential Monte Carlo point selection procedure. Further efficiencies can be obtained by adapting the probability distribution of the sampling procedure according to the relative magnitude of past measurements. This sequential Monte Carlo sampling approach, developed in section 2.5, is an efficient method of obtaining estimates of various information measures for a variety of source processes. Finally, in section 2.6 we will present

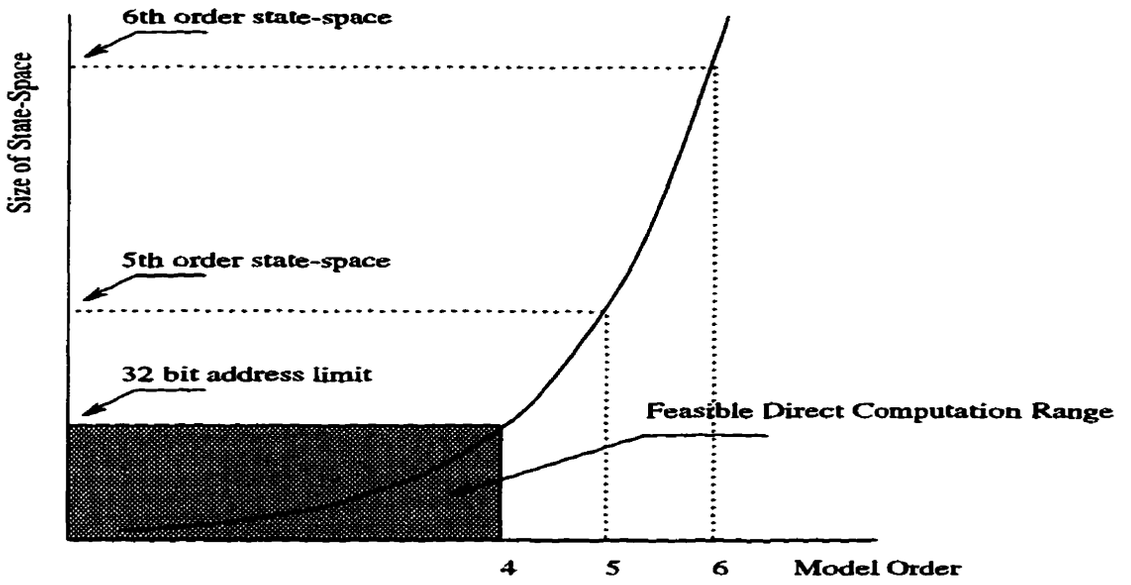


Figure 2.1: State-Space for nth Order Entropy Calculations

the mathematical basis for a method to normalize a set of partial conditional entropy measures in anticipation of computational difficulties which may arise in the experimental work required for entropy estimation.

2.2 Entropy Estimation Technique

In this section we will develop a method to estimate a high order information measure for any source process. The specific measure selected for our development is the kth order conditional entropy as defined below in equation (2.1), but in fact, this approach could be applied to similar measures such as the joint entropy of a process with memory. The conditional entropy measure was selected because it can be shown that the entropy rate of a process with memory is the $(k+1)$th order conditional entropy, where k is the maximum model order needed to completely specify the process. In Appendix A we provide a proof that is based on the Shannon-McMillan-Breiman theorem [1] for stationary ergodic processes.

2.2.1 Derivation of a Useful Entropy Measure

We define a set of n random variables from the source process by $X_1^n = \{X_1, X_2, \ldots, X_n\}$. These variables represent the set of all outcomes of n consecutive observations of the source process. A particular realization of these random variables will be denoted by $x_1^n = \{x_1, x_2, \ldots, x_n\}$. We will represent the set of possible realizations of a random variable by $x \in \mathcal{X}$. In general we will write, $x_1^n \in \mathcal{X}^n$.

The object of our development, the conditional entropy measure, is defined as follows [2]:

Definition 1 (nth order conditional entropy)

$$H(X_n|X_1^{n-1}) \triangleq -\sum_{x_1^n \in \mathcal{X}^n} p(x_1^n) \log p(x_n|x_1^{n-1}). \tag{2.1}$$

This calculation requires a large number of iterations and a large state space as indicated in section 2.1. Solution of a high order entropy estimation problem requires a technique that may be described as follows,

Proposition 1 *We can approximate an entropy measure by selecting a subset of the complete calculation space for exact computation and estimating the values not contained in the subset. The subset of computed measures can be chosen to adequately represent the conditional entropy measure by a Monte Carlo selection procedure. Further efficiencies can be obtained by adapting the probability distribution of the sampling procedure according to the relative magnitude of past measurements.*

In the following discussion we will develop a measure of the conditional entropy of a source process with memory using a sequential Monte Carlo sampling procedure [3] for selecting a portion of the calculation space for direct measurement.

Definition 2 (Partial conditional entropy measure) *We define a partial conditional*

entropy (PCE) measure, $\mathfrak{h}(n, x_1^j)$*, with a complexity of* $O(256^{n-j})$,

$$\mathfrak{h}(n, x_1^j) \triangleq - \sum_{x_{j+1}^n \in \mathcal{X}^{n-j}} p(x_1^n) \log p(x_n | x_{j+1}^{n-1}, x_1^j), \quad \textit{for } n > j \geq 1. \tag{2.2}$$

As noted above, the conditional entropy measure can be too complex for direct high order conditional entropy calculation. Using the partial conditional entropy measure, $\mathfrak{h}(n, x_1^j)$, we can express the conditional entropy, $H(X_n|X_1^{n-1})$, such that each of the partial entropy components is of sufficiently low complexity to make calculation of that component feasible. We write equation (2.1) in terms of a set of $|\mathcal{X}^j|$ $O(256^{n-j})$ PCE measurements,

$$H(X_n|X_1^{n-1}) = \sum_{x_1^j \in \mathcal{X}^j} \mathfrak{h}(n, x_1^j), \tag{2.3}$$

The PCE measure is useful tool for entropy estimation because it allows the conditional entropy measure of equation (2.3) to be separated into a set of $|\mathcal{X}^j|$ independent measures. This is the reason for our choice of the description *partial* conditional entropy.

Note 1 *We should point out that the quantity* $\mathfrak{h}(n, x_1^j)$ *is not equivalent to the conditional entropy measure,* $H(X_n|X_{j+1}^{n-1}, x_1^j)$*. We note that,*

$$H(X_n|X_{j+1}^{n-1}, x_1^j) = - \sum_{x_{j+1}^n \in \mathcal{X}^{n-j}} p(x_n|x_1^{n-1}) \log p(x_n|x_{j+1}^{n-1}, x_1^j).$$

Although the formulations are similar, since $p(x_1^n) \neq p(x_n|x_1^{n-1})$*, it is clear that* $\mathfrak{h}(n, x_1^j) \neq H(X_n|X_{j+1}^{n-1}, x_1^j)$*. We make this point in order to clarify our choice of description for the measure* $\mathfrak{h}(n, x_1^j)$*, which has a particular significance to our approach to entropy estimation.*

2.2.2 Probability Model Estimation

Until this point we have assumed an exact knowledge of the probability distributions needed for entropy calculation, but in fact, we must estimate these distributions from observations of the process of interest. In performing a study of the real conversational English speech process we found that developing these probability models was the most important and most difficult aspect of this research.

To develop a probability model, a data sequence of S consecutive speech samples was classified into $(S-n)$ n-tuples representing the set of random variables of interest, $x_1^n = \{x_1, x_2, \ldots, x_n\}$. Statistics were compiled for every unique n-tuple occurring in the data sequence. The initial set of observations produced maximum likelihood estimates of the joint probability distributions, and post-processing of these distributions produced maximum likelihood estimates of the conditional probability distributions. Clearly, these probability measures are determined by the statistics of the source process they represent. It is also clear that the probability estimations are dependent on the amount of data used to derive them. We will retain information about the size of the data sequence in our descriptions of these two probability measures as it is critical to the development of our entropy estimations.

We will refer to the maximum likelihood estimation of an nth order joint probability measure based on observations of S speech samples as $p_S(x_1^n)$, and the related conditional measure as $p_S(x_n|x_1^{n-1})$.

The conditional entropy measures obtained using these maximum likelihood estimations must also be distinguished by the observed sample size. We will denote the partial conditional entropy estimation computed at a sample size of S by,

$$\mathfrak{h}(S, n, x_1^j) \triangleq - \sum_{x_{j+1}^n \in \mathcal{X}^{n-j}} p_S(x_1^n) \log p_S(x_n|x_{j+1}^{n-1}, x_1^j), \quad \text{for } n > j \geq 1, \tag{2.4}$$

and the resulting estimate of the conditional entropy for an nth order model computed from

S observations as,

$$H_S(X_n|X_1^{n-1}) = \sum_{x_1^j \in \mathcal{X}^j} \mathfrak{h}(S, n, x_1^j), \tag{2.5}$$

2.3 Entropy Estimation with the PCE Measure

We will now define a set of partial conditional entropy measures as $(j+1)$ dimensional vectors. This set can be viewed as comprising points on a surface in $\mathbb{R}^+ \times \mathcal{X}^j$. We will take advantage of this interpretation of the PCE measures to illuminate our approach to conditional entropy estimation, which is to interpolate an estimate of the complete conditional entropy surface from the features of a set of known conditional entropy vectors. This estimate of the complete conditional entropy surface will then be used to compute an estimate of the conditional entropy of the process it represents.

We define a mapping, ψ, by vector function $\mathfrak{h}(S, n, x_1^j)$,

$$\begin{gathered} \psi : S^* \times \mathbb{Z}^+ \times \mathcal{X}^j \mapsto \mathbb{R}^+ \times \mathcal{X}^j \quad \text{s.t.,} \\ \forall\ \mathbf{a} = (S, n, x_1^j) \in S^* \times \mathbb{Z}^+ \times \mathcal{X}^j, \\ \psi(\mathbf{a}) = (\mathfrak{h}(S, n, x_1^j), x_1^j), \end{gathered} \tag{2.6}$$

where $S^* = \{S_1, S_2, \ldots, S_m\}$ is the set of sample sizes at which PCE measures may be computed, $\mathbb{Z}^+ = \{1, 2, 3, \ldots\}$, is the positive integers, and $\mathbb{R}^+ = [0, \infty)$.

This mapping allows us to express the results of our PCE calculations as a set of $(j+1)$ dimensional vectors in $\mathbb{R}^+ \times \mathcal{X}^j$. Let $\Upsilon_{\kappa_i,j} \subseteq \mathcal{X}^j$ be a set of κ_i j-dimensional vectors. Let $S_l \in S^*$ be a particular sample set size in S^* for which we will compute partial conditional entropy measures on the set $\Upsilon_{\kappa_i,j}$. This implies there can be a maximum of $|S^*| = m$ different sets $\Upsilon_{\kappa_i,j}$. Taking the image of $(S_l, n, \Upsilon_{\kappa_i,j})$ in ψ forms the set of partial conditional

entropy vectors computed at sample size S_l. We denote this set $\mathcal{H}_{S_l,n,\Upsilon_{\kappa_i,j}}$, where,

$$\mathcal{H}_{S_l,n,\Upsilon_{\kappa_i,j}} = \psi(S_l, n, \Upsilon_{\kappa_i,j}), \tag{2.7}$$

and note that $|\mathcal{H}_{S_l,n,\Upsilon_{\kappa_i,j}}| = \kappa_i$.

The set $\mathcal{H}_{S_l,n,\Upsilon_{\kappa_i,j}}$ contains κ_i PCE measures computed from probability distributions derived from sample sets of size S_l an the points in $\Upsilon_{\kappa_i,j}$. We will define a superset, $\mathcal{H}_{n,\Upsilon_{K,j}}$, consisting of sets of PCE vectors for all possible sample set sizes $S_l \in S^*$. This complete set is defined by,

$$\mathcal{H}_{n,\Upsilon_{K,j}} = \bigcup_{i=1}^{m}\bigcup_{l=1}^{m} \mathcal{H}_{S_l,n,\Upsilon_{\kappa_i,j}}$$

where $\Upsilon_{K,j} = \bigcup_{i=1}^{m} \Upsilon_{\kappa_i,j}$.

Our definitions to this point do not require uniqueness among the vectors in $\mathcal{H}_{n,\Upsilon_{K,j}}$. This allows the elements of $\Upsilon_{K,j}$ to define multiple PCE vectors in $\mathcal{H}_{n,\Upsilon_{K,j}}$, each distinguished by a unique sample size. We can ensure uniqueness among the entropy vectors by restricting the components of $\mathcal{H}_{n,\Upsilon_{K,j}}$ to those defined by one sample size for each element in $\Upsilon_{K,j}$. We define $\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$ to be the unique set of PCE vectors obtained from the entropy measurements performed on probability distributions derived from the largest available sample set for each element in $\Upsilon_{K,j}$. We write,

$$\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}} = \bigcup_{i=1}^{K} \psi(S^+, n, \mathbf{x}_i), \text{ where} \tag{2.8}$$

$$\forall\ \mathbf{x}_i \in \Upsilon_{K,j}, \quad S^+ = \max_{S^*} S_l,$$

$$\text{s.t.,}\ \ \psi(S_l, n, \mathbf{x}_i) \neq \emptyset.$$

Ideally, we would simply compute all conditional entropy measurements from probability distributions obtained from the largest available sample set, but in practice this may not always be feasible. In performing a study of the real conversational English speech

process, for example, we encountered computational difficulties that precluded a constant sample size. To allow for the variation in sample set sizes in the experimental work detailed in chapter 3, sections 3.4 and 3.5 we have derived a general methodology based on the set $\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$. This set comprises the best available estimate of these components of the conditional entropy surface.

2.4 Interpolating the Entropy Surface

In this section we will describe the process of interpolating the known vectors in $\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$ to obtain an approximation of the complete conditional entropy surface, $\tilde{\mathcal{H}}_{n,\mathcal{X}^j}$.

The exact nth order conditional entropy surface is the set of all partial conditional entropy vectors computed from exact models of the probability distribution of the process. This exact conditional entropy surface, $\mathcal{H}_{n,\mathcal{X}^j}$, is defined by,

$$\mathcal{H}_{n,\mathcal{X}^j} = \psi(S^\infty, n, \mathcal{X}^j), \tag{2.9}$$

where S^∞ represents a sample set large enough to yield an exact measure of the process statistics.

We approximate this surface by interpolating our best estimate of a subset of the complete surface, $\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$, to form the set $\tilde{\mathcal{H}}_{n,\mathcal{X}^j}$. This surface consists of the set of all $|\mathcal{X}^j|$ vectors necessary to compute an approximation of the conditional entropy.

The process required to interpolate the remainder of the conditional entropy surface is dependent on its dimensionality and shape. A variety of techniques are feasible and equally valid. In fact, *in the absence of knowledge of the exact nth order conditional entropy surface we cannot compare the accuracy of one interpolation technique to any other.* However, we can define the distance between two surfaces derived from different interpolation routines, say , $\tilde{\mathcal{H}}^A_{n,\mathcal{X}^j}$, and $\tilde{\mathcal{H}}^B_{n,\mathcal{X}^j}$, with a surface distance metric, $\rho(\tilde{\mathcal{H}}^A_{n,\mathcal{X}^j}, \tilde{\mathcal{H}}^B_{n,\mathcal{X}^j})$,

Definition 3 (Surface distance metric) *The distance between two surfaces, $\tilde{\mathcal{H}}^A_{n,\mathcal{X}^j}$ and $\tilde{\mathcal{H}}^B_{n,\mathcal{X}^j}$, is defined as*

$$\rho(\tilde{\mathcal{H}}^A_{n,\mathcal{X}^j}, \tilde{\mathcal{H}}^B_{n,\mathcal{X}^j}) = \sum_{\mathbf{x}_1^j \in \mathcal{X}^j} ||\tilde{\mathcal{H}}^A_{n,\mathcal{X}^j} - \tilde{\mathcal{H}}^B_{n,\mathcal{X}^j}||,$$

where $||\cdot||$ is the Euclidean norm.

For some interpolation function, f, we define a mapping from the best estimate of K entropy vectors, $\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$, to the set of all $|\mathcal{X}^j|$ vectors defining our best approximation of the conditional entropy surface, $\tilde{\mathcal{H}}_{n,\mathcal{X}^j}$. We define the mapping, f, by,

$$f : \mathbb{R}^+ \times \mathcal{X}^j \mapsto \mathbb{R}^+ \times \mathcal{X}^j \quad \text{s.t.,}$$
$$f(\psi(\cdot)) = \tilde{\mathcal{H}}_{n,\mathcal{X}^j}. \tag{2.10}$$

In this way our general interpolation function maps any set of $(j+1)$ dimensional PCE vectors, $\psi(\cdot)$, to a set of $|\mathcal{X}^j|$ $(j+1)$ dimensional vectors representing an approximation of the conditional entropy surface. Hence we may compute, $f(\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}) = \tilde{\mathcal{H}}_{n,\mathcal{X}^j}$.

The arbitrary mapping described above allows the entropy estimation technique to be formulated in a general manner. In chapter 3, section 3.5 we will describe a specific linear prediction algorithm for surfaces where $j = 1$, and a more complex interpolation routine involving a Delaunay triangulation of the set of computed vectors for surfaces where $j = 2$.

2.4.1 Interpolation Efficiency

The shape of the conditional entropy surface, which is determined by the statistics of the source process, affects the accuracy and efficiency of our entropy estimation technique. We define a property of smoothness for a conditional entropy surface,

Definition 4 (Smoothness) *A conditional entropy surface is smooth if for vector indices* $\{x_1^j, y_1^j\} \in \mathcal{X}^j$, *the distance between adjacent partial conditional entropy measures is bounded by some measure,* γ. *If we define adjacent vectors as those being separated by some maximum measure,* ϵ, *then a smooth conditional entropy surface is one that satisfies,*

$$\begin{aligned} ||\tilde{\mathcal{H}}_{n,x_1^j} - \tilde{\mathcal{H}}_{n,y_1^j}|| < \gamma \quad & s.t., \\ \forall \ \{x_1^j, y_1^j\} \in \mathcal{X}^j, & \\ ||x_1^j - y_1^j|| < \epsilon. & \end{aligned}$$

Obviously, processes demonstrating the property of smoothness in the conditional entropy surface can be better modelled than those without this property, since the predicted conditional entropy surface will be closer to the exact entropy surface for a given number of samples, K.

2.4.2 Computational Efficiency of the Estimation Technique

To simplify presentation of the conditional entropy estimate we define one more mapping, φ, to separate the scalar conditional entropy measure from the computed conditional entropy vector. Let φ be defined by,

$$\begin{aligned} \varphi : \mathbb{R}^+ \times \mathcal{X}^j \mapsto \mathbb{R}^+ \quad & \text{s.t.}, \\ \forall \ \mathbf{a} = (\mathfrak{h}(S, n, x_1^j), x_1^j) \in \mathbb{R}^+ \times \mathcal{X}^j, & \\ \varphi(\mathbf{a}) = \mathfrak{h}(S, n, x_1^j). & \end{aligned} \tag{2.11}$$

Our approximation of the conditional entropy measure, $\tilde{H}(X_n|X_1^{n-1})$, is simply the sum of all the computed and interpolated partial conditional entropy measures in $\tilde{\mathcal{H}}_{n,\mathcal{X}^j}$. Following the definitions leading to the conditional entropy measure in equation (2.5) we

compute,

$$\tilde{H}(X_n|X_1^{n-1}) = \sum_{x_1^j \in \mathcal{X}^j} \varphi(\tilde{\mathcal{H}}_{n,x_1^j}), \tag{2.12}$$

to determine an approximation of the conditional entropy based on the interpolated entropy surface. This estimate of the nth order conditional entropy of the process is obtained by interpolating our best estimates of a set of K PCE vectors computed over $\mathcal{X}^j$ to an estimate of the set of all PCE vectors in $\mathcal{X}^j$. The conditional entropy estimate is the sum of all these estimated PCE measures.

Our approximation to the conditional entropy measure was computed from just K samples of the full conditional entropy sample space. This approach to entropy estimation reduces the computation cost of the calculation by a factor of $\frac{K}{|\mathcal{X}^j|}$.

2.5 A Sequential Monte Carlo Procedure for Selecting Sample Points in $\mathcal{X}^j$

In this section we will outline an efficient algorithm for choosing the sample points $\Upsilon_{K,j} \subseteq \mathcal{X}^j$. This algorithm makes no prior assumptions about the shape of the entropy surface and begins as an ordinary Monte Carlo sampling procedure by selecting a sample point with a uniform probability distribution over the sample range, $\mathcal{X}^j$. We then increase the efficiency of the algorithm and the accuracy of our entropy estimation procedure by modifying the resolution of the sampling algorithm according to the importance of the sampled features to the computed entropy estimate. This is accomplished by weighting the sampling probability distribution according to the magnitude of the entropy vectors computed in prior PCE calculations.

This sequential Monte Carlo sampling procedure adapts the point selection distribution according to the features of the conditional entropy surface. The random nature of the algorithm ensures that it will select points from the entire range $\mathcal{X}^j$. This allows it to identify

the key features of the entropy surface. The adaptive component of the algorithm focuses the selection of PCE measurements around those regions that have the most significant impact on the conditional entropy measure. Even without *a priori* knowledge of the shape of the entropy surface, this sequential Monte Carlo selection algorithm ensures that the set of sample points will be concentrated around the most relevant features of the entropy surface. This algorithm helps to increase the accuracy of the entropy estimation obtained from a set of K observations.

We define a sample selection probability distribution as a function of the approximate conditional entropy surface computed from observations of $\kappa \leq K$ measurements. Let the set of κ sample points be $\Upsilon_{\kappa,j}$. We represent the intermediate entropy surface formed from this set of κ points by $\tilde{\mathcal{H}}^{\kappa}_{n,\mathcal{X}^j} = f(\psi(\Upsilon_{\kappa,j}))$ and the intermediate conditional entropy measure by $\tilde{H}^{\kappa}(X_n|X_1^{n-1}) = \sum_{x_1^j \in \mathcal{X}^j} \varphi(\tilde{\mathcal{H}}^{\kappa}_{n,\mathcal{X}^j})$. Note that we have omitted elements from the definition of $\psi(\cdot)$ to simplify this description. Given these intermediate measures we can define the probability of selecting x_1^j as the $(\kappa+1)$th sample point by $p((\kappa+1), x_1^j)$,

$$p((\kappa+1), x_1^j) = \begin{cases} \frac{1}{|\mathcal{X}^j|} & \text{if } \kappa = 0, \\ g\left(\frac{\varphi(\tilde{\mathcal{H}}^{\kappa}_{n,x_1^j})}{\tilde{H}^{\kappa}(X_n|X_1^{n-1})}\right) & \text{if } \kappa > 0,\ x_1^j \notin \Upsilon_{\kappa,j}, \\ 0 & \text{if } \kappa > 0,\ x_1^j \in \Upsilon_{\kappa,j}, \end{cases} \tag{2.13}$$

where $g(y)$ is any monotonically nondecreasing function defined over the range $0 \leq g(y) \leq 1$ for the range of inputs $0 \leq y \leq 1$. By defining $g(y)$ to have some of the properties of a logarithm, for instance, we can ensure good coverage of the range $\mathcal{X}^j$, even when the conditional entropy surface is highly peaked.

2.6 Sample Size Normalization

The best estimate of the conditional entropy surface, $\tilde{\mathcal{H}}_{n,\chi^j}$, was obtained from a set of vectors, $\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$, that was defined from probability measures based on a variety of sample sizes. This was done in anticipation of computational difficulties arising during the development of the maximum likelihood estimates of the probability distributions used in the PCE measures.

The inclusion of PCE measures from sample sets of varying sizes could result in inaccuracies in the predicted conditional entropy surface. If a probability model, $p_S(x_1^n)$, was computed from a sample that was not sufficiently large to ensure convergence, then the associated partial conditional entropy measure, $\mathfrak{h}(S, n, x_1^j)$, will not be exact. Furthermore, if the convergence characteristics of the source process *depend* on the vector x_1^j, then for a given sample size some partial conditional entropy measures may be overestimated while others are underestimated. We must determine these convergence characteristics and normalize the PCE measurements accordingly.

Our objective is to determine a scaling factor for each point $\mathfrak{h}(S^+, n, x_1^j) \in \widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$, as a function of the sample size, S^+, and the vector, x_1^j. We will then apply this factor to predict the PCE measure that would be obtained at some maximum feasible sample size, S_m. Applying this set of scaling factors to all points in $\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$ will yield a set of PCE vectors normalized to a single, maximal, sample size. This set will be called $\widehat{\mathcal{H}}_{S_m,n,\Upsilon_{K,j}}$. It represents a better estimate of the PCE measures in $\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$ as it utilizes the known convergence characteristics of the process to predict the measures that would be obtained at the largest feasible sample set size.

From this set we can compute the conditional entropy surface predicted by measures at the maximum feasible sample size,

$$\tilde{\mathcal{H}}_{S_m,n,\chi^j} = f(\widehat{\mathcal{H}}_{S_m,n,\Upsilon_{K,j}}),$$

and we can compute the conditional entropy estimate according to equation 2.12,

$$\tilde{H}_{S_m}(X_n|X_1^{n-1}) = \sum_{x_1^j \in \mathcal{X}^j} \varphi(\tilde{\mathcal{H}}_{S_m,n,x_1^j}).$$

We will now describe a general algorithm for normalizing the sample set sizes of the entropy measures in the set $\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$. First, we determine the convergence characteristics of the source process for a set of r sample points $\Upsilon_{r,j} \in \mathcal{X}^j$. We select these r sample points to give a good representation of the features of the entropy surface by applying the sequential Monte Carlo point selection algorithm discussed in section 2.5. We then compute the PCE measure for each point from probability models derived for all sample set sizes $S^C = \{S_1^C, S_2^C, \ldots, S_m\} \not\subseteq S^*$. This notation is meant to indicate that the elements $\{S_1^C, S_2^C, \ldots\}$ do not necessarily correspond to the elements $\{S_1, S_2, \ldots, S_m\} \in S^*$, but that the maximal sample size, S_m, contained in both sets is identical.

The set of $r * |S^C|$ partial conditional entropy measures will be represented by the set, $\mathcal{H}_{S^C,n,\Upsilon_{r,j}}$, where,

$$\mathcal{H}_{S^C,n,\Upsilon_{r,j}} = \bigcup_{S_i \in S^C} \psi(S_i, n, \Upsilon_{r,j}).$$

We may form a set of approximate conditional entropy surfaces representing the convergence characteristics of all points in $\mathcal{X}^j$ by interpolating this set of convergence characteristics. These convergence surfaces, as we shall refer to them, are computed with the interpolation function, f, discussed in section 2.4. We write the set of $|S^C|$ convergence surfaces as,

$$\tilde{\mathcal{H}}_{S^C,n,\mathcal{X}^j} = \bigcup_{S_i \in S^C} f(\psi(S_i, n, \Upsilon_{r,j})). \tag{2.14}$$

For each point in $\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$, we can now compute a normalizing factor to project the associated PCE measure to the maximal sample size, S_m. Let $\psi(S^+, n, x_1^j) \in \widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$ be a

vector computed at sample set size $S^+ \in S^*$ and point $x_1^j \in \Upsilon_{K,j}$. Let $\{S_u, S_l\} \in S^C$ be a pair of sample sizes bracketing S^+ such that,

$$|S_u, S^+| \leq |S_j, S^+| \quad \text{s.t.,} \ \forall \ S_j \in S^C, S_u \geq S^+,$$

and,

$$|S_l, S^+| \leq |S_k, S^+| \quad \text{s.t.,} \ \forall \ S_k \in S^C, S_l < S^+.$$

We use linear interpolation to predict a convergence surface, $\tilde{\mathcal{H}}_{S^+,n,\chi^j}$, for each sample set size $S^+ \in S^*$ as follows,

$$\tilde{\mathcal{H}}_{S^+,n,\chi^j} = \bigcup_{x_1^j \in \chi^j} (1-\alpha)\varphi(\tilde{\mathcal{H}}_{S_u,n,x_1^j}) + \alpha\varphi(\tilde{\mathcal{H}}_{S_l,n,x_1^j}),$$

where, $\alpha = \frac{S_u - S^+}{S_u - S_l}$, $S_u \geq S^+ \geq S_l$, and, $\{\tilde{\mathcal{H}}_{S_u,n,\chi^j}, \tilde{\mathcal{H}}_{S_l,n,\chi^j}\} \in \tilde{\mathcal{H}}_{S^C,n,\chi^j}$.

We apply this procedure to form the set of all $|S^*|$ convergence surfaces,

$$\tilde{\mathcal{H}}_{S^*,n,\chi^j} = \bigcup_{S^+ \in S^*} \tilde{\mathcal{H}}_{S^+,n,\chi^j}.$$

Finally, we normalize the set of all PCE vectors by projecting the measure that would be obtained at the maximal sample size, S_m. Reverting to the formulation of equation 2.8,

$$\hat{\mathcal{H}}_{S_m,n,\Upsilon_{K,j}} = \bigcup_{i=1}^{K} \psi(S^+, n, \mathbf{x}_i) \cdot \frac{\tilde{\mathcal{H}}_{S_m,n,\mathbf{x}_i}}{\tilde{\mathcal{H}}_{S^+,n,\mathbf{x}_i}}, \quad \text{where} \tag{2.15}$$

$$\forall \ \mathbf{x}_i \in \Upsilon_{K,j}, \quad S^+ = \max_{S^*} S_l,$$

$$\text{s.t.,} \ \ \psi(S_l, n, \mathbf{x}_i) \neq \emptyset, \ \text{and}$$

$$\{\tilde{\mathcal{H}}_{S_m,n,\mathbf{x}_i}, \tilde{\mathcal{H}}_{S^+,n,\mathbf{x}_i}\} \in \tilde{\mathcal{H}}_{S^*,n,\chi^j}. \tag{2.16}$$

We then derive the normalized conditional entropy surface,

$$\tilde{\mathcal{H}}_{S_m,n,\mathcal{X}^j} = f(\widehat{\mathcal{H}}_{S_m,n,\Upsilon_{K,j}}),$$

and compute our best estimate of the nth order conditional entropy measure of the process,

$$\tilde{H}_{S_m}(X_n|X_1^{n-1}) = \sum_{x_1^j \in \mathcal{X}^j} \varphi(\tilde{\mathcal{H}}_{S_m,n,x_1^j}). \tag{2.17}$$

This formulation utilizes information obtained from real PCE calculations and observed convergence characteristics to predict the outcome of the entropy calculation as if it were computed at S_m, the maximal sample size. This is a flexible scaling transformation that adapts the scaling factor depending on the dual factors of sample size and the point of interest. In chapter 3, section 3.5 we will have the opportunity to observe the effect of this normalizing transformation when we apply it to the results of our experimental work on the real speech process.

In this chapter we have defined the mathematical model for our approach to estimating the nth order conditional entropy of a general process. We have presented a formulation of this method in a very general manner to allow for its application to a variety of source processes and entropy measures. The model may be used to describe an entropy estimation of any order, n, from any number, K, of partial computations of complexity $O(|\mathcal{X}^j|)$.

Chapter 3

Entropy Rate Estimation of the Speech Process

Ease and speed in doing a thing do not give the work lasting solidity or exactness of beauty.

Plutarch. Life of Pericles.

3.1 Introduction

We desire a measure of the entropy rate of the speech process in order to define a lower bound on the distortion-free compressibility of the digital speech signal. This measure can serve as a reference, and perhaps a goal, for the designers of distortion-free and even low-distortion speech coders. The entropy rate of the source process is also required to determine the statistical redundancy of the digital speech signal. A measure of redundancy is useful for determining an information-theoretic measure of security for digitally encrypted source processes, as noted by Shannon in [4], and Massey in [5]. We will require this measure for several aspects of the development of a general model for secure speech communication

systems. The technique developed in chapter 2 will be applied here to determine an entropy rate estimate for speech. This measure will subsequently be used to develop quality and redundancy models in chapter 5, and security models in chapters 4 and 6.

To date, the exact entropy rate of the speech signal remains unknown because no analytic function has been derived to exactly characterize the process. In Appendix J we summarize some of the early studies of analytic models of the speech process. Work in the field of speech recognition has produced some representations based on mel-frequency cepstrum [6] and LPC cepstrum [7], and more recent models are based on human auditory characteristics [8], [9]. None of these methods has yet been completely successful at describing the spontaneous speech process of interest to the general model [10].

In our prior work [11], [12], [13] we approached the problem of entropy rate estimation by computing the exact conditional entropy of the male speech signal for first, second, third, and fourth order models of the process. We then fit an exponential curve of the form $y = A + \alpha e^{-\gamma n}$ to the points using minimum mean squared error as the matching criterion. The asymptote of this curve was taken to be our best estimate of the entropy rate. It was recognized that the accuracy of this prediction could be greatly increased by the addition of higher order measures to our model of the conditional entropy function.

The 64 kbps μ-law PCM format process we will define in sections 3.2 and 3.3 to be representative of conversational English speech consists of 8 bit samples. This results in a signal set cardinality of $|\mathcal{X}| = 2^8 = 256$, which in turn results in a total of 256^n states and $O(256^n)$ operations being required to specify an nth order probability model and process an nth order entropy calculation. Given these characteristics, we were previously unable to compute measures higher than fourth order in a reasonable time with the available computing resources.

The technique presented in chapter 2 will allow computation of conditional entropy measures that were previously infeasible. We begin the entropy estimation process in section 3.4 with the development of high order maximum likelihood estimates for the speech process

probability distribution. In section 3.5 we then apply the sequential Monte Carlo entropy estimation procedure to determine estimates of the fifth and sixth order conditional entropy measures of the speech process. In this section we also apply the conventional direct entropy measure to determining the first, second, third, and fourth order conditional entropies of the process.

In section 3.6 we present an extrapolated measure of the entropy rate and a set of bounds on that rate based on a non-linear weighted regression of the computed conditional entropy estimates. Our results predict an entropy rate of 22.3 kbps for distortion-free transmission of speech samples of 64 kbps μ-law PCM fidelity.

The material presented in this chapter is focused on predicting the entropy rate of the speech process, but the technique applied to the entropy estimation problem can clearly be applied to a wider variety of redundant source types. In future work, this approach may be applied to entropy estimation of a broader class of sources with memory, including facsimile and video. In addition, it was found that the technique for estimating a measure based on a large probabilistic model of a stochastic process could be applied to a wider variety of measurement requirements. In chapter 5, for instance, we found the estimation technique to be appropriate to determining a high order Kullback-Leibler information divergence measure. In chapter 6 we found it necessary to apply the estimation technique to a process consisting of compressed and encrypted speech, with significantly different results.

3.2 The Speech Process

The following is a precise description of the speech data we have used for all experimental aspects of this research.

Definition 5 (The speech process) *We define the speech process to be the set of all conversational English speech utterances collected with the fidelity of a telephone bandwidth channel, quantized into 8 bit μ-law PCM samples ($\mu = 255$) at a sampling rate of 8 kHz, according to CCITT recommendation G.711 [14].*

To derive accurate and relevant models of this process we obtained a very large corpora of speech data in a database of conversational English speech called SWITCHBOARD [15]. The details of the recording and data selection procedure for the SWITCHBOARD database are discussed in Appendix B. This particular representation of the speech process was chosen because it is the accepted standard for toll quality telephone bandwidth speech [16], [17], [18], [19].

The process chosen for this research consists of the most general realization of the speech process. It consists of speech samples collected from spontaneous conversations in English by speakers from every major dialect region in North America. The speech segments consist of male and female speakers of varying dialects and speech patterns having normal telephone conversations, complete with interruptions and idiosyncratic speech utterances.

3.3 Statistical Characteristics of the Speech Process

The conversational English speech process defined above has the following statistical characteristics:

3.3.1 Stationarity

Imagine if we were to somehow collect the entire history of all telephone conversations meeting our definition of the speech process. This collection would have a single mean value over the set of all samples, and similarly would have a constant second, third, and fourth moment, and so on, if the statistics for these measures were computed over the whole of the sample set. These qualities, the reader will observe, describe a process exhibiting strict-sense stationarity [20].

While it is true that the speech process as a whole is strictly stationary, individual segments of the process, for example an individual conversation, can have statistics that differ from those of the whole process. This is not a contradiction; the stationarity of the

speech process is defined in the abstract sense by assuming statistics can be computed at any time from an infinite set of process observations, while a short segment of speech yielding differing statistics can be viewed as an infinitesimally small portion of the entire process.

The maximum likelihood estimation procedure used to derive an estimate of the statistics of the general source process will produce a model that reflects this property of stationarity. It is difficult to prove the stationarity of our sample space by an experimental approach, however we argue that by observing a large portion of a process known to be stationary, with no preconditioning on the type of data collected for our sample, we can achieve a subset of the speech process that is also effectively stationary. The model obtained from this process is the worst possible case for effective source coding as it does not recognize any short-term temporal characteristics in the source process. The results of our calculations based on this model will therefore form the worst case, or upper bound, on entropy estimation of the speech process.

It is interesting to consider if we could achieve lower entropy estimations by adapting our modelling and entropy computation procedures to a set of specific features in the speech process. For example, by subdividing the speech process into male and female subsets and computing the entropy of each we could achieve more accurate measures of entropy for each type of speech. To compute the entropy of the process as a whole, we would then compute the sum of the two entropies, each weighted according to its relative frequency in the speech process.

Shannon derived bounds for the entropy of processes consisting of a mix of distinct sub-processes in [4]. If we have N distinct sub-processes that comprise the complete process in the ratios $\{p_1, p_2, \ldots, p_N\}$ such that $\sum_{i=1}^{N} p_i = 1$, and each sub-process has an entropy rate H_i, then the entropy of the complete process, H, is bounded by,

$$\sum_{i=1}^{N} p_i H_i \leq H \leq \sum_{i=1}^{N} p_i H_i - \sum_{i=1}^{N} p_i \log p_i.$$

An analogous case would be the use of two types of speech coders, one for male, and one for female speech signals. The data stream would consist of compressed data and an identifier for the type of traffic, male or female. Extending this analogy, we could obtain better and better rates of compression or lower entropies by subdividing the speech process into smaller and smaller categories of speech type. In the limit, we would achieve a large set of speaker-specific speech coders or probability models and an index of identifiers for all possible traffic types. This scenario, though infeasible, represents a best case, or lower bound on an estimate of the entropy measures of the speech process.

Speech coding algorithms typically take advantage of the varying short-term statistics of the speech signal to achieve good rates of compression [2], [21], [18], [22], [23]. In chapter 5 we will examine a variety of algorithms for inclusion as sample points in the general model. These algorithms are effective because they adapt to the local characteristics of the speech signal, regardless of the expected general averages that may be true for the speech process as a whole. Their behaviour is a compromise on the speaker-specific speech coding suggested above. Practical speech coders adapt to the characteristics of the current speaker in order to achieve good rates of compression, but do so imperfectly as they cannot feasibly model the exact characteristics of the speaker and so fail to achieve the true entropy rate of the source process.

We recognize that the entropy estimates obtained by our research form an upper bound for the speech process. The reader will note that the predicted entropy rate still compares favourably with low-distortion speech coder rates, and is considerably lower than the rate achieved by distortionless coders [24], [25]. It remains a topic for further research to determine if the tactic of classifying by speech types could yield a feasible technique for more specific entropy estimation, and to determine if the additional data necessary to index the set of speech types would counteract the apparent advantage of independent classification.

3.3.2 Limited Memory

Our observations of the speech process indicate that there is finite memory in the speech process. This is a natural result of the articulatory mechanism that produces the speech signal. Signal memory is most prominent in the production of vowel sounds, for example, but even these highly correlated phenomena are eventually terminated by a transition to another unit of speech [23], [26], [27]. Referring to the correlation statistics of the conversational speech process presented in figure 3.1 we see that for some index k, random variable X_{k+j+1} is independent of random variable X_j for all $j > 0$. These correlation statistics lead to the definition of the following property in the speech process pmf:

Definition 6 (Speech process pmf)

$$p(X_i|X_1^{i-1}) = p(X_i|X_{i-k}^{i-1}) \quad \forall \quad i > k, \tag{3.1}$$

3.4 Modelling the Speech Process for Conditional Entropy Calculations

In this section we will discuss the details of developing maximum likelihood estimates of the probability distributions of the speech process required for calculation of conditional entropy measures. We will identify two different types of modelling activity, one for an exact and direct calculation of a conditional entropy measure, and one for an indirect calculation based on entropy estimation by the Monte Carlo sampling approach discussed in chapter 2. For each of these two types of modelling activity we will consider the issue of model convergence discussed in section 2.2.2. We will present data indicating the convergence characteristics of the modelling process as a function of the sample set size that will be used in section 3.5.5 to normalize the conditional entropy estimates to a constant, maximal, sample size.

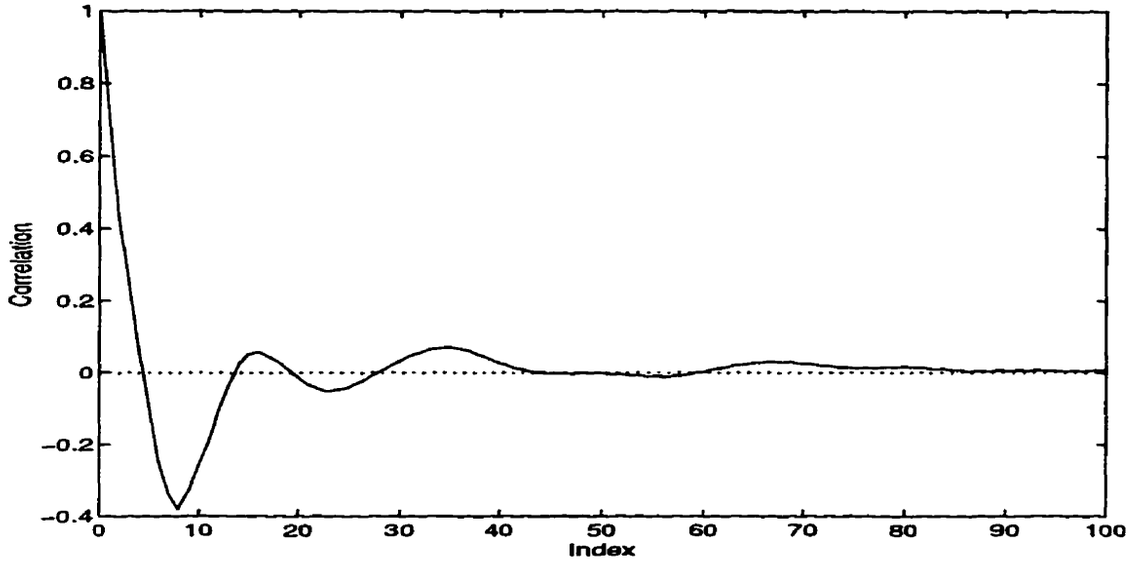


Figure 3.1: Long Term Correlation Statistics of the Speech Process

3.4.1 The Effect of Model Order on the Entropy Calculation

Our approach to the conditional entropy calculation was determined by complexity. As discussed in section 3.1, our definition of the conversational English speech process results in a total of 256^n states and $O(256^n)$ operations being required to specify an nth order probability model and process an nth order entropy calculation. At this level of computational complexity and storage requirements, we found the maximum feasible direct calculation to be of order 4. A typical fourth order conditional entropy calculation required approximately 40 hours elapsed time on a SparcStation II computer. A fifth order calculation would therefore require approximately 14 months to complete on the same computer. Were this feasible, it would still be necessary to address the problem of the greatly expanded state space of 256^5 elements.

The fourth order model therefore became the dividing line between calculations that could be computed directly, and those that would require the indirect approach of the Monte Carlo method to approximate.

3.4.2 Direct vs. Indirect Modelling Approaches

There were two steps to the procedure for calculating the various entropy measures required for entropy rate prediction. The first was to develop the appropriate probabilistic model for the desired entropy measure, and the second was to compute a conditional entropy measure based upon either the direct formula of equation (2.1) or the indirect Monte Carlo PCE formula developed in section 2.2 and summarized in equation (2.17).

Modelling the probability distributions as described in section 2.2.2 was found to be a challenging activity due to the large number of rare events in the process. In the interest of efficiency, it was necessary to determine the sample size beyond which additional observations no longer contributed significantly to the accuracy of entropy calculation. This question arose for both the direct and indirect calculation approaches, but caused the most difficulty in the indirect case where convergence was not always achieved with the available sample set sizes.

3.4.3 Direct Approach Convergence Characteristics

Models of the probability distributions of the speech process were computed in their entirety up to the complexity limit of fourth order. Presented in figures 3.2, 3.3, and 3.4 are views of the first, second, and third order probability distributions collected for the speech process. The first order probability mass function (pmf) is a complete rendition of the probability data, while the second and third order pmfs in figures 3.3, and 3.4 consist of only a subset of the total data defining the probability distributions. These two figures are intended to present an impression of the shape of the respective pmfs and not an exhaustive view of the complete data structure. It was not possible to present the vast amount of data collected for the fourth order probability distribution.

To determine the convergence of the maximum likelihood estimations of these probabilistic models, we chose to observe the effect of increasing sample set size on the resulting

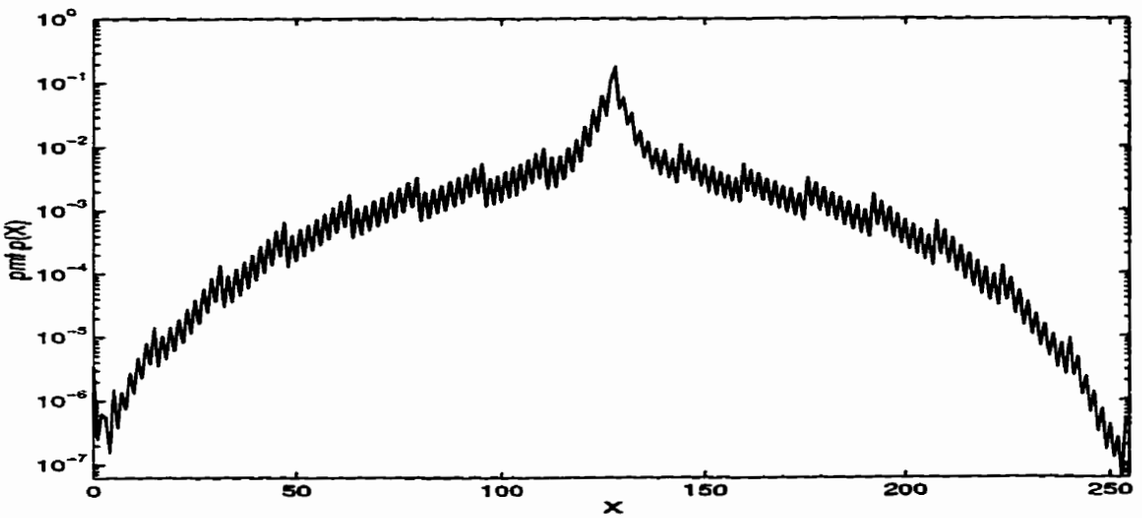


Figure 3.2: First Order pmf $p(X)$

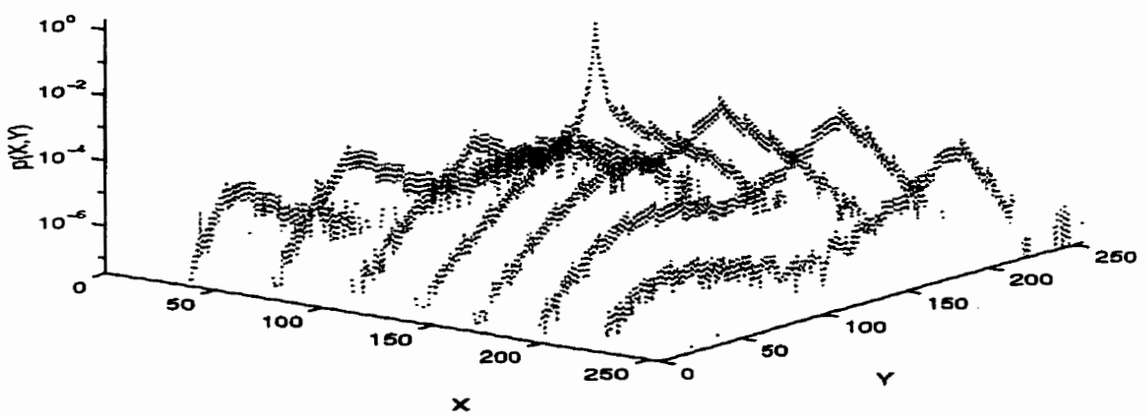


Figure 3.3: Second Order pmf Excerpts $p(X_1^2)$

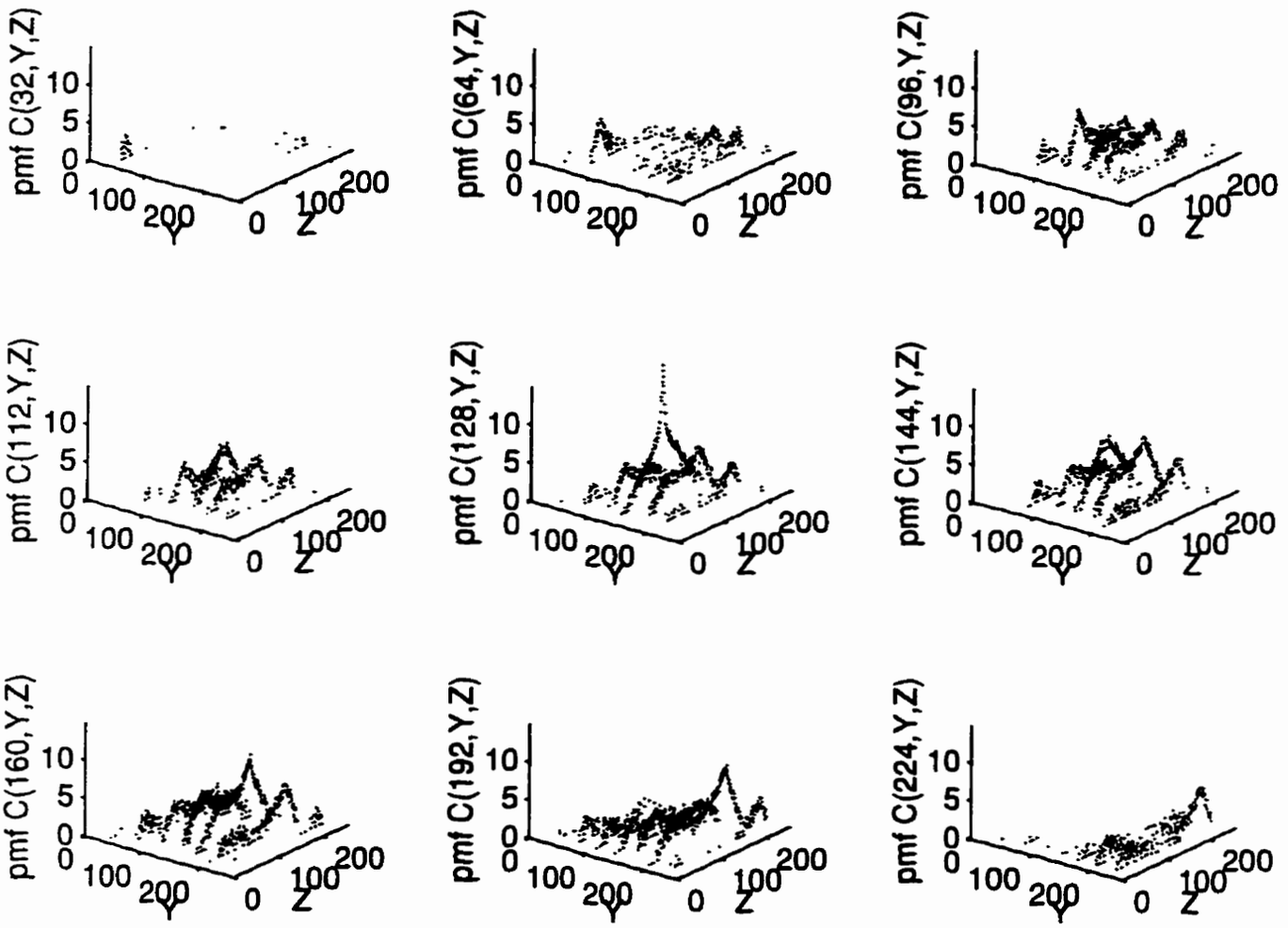


Figure 3.4: Third Order pmf Excerpts $p(X_1^3)$

conditional entropy measure. We computed first, second, third, and fourth order conditional entropy measures for progressively larger samples of speech according to equation (2.1). This approach did not measure the convergence of the probability models directly, but rather measured the effect of the sample set on a measure derived from the models.

The results of this study, presented in figure 3.5, indicate increasing stability in all four conditional entropy measures with additional sample data. The figure details average conditional entropy measures for all four model orders and the 95% confidence interval computed for those means. This data was compiled from observations of conditional entropies computed from multiple disjoint sample sets. The number of observations available for each data point was inversely proportional to the sample set size except at the maximal sample set size where we computed extra conditional entropy measures in order to obtain good confidence intervals. Table 3.1 details the number of observations obtained at each sample set size.

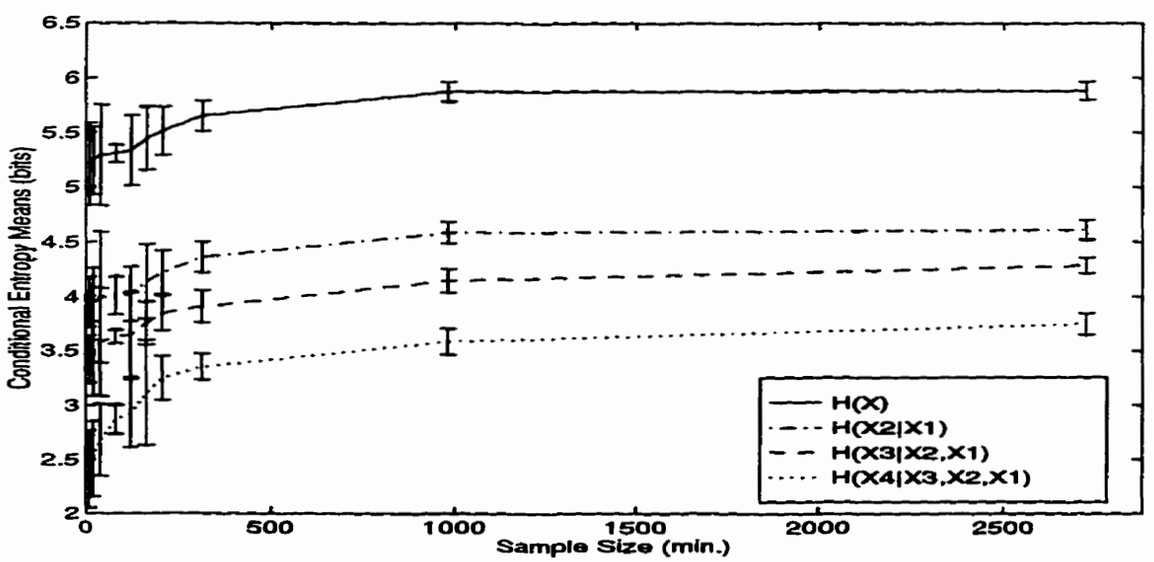


Figure 3.5: Convergence Properties of the Direct Entropy Calculations

We note that the measures computed for sample sets of less than 400 minutes indicate a

Model Order	Size (minutes)									
	11	21	42	84	126	167	209	314	982	2725
1	10	10	10	8	8	4	4	4	2	4
2	10	10	10	8	8	4	4	4	2	4
3	10	10	10	8	8	4	4	4	2	3
4	10	10	10	8	8	4	4	4	2	2

Table 3.1: Observation Count for Confidence Intervals in Figure 3.5

high variance, as would be expected for small samples, and generally underestimate the real conditional entropy. For larger sample sets, the probability of observing rare, but highly informative, events is increased and this affects the related conditional entropy measures. At a sample size of 2725 minutes of speech data, all four conditional entropy measures indicate a general convergence to a final value.

This simple study yielded valuable information about the speech process. From figure 3.5 we determined that we required at least 2725 minutes of conversational speech data to effectively represent the speech process for a conditional entropy calculation. We also acquired direct measures of the first, second, third, and fourth order conditional entropy measures at this maximal sample size. In section 3.5.1 we will summarize these measurements for use in predicting the entropy rate of the speech process.

3.4.4 Indirect Approach Convergence Characteristics

It was infeasible to compute $H(X_n|X_1^{n-1})$ for $n \geq 5$ due to the high computational complexity of the calculation. We therefore employed the indirect Monte Carlo entropy estimation approach presented in section 2.2 to estimate the fifth and sixth order conditional entropies. No conditional entropy measures were computed above a sixth order model, but the results of our research indicate that these measures are feasible using the Monte Carlo technique.

In order to minimize the dimensionality of the resulting conditional entropy surfaces, the PCE measures, $\mathfrak{h}(S, n, x_1^j)$, were chosen to have the maximum feasible complexity of $O(256^4)$. This is the highest complexity measure that we could compute in a reasonable

time. We selected this maximal feasible measure to minimize the dimensionality of the resulting conditional entropy surfaces and simplify the interpolation algorithms needed to define the approximate conditional entropy surfaces. A measure of lower complexity could have been used, but would have required a greater number of surface points to be computed for each entropy estimate.

From the definition in equation (2.9) we see that an exact nth order conditional entropy surface will consist of 256^{n-4} vectors. This means that the fifth order conditional entropy surface is defined by a set of 256 PCE vectors,

$$\mathcal{H}_{5,\mathcal{X}} = \bigcup_{x_1 \in \mathcal{X}} \psi(5, x_1),$$

and the sixth order surface is defined by 256^2 PCE vectors,

$$\mathcal{H}_{6,\mathcal{X}^2} = \bigcup_{x_1^2 \in \mathcal{X}^2} \psi(6, x_1^2).$$

Our fifth order entropy surface is therefore two dimensional and our sixth order surface is three dimensional.

Were we to extend our estimates to seventh order and higher, we would simply obtain higher dimensional surfaces. Conversely, were we able to perform computations of complexity $O(256^5)$ in reasonable time, we could reduce the dimensionality required for higher order estimates and compute the fifth order conditional entropy directly.

The indirect Monte Carlo approach outlined above requires the twin steps of the development of a probabilistic model followed by the calculation of a PCE vector to be repeated until there is a sufficient coverage of the vector space to allow accurate interpolation of the conditional entropy surface. To measure convergence in the indirect modelling and calculation approach we chose to observe the effect of an increasing sample set size on the results obtained by a small subset of partial conditional entropy measures.

Fifth Order Convergence

To determine the convergence of the fifth order entropy approximation we chose a subset of $r_1 = 5$ points, $\Upsilon_{r_1,1} = \{48, 90, 117, 121, 127\} \in \mathcal{X}$ by the Monte Carlo selection procedure detailed in section 2.5. We then computed PCE measures for each at these points at each of the sample set sizes in $S^{C_1} = \{540, 1081, 2162, 4324, 8648\}$ to obtain a set of convergence characteristics for the process,

$$\mathcal{H}_{S^{C_1},5,\Upsilon_{r_1,1}} = \bigcup_{S_i \in S^{C_1}} \psi(S_i, 5, \Upsilon_{r_1,1}).$$

This set gives good coverage of the region contributing most to the conditional entropy estimate and an overview of the convergence characteristics of other measures in that range.

The results, presented in figure 3.6, show a steady convergence to some final PCE value for all measures in $\Upsilon_{r_1,1}$. However, the convergence was considerably slower than that of the direct measurements in figure 3.5 and required a significantly larger sample size. Where the direct calculation measures began to converge with 2725 minutes of observed data, the indirect approach required at least 8648 minutes to stabilize the PCE measures.

Sixth Order Convergence

The Monte Carlo selection procedure was also used to determine the convergence characteristics of the sixth order entropy approximation. A subset of $r_2 = 6$ points, $\Upsilon_{r_2,2} = \{(127, 126), (127, 120), (108, 118), (76, 104), (53, 61), (25, 26)\} \in \mathcal{X}^2$ was selected. We then computed PCE measures for each of these points at each of the sample set sizes in $S^{C_2} = \{S^{C_1}, 17296\}$ to obtain a set of convergence characteristics for the process,

$$\mathcal{H}_{S^{C_2},6,\Upsilon_{r_2,2}} = \bigcup_{S_i \in S^{C_2}} \psi(S_i, 6, \Upsilon_{r_2,2}).$$

The Monte Carlo point selection procedure ensures that these points give an accurate representation of the important features on the conditional entropy surface. When we present

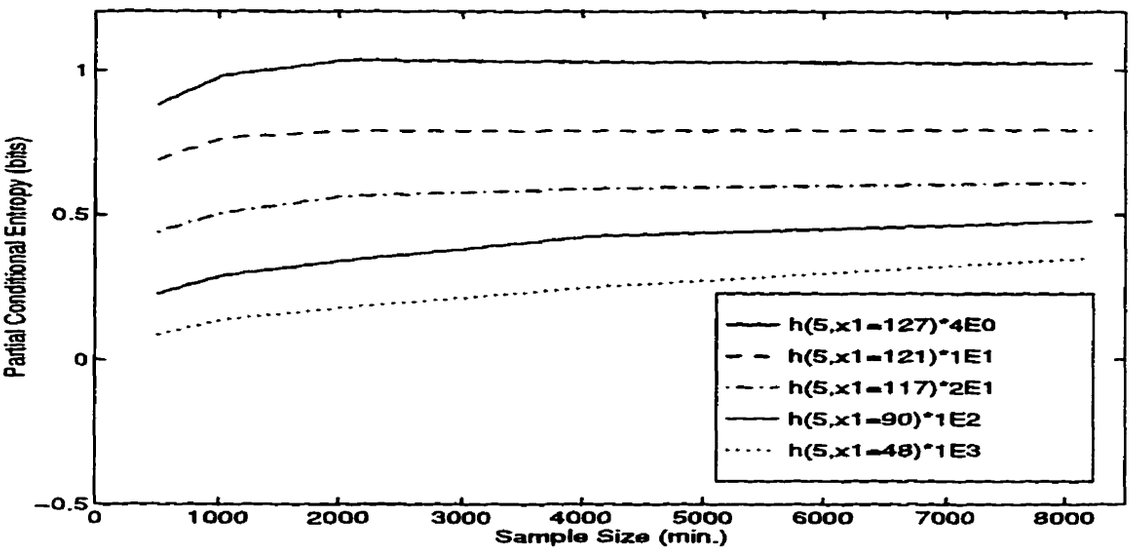


Figure 3.6: Convergence of Fifth Order Partial Conditional Entropy Measurements

the resulting entropy surface in section 3.5 we will see that some of these points were taken along the spine of a significant ridge of entropy contribution while others represent points away from this area of concentration.

The results of these measures, presented in figure 3.7, indicate an even slower convergence than the fifth order measures and a significant increase in the sample set size required to obtain convergence. The sixth order measures were found to exhibit an acceptable level of stability at a maximum sample size of 17296 minutes of speech data for only a subset of the observed points. This convergent subset consists of the measures taken at the points $\{(127,126),(127,120),(108,118),(76,104)\}$. The points observed outside this set do not appear to have converged to stable PCE measures at this sample size. It was not feasible to compute PCE measures for sample sizes beyond 17296 minutes in size, so the measures obtained at this maximal sample size have had to suffice for our best estimates of the PCE measures for points in $\Upsilon_{r_2,2}$.

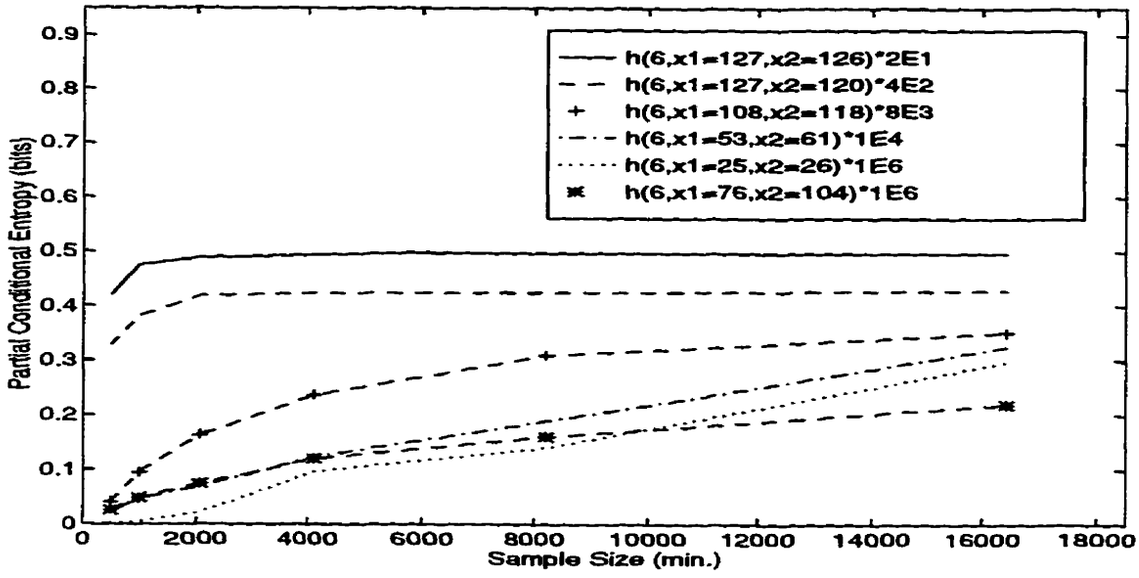


Figure 3.7: Convergence of Sixth Order Partial Conditional Entropy Measurements

A Note on Rates of Convergence

The slow convergence of the indirect measures presented here can be attributed to the definition of the PCE measure for a sample of size S in equation (2.4). Computing a probability measure for a PCE calculation consists of filtering a sample set of size S for n-tuples prefaced with the pattern of interest, $x_1^j \in \Upsilon_{K,j}$. Since only these filtered n-tuples are admitted to the probability model, it is necessary to provide greater volumes of speech data to the modelling procedure in order to capture sufficient statistics for an accurate representation of the process.

3.4.5 Sample Set Size Ranges for Speech Models

The data used for these experiments consisted of a large corpora supplied on a set of CDROMs, each containing approximately 1500 minutes of conversational speech data. This amounted to approximately 695 MBytes per CDROM, and a total of 7.3 GBytes of data for the maximal set used to determine the sixth order convergence characteristics. It was not

feasible to store this amount of data on magnetic disk, and as a result, accessing the data to build each conditional probability model required a significant amount of time. As it was necessary to extract the data from a set of CDROMs, each mounted sequentially, the sixth order convergence measures in figure 3.7 required an elapsed time of approximately 16 days for each point $x_1^2 \in \Upsilon_{r_2,2}$. The fifth order measures of figure 3.6 required approximately 9 days for each $x_1 \in \Upsilon_{r_1,1}$.

Due to these practical necessities, it was not feasible to use a sample set of the maximal size for every measure needed to form the conditional entropy surfaces, $\mathcal{H}_{S^*,n,\Upsilon_{K,j}}$. Smaller sample sets were used to calculate PCE vectors for all fifth and sixth order measures, except for those measures obtained from the study of convergence statistics. The sample set sizes used to generate conditional entropy points varied, depending on external conditions such as memory utilization, system load, and system failures. In all cases, the sample sets contained an identical core corpora of 2200 minutes of speech and additional data as was necessary to obtain accurate entropy measurements. Table 3.2 details the upper and lower sample size bounds, $S_{\min}$, and $S_{\max}$, for the set of sample sizes $S^* = \{S_{\min}, \ldots, S_{\max}\}$ applied to the set of all PCE measurements. As noted above, this does not include sample sizes for points computed at maximal sample set sizes in the convergence study.

Model Order	$S_{\min}$ (min.)	$S_{\max}$ (min.)
Fifth	2200	4500
Sixth	6500	13000

Table 3.2: Sample Size Ranges for Indirect Entropy Calculations

3.5 Computing Conditional Entropy Measurements

In this section we will utilize the conditional entropy measures obtained in section 3.4 to compute conditional entropy estimates of first, second, third, fourth, fifth, and sixth order realizations of the speech process. We will begin by summarizing the exact and direct

conditional entropy measures obtained in section 3.4.3. We will then apply the Monte Carlo sampling algorithm defined in section 2.2 to obtain estimates of the fifth and sixth order conditional entropy measures. To do this we will need to define two interpolation algorithms for generating an entropy surface from the set of entropy vectors obtained by Monte Carlo sampling. The first is based on simple linear interpolation in two dimensions, and the second, required for surfaces of three dimensions, is derived from a Delaunay triangulation of the computed vectors. Finally, in this section we will address the issue of normalizing the set of PCE measurements to a constant, maximal, sample set size as discussed in section 2.6. We will apply the knowledge obtained about the convergence characteristics of the process in section 3.4.4 to obtain estimates of the conditional entropy surfaces that would have been obtained at the maximum feasible sample size. We will use these modified surfaces to recompute our best estimate of the fifth and sixth order conditional entropies.

3.5.1 Exact Conditional Entropy Calculations

In section 3.4.3 we discussed the convergence characteristics of the first, second, third, and fourth order conditional entropy measures as a function of the sample set size. In figure 3.5 we presented the results of that study. We concluded that all four entropy measurements indicated a sufficient level of stability to be considered accurate representations of the first four nth order conditional entropy measures. In table 3.3 we present the means and confidence intervals of the sets of conditional entropy measures obtained at the maximum sample size of 2725 minutes of conversational speech data.

$H(X)$	$H(X_2\|X_1)$	$H(X_3\|X_1^2)$	$H(X_4\|X_1^3)$
$5.885 \pm .08$	$4.613 \pm .09$	$4.287 \pm .07$	$3.751 \pm .10$

Table 3.3: Conditional Entropy Means and 95% Confidence Intervals of the Speech Process

3.5.2 Interpolation Algorithms

The interpolation function, f, was defined in section 2.4 to be an arbitrary mapping from a set of K PCE vectors to the set of all $\mathcal{X}^j$ vectors defining the conditional entropy surface. In this section we will define the specific interpolation techniques used to estimate the conditional entropy surfaces, $\tilde{\mathcal{H}}_{5,\mathcal{X}}$, and $\tilde{\mathcal{H}}_{6,\mathcal{X}^2}$, from the best estimates of the PCE vectors, $\widehat{\mathcal{H}}_{5,\Upsilon_{K_1,1}}$ and $\widehat{\mathcal{H}}_{6,\Upsilon_{K_2,2}}$, or the sets of best estimates normalized to a maximal sample size, $\mathcal{H}_{\mathcal{S}_{m_1},5,\Upsilon_{K_1,1}}$ and $\mathcal{H}_{\mathcal{S}_{m_2},6,\Upsilon_{K_2,2}}$.

As discussed above, any number of more sophisticated interpolations could be applied to the data, but the advantage of these alternatives can not be quantified without a better model of the process under study. We are satisfied with taking the simplest approach to predicting the surface: a linear interpolation.

Two Dimensional Interpolation of the Entropy Surface

To interpolate the conditional entropy surface when $j = 1$ requires an interpolation function, f_2, in two dimensions. This function will take the set of computed PCE vectors as input and interpolate the set of all surface points, $\tilde{\mathcal{H}}_{5,\mathcal{X}^1}$.

The conditional entropy surface is partitioned into the set of regions defined by $\Upsilon_{K_1,1}$. Let $\{a, b, c, \ldots, y, z\} = \Upsilon_{K_1,1}$ define the ordered set of points such that $a < b < c < \ldots < y < z$. Then, $\{\mathcal{X}_{ab}, \mathcal{X}_{bc}, \ldots, \mathcal{X}_{yz}\} = \mathcal{X}_{az}$ defines all points in $\mathcal{X}$ between the minimum point, a, and the maximum point, z.

We can interpolate the portion of the conditional entropy surface within any of these regions, $\mathcal{X}_{ij} \in \mathcal{X}_{az}$ as follows,

$$\tilde{\mathcal{H}}_{5,\mathcal{X}_{ij}} = \bigcup_{p=i}^{j} \left\{ \varphi(\psi(5,i)) + (p-i)\frac{\varphi(\psi(5,i)) - \varphi(\psi(5,j))}{i-j}, p \right\}.$$

The interpolation function for the conditional entropy surface between points a and z

can then be described by the mapping,

$$f_2 : \mathbb{R}^+ \times \mathcal{X}^2 \mapsto \mathbb{R}^+ \times \mathcal{X}^2 \text{ s.t.,} \tag{3.2}$$
$$f_2(\cdot) = \bigcup_{\chi_{ij} \in \mathcal{X}_{az}} \tilde{\mathcal{H}}_{5,\chi_{ij}}$$

By ensuring that the minimum point is $a = 0$ and the maximum point is $z = |\mathcal{X}|$ we can interpolate all points on the conditional entropy surface.

Three Dimensional Interpolation of the Entropy Surface

To interpolate the conditional entropy surface when $j = 2$ requires an interpolation function, f_3, in three dimensions. This function will take as input the set of computed PCE vectors for the sixth order surface and interpolate the set of all points, $\tilde{\mathcal{H}}_{6,\mathcal{X}^2}$.

Interpolating in three dimensions is not as simple as the two dimensional linear interpolation procedure we applied to the fifth order estimate. We approach the problem by defining a set of triangular planes in $\mathbb{R}^+ \times \mathcal{X}^2$ to approximate the conditional entropy surface. If we define the vertices of these planes to be the set of computed PCE vectors, then the surface is defined by evaluating the planar equations at all points in $\mathcal{X}^2$.

To form the set of planes in $\mathbb{R}^+ \times \mathcal{X}^2$ we begin by triangulating the set of points, $\Upsilon_{K_2,2} \in \mathcal{X}^2$ with an unconstrained Delaunay triangulation algorithm [28], [29]. This procedure yields a unique set of τ triangles, $\mathcal{T}$ in the plane $\mathcal{X}^2$. We define the set of triangles $\mathcal{T}$ in terms of their vertices,

$$\mathcal{T} = \bigcup_{i \in \tau} \mathcal{T}_i = \bigcup_{i \in \tau} \{v_i^1, v_i^2, v_i^3\} \text{ s.t.,} \tag{3.3}$$
$$\{v_i^1, v_i^2, v_i^3\} \in \Upsilon_{K_2,2}.$$

For each triangle $\mathcal{T}_i \in \mathcal{T}$ we may form a projection in $\mathbb{R}^+ \times \mathcal{X}^2$ by taking the image of

each vertex in $\psi(\cdot)$. This operation forms the mapping,

$$\{(0, v_i^1), (0, v_i^2), (0, v_i^3)\} \mapsto \{\psi(v_i^1), \psi(v_i^2), \psi(v_i^3)\} \tag{3.4}$$
$$\forall \quad i \quad \text{s.t.,} \quad \mathcal{T}_i \in \mathcal{T}.$$

Note that we have omitted elements from the definition of $\psi(\cdot)$ to simplify this description.

We define the set of surface planes by,

$$\psi(\mathcal{T}) = \bigcup_{i \in \mathcal{T}} \psi(\mathcal{T}_i),$$

and note that the vertices of the triangular planes in $\mathbb{R}^+ \times \mathcal{X}^2$ are completely specified by the PCE vectors $\psi(6, \Upsilon_{K_2,2})$.

We define the three dimensional interpolation procedure as follows: The set $\mathcal{T}$ represents a unique and disjoint set of triangular planes in $\mathcal{X}^2$ that is generated from the Delaunay triangulation of $\Upsilon_{K_2,2}$. If we ensure that the corner points of the plane $\mathcal{X}^2$, $\{(0,0), (0, |\mathcal{X}|), (|\mathcal{X}|, 0), (|\mathcal{X}|, |\mathcal{X}|)\}$, are included in the set $\Upsilon_{K_2,2}$, then it can be shown that all points in $\mathcal{X}^2$ are inside a triangle $\mathcal{T}_i \in \mathcal{T}$ [30]. Hence, the set $\psi(\mathcal{T})$, which is formed from the projection of all triangles in $\mathcal{T}$ onto $\mathbb{R}^+ \times \mathcal{X}^2$ forms a unique surface that is defined for all points in $\mathcal{X}^2$. An illustration of this process is presented in figure 3.8.

We define the three dimensional interpolation function, f_3, as the projection of all points $x_1^2 \in \mathcal{X}^2$ onto the surface defined by $\psi(\mathcal{T})$. If we represent the projection of a point $x_1^2 \in \mathcal{X}^2$ onto the surface by $\psi(\mathcal{T}(x_1^2))$ then,

$$f_3 : \mathbb{R}^+ \times \mathcal{X}^2 \mapsto \mathbb{R}^+ \times \mathcal{X}^2 \quad \text{s.t.,} \tag{3.5}$$
$$f_3(\cdot) = \bigcup_{x_1^2 \in \mathcal{X}^2} \psi(\mathcal{T}(x_1^2)).$$

This procedure corresponds to a linear interpolation in three dimensions.

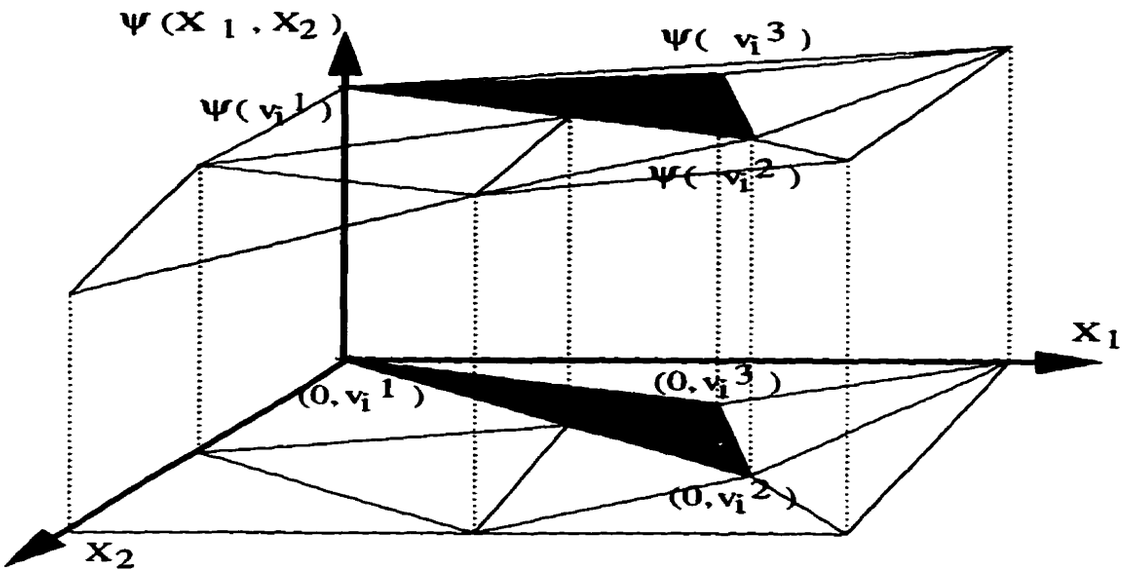


Figure 3.8: Example of Forming a Set of Triangular Surface Planes

3.5.3 Symmetry in the Speech Process

In developing the probability distributions of the speech process indicated in figures 3.2, 3.3 and 3.4 we noted strong evidence of symmetry as a function of the vector x_1^j. This same property of symmetry was found in sets of first, second, third, and fourth order PCE vectors. For example, we computed the complete two dimensional conditional entropy surfaces, $\mathcal{H}_{1,\mathcal{X}^1}$, $\mathcal{H}_{2,\mathcal{X}^1}$, $\mathcal{H}_{3,\mathcal{X}^1}$, $\mathcal{H}_{4,\mathcal{X}^1}$ from a 1300 minute sample of the speech process. The results, presented in figure 3.9, indicate a strong symmetry in x_1 and a close correspondence to the first order probability distribution of figure 3.2. This property of symmetry was also found to be present in the three dimensional conditional entropy surface developed to approximate the sixth order conditional entropy measure. The three dimensional surface, $\tilde{\mathcal{H}}_{6,\mathcal{X}^2}$, was found to resemble the second order probability distribution of figure 3.3.

We may take advantage of this property of the speech process to increase the accuracy of our entropy prediction method. We avoid performing calculations made redundant by the property of symmetry by reflecting all two and three dimensional PCE vectors across

the line of symmetry.

For two dimensional PCE vectors we perform the mapping,

$$\tilde{\mathcal{H}}_{n,x_1} \Leftrightarrow \tilde{\mathcal{H}}_{n,\{|\mathcal{X}|-x_1\}} \quad \forall \quad x_1 \in \mathcal{X}^1,$$

and for three dimensional PCE vectors we perform the mapping,

$$\tilde{\mathcal{H}}_{n,\{x_1,x_2\}} \Leftrightarrow \tilde{\mathcal{H}}_{n,\{|\mathcal{X}|-x_1\},|\mathcal{X}|-x_2\}} \quad \forall \quad \{x_1,x_2\} \in \mathcal{X}^2.$$

These simple transformations increase the efficiency of our calculations by effectively doubling the number of PCE vectors available to predict the conditional entropy surface of the speech process.

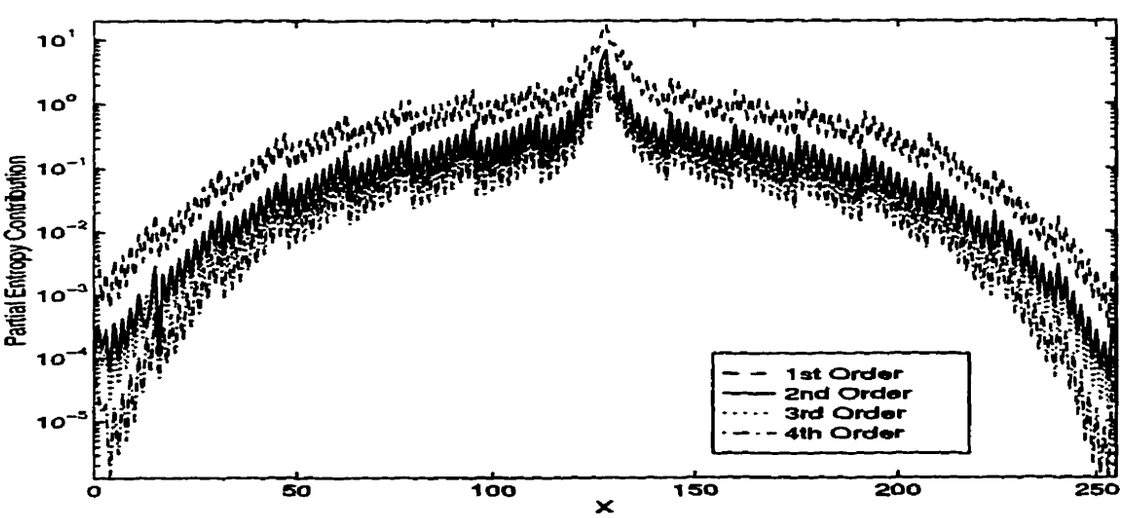


Figure 3.9: Symmetry in the Conditional Entropy Surfaces

3.5.4 Results of Entropy Estimation Measures

Verification of the Indirect Approach

Before presenting the results obtained for the fifth and sixth order conditional entropies, we wish to demonstrate the accuracy and efficiency of the indirect Monte Carlo approach by using it to calculate a known entropy value.

A set of 51 unique third order PCE vectors, $\mathcal{H}_{4,\Upsilon_{51,1}}$ was computed from a 1500 minute sample of the speech process. Due to the symmetry inherent in the source process, we could reflect these measures across the line of symmetry to form a set of 102 PCE vectors describing the entropy surface, $\mathcal{H}_{4,\Upsilon_{102,1}}$. The PCE vectors were then interpolated according to the interpolation function, f_2, described in section 3.5.2 to form the approximate conditional entropy surface,

$$\tilde{\mathcal{H}}_{4,\mathcal{X}^1} = f_2(\mathcal{H}_{4,\Upsilon_{102,1}}).$$

This surface is presented in figure 3.10. We then computed an approximation of the fourth order conditional entropy from this surface,

$$\tilde{H}(X_4|X_1^3) = \sum_{x_1 \in \mathcal{X}^1} \varphi(\tilde{\mathcal{H}}_{4,\mathcal{X}^1}).$$

The result of this calculation was a conditional entropy estimate of 3.405 bits/sample. Comparing this approximate measure to the exact fourth order conditional measure of 3.751 bits/sample presented in table 3.3 reveals a close correspondence between the direct and indirect approaches. Assuming the direct result to be accurate, we see that the indirect approach incurs an error of 9.22 % at an expense of only $\frac{51}{256} = 19.9\%$ of the work required for the direct calculation.

This comparison was obtained on the basis of a sample size which is shown in figure 3.5 to be less than that required for convergence of the direct measure. Were we to normalize the set of PCE vectors, $\tilde{\mathcal{H}}_{4,\mathcal{X}^1}$, to a maximal sample size according to the algorithm described

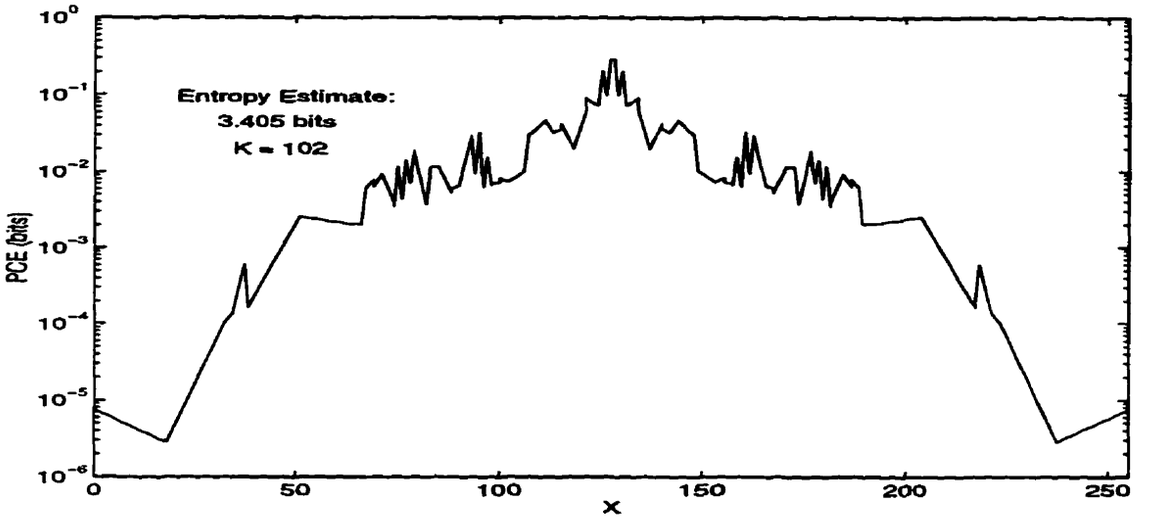


Figure 3.10: Approximate Fourth Order Partial Conditional Entropy Surface

in section 2.6 we would obtain a more accurate approximation of the fourth order measure. We could also compare the predicted fourth order measure to a measure computed directly from a sample set of similar size. An exact fourth order conditional entropy was computed to be 3.324 bits/sample for the sample set of 1300 minutes used to develop the results of figure 3.9. This similarly sized sample set yields an error estimate of only 2.44 %. We conclude that the indirect approach can achieve an accurate measure of the conditional entropy when a direct calculation is not feasible.

Raw Fifth Order Conditional Entropy Calculation

Having verified the feasibility of entropy estimation from a Monte Carlo sampling approach with the fourth order measure, we applied the same technique to develop a set of PCE vectors, $\mathcal{H}_{S^*,5,\Upsilon_{K_1,1}}$. In total, 41 unique fourth order PCE vectors were computed over a range of sample sizes, S^*. Due to the symmetry in the source process, this resulted in a total of $K_1 = 82$ PCE vectors to define the surface. Due to computational difficulties, the

elements of $\mathcal{H}_{S^*,5,\Upsilon_{K_1,1}}$ were computed at varying sample sizes in S^* such that,

$$\mathcal{H}_{S^*,5,\Upsilon_{K_1,1}} = \bigcup_{S_i \in S^*} \psi(S_i, 5, \Upsilon_{K_1,1}).$$

Selecting the set of measurements obtained at the maximum available sample set size for each element in $\Upsilon_{K_1,1}$, we formed our best estimate of the raw PCE data, $\widehat{\mathcal{H}}_{5,\Upsilon_{K_1,1}}$ as described by equation 2.8.

The fifth order surface defined by $\widehat{\mathcal{H}}_{5,\Upsilon_{K_1,1}}$ had the same dimensionality as the fourth order surface described above, allowing us to proceed in the same manner to compute the fifth order conditional entropy estimate. The PCE vectors were interpolated according to the interpolation function, f_2, to form the approximate conditional entropy surface,

$$\tilde{\mathcal{H}}_{5,\mathcal{X}^1} = f_2(\widehat{\mathcal{H}}_{5,\Upsilon_{K_1,1}}).$$

This fifth order conditional surface approximation is presented in figure 3.11. We then computed an approximation of the fifth order conditional entropy from this surface,

$$\tilde{H}(X_5|X_1^4) = \sum_{x_1 \in \mathcal{X}^1} \varphi(\tilde{\mathcal{H}}_{5,\mathcal{X}^1}).$$

The result of this calculation was a fifth order conditional entropy estimate of 2.994 bits/sample. This estimate was computed from the non-ideal set of PCE vectors computed at varying sample sizes. In section 3.5.5 we will compute a better estimate by first normalizing the set of PCE vectors to a constant, maximal, sample size.

Raw Sixth Order Conditional Entropy Calculation

To compute an estimate of the sixth order conditional entropy we produced a set of 101 unique fourth order PCE vectors from samples sizes in the set S^*. Applying the property

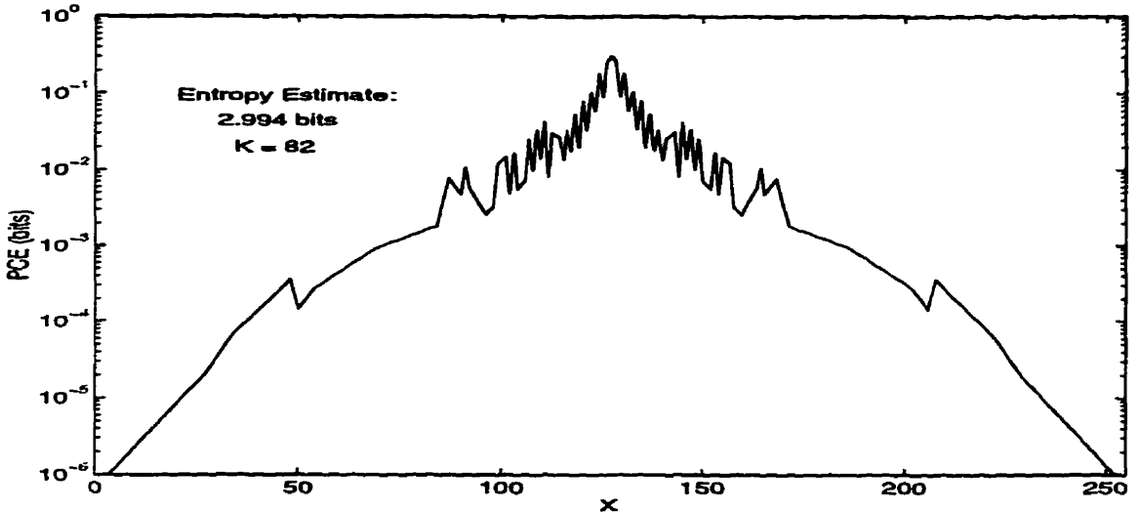


Figure 3.11: Approximate Fifth Order Partial Conditional Entropy Surface

of symmetry described for three dimensional vectors in section 3.5.3 yielded at total of $K_2 = 202$ PCE vectors, $\mathcal{H}_{S^*,6,\Upsilon_{K_2,2}}$, with which to predict the sixth order conditional entropy surface. Like the fifth order PCE vectors, this set of measures was computed at varying sample sizes such that,

$$\mathcal{H}_{S^*,6,\Upsilon_{K_2,2}} = \bigcup_{S_i \in S^*} \psi(S_i, 6, \Upsilon_{K_2,2}).$$

Selecting the set of measurements obtained at the maximum available sample set size for each element in $\Upsilon_{K_2,2}$, we formed our best estimate of the raw PCE data, $\widehat{\mathcal{H}}_{6,\Upsilon_{K_2,2}}$ as described by equation 2.8.

The sixth order surface defined by $\widehat{\mathcal{H}}_{6,\Upsilon_{K_2,2}}$ is three dimensional. We therefore proceeded by interpolating an estimate of the conditional entropy surface with the three dimensional interpolation function, f_3. We computed the sixth order conditional entropy surface ap-

proximation as,

$$\tilde{\mathcal{H}}_{6,\mathcal{X}^2} = f_3(\widehat{\mathcal{H}}_{6,\Upsilon_{K_2,2}}).$$

Presented in figure 3.12 is the Delaunay triangulation of the points $\Upsilon_{K_2,2}$ used to interpolate the approximate conditional entropy surface.

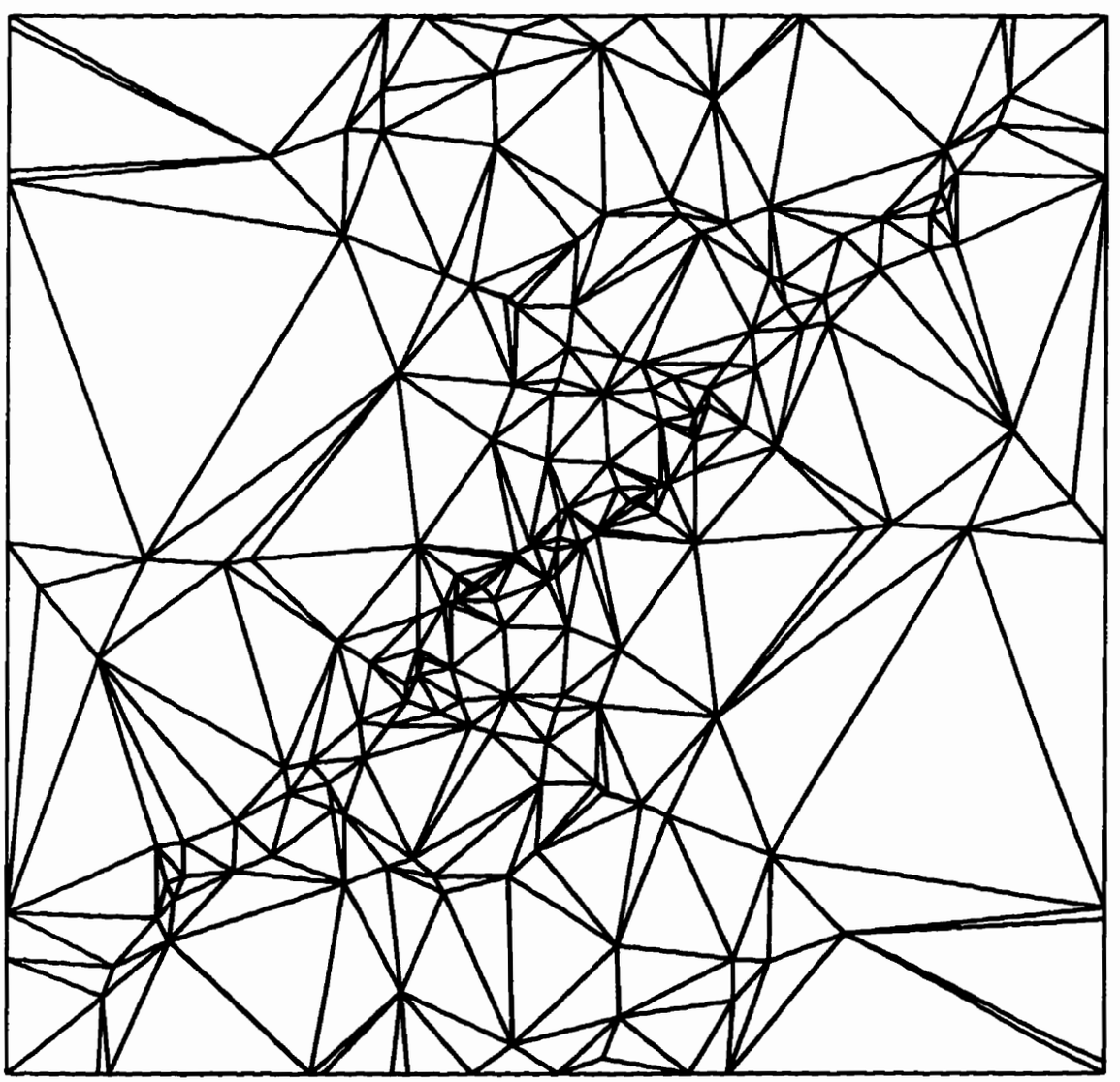


Figure 3.12: Delaunay Triangulation of PCE Points in $\Upsilon_{K_2,2}$

The resulting approximate conditional entropy surface is detailed in figures 3.14, and 3.16. These figures indicate the shape of the surface and include a set of contours on the surface at levels 10^{-i}, $i = 2, 3, \ldots 6$. Figure 3.16 also presents a measure of the conditional entropy volume contained in each of the contours displayed on the entropy surface. We computed an approximation of the sixth order conditional entropy from this surface,

$$\tilde{H}(X_6|X_1^5) = \sum_{x_1^2 \in \mathcal{X}^2} \varphi(\tilde{\mathcal{H}}_{6,\mathcal{X}^2}).$$

The result of this calculation was a sixth order conditional entropy estimate of 2.864 bits/sample. Again we note that this estimate was computed from the non-ideal set of PCE vectors computed at varying sample sizes. In section 3.5.5 we will compute a better estimate from a set of PCE vectors normalized to a constant, maximal, sample size.

3.5.5 Normalization to the Maximal Sample Size

The results of the direct entropy calculations presented in figure 3.5 were noted to converge to the final values noted in table 3.3 at a sample size of 2725 minutes of speech. Consequently, these results can be applied without modification to our estimate of the entropy rate of the speech process.

Our observations of the convergence characteristics of the fifth and sixth order calculations indicate that the results obtained by measurements on the sample sizes outlined in table 3.2 are not exact. More accurate measures can be obtained by normalizing the measures in the set $\mathcal{H}_{S^*,n,\Upsilon_{K,j}}$ to a constant, maximal, sample set size following the technique outlined in section 2.6. This procedure allows us to predict the entropy measure that would be obtained if the calculation had been performed on the maximal sample set.

Fifth Order Surface Normalization

The set of fifth order PCE vectors $\mathcal{H}_{S^*,5,\Upsilon_{K_1,1}}$, compiled from sample sets of sizes over the range indicated in table 3.2, was normalized to a maximal sample size of $S_{m_1} = 8648$ minutes.

This was accomplished by applying the standard 2 dimensional interpolation technique described in section 3.5.2 to the convergence data in figure 3.6 to produce a set of convergence surfaces, $\tilde{\mathcal{H}}_{S^*,5,\mathcal{X}^1}$. A scaling factor was then computed for each point $\psi(S,5,x_1) \in \mathcal{H}_{S^*,5,\Upsilon_{K_1,1}}$ to form the set $\widehat{\mathcal{H}}_{S_{m_1},5,\Upsilon_{K_1,1}}$ as described by equation (2.15).

The interpolation function was then applied to this normalized set to obtain our best estimate of the fifth order conditional entropy. The conditional entropy surface plotted in figure 3.13 was obtained from,

$$\tilde{\mathcal{H}}_{S_{m_1},5,\mathcal{X}^1} = f_2(\widehat{\mathcal{H}}_{S_{m_1},5,\Upsilon_{K_1,1}}),$$

and the fifth order conditional entropy estimate of 3.262 bits/sample was computed from the equation,

$$\tilde{H}_{S_{m_1}}(X_5|X_1^4) = \sum_{x_1 \in \mathcal{X}^1} \varphi(\tilde{\mathcal{H}}_{S_{m_1},5,\mathcal{X}^1}).$$

The effects of the normalization procedure may be observed by comparing the original fifth order surface in figure 3.11 to the normalized surface in figure 3.13. We note a slight reduction in the peakedness of the surface and a general increase in the magnitude of the most significant PCE vectors near the mean of the process. These modifications were due to the observed convergence characteristics of the more probable vectors and had the most significant impact on the entropy estimate. We also note a reduction in the variance of some PCE vectors, particularly in the regions of lower probability. This smoothing effect is an indication of the normalization process.

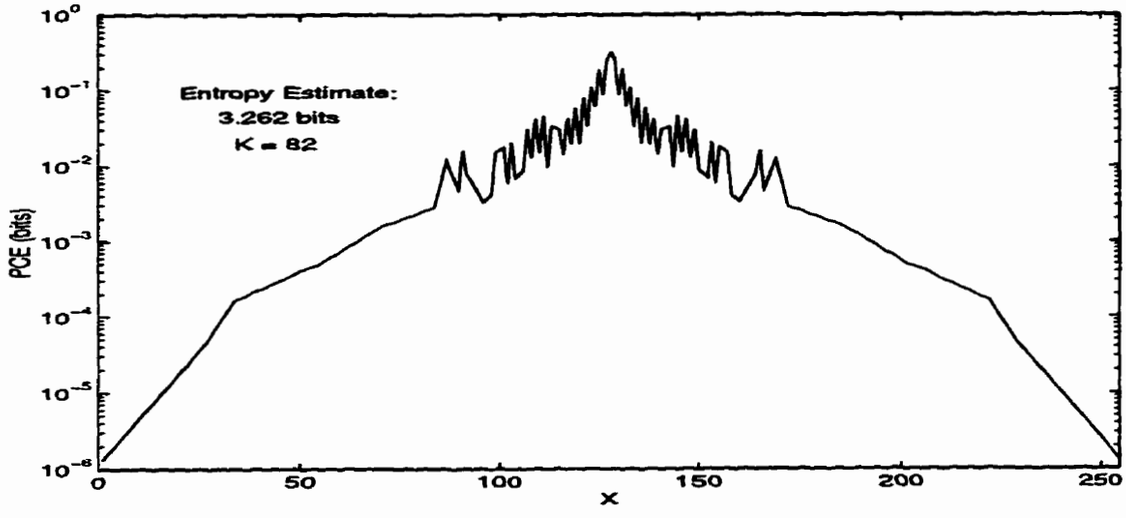


Figure 3.13: Normalized Fifth Order Conditional Entropy Surface

Sixth Order Surface Normalization

Similarly, the set of sixth order PCE vectors, $\mathcal{H}_{S^*,6,\Upsilon_{K_2,2}}$, was normalized to a maximal sample size of $S_{m_2} = 17296$ minutes. This was accomplished by applying the standard 3 dimensional interpolation technique presented in section 3.5.2 to produce a set of convergence surfaces, $\tilde{\mathcal{H}}_{S^*,6,\chi^2}$, from the convergence measures presented in figure 3.7. A scaling factor was then computed for each point $\psi(S,6,\boldsymbol{x}_1^2) \in \mathcal{H}_{S^*,6,\Upsilon_{K_2,2}}$ to form the set $\widehat{\mathcal{H}}_{S_{m_2},6,\Upsilon_{K_2,2}}$ as described by equation (2.15).

The interpolation function was then applied to obtain a best estimate of the sixth order conditional entropy. The conditional entropy surface was obtained from

$$\tilde{\mathcal{H}}_{S_{m_2},6,\chi^2} = f_3(\widehat{\mathcal{H}}_{S_{m_2},6,\Upsilon_{K_2,2}}),$$

and the sixth order conditional entropy estimate of 3.121 bits/sample was computed from

the equation,

$$\tilde{H}_{S_{m_2}}(X_6|X_1^5) = \sum_{x_1^2 \in \mathcal{X}^2} \varphi(\tilde{\mathcal{H}}_{S_{m_2},6,\mathcal{X}^2}).$$

Figures 3.15, and 3.17, detail the shape of the normalized entropy surface. Figure 3.15 presents a three dimensional image of the surface with a set of contours at levels 10^{-i}, $i = 2, 3, \ldots, 6$. Figure 3.17 details the volume of conditional entropy contained within each of the contours on the entropy surface. We note a smoother and more continuous form to the surface that results from scaling the set of PCE measures to a single sample set size. Comparing the volumes contained in contours of the unscaled conditional entropy surface in figure 3.16 to the volumes in the scaled conditional entropy surface of figure 3.17, we see how the normalization routine reshapes the surface. In this case, the central peak of the entropy surface has been decreased, while the measures in the diagonal region surrounding the peak have been increased. These changes are consistent with the convergence characteristics observed in figure 3.7.

Summary of Best Conditional Entropy Estimates

In table 3.4 we summarize the best estimates obtained for conditional entropy measures from first to sixth order. This table includes observations on the convergence characteristics of the entropy measures and a measure of the fraction of the complete $O(256^n)$ calculation computed to obtain each entropy estimate.

Order	Measure	Type	Computed	Convergence
$H(X)$	5.885	Direct	100%	Good
$H(X_2\|X_1)$	4.613	Direct	100%	Good
$H(X_3\|X_1^2)$	4.287	Direct	100%	Good
$H(X_4\|X_1^3)$	3.751	Direct	100%	Good
$H(X_5\|X_1^4)$	3.262	Indirect	16.0%	Good
$H(X_6\|X_1^5)$	3.121	Indirect	0.15%	Fair

Table 3.4: Summary of Best Conditional Entropy Estimates

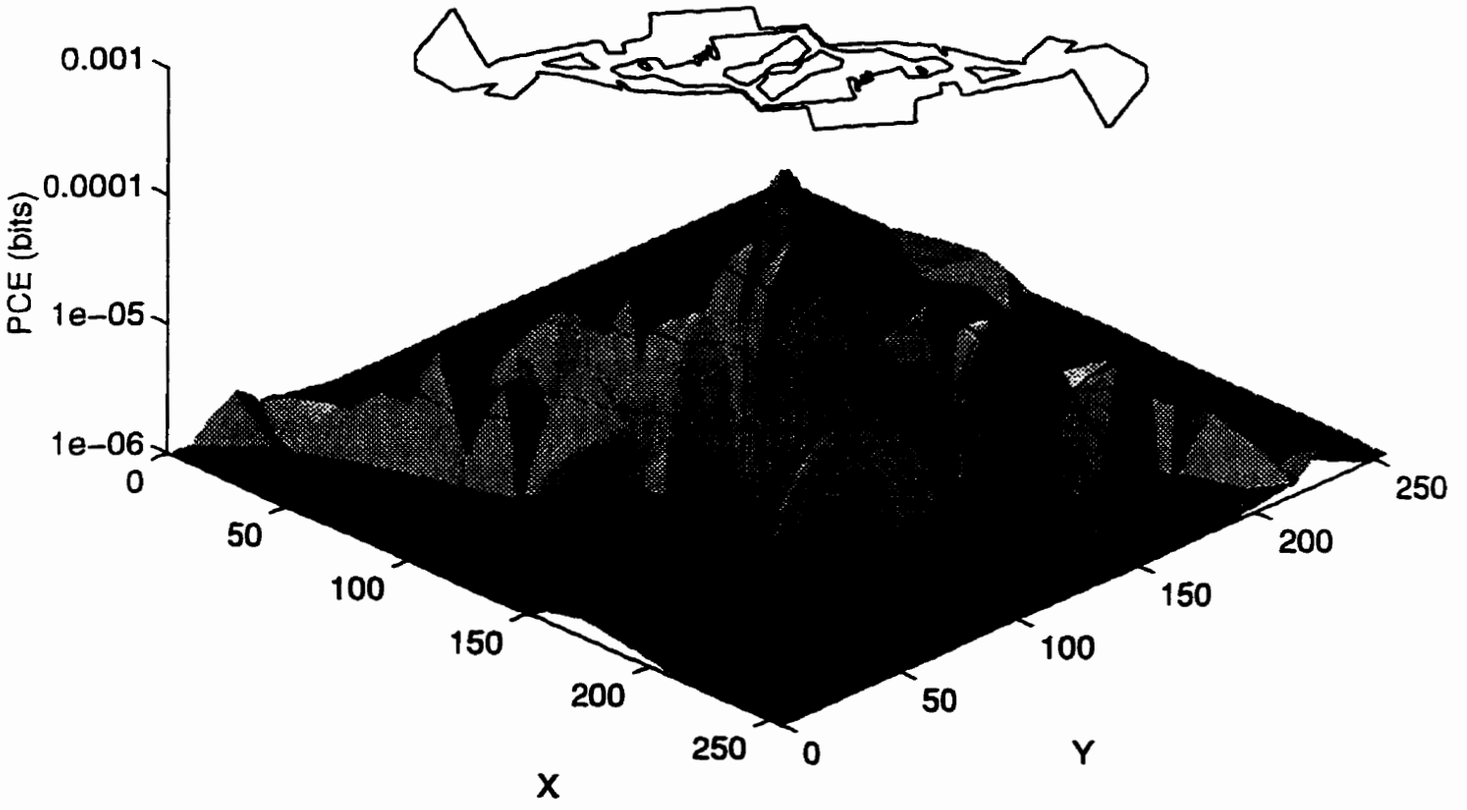


Figure 3.14: Approximate Sixth Order Conditional Entropy Surface with Contour

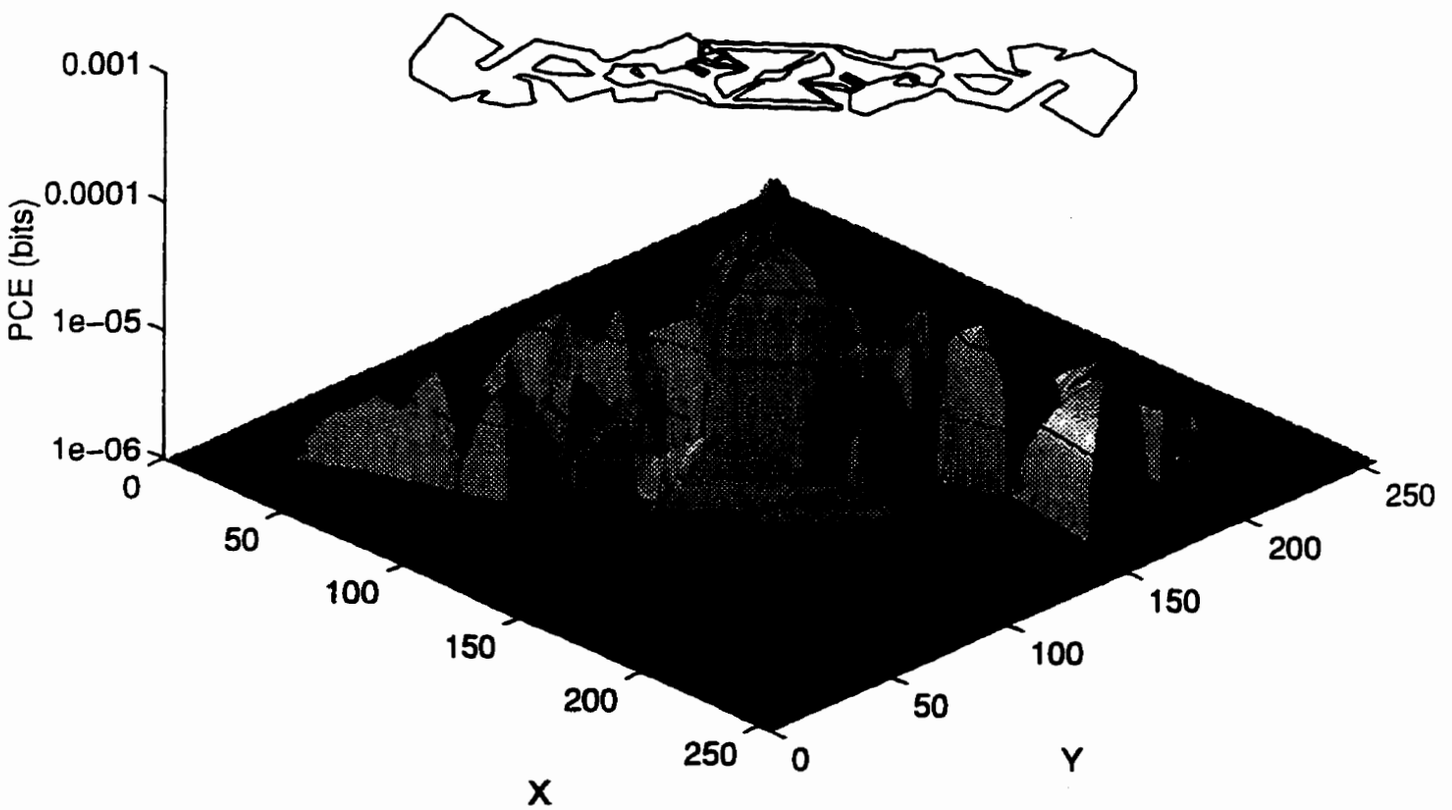


Figure 3.15: Normalized Sixth Order Conditional Entropy Surface with Contour

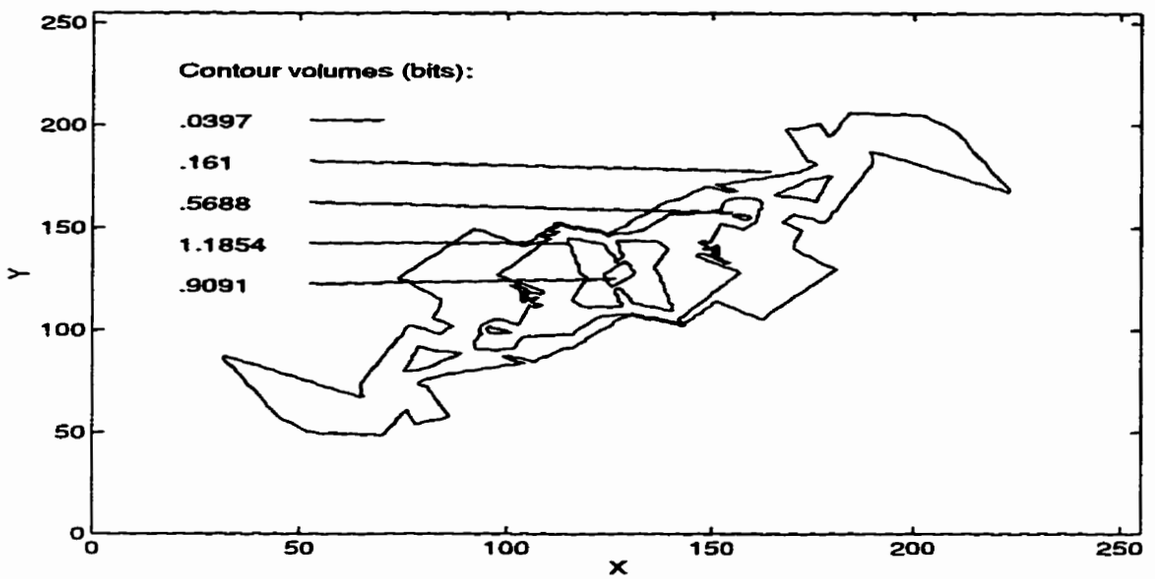


Figure 3.16: Approximate Sixth Order Conditional Entropy Contour

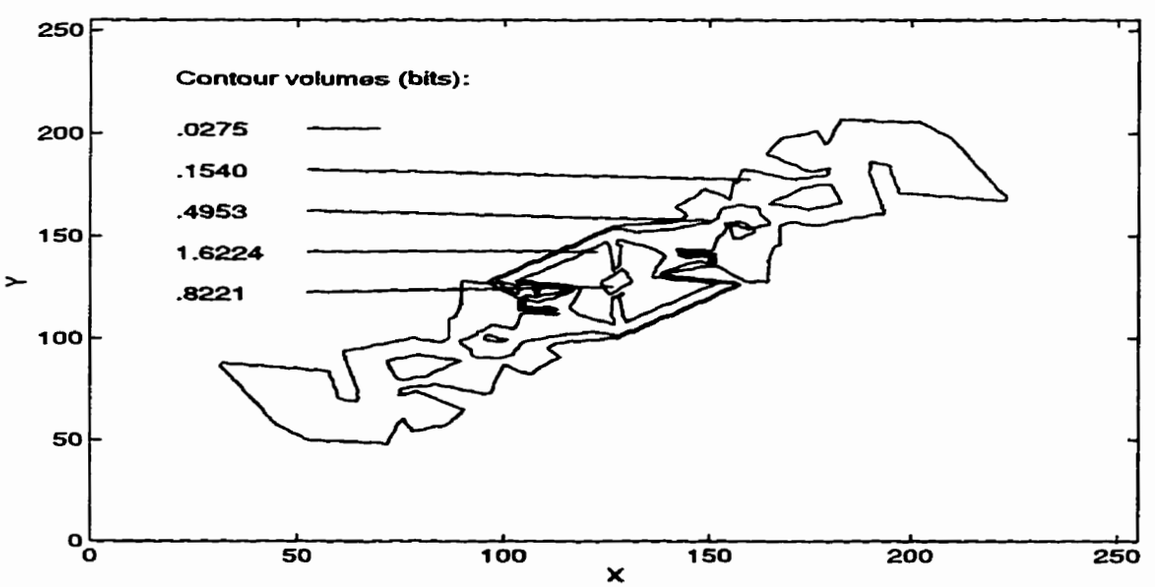


Figure 3.17: Normalized Sixth Order Conditional Entropy Contour

3.6 Entropy Rate Estimation

We have computed the first to fourth order conditional entropy of the speech process exactly, and have obtained estimates of the fifth and sixth order conditional entropies. A summary of our results is presented in table 3.4. In Appendix A we present a proof indicating that the entropy rate is equivalent to the $(k+1)$th order conditional entropy for processes that can be fully specified with a k dimensional probability distribution. This proof is an extension of the general result of the Shannon-McMillan-Breiman theorem [1], that reduces the computational requirement for an entropy rate calculation to a finite $(k+1)$th order conditional entropy calculation.

To determine the maximum order, k, required to specify the probabilistic model of the speech process, we refer to the long term correlation statistics of the speech process presented in figure 3.1. Recalling that these statistics were computed for the very broad range of English speakers allowed by our definition of the general speech process, we are not surprised to observe that the full specification of the speech process pmf would require a model of at least 40th order. In Appendix E we have tabulated a conservative estimate of the number of operations required for the first six conditional entropy measures to be on the order of 1.2×10^{14} integer and 1.8×10^{13} floating point operations. Further conditional entropy calculations are infeasible with the available computing resources, and a 40th order calculation is clearly beyond any reasonable expectations.

To predict the entropy rate of the speech process we will use the set of computed conditional entropy measures to estimate a model of the conditional entropy as a function of model order. Our data consists of a set of measurements from first to sixth order in which we have varying degrees of confidence. This situation leads to the choice of a model fitting method that weights the error residual of each measurement according to its expected relative accuracy. We have chosen a weighted regression analysis to fit a model to the computed conditional entropy measurements.

In the following sections we will propose a model of the conditional entropy function to be fitted to the known data points. We will then quantify the relative weights to be assigned to the error residuals of the data points in terms of their confidence intervals. Finally, we will use the results of the regression analysis to define a model of the conditional entropy function and a 95% inference region around the model. From this model we will predict the entropy rate and a set of 95% confidence intervals for the speech process.

3.6.1 Defining the Conditional Entropy Model

We can apply knowledge of the theoretical behaviour of the conditional entropy measures to determine a set of constraints for the function, $y(n)$, used to model the conditional entropy as a function of model order. We first observe that additional information can only decrease the uncertainty about the source process. It can be shown that the sequence of conditional entropy points, $\{H(X), H(X_2|X_1), H(X_3|X_1^2), \ldots\}$ is positive and non-increasing since,

$$H(X_n|X_1^{n-1}) \geq H(X_{n+1}|X_1^n) \geq 0 \quad \forall \quad n \geq 1.$$

We also note that the correlation function of the speech process plotted in figure 3.1 indicates generally decreasing correlation as a function of the model order, n. Since the dependence of the random variables in the speech process demonstrably decreases with increasing separation, we can expect an accompanying decrease in the information obtained by additional observations. This will result in a gradual reduction in the rate of change of the conditional entropy function with increasing model order.

To summarize, we require the following three characteristics in the conditional entropy

model,

$$y(n) \geq 0 \quad \forall\ n > 0 \tag{3.6}$$

$$\frac{dy(n)}{dn} \leq 0 \tag{3.7}$$

$$\frac{d^3y(n)}{dn^3} \leq 0 \tag{3.8}$$

These three constraints define the known characteristics of a model for the conditional entropy function. In the absence of additional information about the conditional entropy function we apply the principle of Occam's razor [1] and select the simplest function which satisfies the constraints and appears to fit the data points. We plotted our best estimates of the conditional entropy as a function of model order in figure 3.18 in order to assess the general shape of the function. These points indicate a pattern similar to a simple exponential decay, suggesting the choice of a 3-parameter exponential curve of the form $y(n) = A + \beta e^{-\alpha n}$ to model the conditional entropy function. This curve satisfies the constraints of equations (3.6), (3.7), and (3.8) for all $\{A, \beta, \alpha\} > 0$. A curve of this form will necessitate the use of a non-linear regression procedure to fit the model to the known data points.

3.6.2 Confidence Intervals for Weighted Regression Analysis

We will use the 95% confidence intervals of each conditional entropy measurement to determine the weight that will be assigned to each error residual in the regression analysis. Specifically, we will assign a relative weighting for the error residual of each data point in proportion to the inverse of the confidence interval computed for that model order.

[1] William of Ockham, an influential 14th century philosopher proposed the idea that "plurality should not be assumed without necessity" in his controversial treatises on papal power and civil sovereignty. This principle came to be known as Occam's razor.

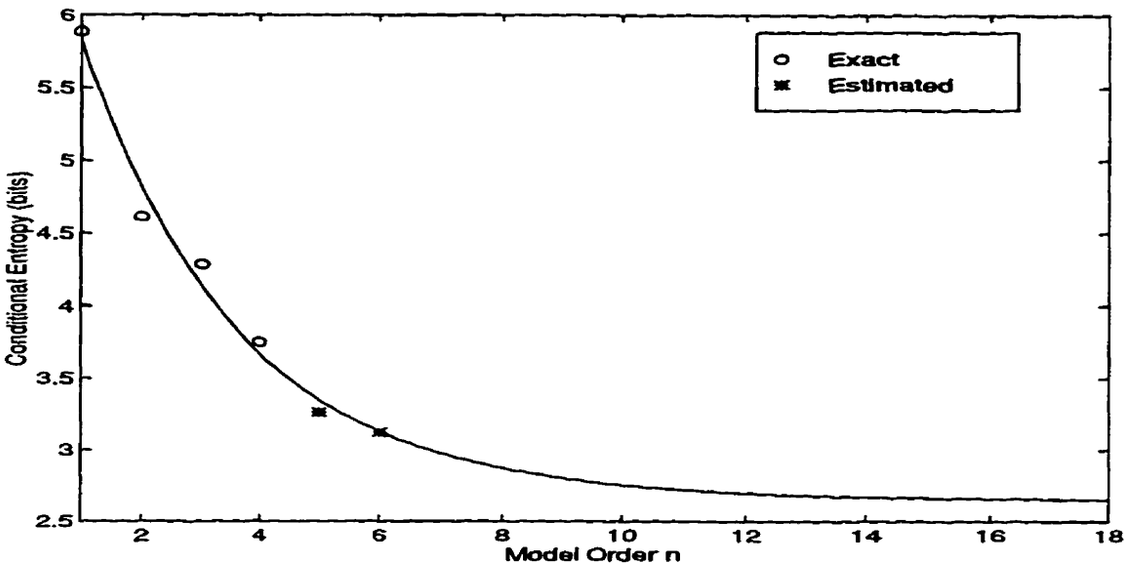


Figure 3.18: Conditional Entropy Estimates and a Simple Exponential Model

Direct Measures

We computed 95% confidence intervals for the results of the independent direct conditional entropy calculations as summarized in table 3.3. These measurements were all computed at the maximum sample size of 2725 minutes, and comprise our best estimates of the four direct conditional entropy measures.

Indirect Measures

In deriving the estimates of the fifth and sixth order conditional entropy measures we applied a sampling strategy, and then interpolated the sampled points to achieve an entropy estimate. To compensate for variations in the sampled data measures arising from computational difficulties, we applied a normalization routine to the estimated entropy measures. The result of this procedure is a pair of measures for which it is very difficult to quantify an error estimate. In this section we will determine a set of reasonable bounds for the results to serve as 95% confidence intervals.

We note that the interpolation routines for both fifth and sixth order surfaces are unbiased for large sets of PCE measures: they can be expected to overestimate the same number of surface points that they underestimate. We will assume then, that the interpolation process is not the dominant contributor to error in the entropy estimates. It has been noted, however, that modelling the speech process was difficult and that compromises were required which resulted in non-ideal models of varying sample sizes. We believe the majority of the error in entropy estimation occurred due to the modelling process.

As a conservative lower bound on the fifth and sixth order conditional entropy estimates, we select the measures obtained from unnormalized PCE vectors in sections 3.5.4 and 3.5.4, respectively. The models of the speech process used to derive these estimates did not all converge to stable measures. The choice of the unnormalized measures as a lower bound is supported by our analysis of the convergence characteristics of the process, which indicates that these measures underestimate the conditional entropy of the speech process. Accurate measures will therefore be at least as large as the raw results.

We will assume the upper bound on the fifth and sixth order conditional entropy is also largely determined by modelling errors. To simplify the regression analysis we will define the upper bounds such that we have a symmetric confidence interval around the computed conditional entropy measures. The fifth and sixth order conditional entropy estimates and the associated 95% confidence intervals to be used in the regression analysis are summarized in table 3.5.

3.6.3 Non-linear Weighted Regression Analysis

To fit the exponential model to our experimental data we applied a non-linear weighted regression analysis with a least squares error criterion. The regression analysis was performed on the set of all conditional entropy measures computed at the maximum available sample sizes, and error residuals were weighted according to the estimated 95% confidence intervals on the average of the experimental measures.

Our best estimates of the first to sixth order conditional entropies, and the associated estimated confidence intervals are summarized in table 3.5. We note that the confidence intervals on the direct measures are all approximately .08 bits, while the the fifth and sixth order confidence intervals are approximately 0.26 bits. This allows us to assign an error weighting criteria in the ratio 3.25:1 to the non-linear weighted regression analysis of the results. Using these relative weights we applied a weighted least squares analysis to the data points to estimate the parameters of the non-linear conditional entropy model. The resulting model and its associated 95% inference regions is presented in figure 3.19.

The regression analysis yielded the model described by the equation $y(n) = 2.79 + 3.03e^{-0.407n}$. If our choice of model is accurate, then the regression analysis has provided a means to predict the nth order conditional entropy with some degree of confidence.

Order	Best Estimate	No. Points	Rel. Weight
1	5.885 ± 0.08	4	3.25
2	4.613 ± 0.09	4	3.25
3	4.287 ± 0.07	3	3.25
4	3.751 ± 0.10	2	3.25
5	3.262 ± 0.268	1	1
6	3.121 ± 0.257	1	1

Table 3.5: Summary of Means, 95% Confidence Intervals, Observations, and Relative Weights of Computed Entropy Measures

3.6.4 Entropy Rate Prediction Results

As discussed in Appendix A, the entropy rate of the speech process can be estimated by computing a high order conditional entropy measure. Using our model of the conditional entropy function, $y(n)$, we compute

$$H = \lim_{n \to \infty} y(n) = \lim_{n \to \infty} 2.79 + 3.03e^{-0.407n} = 2.79. \tag{3.9}$$

Thus, the first parameter of our model represents our best estimate of the entropy rate

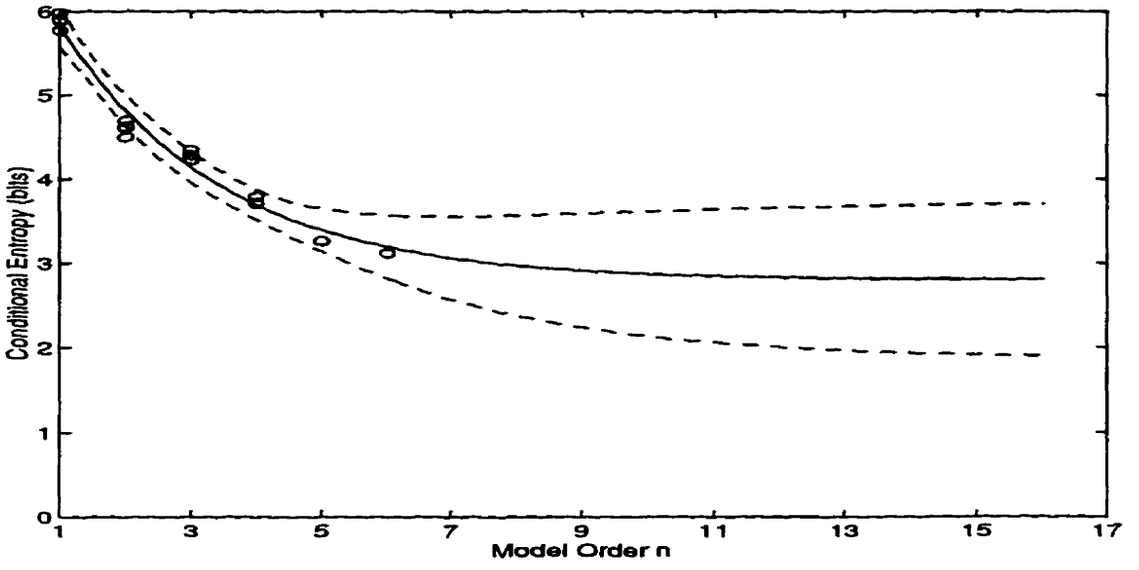


Figure 3.19: Regressed Non-linear Model and 95% Inference Region

of the speech process, and an analysis of the inference regions around the model yields 95% confidence intervals. These results are summarized in table 3.6. Our analysis predicts an entropy rate of 22.3 kbps for conversational English speech of telephone bandwidth and 64 kbps μ-law PCM fidelity. Our inference regions allow a fairly generous range of $\pm$ 7 kbps around this estimate to account for the sources of error encountered in the modelling process.

We note that these results have been obtained from a speech process defined to be stationary. As discussed in section 3.3.1, we would expect this entropy rate to represent the upper limit of the entropy rate of the speech process because our modelling approach could not adapt to the short-term temporal statistics of the source. Were we to follow our suggestion of categorizing the speech process by speaker and computing a weighted average of speaker-specific entropy measures we could obtain a lower limit on the entropy rate.

Bounds	Lower	Predicted	Upper
Measure (bits)	1.91	2.79	3.70

Table 3.6: The Entropy Rate of the Speech Process and Bounds on the 95% Confidence Interval

3.7 Summary and Observations on the Entropy Calculations

We have presented a method of estimating high order entropy measures for any type of source process. This method can reduce the high computational expense of an entropy calculation to a feasible level by using an interpolation function to estimate portions of the full calculation. The method has an additional advantage in that it allows an entropy calculation to be partitioned in time and space across multiple processing units. Finally, it allows successive refinements in the accuracy of the entropy estimate with the addition of more sample points. In the limit the method can provide an exact measurement of the source process entropy.

A number of interesting details were revealed in applying this technique to the speech process. First, we noted that this technique was most effective for processes exhibiting a high degree of memory. The entropy surface of these processes is highly peaked and can be accurately modelled by the Monte Carlo point selection procedure of equation (2.13). By concentrating the point selection probability distribution in the peaked region we can obtain a high resolution sampling of the vectors contributing most to the conditional entropy of the process.

The entropy surface of the speech process was suitable to efficient sampling and interpolation by the Monte Carlo selection procedure. We found the surface to be relatively smooth and highly peaked around the mean of the process. Both characteristics contribute to a good estimate of the entropy rate from a relatively small number of PCE measures.

A limitation of this technique was found in modelling processes that do not possess

a smooth entropy surface, as defined in section 2.4.1. To obtain an accurate measure of such a process requires a large number of PCE vectors to be computed in order to identify the relevant features of the entropy surface. Our technique has no particular advantage in modelling processes of this type.

In general, we found characterizing the shape of the entropy surface to be a critical step in efficient entropy estimation. If we could predict the general shape of the entropy surface of a process, say a facsimile or video source, we could design a better *a priori* sampling distribution and achieve an accurate and efficient entropy estimate. A better understanding of these surfaces would also result in more accurate interpolation routines and consequently, better entropy measures.

In developing estimates of the probability distributions of the speech process we encountered a variety of difficulties due to the large volume of speech data being processed and the long execution times required of our experiments. These problems precluded a constant and maximal sample size for the speech process under study. The following are a few areas in our experimental work where we noted the opportunity for improvements that would enhance the accuracy and confidence interval of the predicted entropy rate,

1. More accurate measures of the conditional entropy of the speech process could be obtained by computing all PCE vectors from a single large sample set. A larger sample set would improve the convergence of PCE vectors outside the more probable regions, and a constant sample set size would eliminate the need to normalize the PCE measures with the set of computed convergence surfaces. This procedure would require the use of larger, and more reliable, computing facilities than were available for our initial prediction.

2. Were this not feasible, we could still increase the accuracy of our results by improving the resolution of the set of convergence surfaces. This could be achieved by expanding the set of convergence measures to include a larger number of points over the calcula-

tion space. This process is feasible with the available computing resources and simply requires the expenditure of a greater amount of time to collect the convergence data.

3. The accuracy of the entropy measure could also be improved by including more PCE vectors on the entropy surface. A larger number of PCE vectors would directly improve accuracy by providing a higher resolution set of measures for interpolating the complete entropy surface.

This entropy estimation technique can be used to develop high order entropy measures for a wide variety of source processes, particularly those with a strong property of memory. It is hoped that the techniques presented here can find wider use in entropy rate estimation for a broader class of interesting source processes.

Chapter 4

Security Models for Secure Speech Systems

We dance round in a ring and suppose,
But the Secret sits in the middle and knows.

Robert Frost – The Secret Sits

4.1 Introduction to Cryptology

We will begin with an introduction to cryptology that allows us to describe the systems commonly used to obtain secure communications, and define the relevant parameters for models of cryptographic security.

The science of cryptology can be divided into two areas of activity: cryptography and cryptanalysis. The cryptographer seeks to design systems capable of ensuring the secrecy or authenticity of messages. The cryptanalyst seeks to determine the content of the encoded messages or to deceive a valid receiver through impersonation or other fraudulent activities. The cryptographer always employs a secret key to control the enciphering process. In

some cryptographic applications, those known as symmetric key algorithms, this secret key must be exchanged by a secure channel with the authorized receiver prior to initiating secret communications. In asymmetric key algorithms, there is no requirement for the prior exchange of secret information. The receiver generates a unique pair of keys, one for encryption and the other for decryption, and makes the encrypting key generally available to anyone who wishes to send him a secret message.

4.1.1 Symmetric Key Cryptography

There are two fundamental forms of symmetric key cryptosystems, block ciphers and stream ciphers. Both are capable of encrypting data at rates on the order of 100 Mbps or more. In North America, cryptographic research and development has concentrated on block ciphers, while in Europe fortune has tended to favour the stream cipher. The two cryptosystems transform the message into ciphertext in significantly different ways, but can provide equivalent security. It is usually just a preference for one set of system characteristics that determines the choice of cryptosystems. We will describe the two types and discuss their differences in the next two sections.

Block Ciphers

The first sophisticated treatment of block ciphers in the open literature was given by Shannon in [4]. Later work by Feistel in [31] and Feistel, Notz and Smith in [32] helped to define the current form of block cipher designs. A block cipher can be described as a set of transformations from a message space to a cipher space. Each transformation corresponds to the use of a particular key. In order for each message to be retrievable, we must have an invertible transform from message space to cipher space. If each message block of m characters is mapped into a ciphertext block of an identical m characters, we would have a total of $b^m!$ possible reversible transformations, where b is the size of the key alphabet. Most modern ciphers operate in Z_2 and hence have a potential key space of $2^m!$.

We can view the block cipher we have defined as a device for substituting a ciphertext block for each possible block of plaintext input. The choice of output block is dictated by the particular key used for that encipherment. If this device were built to handle a five-digit binary input, for example, we would be able to encipher an alphabet of $2^5 = 32$ characters. The number of possible connection patterns between inputs and outputs would then be 32!. While this may seem a very large number, there would still be only 32 input-output matchings for each key and the cipher would be vulnerable to a frequency-analysis attack or an exhaustive search. What we require is to have a message space that is so large it is impractical to mount such an attack. Current technology would be incapable of an exhaustive search if block sizes were on the order of 128 bits. Unfortunately, such a substitution device would require 2^{128} inputs and outputs, a technological impossibility by today's standards. Given this constraint, it is necessary to find a way to use the available technology to achieve at least a significant subset of all possible transformations.

Another possibility would be to permute the message bits according to a predetermined routing scheme. Due to their simplicity, these devices are feasible to build for large sets of inputs. Unfortunately, that same simplicity makes these devices highly vulnerable to chosen-plaintext analysis. Any permutation device designed for n inputs can be analysed in $(n-1)$ steps, so clearly these devices will not suffice for a cryptographic algorithm.

Shannon recognized a solution to the problem in the concept of a *mixing transformation* [4]. Such a transformation can be used to map a relatively high probability region, **R**, of a probability space, Ω, into another region in Ω. If applied a large number of times the transformation will result in the initial region, **R**, becoming uniformly mixed throughout the entire space. Strictly speaking, a mixing transformation can only occur over a space with an infinite number of points, but the general process can be applied in a cryptographic application to redistribute a region of high probability messages over the entire cipherspace.

Good mixing transformations are ones in which a small variation in any of the inputs results in a large change in the output. It happens that a permutation followed by a

substitution forms an effective mixing transformation. Figure 4.1 is an example of a 15 bit block cipher with 5 rounds of a substitution-permutation transformation. The application of a large number of successive rounds of this transformation achieves a very complicated mixing of the data, as is evidenced by the example. A further advantage of this arrangement is that it allows the use of small, manageable substitution boxes and protects them from frequency analysis within the overall structure of the network. The example in figure 4.1 has no key and so would not suffice in a practical application. Real systems employ a secret key that in some manner modifies the data or controls the permutation or substitution components.

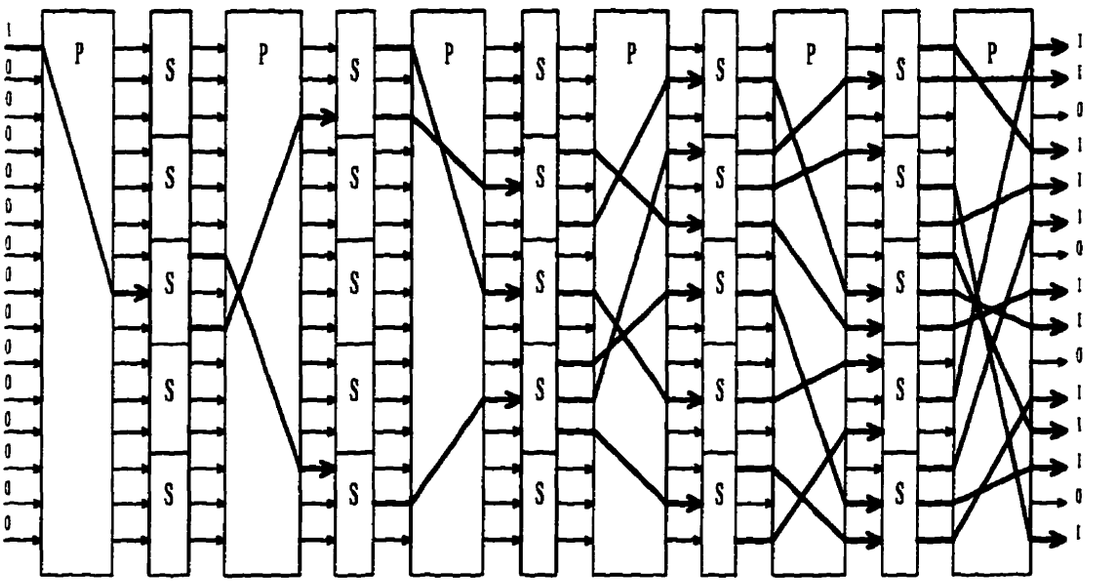


Figure 4.1: Substitution-Permutation Network

This particular combination of operations is effective because it applies two principles that Shannon found to be important to the encryption of data. Substitution adds a characteristic he called *confusion* to the ciphertext. By replacing each character by an unknown, key-dependent character, we have increased the attacker's uncertainty about the pattern he is trying to track. The permutation device then adds a second twist to the puzzle by

adding what Shannon called *diffusion*. This operation has the characteristic of spreading the effect of each message character over a wider area of the cipher block, to the desirable end of concealing some statistical relationships within the data.

The characteristics of a mixing transformation allow us to build strong cryptographic systems with relatively simple and inexpensive devices. By combining large permutation devices with sets of small substitution boxes, the individual weaknesses of each approach are avoided and the resulting ciphertext blocks can achieve a strong inter-symbol dependence known as the *strict avalanche criterion*. This effect was defined by Webster and Tavares in [33]. Higher order characteristics were examined by Adams in [34], and by Forré in [35].

The avalanche effect is a desirable characteristic for a block cipher, but it is not without its hazards. By making block ciphers highly sensitive to changes in input parameters, the designer has introduced a hard limiter into the communications path. The alteration of a single bit in an enciphered message will result in the complete corruption of the message block upon decipherment. This may be desirable for some reasons of security, but it can have a significant impact on the quality of the received signal. The strict avalanche effect also imposes a requirement for perfect synchronization between sender and receiver. Both must be in agreement over the block alignment or the decrypted messages will be completely garbled.

Stream Ciphers

Block ciphers are essentially substitution devices that rely on extremely large block sizes to prevent cryptanalysis by brute force. They are memoryless devices in that a certain plaintext block and key combination will always result in the same ciphertext block output. Stream ciphers, in contrast, encipher each message character individually with a time-varying function that is governed by the internal state of the device. The system state is a memory device and it comprises the principle difference between the two symmetric key encryption approaches. After each character is enciphered, the device's state is advanced

according to some rule. As a result, repeat occurrences of the same plaintext character will rarely result in the same ciphertext output.

One of the most remarkable of all ciphers is the one-time-pad, or Vernam cipher [36]. It is capable of producing a perfectly secure ciphertext stream by adding each character of the plaintext message to a corresponding character in a random sequence of the same length. By perfectly secure we mean that it is theoretically (and practically) impossible to determine the message from an analysis of the ciphertext. We will discuss perfect security when we derive a model of cryptographic security in section 4.5. This highly desirable arrangement is achieved by introducing a completely random property to the encryption process. By adding a random character to a message character we produce a cipher character from which it is impossible to infer the message character. Applying this operation to all characters in the message, using an equal number of randomly chosen characters, yields ciphertext from which the cryptanalyst can garner no information. The best he can do is make a guess based on the *a priori* probability distribution of the set of all possible messages of that length.

Why then do we not use one-time-pads for all our encryption needs? The reason lies in the expense of distributing and protecting the huge amounts of random data, the key, that is required for every pair of users wishing to engage in secret communications. The only way to reproduce an identical stream of random data at the sender and receiver is to pre-record it and distribute copies in a secure manner to the two intended recipients. This is typically not feasible and so this method has been limited primarily to applications regarding diplomatic exchanges and espionage activities [37].

Nonetheless, the process of combining something like a random stream with the message stream suggests an attractive encryption method. Stream ciphers attempt to replace the completely random stream with a simpler approximation, one that can be more easily exchanged between the communicating parties. This approximation is the output of a deterministic device that is shared by both parties. The device utilizes a shared key to

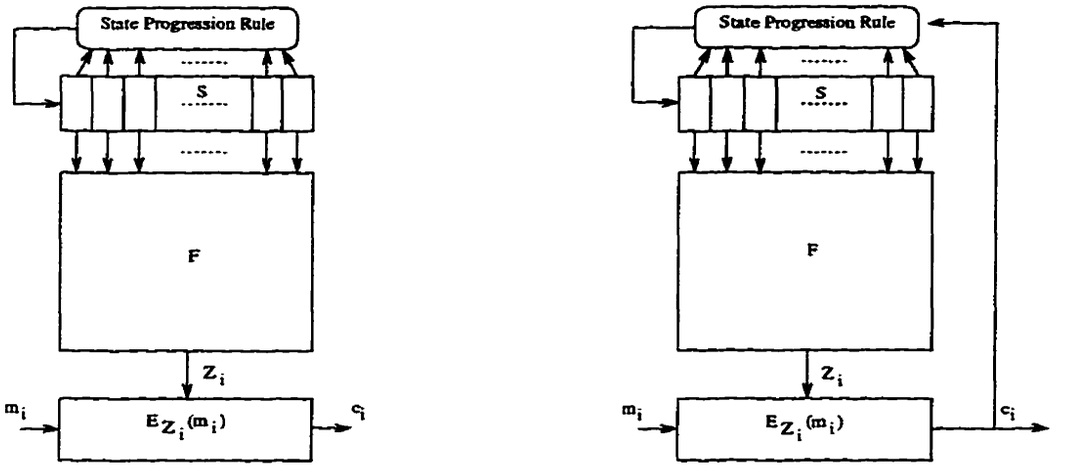
produce a much longer sequence of characters which are then used to encipher the plaintext. This sequence, the running key, must be unpredictable to ensure security for the system.

The difficulty of course, is that the devices are deterministic by nature and therefore not truly able to produce an unpredictable stream. Any running key sequence would exhibit some period of repetition. Given this period, say some number 2^n, and a sequence of $2n$ running key values, the Berlekamp-Massey algorithm [38] can compute the coefficients of a linear feedback shift register (LFSR) capable of reproducing the entire linear sequence. The length of the shortest LFSR required to duplicate a given sequence defines what is called the *linear complexity* of a running key. It is generally possible to achieve a higher linear complexity using a non-linear combination of memory states [39]. It is a considerably more difficult task to determine the smallest non-linear device capable of reproducing a given sequence, and in general, the sequences produced by non-linear devices have a linear complexity that is greater than the actual number of memory states available to the device. These characteristics make non-linear feedback shift registers the device of choice for most stream cipher designs.

A large linear complexity is a necessary but not sufficient condition for cryptographic security. Simple sequences such as $(0, 0, 0...0, 0, 1)$ have a high linear complexity for large run lengths, but are ineffective at concealing a message. In order to approximate a random sequence the running key must have a uniform distribution and the sequence must contain an equivalent distribution of single bits, pairs, triples, etc.. A detailed study of complexity issues in stream ciphers may be found in Rueppel's work [40]. Another general treatment of stream cipher design issues may be found in [41].

Stream ciphers may be divided into synchronous and self-synchronizing systems. In synchronous stream ciphers, the next state of the device depends only on the previous state and not on the input. Self-synchronizing ciphers employ past encryptions to define the current memory state. As a result, self-synchronizing ciphers rely in part on the characteristics of the input sequence for ciphertext generation.

A block diagram of both types of cipher is given in figure 4.2. The subsystem F is typically a nonlinear device that uses the system state, S, to produce an encryption character, Z_i. In its most general form the stream cipher combines the plaintext message character, m_i, and the key character, Z_i, in an invertible function $E_{Z_i}(m_i)$. The resulting character, c_i, is the ciphertext. In practical applications the strength of the encryption algorithm lies in the unpredictability of Z_i and not with $E_{Z_i}(m_i)$, so this function is often chosen to be the exclusive OR operation, $c_i = m_i \oplus Z_i$.



(a) Synchronous Stream Cipher

(b) Self-Synchronizing Stream Cipher

Figure 4.2: Basic Stream Ciphers

It should be apparent that stream ciphers operate as substitution devices on the character level. If the running key subsystem F is well designed, the stream cipher introduces Shannon's confusion into the ciphertext. As for the property of diffusion, the synchronous cipher offers none at all. Each message character affects only one corresponding ciphertext character. The self-synchronizing cipher offers some diffusing properties in that previous outputs are used to produce the new running key character Z_i. Unfortunately, the impact of this diffusion is limited because the device can only transfer information in the forward

direction and only for a fixed number of states.

Synchronous stream ciphers require perfect synchronization between the encrypting and decrypting devices. If the running keys of the communicating parties differ by as little as one bit the resulting decryption will be completely meaningless, just as with block encryption devices. In order to re-establish secure communications, the receiver must search over all possible offsets of the running key or notify the sender that re-synchronization is required. This sensitivity has obvious advantages to security in that it is impossible for an attacker to insert, delete, or replay ciphertext without being detected. On the other hand, the need for strict synchronization places a larger burden on the devices used for channel coding.

Self-synchronous stream ciphers require only that the last n ciphertext characters were received correctly in order to reproduce the correct Z_i character. In the case of a channel error the receiver will re-synchronize automatically after the next n consecutive ciphertext characters are received. These ciphers are more tolerant of poor channel conditions than the synchronous variety. Unfortunately, the self-synchronous cipher will also re-synchronize after an attacker has attempted to insert, delete, or replay ciphertext. By their very nature they have an additional weakness in that they transmit the inputs to the encryption device F and allow a comparison to be made of inputs to outputs.

Rueppel noted in [40] that stream ciphers are generally more difficult to analyse than block ciphers because of their internal memory, non-linear transformations, and, in the case of self-synchronous ciphers, because of the statistical dependence between the output stream and the message stream. He feels that it is perhaps for this reason that there is considerably less material available on the analysis of stream ciphers. It is interesting to note then, that stream ciphers still make up the majority of practical encryption applications.

4.1.2 Asymmetric Key Cryptography

In their famous paper "New Directions in Cryptography" [42] Diffie and Hellman surprised the cryptologic community with the introduction of a method to exchange secret messages

without a symmetric key. They postulated the use of a function which is easy to compute in one direction but very difficult to invert without the knowledge of some additional property. Their ideas were a catalyst for the development of a number of practical implementations of asymmetric key cryptosystems. Asymmetric key cryptosystems, or public key cryptosystems as they are more commonly called, require a pair of encryption keys with special properties. The asymmetric key pair, e and d, allow a cryptographic algorithm to perform a transformation of a message into ciphertext with either key, and another transformation back to the original message with the other key. This requires a special relationship between the two keys, but to ensure security an asymmetric key cryptosystem must be designed so that it is computationally infeasible to determine the decryption key from knowledge of the encryption key, or encryption function.

This asymmetric key characteristic allows implementations of cryptosystems where one half of the key pair is made public to anyone who wishes to send a message. Message encryption is performed using the public key, and anyone can perform this encrypting operation, but message decryption is designed to be feasible only for the holder of the other half of the key pair.

This characteristic of asymmetric key cryptosystems makes it possible to provide a secure channel to two users who have never met or exchanged any prior secrets. This is a strong advantage of asymmetric key over symmetric key systems where it is necessary to exchange a secret key in advance of the secure conversation.

In addition, asymmetric key systems have the advantage of providing strong authentication protocols and digital signature implementations through modification of the asymmetric key protocol. Implementations of asymmetric key cryptography include the RSA system [43], the El Gamal system [44] the McEleice system [45] and elliptic curve cryptosystems [46]. Digital signatures are discussed in [47], [48].

Public key systems are generally too complex to allow real-time encryption of digitized speech signals. They can be very effective, however, at providing a medium to exchange a

session key between two parties who wish to communicate by voice. The session key would then serve in a symmetric key system to encrypt the speech traffic for transmission on an unsecured channel.

An asymmetric key system can serve in the place of a secure channel for symmetric key exchange. In the asymmetric key protocol indicated in figure 4.3, the initiator of the communication session will acquire an encrypting key for his intended recipient, E_{Dest}, from a trusted public key directory in the public domain. He encodes a request to establish a session key, $\mathbf{K}$, for a secure speech communication session by using E_{Dest}. The receiver will then retrieve the sender's public key, E_{Source}, from the public directory and the two will negotiate a session key under the protection of the asymmetric key encryption algorithm. Both sender and receiver then apply the negotiated session key, $\mathbf{K}$, to encrypt and decrypt the speech transmission by means of a symmetric key encryption algorithm.

In the outlined protocol, the asymmetric key system serves as a replacement for the secure key exchange channel. It provides the additional benefit of allowing users who have had no prior contact to exchange secret information. It also eliminates the need to maintain private databases of the $n(n-1)$ possible key pairs necessary to allow any of n users in a network to communicate with any other users by a purely symmetric key protocol.

We must point out that several assumptions have been made about the asymmetric key exchange protocol in this simple outline. First, we have assumed that the attacker may request public keys from the public key directory, but that he may not substitute his own public keys in place of valid keys. This is a reasonable assumption that may be ensured by a slightly more elaborate protocol for obtaining public keys from a trusted public key authority. Second, we have assumed that the symmetric key negotiation procedure can not be manipulated in any undetectable manner by the attacker. This can also be ensured by careful implementation of the negotiation protocol. Finally, we have assumed that the attacker can not enter the private domains of either the sender or receiver. We note, however, that we allow the attacker to observe all of the encrypted key negotiation session

information, the encrypted speech transmission, and the public keys of both sender and receiver.

Under this type of security protocol the system security level is determined by the minimum security afforded by either the asymmetric or symmetric key components. Denoting the security index for the asymmetric and symmetric cryptosystems by $S_{\text{asymmetric}}$ and $S_{\text{symmetric}}$, respectively, the security index for the combined protocol of figure 4.3 is,

$$S_{\text{system}} = \min\left(S_{\text{asymmetric}}, S_{\text{symmetric}}\right),$$

since we can assume the attacker will concentrate on whichever aspect of the system is easiest to compromise. Typically, breaking the asymmetric key security system will yield the most value to the attacker since once it is broken he might obtain access to all session keys. Conversely, breaking a single session key will reveal only the current conversation and not endanger the asymmetric key exchange mechanism.

In many practical applications we rely on the ability to provide the user with an incorruptible version of a common public key. This public key is generally associated with a Certificate Authority (CA). The CA can ascertain the identity of a user and form a verifiable certificate of authenticity linking the user's identity to the user's public key.

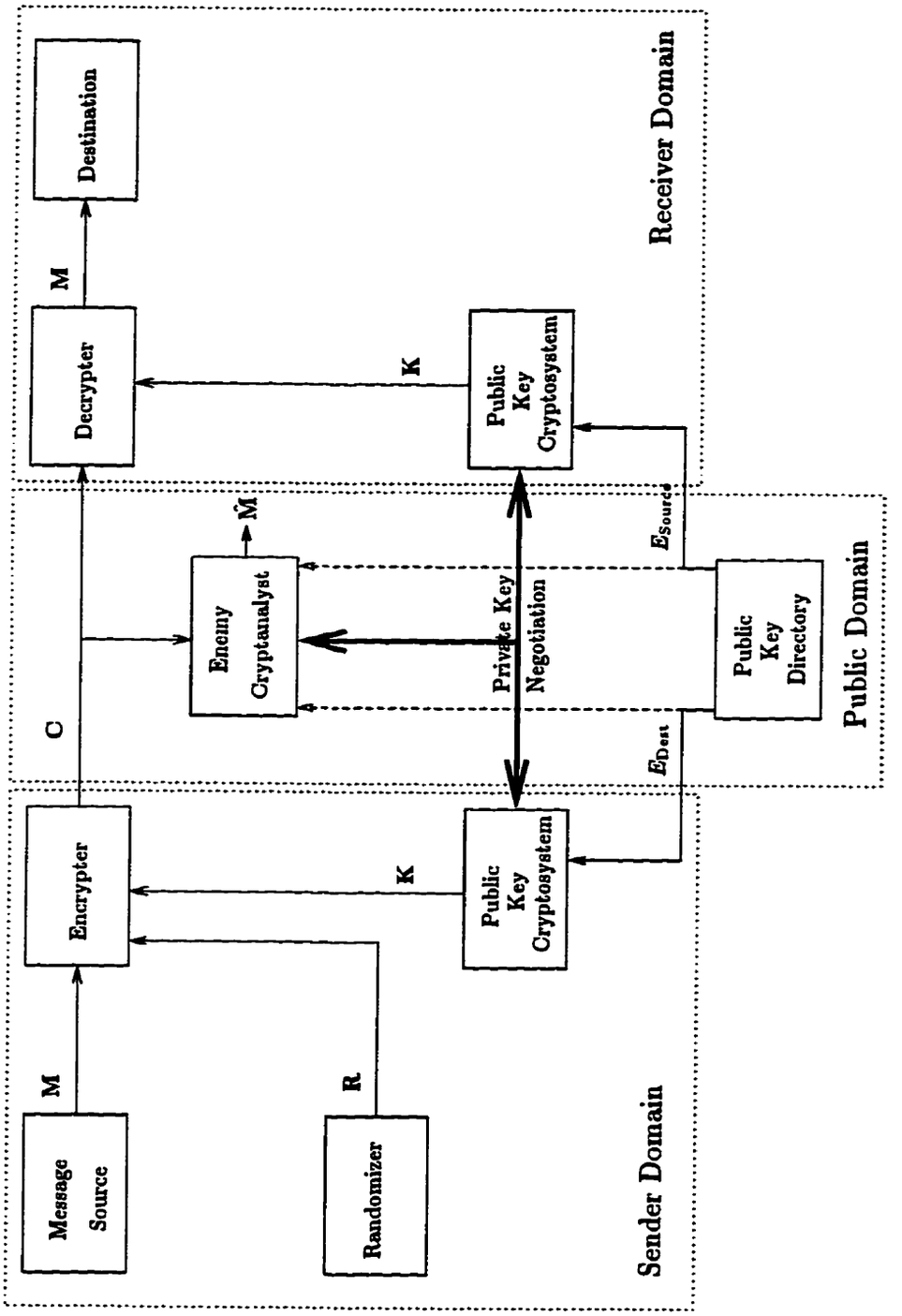


Figure 4.3: Use of Public Key to Exchange a Session Key

4.2 Cryptanalytic Capabilities

A cryptanalyst will be assumed to take advantage of any information he can gain about the encryption system to decipher the message. It is a common assumption, first suggested by Kerckhoff [49], that the secrecy afforded by a cryptosystem must reside entirely in the key. Our first assumption about security will be that the attacker knows the encryption algorithm but not the secret key being used.

The cryptanalyst may be able to engage in a range of activities, from passive listening to active interference with the communications channel. The purely analytic attacks include the following, in order of their value to the cryptanalyst,

1. **Ciphertext-only attack.** The cryptanalyst can observe the ciphertext but has no direct knowledge of the message being encrypted.

2. **Known-plaintext attack.** The cryptanalyst has complete or partial knowledge of the message being encrypted and can observe the resulting ciphertext.

3. **Chosen-plaintext attack.** The cryptanalyst can choose messages to be encrypted and observe the resulting ciphertext. Carefully chosen messages may help to reveal more information about the key.

The cryptanalyst may in some circumstances be capable of more active attacks on the cryptosystem. These attacks may be intended for purposes other than deciphering the message [50]. Active attacks include,

1. **Message repetition.** The attacker may be able to deceive the receiver by replaying a valid ciphertext message. This attack does not require the cipher to be broken, but may be beneficial to the attacker if the contents of the cipher are known.

2. **Message insertion or deletion.** Similar to the message repetition attack, the attacker may attempt to modify the contents of the received message by adding or

deleting ciphertext blocks in transmission.

3. **Denial of Service.** The attacker may simply choose to ensure that the message cannot be recovered by the receiver by introducing noise to the communications channel.

4. **Cloak and Dagger attacks.** The most efficient approach available to an attacker may be to acquire the message or key by theft, bribery, or defeating a weaker part of the cryptosystem protocol. Software implementations of encryption algorithms, for example, can be compromised by obtaining access to the memory space used to hold the key or message during encryption.

In developing models of the secure communication system we will generally assume that only the passive forms of attack are available to the cryptanalyst. We can, however, allow for analysis of an active denial of service attack where the attacker is capable of only partial interference with the communications channel. The effect of noise on the communications channel under cryptographic protection will be studied in detail in chapter 5.

4.3 Motivation for Security Index Development

Our brief introduction to cryptology has attempted to give some indication of the sophistication applied to the design of cryptographic systems and the resulting need for similarly sophisticated methods of cryptanalysis. The development of a new cryptographic algorithm is usually based on a thorough understanding of a set of principles considered essential for resistance to cryptanalytic attack. These principles have been developed over time as a result of significant effort towards the design and analysis of practical security systems.

Many of these principles have arisen from a complexity-theoretic approach to cryptographic design [51]. For example, symmetric key block ciphers owe their security to the high computational complexity required to invert the operations of a large substitution-permutation network, while symmetric key stream ciphers rely on high linear complexity in

the running key generator to approximate Vernam cipher performance. Similarly, the set of asymmetric key algorithms owe their security to the supposed complexity of a relatively small number of "difficult" problems, such as taking discrete logarithms or factoring the product of large prime numbers.

Massey recognized in [5] the unsettling fact that the security of these algorithms often rests on a conjecture that the underlying problem is difficult. The design principle for these algorithms can best be described by the statement: *If* this problem is difficult, *then* this secrecy system is secure against that form of attack. Clearly, if at some point in the future the solution to a particular type of "difficult" problem is found to be feasible, then encryption algorithms of that type may prove to be insecure. There are two dangers in this scenario: first, if a feasible solution technique becomes known, then not only is the system compromised for future use, but all messages previously encrypted by that system may be deciphered by an attacker who has recorded past messages. Second, if there is a feasible solution technique, perhaps it is already being used secretly by an attacker.

It would be valuable to provide some measure of cryptographic security that was independent of assumptions about the cryptanalytic problem and based purely on an objective assessment of the performance of the cryptographic algorithm. This measure would constitute a design and evaluation tool for cryptographic algorithms to be used in conjunction with the general principles gained so painstakingly by direct analysis of each new class of cipher. In this chapter we will propose a set of relevant information-theoretic cryptographic security indices to provide such a measure.

4.3.1 Practical and Theoretical Security

Practical security refers to the notion that the cryptosystem is secure against attack on the basis of its computational complexity. A measure of practical, or computational security as it is sometimes called, would indicate the order of operations required to invert the enciphering operation.

Theoretical security, or unconditional security as it is also called, is achieved when the cipher is immune to cryptanalysis even when the cryptanalyst has unlimited computational ability. Theoretical security is determined by an information-theoretic approach that has led to some pessimistic conclusions about the requirements for cryptographic security [5], [52]. We will show in section 4.5.1, for instance, an information-theoretic proof that perfect secrecy can only be attained by a certain costly and inconvenient protocol. We will also develop the unicity distance measure in section 4.4, a classic theoretical security index which indicates the minimum amount of information that a cryptanalyst would need to intercept in order to decipher the message.

The results of a unicity distance calculation are generally very low and have led many designers to conclude that purely computationally motivated design criteria are sufficient to achieve a good cryptosystem. We wish to demonstrate that theoretic security indices can provide relevant measures for evaluating the quality of an encryption algorithm. Information-theoretic design criteria for substitution devices in block ciphers have been studied in work by Tavares, Dawson, Zhang, and others [33], [53], [54], and [55]. In section 4.6 we will introduce two information-theoretic security indices that can distinguish the quality and efficiency of different cryptographic algorithms on the basis of the operations performed on the ciphertext.

We make the following proposition on the relation between practical and theoretical security measures,

Proposition 2 (Relative theoretic and practical security measures) *The relative practical security of two cryptosystems can be reflected in a set of relative theoretic security measures for cryptosystems based on these fundamental general design criteria,*

- **Diffusion.** *The cipher should introduce the quality of* diffusion *defined in section 4.1. This will have the effect of distributing the value of a single ciphertext digit or a single key digit over a broad range of adjacent ciphertext characters. Diffusion is essential*

to a good cipher as it helps to conceal the statistical structure of the mappings from plaintext to ciphertext and increase the difficulty of predicting the key.

- **Confusion.** *The cipher should be designed to introduce the quality of* confusion *as defined in section 4.1. This requires the ciphertext to be produced from a complex transformation of message and key information. Confusion is essential to a good cipher as it allows unpredictable key information to be applied in such a manner as to increase the entropy of the ciphertext.*

- **Mixing Transformations.** *In block ciphers the application of successive rounds of substitution and permutation operations has been shown to increase the security of a cipher by creating a ciphertext product that is well distributed over the cipherspace.*

- **Nonlinearity.** *Nonlinear transformations of message and key increase security by increasing the complexity of the operations required to invert the algorithm. Stream ciphers in particular can achieve higher linear complexity by computing a running key by means of a nonlinear operation.*

- **Uniformity.** *In symmetric key systems security is enhanced by ensuring the key source produces keys with a uniform probability distribution over the key space. This maximizes key entropy. In asymmetric key systems such as RSA, security is enhanced by selecting prime numbers for the generation of the asymmetric key modulus according to a uniform probability distribution over a range of eligible primes [56].*

- **Complexity.** *The number of possible transformations of message to ciphertext must be sufficiently large to prevent an exhaustive search of the key space. The complexity of the cryptographic transform must also be sufficiently high to defeat all known cryptanalytic techniques for reducing the problem. Examples of complexity-reducing attacks include factoring and discrete log computation techniques and differential cryptanalysis.*

It is difficult to provide a proof of this proposition because of the variety of possible cryptographic and cryptanalytic methods. We must consider both the transformations applied by the encryption algorithm and the sophisticated mathematical and statistical tools available to the cryptanalyst when we assess practical security. That is why we have limited our statement of correspondence between practical and theoretical measures to the class of ciphers that have been designed to be resistant to the techniques employed by modern cryptanalysts. In fact, this list is not exhaustive and only contains those fundamental principles that are known to improve cryptographic security. Should other principles of good design be discovered, it would be prudent to add them to the list and further reduce the set of ciphers for which we can assume correspondence between security measures.

Our rationale for this proposition has been to include only those ciphers for which we can be reasonably certain that there is no trivial, hidden, or low-complexity solution to the cipher in addition to the intended inverse transform. For this type of well designed cipher, the theoretic security measure can be indicative of the relative strength of the cryptographic algorithm. By excluding ciphers with hidden weaknesses, we have avoided the trap of computing high security indices for ciphers that could be decrypted by some simple approach.

As an example of a poor cipher design yielding a high security index we consider a cipher that applies the key in a highly nonuniform fashion, concealing some parts of the message while revealing the rest. Some theoretic security indices would consider only the average effect of the key entropy and misrepresent the system security as being higher than actually warranted. The design criteria required for the correspondence proposition exclude this type of cipher and so avoid this kind of evaluation error.

Ciphers which do follow the design criteria required for the correspondence proposition will tend to apply a complex transformation of key and message characters in a manner which is difficult to invert without knowledge of the key. In the absence of hidden weaknesses, the averaging characteristics of most theoretical security indices can be better

justified because the cipher characteristics are, by design, more uniformly distributed.

4.4 Shannon's Unicity Distance

Following publication of his landmark work on Information Theory, [57], Shannon published the framework for information-theoretic evaluation and design of secrecy systems in [4] [1]. In this section we will present Shannon's security index for the general class of symmetric key cryptosystems described by the block diagram of figure 4.4. These systems will be assumed to process discrete information only, in accordance with the vast majority of practical encryption applications.

The operation of symmetric key systems is centered around the exchange of a key, $\mathbf{K} = \{K_1, K_2, \ldots, K_K\}$, by means of a channel assumed to be protected from interception by the enemy cryptanalyst. The key is assumed to be generated by some source process, as is the message, $\mathbf{M} = \{M_1, M_2, \ldots\}$, and a random stream of information $\mathbf{R} = \{R_1, R_2, \ldots, R_R\}$. The resulting cryptograms will be denoted by $\mathbf{C} = \{C_1, C_2, \ldots\}$. The statistics of $\mathbf{M}$ are determined by the source process, and the statistics of $\mathbf{K}$ and $\mathbf{R}$ are considered to be determined to benefit the cryptographer. We will assume throughout this development that the quantities are statistically independent, this being the choice yielding maximum advantage to the cryptographer in frustrating an attacker.

Shannon did not consider a randomizer in his model of the secure system but he did recognize its purpose, which is to blur the statistics of the source process. This activity is commonly referred to as homophonic coding. The randomizer allows highly probable message elements to be replaced with elements randomly selected from a larger group of synonyms for the popular element. Homophonic coding can frustrate a frequency analysis of the encrypted message stream, but is easily reversed by the intended receiver who simply

[1] Shannon actually published his work on secrecy systems in 1945, prior to the Information Theory paper, but this paper was classified as Secret by the U.S. government until 1949.

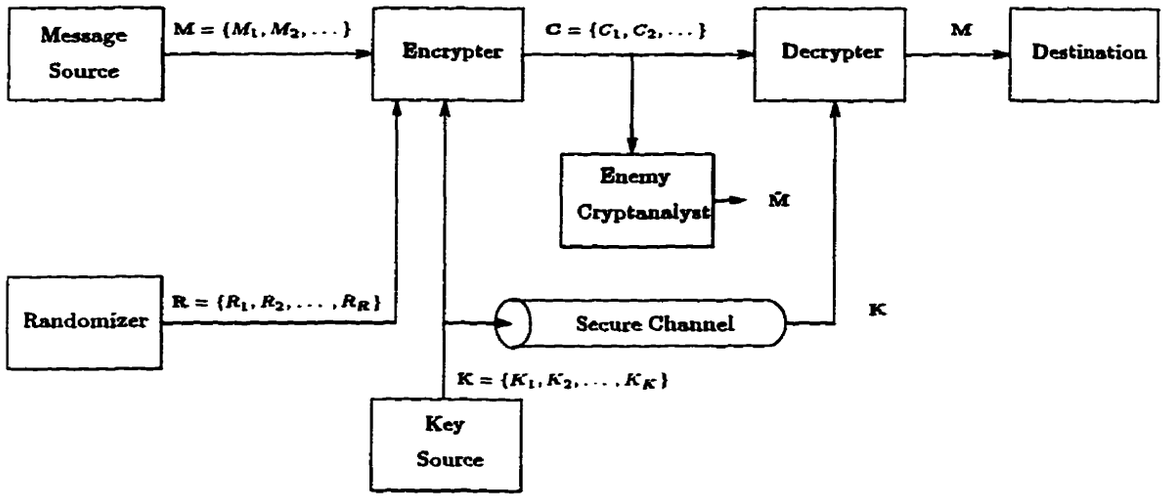


Figure 4.4: Model of a Private-key Cryptosystem

replaces the decrypted synonyms with the intended message element.

We define the encrypting operation as a function performed on the message according to the parameters $\mathbf{K}$ and $\mathbf{R}$,

$$\mathbf{C} = E_{\mathbf{KR}}(\mathbf{M}),$$

and the decrypting operation as,

$$\mathbf{M} = D_{\mathbf{K}}(\mathbf{C}),$$

where again we note that knowledge of the randomizer is not necessary to the receiver to decode a unique image of the message.

Shannon originally considered the case of a ciphertext-only attack, but we can extend his analysis to allow for those cases where the attacker has access to additional information. These approaches may be classed as known-plaintext and chosen-plaintext attacks. If a known-plaintext or chosen-plaintext attack is applied to a cryptosystem it may be successful at yielding information about the message that is unavailable in a ciphertext-only attack. We can account for these types of attack in our security indices by considering the extra information to have the effect of increasing the redundancy of the message source. We will

show how redundancy is a fundamental measure in the development of a security index.

From the available sources of information the attacker constructs his estimate of the transmitted message, $\hat{\mathbf{M}}$. The goal of the cryptographer is to ensure that $p(\hat{\mathbf{M}} = \mathbf{M}) = p(\mathbf{M})$. This would imply that the *a posteriori* probabilities of the various messages, $p(\mathbf{M}|\mathbf{C})$ are equivalent to their *a priori* probabilities, $p(\mathbf{M})$ and the attacker has gained no information about the message from his observations. Strictly speaking, the attacker has learned two things: first, he knows that a message was sent, and second, he has determined the length of that message. The value of this information can be negated by the cryptographer, however, if he designs the system to occasionally transmit "null" messages that have no value and if the system appends "null" information to some messages in order to conceal their true length.

If the attacker is to compromise the secrecy system he will utilize the information gained about the key, or the message, from observations of the ciphertext. The unicity distance, N_o, is defined as the minimum number of ciphertext characters that the attacker must observe before he may deduce a unique key or message corresponding to the ciphertext. This is equivalent to reducing the key or message equivocation to approximately zero.

In Appendix F we define the key and message equivocations and present a detailed development of the unicity distance index. The appendix develops a precise definition of the general unicity distance index of the form,

$$N_o = \frac{K \log L_K}{\log L_C - R_o \log L_M}, \tag{4.1}$$

and presents a simplified equation of the form,

$$N_o = \frac{K}{D}, \tag{4.2}$$

for a cryptosystem with alphabet sizes $L_K = L_C = L_M$, and where K is the size of the key

in bits, and $D = 1 - R_o$ is the percentage redundancy of the source process.

Also presented in Appendix F is a development of an expression for the unicity distance in ciphers utilizing a randomizer as defined in figure 4.4. If the randomizing stream, $\mathbf{R}$, is chosen independently of the message and key, then the key equivocation for the cryptosystem can be expressed as,

$$N_o = \frac{K + H(\mathbf{R})}{D}, \tag{4.3}$$

where $H(\mathbf{R})$ is the entropy of the randomizing process. This development reveals that a randomizer can increase the unicity distance of a cipher.

4.5 Unicity Distance as a Security Index

Shannon's derivation of the unicity distance assumed a particular form that he called a "random" cipher. The result of equation (4.1) has been found applicable to ordinary ciphers wherever it has been possible to test the correspondence [58]. Furthermore, it was shown by Hellman in [52] that Shannon's random cipher model was unduly pessimistic about security. To paraphrase Hellman's results, he found that the unicity distance predicted by equation (4.1) was the minimum achievable by ordinary, "non-random" ciphers.

Let us now examine the use of the unicity distance as a security index. The unicity distance of a cipher is determined by the key entropy, $H(\mathbf{K})$, the redundancy of the source process being encrypted, D, and in some cases the entropy of the randomizer, $H(\mathbf{R})$. The unicity distance indicates the number of ciphertext characters that the attacker must observe in order to determine a unique solution to the cipher. If the attacker observes a number of characters lower than the unicity distance he may at best construct a residue class of messages that could have produced the ciphertext. Beyond the unicity point an attacker can always determine a unique solution to the cipher. The capture of additional ciphertext serves to increase his efficiency by providing more data with which to check the consistency

of his solution.

We can explain this effect by noting that redundancy in the source process works to impose a structure on the cryptogram. This structure is counteracted by the encryption algorithm, which uses the key to add entropy to the ciphertext. Creation of the ciphertext uses incremental amounts of the key. As more of the ciphertext is transmitted more of the key is used, to the point where all of the entropy afforded by the use of a particular random key has been applied to the encryption routine. Beyond this point the ciphertext is vulnerable to decryption.

The unicity distance neatly captures these qualities. The designer may seek to increase security by increasing the amount of key entropy available to the encryption operation, he may attempt to reduce the redundancy of the source process prior to encryption, and he may use a randomizer to provide additional uncertainty for the attacker.

4.5.1 Security by Key Selection

We can describe the security of a cipher system formally by the classifications of perfect, imperfect, and ideal security in [51]. In this section we will consider the effect that controlling the key entropy can have on the unicity distance for each of these categories of cipher.

Perfect Ciphers

The Vernam cipher described in section 4.1.1 can achieve perfect security. These ciphers have practical use when the greatest importance in the system design considerations is security, and the cost or inconvenience of the implementation is secondary.

The definition for perfect secrecy is that the ciphertext reveals no information about the message,

$$H(\mathbf{M}|\mathbf{C}) = H(\mathbf{M}). \tag{4.4}$$

For any cryptosystem we have,

$$\begin{aligned} H(\mathbf{M}|\mathbf{C}) &\leq H(\mathbf{M},\mathbf{K}|\mathbf{C}) \\ &= H(\mathbf{K}|\mathbf{C}) + H(\mathbf{M}|\mathbf{K},\mathbf{C}) \\ &= H(\mathbf{K}|\mathbf{C}) \\ &\leq H(\mathbf{K}). \end{aligned} \tag{4.5}$$

Combining equations (4.4) and (4.5) we find that for perfect secrecy the entropy of the key source must be larger than that of the message source,

$$H(\mathbf{K}) \geq H(\mathbf{M}), \tag{4.6}$$

so that we will not exhaust the supply of random information that is necessary to counteract the redundancy of the message. This result is the reason for the high cost and potential inconvenience of a perfect cipher: to encipher an information transmission, an equal amount of key information must be exchanged by means of an alternate secure channel in advance. The security of the alternate channel must also be perfect in order to ensure a secure transmission, so this system typically requires the physical exchange of a large volume of key information.

By equations (4.5) and (4.6) we can bound the key equivocation of the perfect cipher,

$$H(\mathbf{M}) \leq H(\mathbf{K}|\mathbf{C}) \leq H(\mathbf{K}).$$

If the key source generates random information at a rate of R_K, then these bounds may be expressed as,

$$R_o \log L_M \leq \frac{1}{j} H(\mathbf{K}|\mathbf{C}) \leq R_K \log L_K.$$

For $\{R_o > 0, L_M > 1\}$ there is no solution to the unicity equation (F.9), such that

$H(\mathbf{K}|C_1^j) \approx 0$. Thus the unicity distance of a perfect cipher is infinite.

Imperfect Ciphers

The class of imperfect ciphers are those ciphers not meeting the key entropy requirement defined in equation 4.6. Imperfect ciphers are vulnerable to analysis because the amount of key information available to the encryption algorithm is insufficient to conceal all of the message information. These systems are described by the equation,

$$I(M_1^j; C_1^j) = H(M_1^j) \quad \text{for } j \geq N_o,$$

which indicates that the information leakage from the ciphertext is sufficient to reveal the message beyond the unicity point, N_o.

The majority of practical ciphers fall into this category as they are designed to encrypt large amounts of information using a conveniently sized key. The cipher designer cannot avoid this weakness, but he can increase the security of the cipher by increasing the entropy of the key. It is clear, by equation (4.2), that the unicity distance increases in direct proportion to the key entropy.

It is interesting to note that, in this regard, the theoretical security index of the unicity distance corresponds to the notion of practical security: the practical security of a system is increased by expanding the key space that would need to be searched by an exhaustive attack.

Ideal Ciphers

There is a third classification for ciphers that are limited to a finite key size, but still achieve an infinite unicity distance. These ideally secure ciphers are described by the bounds,

$$0 < I(M_1^j; C_1^j) < H(M_1^j) \quad \text{for large j.}$$

In these systems there remains a residual uncertainty about the plaintext which cannot be resolved by analysing large volumes of ciphertext, hence the key equivocation is non-zero for any amount of intercepted text and the unicity distance is infinite.

It is important to note that ideal ciphers do not ensure unconditional security as the uncertainty may be limited to only a portion of the message space, or that uncertainty may be immaterial to determining the content of the message. We note that a unique description of intercepted ciphertext is not always necessary to compromise a security system. Development of a residue class of likely messages can lead to a distorted, but still meaningful, decryption of some types of source process.

4.5.2 Security by Source Coding

It is clear from the development of the unicity distance measure of equation (4.2) that redundancy plays a key role in determining the theoretical security of a cipher system. Cryptographers have long been aware of the advantages of removing redundancy prior to encrypting the message stream, and cryptanalysts have for just as long taken advantage of the portions of the message they could predict to compromise the system. Kahn presents an entertaining historical account of the cryptanalysis of many classical ciphers through the use of redundant information in [49].

If we allow the cryptographer the option of including a source coding operation prior to the encryption operation we may observe two characteristics that improve the security of the cipher system. First, source coding increases the unicity distance by decreasing the message redundancy, as indicated by equation (4.2). Second, source coding reduces the transmission rate required for the message. This improves secrecy by decreasing the amount of ciphertext available to the attacker for analysis.

It is interesting to note that the unicity distance for an entirely non-redundant message is infinite. This implies that through effective source coding alone, we may obtain perfect secrecy. As an example, consider a non-redundant source process that is encrypted by very

simple encryption algorithm, such as a Caesar cipher[2]. In this case, we are clearly not providing as much key information as there is message information since $H(\mathbf{K}) \not\geq H(\mathbf{M})$, so the system does not meet the definition for perfect secrecy. Despite the weakness in the encryption algorithm, the attacker cannot determine a unique solution to a cipher since any message of a given length from a non-redundant source is equally likely. He may invert the Caesar cipher for all keys in the key space, but will still be unable to determine which of the set of messages could have produced the ciphertext, since all are equally capable.

It is instructive to note the effect of a known-plaintext, or chosen-plaintext attack on the system described above. These attacks will easily reveal the key being used by the cipher, and so compromise even the non-redundant message. This breach of security is identified by the unicity distance model of security if we consider the information gained by the attack to add redundancy to the ciphertext. A redundant ciphertext, encrypted by a low entropy key, will have a finite unicity distance.

4.5.3 Security by Randomization

It is clear, by equation (4.3), that the unicity distance of a cipher is linearly proportional to $H(\mathbf{K}) + H(\mathbf{R})$. Thus, we can increase the unicity distance by increasing the entropy of the randomizer. Through the use of a randomizer, the designer increases security by increasing the number of possible ciphertexts seen by the attacker. The randomizer cannot reduce the real redundancy of the message source, however, so its affect on security is linear in proportion to the size of the randomizer stream. We note that randomization effectively increases the size of the message space, and consequently the cipherspace may have to be increased to transmit the random information. The higher the proportion of random information in the cipher stream, the lower the efficiency of the transmission channel. Security by randomization is therefore constrained by bandwidth and efficiency considerations in the overall system design.

---

[2]The Caesar cipher is a monoalphabetic substitution algorithm where the ciphertext is produced by a key-dependent cyclic shift of the source alphabet

4.6 New Security Indexes for Encryption Devices

4.6.1 Limitations of the Unicity Distance Index

We have shown the unicity distance to be a valid security index for private key ciphers and have noted a correspondence in the parameters of the index to some of the notions of practical security. Notably, we found that both theoretical and practical security depend on a high key entropy and low redundancy in the message source.

It is clear, however, that the unicity distance index is not sufficient to define the relative security of all cipher systems. Specifically, we note that the unicity index can not distinguish between different ciphers that happen to use the same key space and operate on the same source process. It is possible for two ciphers, one providing excellent security and one revealing significant portions of the message, to have the same unicity distance.

As an example, consider two systems with the same keyspace enciphering the same message source. A system with a complex substitution-permutation network and key inclusion strategy can provide good security. Alternatively, we could design a poor system, one that applies an encryption algorithm in a secure fashion to every nth character but leaves the subsequent $n-1$ characters unchanged. In this extreme case, both encryption algorithms will yield the same unicity distance measure, but only the first one is secure if the source process exhibits an appreciable degree of redundancy. The set of ciphers that meet the design criteria of Proposition 2 will not include the poor system indicated in the extreme example above, but can contain variation in the extent to which the design principles are implemented. It is expected that in the case of block ciphers, for instance, a greater number of substitution-permutation rounds will result in greater security. The unicity distance measure is insensitive to these characteristics of cipher design.

4.6.2 Encryption Efficiency Index

We have demonstrated how the security of an encrypted transmission is partially indicated by the unicity distance index, but have identified that additional criteria are necessary to define the security of a cipher.

Clearly, the nature of the transformation applied to create the ciphertext must affect the security of the system. We expect the encryption algorithm to apply key and randomizer information to message information in such a way as to increase the entropy of the ciphertext. If the encryption algorithm is effective, then the ciphertext will have a higher entropy than the plaintext. The difference between the two measures is indicative of an increase in security, but may be misleading in that it does not reveal the amount of key and randomizer information that was expended to achieve the improvement. We propose to study a security index first examined in [59] and extended here to consider randomizers.

Proposition 3 (Efficiency Index) *A general measure of the efficiency of an encryption algorithm is given by normalizing the increase in entropy achieved in transforming the message to ciphertext by the amount of key entropy and randomizer entropy needed to obtain the increase. We define an efficiency index, $\mathcal{E}$, for a cipher by,*

$$\mathcal{E} \triangleq \frac{H(\mathbf{C}) - H(\mathbf{M})}{H(\mathbf{K}) + H(\mathbf{R})}.$$

Systems employing randomizers will not be considered in our experimental models, so we may simplify the efficiency index to,

$$\mathcal{E} \triangleq \frac{H(\mathbf{C}) - H(\mathbf{M})}{H(\mathbf{K})}, \tag{4.7}$$

and consider only the effect of key entropy.

We may apply this index to compare cipher algorithms for encrypting a given message source with a given key space. The cipher that adds more entropy to the ciphertext per unit

of key entropy may be considered to be performing a more efficient encryption operation. The normalization operation removes the dependency of the index on the amount of key entropy, so the security index, $\mathcal{E}$, may also be applied to determining the relative efficiency of ciphers operating with different key spaces.

The secure algorithms considered in this research may be found to operate in the range $0 \leq \mathcal{E} \leq 1$, where key information is added to the ciphertext by the encryption operation. The degree of key inclusion is indicated by the size of $\mathcal{E}$. The encryption efficiency index constitutes a measure of the relative efficiency of a cipher and allows ciphers to be distinguished on that basis.

Note 2 (Reference to unicity derivation) *We should point out that the efficiency distance defined above differs from the derivation of the unicity distance in Appendix F in that it does not assume the ciphertext entropy, $H(C_1^j)$, can achieve the upper bound of $j \log L_C$ as expected in the unicity distance proof. Instead, for encryption algorithms A and B, operating on the same message and key spaces we are accounting for the possible variation in ciphertext entropy,*

$$H(C_A) \neq H(C_B) \leq \log L_C,$$

where C_A, and C_B represent ciphertext produced by algorithms A and B, respectively.

4.6.3 Encryption Quality Index

A second theoretic security index was presented in [59] to evaluate encryption systems on the basis of their relative resistance to redundancy. We propose to include this index as a theoretic security measure.

Proposition 4 (Quality Index) *A general measure of the quality of an encryption algorithm is given by normalizing the increase in entropy achieved in transforming the message to ciphertext by the amount of redundancy in the message process being encrypted. We*

define a quality index, $\mathcal{Q}$, for a cipher by,

$$\mathcal{Q} \triangleq \frac{H(\mathbf{C}) - H(\mathbf{M})}{D}, \tag{4.8}$$

where D is the redundancy of encrypted source process, $\mathbf{M}$.

This measure indicates the incremental increase in ciphertext entropy compared to the message redundancy. It expresses how well a cipher counteracts the redundancy inherent in the message. Ciphers that add a large amount of entropy relative to source redundancy will have high quality indices. We note that source processes with a low redundancy require less additional entropy to obtain security; ciphers for low redundancy processes can attain high quality indices with the addition of little extra entropy to the ciphertext.

4.6.4 Relevance of the New Security Indices

It is interesting to note that the unicity distance may be expressed as a function of the two new security indices, $\mathcal{E}$ and $\mathcal{Q}$. Combining equations (4.7) and (4.8) we find,

$$N_o = \frac{\mathcal{Q}}{\mathcal{E}} = \frac{H(\mathbf{K})}{D}. \tag{4.9}$$

Whereas the unicity index yields a measure of the absolute security of a cipher, the values of the efficiency and quality indices have no direct relation to a quantifiable level of security. In fact, there are other possible arrangements of the terms in each index that would yield measurements that may be classified as efficiency or quality indices. For instance, we may compute $\mathcal{E}' = \frac{(H(\mathbf{C})-H(\mathbf{M}))^2}{H(\mathbf{K})}$, or $\mathcal{Q}' = \frac{\sqrt{H(\mathbf{C})-H(\mathbf{M})}}{D^2}$ for the required measures.

It should be stressed, therefore, that the efficiency and quality indices may only be used to evaluate the *relative* behaviour of different encryption coders.

4.6.5 Entropy as a Security Measure

The efficiency index and the quality index are based in part on a measurement of the amount of entropy added to the message by the encryption operation. The unicity index and the efficiency index consider a measurement of the entropy of the key space of the cipher. Even the redundancy measure is based on the entropy rate of the source process. Entropy measures quantify the uncertainty an observer would face about guessing the outcome of a random variable. Using entropy in the context of a theoretical security measurement is therefore appropriate when the security index incorporates entropy to express in some manner the amount of work the attacker must perform to break the system.

Shannon employed measures of entropy and conditional entropy in deriving the unicity index [4]. More recently, Maurer employed entropy as an indicator of security in random bit generators [60]. The efficiency and quality indices employ entropy to express measures that are dependent on the success of the encryption algorithm at adding to the amount of analysis required by the attacker.

4.7 Summary of Proposed Security Models

We have 3 theoretic indices we could apply to model the cryptographic security of speech encryption systems. They are,

1. **Unicity distance.** $N_o = \frac{H(\mathbf{K})}{D}$.
2. **Efficiency index.** $\mathcal{E} = \frac{H(\mathbf{C})-H(\mathbf{M})}{H(\mathbf{K})}$.
3. **Quality index.** $\mathcal{Q} = \frac{H(\mathbf{C})-H(\mathbf{M})}{D}$.

The three indices require measures of the key entropy, the ciphertext entropy, message entropy, and the associated measure of message redundancy. We have omitted the use of a randomizer. Were all of these processes memoryless these measures could be obtained quite

easily, but the speech process is highly redundant and this redundancy will likely introduce some memory into the ciphertext process as well. Only the key process can be designed to be memoryless and allow an easy entropy calculation.

In chapter 6 we will apply the entropy estimation techniques developed in chapter 2 to obtain measures of ciphertext entropy for a variety of secure speech system implementations. These results will be combined with the speech process entropy estimation results of chapter 3 to create models of secure speech system security.

Chapter 5

Bit Rate and Quality Models for Secure Speech Systems

Come, give us a taste of your quality.

William Shakespeare. Hamlet. Act II, Sc. 2.

5.1 Introduction

A subjective evaluation of the quality of an encrypted speech transmission, and an understanding of the bit rate required to achieve that quality are two parameters of great importance in the design of a secure communication system. We wish to provide an outline for such models of performance within the framework of a speech encryption system.

We begin in section 5.2 with an overview of the methods currently used for coding the speech process. We will review the techniques used in a variety of waveform and vocoder-type speech coders, and summarize the standard performance characteristics. As it is infeasible to perform a timely experimental study of all possible variations of speech coding algorithms, we will define a subset of coders that are generally representative of a range of

coding techniques and capabilities available to designers of secure speech communication systems.

At the limit of optimal coder design, the relationship between quality and rate is clearly defined by the rate-distortion curve of the speech process. We will consider this relationship in section 5.3. In practical speech coder implementations we are limited to a maximum feasible complexity, and so it is unlikely that we can achieve minimal theoretical distortion at a chosen rate. For the broader class of speech coders of interest to designers of practical systems, we must therefore derive operational rate-distortion functions. In particular, we must consider the effect that an encryption process may have on coding distortions under a variety of channel conditions.

We will consider in section 5.4 the problem of defining a distortion measure which can be easily computed and has good correspondence to the results obtained by a subjective evaluation. Following the work of Quackenbush [61] we select a segmental-SNR measure for its simplicity and relative accuracy.

We will then present in section 5.5 the results of some experimental work using the segmental-SNR measure to analyse the operational rate-distortion characteristics of the set of representative speech coders. These results will serve to define the general relationship between quality and bit rate for the set of speech coders deemed representative of the range of coders available to secure system designers. We will then extend the analysis to consider the effect of a variety of channel error types on the perceived quality of encrypted speech.

In section 5.6 we will present a model of bit rate as a function of the residual redundancy in the coded speech signal. We will describe a set of experiments performed on the representative speech coders to determine an information-theoretic measure of the divergence of coded speech from the original. This study will reveal a characteristic relationship between redundancy and the bit rate achieved by typical speech coders.

5.2 Speech Coders

In an integrated secure communication system, the speech coder is responsible for producing a digital representation of the continuous speech waveform. This coding operation determines a number of important characteristics for the communication session, notably speech quality, bit rate, encoding delay, and error sensitivity.

In an unsecured system, the level of sophistication, or complexity, in the speech coder will largely determine the combination of quality and bit rate achieved by the device. In a secure speech communication system, however, the performance of the system is dramatically affected by the encryption device. The encryption operation introduces a hard limiter into the communications path that typically exacts a high penalty for any channel or synchronization errors. Highly compressed speech is quite sensitive to errors and so the combination of encryption and source coding techniques must be carefully balanced.

The following sections will review six common coding algorithms. These include a number of waveform coders and also some of the vocoding type. This is by no means an exhaustive study, but the methods selected are generally representative of the majority of speech coding techniques. Detailed coverage of speech coders may be found in [2], [62], [22], and [23].

5.2.1 Pulse Code Modulation

Pulse Code Modulation (PCM) is the most basic form of speech coding. These coders sample the analog speech waveform and encode it into a binary signal. PCM coders are *instantaneous*, meaning that they require at most one sample period of delay. In PCM, each sample is quantized to one of 2^R amplitude levels, where R is the number of bits used to represent each sample. At a sampling rate of f_s, which must be at least the Nyquist rate [63], the PCM coder requires a bandwidth of Rf_s bps.

As most telephone facilities support a bandwidth of less than 4 kHz, the sampling rate

for speech is generally chosen to be 8 kHz. Subjective tests have found that to achieve toll quality speech requires somewhere in the range of 12 quantization bits. Thus the uniform PCM coder has a bit rate of 96 kbps [23].

Reductions in bit rate with only a minor increase in complexity can be achieved by applying a non-linear quantization rule. It has been observed in many studies of the speech waveform that small signal amplitudes occur much more frequently than large signal amplitudes. By designing a quantizer which has more closely spaced levels at the low signal amplitudes and more widely spaced levels at large amplitudes we will reduce the average quantization error. These non-linear quantizers can greatly improve signal quality and hence allow a smaller number of quantization bits to be used. Two popular non-linear quantizers, the A-law and μ-law devices, are based on a logarithmic function. This form of coding, referred to as log-PCM, is capable of providing communications quality voice at 8 bits/sample for a transmission rate of 64 kbps.

Log-PCM coding at 64 kbps is often used as a standard against which other coders are judged. Due to its simplicity, log-PCM often forms the standard input to a variety of more sophisticated coding algorithms.

5.2.2 Differential Pulse Code Modulation

Differential Pulse Code Modulation (DPCM) takes advantage of correlation in the speech signal to achieve significant gains in signal compression. Instead of quantizing speech samples directly, as in PCM, DPCM quantizes the difference between a current sample and a predicted estimate that is calculated as a weighted average of previous samples. Since part of the input signal is predictable, only the unpredictable residual signal need be transmitted. The residual signal is calculated as the difference between the input signal and the predicted signal, $\hat{d}(n) = Q[x(n) - \hat{x}(n)]$, where $Q[\cdot]$ represents the quantizing operation. The residual signal has a smaller variance than the input signal and so can be quantized more efficiently.

The additional compression permitted by this coding method, and the more sophisticated methods to be discussed below, come at the expense of an increased sensitivity to errors in the data stream. Removing redundancy and predicting on the basis of previously received samples means that the effect of an erroneous sample can propagate through subsequent samples. These types of coders may employ some side information to limit error propagation and to allow re-synchronization with the transmitter.

5.2.3 Adaptive Coding Methods

The short-term statistics of the speech process are non-stationary, which causes static coding algorithms like PCM and DPCM to yield suboptimal results. An adaptive design can improve upon the performance of these schemes by adjusting the quantizer step size according to the magnitude of the previous speech sample. A typical arrangement would be to adjust according to the relation: $\Delta_{n+1} = \Delta_n \cdot M(n)$, where $M(n)$ is an appropriate multiplicative factor. This form of adaptation is applicable to a variety of coders. When applied to a simple PCM system it is called Adaptive PCM (APCM).

In differential-type applications it is also possible to adapt the predictor coefficients with a short-term estimate of the autocorrelation function of speech. By periodically updating these coefficients it is possible to achieve a considerably smaller prediction error and hence a better bit rate. Adaptive Differential Pulse Code Modulation (ADPCM) employs one or both of these adaptations to achieve significant reductions in bandwidth. Toll quality ADPCM samples can be quantized in 4 bits, for a bit rate of 32 kbps. The adaptive design yields better results for higher orders of prediction than are possible for non-adaptive schemes.

5.2.4 Sub-Band Coding

Sub-Band Coders (SBC) achieve signal compression by exploiting speech redundancy in the frequency and time domains. Sub-band coders divide the speech spectrum into a number of

sub-bands and code each bandpass signal using some form of time domain based compression method. Spectral redundancy is reduced by allotting varying amounts of accuracy to the speech bands. The low frequency bands are typically allotted more bits than the high frequency bands due to their greater importance to speech intelligibility. In SBC, each band is usually sufficiently narrow to reduce the dynamic range of the signal. This means the individual bands can be more efficiently coded than in the broad-band waveform coding approaches.

Sub-band coders achieve what is considered to be communications quality speech at rates around 16 kbps. Toll quality SBC can be achieved at a rate of 24 kbps. For toll quality voice at lower rates the coder must be able to dynamically adapt its frequency bands to follow the three major speech formants. These more sophisticated coders save bandwidth and maintain quality by deleting those bands not currently carrying the high-energy formants.

Static sub-band coders generally use Adaptive PCM to quantize the changing amplitude levels as the formants move between bands. DPCM and ADPCM are not particularly effective due to the reduction in spectral redundancy in each of the narrow bands.

5.2.5 Vector Quantization Techniques (Analysis by Synthesis)

Vector Quantization (VQ) is one of the more sophisticated methods for speech signal compression. The VQ system performs a pattern matching operation between a set of some relevant input parameters and a codebook of representative vectors. The system achieves a considerable compression factor by transmitting the vector index to the receiver instead of the vector itself. The receiver performs a look-up operation in a matching codebook to reproduce the vector contents. The system input can be any parameters appropriate to represent the speech signal; PCM samples, LPC or pitch prediction coefficients, or even cepstral coefficients.

System performance is generally limited to a resolution of $r = \log_2 N/k$ bits per vector component, where N is the number of vectors in the codebook and k is the number of

speech parameters in the vector. This approach allows fractional bit rates for individual components, an advantage not found in other coding schemes. VQ also has the effect of reducing the average distortion as compared to scalar quantization methods.

The pattern matching operation is the most challenging aspect of the system. The algorithm must search the codebook for a vector best representing the input parameters. Evaluation is made on the basis of some distortion measure between the input vector x and the candidate vector y. Search size is a limiting factor in these systems. Practical systems typically limit the possible dynamic range of the vector to be quantized.

The most sophisticated of the vector quantization coders employ a technique known as Analysis by Synthesis Predictive Coding. These systems transmit LPC parameters and a vector index for an approximation of the excitation type. Both sender and receiver contain an identical component for synthesising the voice signal from the transmitted parameters. The receiver does just that with the parameters, but the sender uses this subsystem to select the optimal parameters for transmission. The sender employs a closed loop configuration to compare synthesised speech from all possible code vectors to the input vector. The code vector yielding the minimum distortion measure is selected for transmission. These coders make the optimal vector choice for the current system state, resulting in a distortion measure that is on the order of the system resolution.

Analysis by Synthesis systems such as the one described here are referred to as Vector Excitation Coders (VXC) of Code-Excited Linear Predictors (CELP). These systems can provide acceptable quality voice at 16 kbps. They are also capable of much lower bit rates and are often used for secure implementations at rates of 2.4-4.8 kbps. Subjective voice quality at these rates is considerably reduced.

5.2.6 Vocoders

Vocoders identify and model the primary characteristics of speech. They offer a great bandwidth reduction over waveform coders by transmitting only the minimal set a parameters

necessary to re-synthesize the speech waveform. The disadvantage of vocoders is that they are limited to analysing and reconstructing the speech signal with an approximate model of the speech production system. The model can be quite complex, and yet still produce speech lacking the naturalness of a toll quality signal.

It is difficult to compare vocoded speech to waveform-coded speech since the distortions are quite different. Waveform coders suffer primarily from additive noise due to quantization. Vocoders rarely have such background noise but rather, suffer from reconstruction deficiencies such as whistles, burbles, buzziness, harshness, and muffled quality. Increasing the bit rate in vocoded speech does not generally produce a noticeable improvement, as the limitation in these systems is in the vocoder models. To the extent that the two approaches have been compared, it was found that a phase vocoder was approximately equivalent to 3 bit log-PCM speech [23].

Performance improvements can be obtained by the combination of SBC at the more crucial low frequencies and phase vocoding at high frequencies. This hybrid coder can achieve high quality speech results at rates of 16 to 20 kbps.

Currently, the minimum achievable bit rates are obtained by phonetic vocoders. These devices recognize and transmit codes for the phonetic content of the input speech. Assuming approximately 42 phonemes in the English language and a normal speech rate of 10 phonemes/s, the basic minimum speech rate could be as low as $10 \cdot \log_2 42 = 54$ bps. For a broader implementation where there are 1560 possible variations on the basic phonemes (due to articulation constraints) and at a relatively high speech rate of 40 phonemes/s, the maximum rate would be $40 \cdot \log_2(42 \times 1560) = 640$ bps.

Extending compression algorithms beyond the analysis of the physical speech production model leads to a somewhat quixotic specification for the identification of the semantic content of the spoken word. User response to such an ideal device would likely be mixed, however, as many conversations would too easily be identified as containing information rates approaching zero. In any event, such a device is beyond our current capabilities.

5.2.7 Summary of Coder Characteristics

Some of the relevant characteristics of the coders discussed above are outlined in table 5.1. This table indicates the general sense that the higher the speech quality, the higher the corresponding bit rate and that achieving lower bit rates generally requires a more sophisticated algorithm.

Type	Quality	Bit Rate (kbps)	Complexity	Use
PCM	toll	96	very low	common
log-PCM	toll	64	low	common
DPCM	toll	58	low	some
ADPCM	toll	32	low-medium	some
ADPCM-VQ	communications	16	medium	research
SBC	communications	16	medium	research
CELP	communications	4.8	very high	some
LPC	synthetic	2.4	high	common
Formant	synthetic	1.2	very high	research
Phonetic	synthetic	0.2	very high	research

Table 5.1: Properties of Some Speech Coders

Figure 5.1 summarizes the results of table 5.1 in a more qualitative manner (after [16]). The figure indicates in general terms the performance of known coding methods in a test of the subjective voice quality called the Mean Opinion Score (MOS) [64], [65].

There is clearly a relationship between the parameters of bit rate, complexity, speech quality, and error sensitivity in secure speech communication systems. In the following sections we will develop models of quality and bit rate for the class of speech coders commonly used in secure applications.

5.2.8 Representative Speech Coders

It would be infeasible to perform experimental work on all possible variations of waveform coders and vocoders, particularly when researchers are continually introducing improvements to their algorithms. To obtain results that are relevant to the study of secure speech

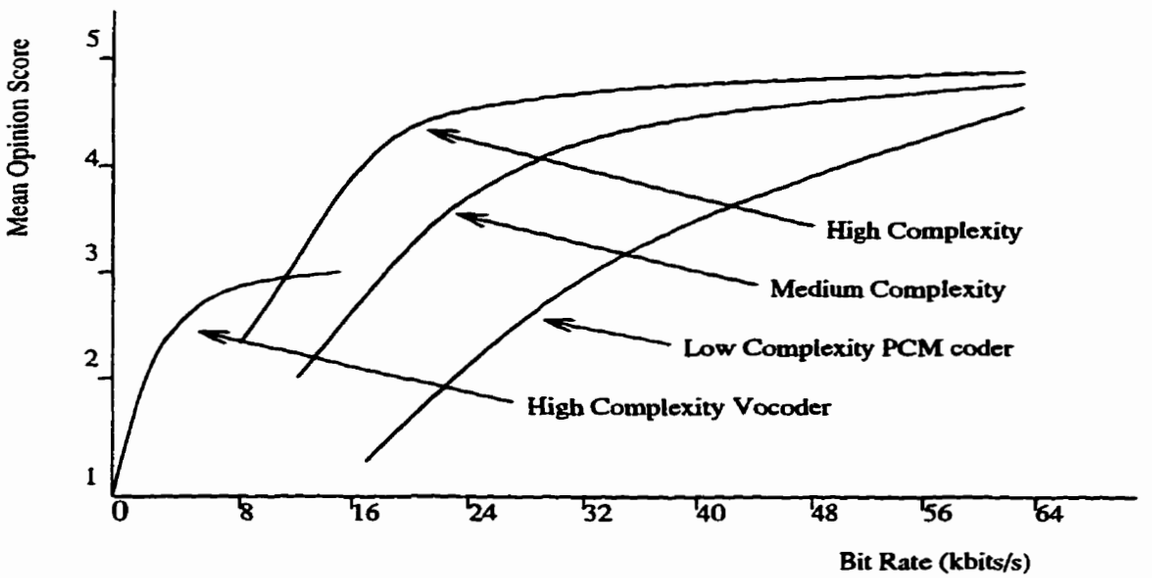


Figure 5.1: Subjective Quality Versus Bit Rate

communication systems, and that also reflect the general state of practical coding achievements we have selected a subset of coders for our experimental work.

The set includes the following coders, for the following reasons,

1. CCITT G.711 standard μ-law PCM [14]. The μ-law PCM quantization forms the standard for toll-quality speech communications and is the standard input to other more sophisticated coders.

2. 32 kbps DPCM [22]. This simple non-adaptive coder represents an intermediate step between PCM and more sophisticated waveform coders such as ADPCM. The non-adaptive predictive filter reduces temporal redundancy.

3. CCITT G.721 ADPCM [66], [21]. This dual-adaptive ADPCM coder represents a high standard in waveform coding that is used in a large number of practical applications. In addition, the CCITT G.723 standard has been applied to provide variable rate ADPCM coding at 24, 32, and 40 kbps. These additional rates allow a wide coverage of potential bit-rate ranges for speech coders.

4. Federal Standard 1016 4.8 kbps CELP coder [67]. This sophisticated design forms an accepted reference point for other vocoder designs and achieves an attractively low bit rate. The coder is widely used for low bit rate applications.

Presented in Appendix G are brief descriptions of the software algorithms used to implement the representative coders. A study of the SNR performance of these coders is provided in Appendix H.

5.3 Rate-Distortion Measures

There is a theoretical basis for the relationship between the rate and quality of a transmission that is defined for an optimal coding of the process. The rate-distortion function, $R(D)$, indicates the minimum rate that may be achieved for a given level of distortion in the source coded signal [57]. This relationship can give insight to the behaviour of rate and quality parameters in the class of non-optimal practical coders of interest to secure system designers.

5.3.1 Memoryless Sources

The quantity $R(D)$ is defined by the minimum value of mutual information between the source coder's input and output that is needed for signal reconstruction with a fidelity of D.

We can derive bounds on $R(D)$ for the class of stationary memoryless source processes. Using a mean squared error distortion measure we can derive the MSE generalized Shannon lower bound, $R^*(D)$. We define the set of admissible transformations from X to Y as those

pairs (x, y) where the MSE distortion, $E[\rho(x, y)]$ is bounded by D. Therefore,

$$\begin{aligned} R^*(D) &= \min_{(x,y):E[\rho(x,y)]\leq D} I(X;Y) \\ &= \min_{(x,y):E[\rho(x,y)]\leq D} [H(X) - H(X|Y)] \\ &= \min_{(x,y):E[\rho(x,y)]\leq D} \left[H(X) - \underset{y}{E}[H(X|Y=y)]\right]. \end{aligned} \tag{5.1}$$

If we let $\phi(D_y) = H(X|Y=y)$, and $\phi(D) = \max_{p(x,y):E[\rho(x,y)]\leq D} H(X|Y)$, then minimization is achieved by,

$$\begin{aligned} R^*(D) &\geq H(X) - \underset{y}{E}[\phi(D_y)] \\ &\geq H(X) - \phi\left(\underset{y}{E}[D_y]\right) \\ &\geq H(X) - \phi(D), \end{aligned} \tag{5.2}$$

where we utilize Jensen's inequality for $H(X|Y)$, a concave function. We obtain a general lower bound by assuming the maximum achievable $\phi(D)$. It can be shown that the conditional entropy, $H(X|Y)$, is maximized when X has a Gaussian distribution. Hence to form our lower bound we choose, $\max\{\phi(D)\} = \frac{1}{2}\log 2\pi eD$, the entropy of a memoryless Gaussian signal with variance D. This yields the the general lower bound,

$$R^*(D) \geq H(X) - \frac{1}{2}\log 2\pi eD\,. \tag{5.3}$$

As Berger noted in [68], an upper bound on $R(D)$ can be defined by the rate-distortion curve of the memoryless Gaussian process, which he proved to be the most difficult source with a fixed second moment to reproduce with respect to the squared error criterion. A proof of this result is presented in Appendix I. The rate-distortion function for any memoryless

distribution is therefore bounded above and below as

$$R^*(D) \leq R(D) \leq R_{Gaussian}(D),$$

where,

$$R_{Gaussian}(D) = \begin{cases} \frac{1}{2}\log_2 \frac{\sigma^2}{D} & 0 \leq D \leq \sigma^2, \\ 0 & D > \sigma^2. \end{cases} \tag{5.4}$$

5.3.2 Sources with Memory

Sources with memory permit greater data compression than memoryless sources as it is often possible to take advantage of the temporal correlation in the signal. For a given level of distortion, these sources may be encoded at a lower rate than similar, but uncorrelated, sources.

A distortion-rate function has been derived for the correlated zero-mean Gaussian source in the work of Kolmogorov [69] and Berger [68]. Simplifying this work yields a distortion-rate function for the Gaussian source under the conditions of small distortions. This simplified distortion-rate function is,

$$D_{Gaussian}(R) = \gamma^2 \sigma^2 2^{-2R},$$

where γ^2 is defined below and distortions are bounded by the minimum of the power spectral density of the process,

$$\phi \leq \min_{\omega}\{S_{xx}(e^{j\omega})\}.$$

This simple function, from [22], makes use of the spectral flatness measure (sfm) presented by Makhoul and Wolf in [70]. The sfm is the ratio of the geometric and arithmetic means of the power spectral density of a process and has the range $0 \leq \gamma^2 \leq 1$. It is defined

as,

$$\gamma^2 = \lim_{N\to\infty} \frac{\prod_{k=1}^{N} S(\omega_k)^{\frac{1}{N}}}{\frac{1}{N}\sum_{k=1}^{N} S(\omega_k)}, \tag{5.5}$$

where $S(\omega_k)$ is the speech energy within a frequency bandwidth of $\pi(k-\frac{1}{2})N$, centered at ω_k. The spectral flatness measure is a useful tool for describing the shape of a power spectral density, and hence the correlation in the source, by a single value. The inverse of the sfm is a measure of the waveform predictability, often referred to as the prediction gain.

As in the case of memoryless sources, we can bound the rate-distortion function of a source with memory. For a given level of memory in the source, as expressed by the sfm measure γ^2, we can upper bound the function with the rate distortion function of a Gaussian source with the same characteristic of memory. As a lower bound, we refer to the derivation of the MSE generalized Shannon lower bound in equation (5.3). Our general lower bound was achieved by assuming the maximum possible value for $\phi(D)$ in order to minimize the rate-distortion function of equation (5.1). The presence of memory in the source does not justify the modification of this assumption in deriving a lower bound.

The rate-distortion function of a general source process with memory is bounded as follows,

$$R^*(D) \le R(D) \le R^{\gamma^2}_{Gaussian}(D),$$

where,

$$R^{\gamma^2}_{Gaussian}(D) = \begin{cases} \frac{1}{2}\log_2 \frac{\gamma^2\sigma^2}{D} & 0 \le D \le \gamma^2\sigma^2, \\ 0 & D > \gamma^2\sigma^2. \end{cases} \tag{5.6}$$

The corresponding distortion-rate bounds are,

$$\frac{1}{2\pi e} 2^{-2(R-H(X))} \le D(R) \le \gamma^2\sigma^2 2^{-2R}. \tag{5.7}$$

5.3.3 Quality-Rate Bounds on the Speech Process

Accurate distortion-rate bounds can be obtained by evaluating equation (5.7) using measures specific to the conversational speech process. We can then derive a normalized measure to represent a quality-rate function for the process that will serve as a reference for the experimental quality measures presented in section 5.5.

To determine an upper bound for the process we computed a spectral flatness measure of $\gamma^2 = 0.2$ according to equation (5.5) for a large sample the speech process. The power spectral density of this process is presented in figure 5.2. Using equation (5.7) we can define the upper bound on the distortion-rate function to be a Gaussian process with memory characteristic equivalent to $\gamma^2 = 0.2$.

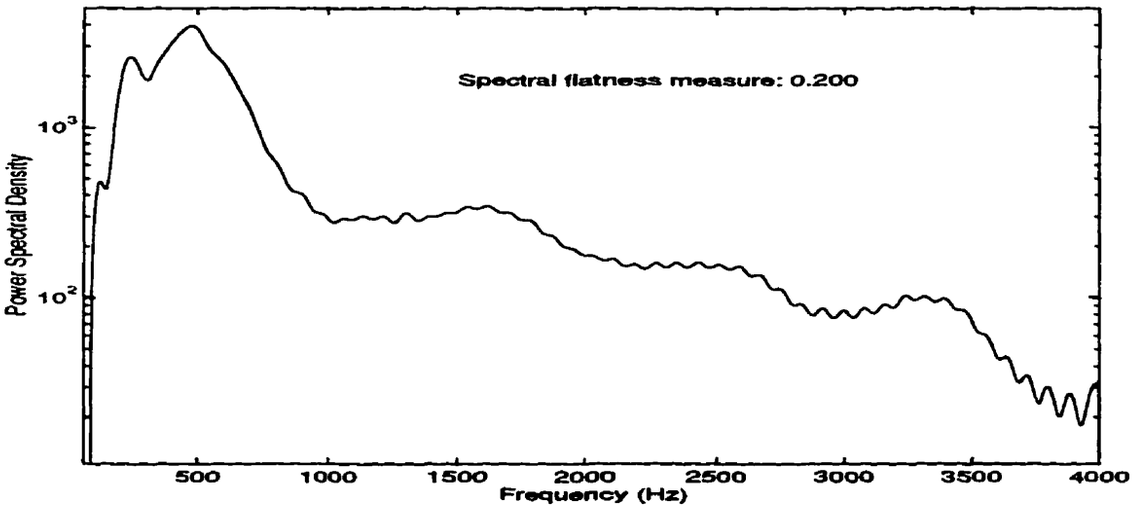


Figure 5.2: Power Spectral Density of the Speech Process

The sfm of 0.2 indicates a speech predictability measure for the source process that is lower than that usually reported for speech. This lower predictability is the product of our definition of the source process as including all conversational English speech at μ-law PCM fidelity. The database used to compute the spectral flatness measure was very large, and

contained many different speakers with different voice characteristics, making generalized prediction of the source difficult.

A lower bound is defined by the MSE generalized Shannon lower bound and requires an accurate estimation of the entropy rate of the speech process. Drawing upon the entropy rate estimate of $H(X) = 2.79$ bits/sample computed in chapter 3 we can evaluate a lower bound for the conversational English speech signal according to equation (5.7).

From these distortion-rate bounds we can derive a pair of quality-rate bounds for the speech signal. Presented in figure 5.3 is a normalized quality measure, $\frac{\sigma^2 - D(R)}{\sigma^2}$, derived from the upper and lower distortion-rate bounds of the speech process.

This figure indicates the region in which we can expect to bound the optimal performance characteristics for coded conversational 64 kbps μ-law PCM speech. The bounds also serve to indicate the apparently logarithmic relationship between the rate and quality in an optimal source coder. We must recognize however, that this level of optimal performance may only be achievable by a coder of infinite, or at least infeasible, complexity. To further our study of secure speech communication systems we must explore the relationship between quality and rate that is achieve by practical complexity-limited source coders.

5.4 Objective Quality Measures for Speech

In order to derive an operational rate-distortion model for real speech coders we must determine an appropriate measure of quality with which to quantify distortions of the speech process. Typically, the best measures of quality are those derived from the subjective evaluations of human listeners. This type of evaluation is difficult, time-consuming and expensive to perform accurately; we therefore desire an objective quality assessment which can be easily computed and shows a high correlation to the results of human preference tests.

More than ten years of research were performed at the Georgia Institute of Technology in

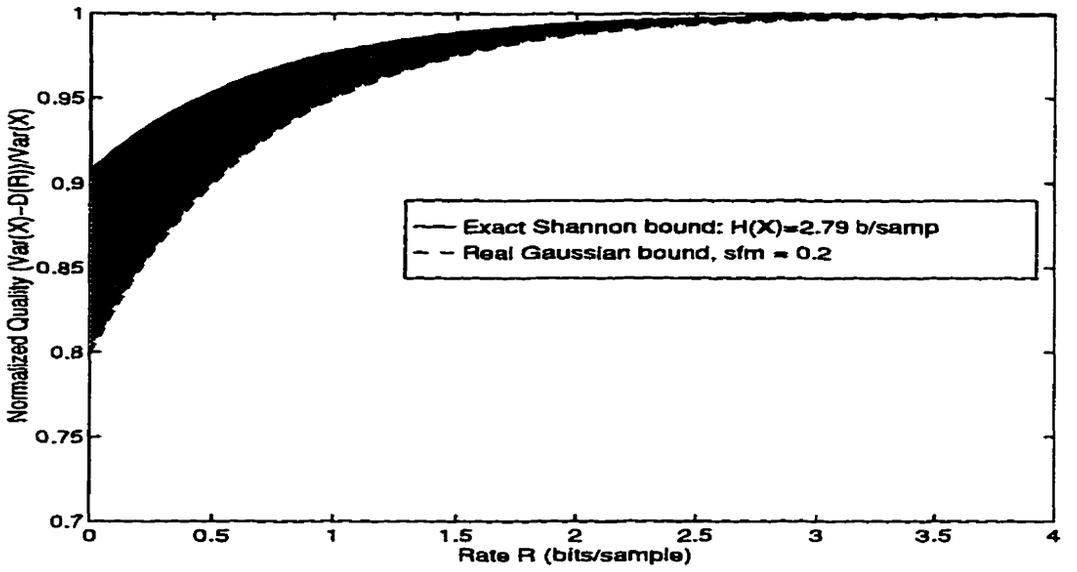


Figure 5.3: Quality-Rate Bounds for the Real Speech Process

pursuit of relevant objective speech quality measures. The major contributions of the work are summarized by Quackenbush, Barnwell, and Clements in [61]. This research subjected a large database of 17.2 hours of speech to a variety of moderate distortions originating from transmission, coding, and other sources of corruption. The distorted and original speech samples were then evaluated for subjective quality with the Diagnostic Acceptability Measure [71] and a suite of objective quality measures.

The authors of [61] identified a large set of speech distortions that occur in the coding and transmission of digital speech. This work was particularly useful for our purposes because it included distortions present in PCM, Adaptive PCM, ADPCM, LPC, Subband and vocoding systems. In addition, pure time and frequency domain effects such as additive noise, clipping, echo, and lowpass, highpass and bandpass filtering were considered for their effect on subjective quality.

In selecting relevant objective speech measures and typical distortion types, the authors referred to a significant body of prior work, including [72], [73], [74], and [75]. A partial

list of the correlation coefficients they obtained between subjective measures and objective evaluations are summarized in table 5.2.

Objective Speech Quality Measure	$\|\rho\|$
SNR	0.24
Segmental-SNR	0.77
Frequency variant Seg-SNR	0.93
Itakura energy ratio	0.59
Linear spectral distance	0.38
Inverse linear spectral distance	0.63
Log spectral distance	0.60
Polynomial regression modelling,	
Spectral distance	0.80
Frequency variant log spec. distance,	
LPC-based	0.68
Filter bank	0.72
Composite measures,	
Simple & freq. var. measures	0.86
Parametric measures	0.82

Table 5.2: Correlation of Objective Measures to Subjective Measures

From the correlation results of table 5.2 it is clear that a single simple objective measure of speech quality is not capable of perfect correlation with subjective measures over all possible speech distortions. Some of the objective measures do, however, indicate a reasonably close correspondence to the subjective results. We must note that the results for the SNR-based correlations were compiled exclusively on the basis of distortions of the class that could be produced by waveform coders. These results do not include distortions classified in [61] as vocoder-type, sub-band, and frequency domain distortions. This restriction was imposed because signal-to-noise measures are only appropriate for distortions that produce a facsimile that can be time-aligned with the original signal. Many of the non-waveform type of distortions produce SNR evaluations that are unjustifiably low, despite having a reasonable subjective quality level. This is typically due to the difficulty of time-aligning the original and coded signals for comparison.

The majority of our representative set consists of waveform coders, with the exception of the CELP algorithm. With this in mind, we have identified the segmental-SNR as a suitable objective measure for evaluating the quality of speech coders under the conditions of a secure communications channel. Experimental evidence to be presented in the following sections will verify that the segmental-SNR results obtained for CELP-coded speech are still reasonably consistent with the results of waveform-coded speech.

The segmental-SNR measure is defined as,

$$SegSNR = \frac{10}{M} \sum_{m=0}^{M-1} \log_{10} \sum_{n=N_m}^{N_m+N-1} \left(\frac{x^2(n)}{[x_d(n) - x(n)]^2} \right), \tag{5.8}$$

where $x(n)$ is the original signal, $x_d(n)$ is the distorted signal, N is the segment length and M is the number of segments in the signal being analysed. The segmental-SNR measure allows an unbiased measure of both high and low amplitude portions of the signal. The segmental-SNR computes short-term SNRs over a set of consecutive sample segments and averages the results. This measurement technique parallels one characteristic of the human auditory system in that a short-term noise burst will have only a limited effect on the overall assessment of quality. This similarity to the natural human response mechanism helps to explain the correlation between the segmental-SNR and subjective listening measurements.

Referring to the correlation results of table 5.2 we see that with a measure of 77%, the segmental-SNR measure has good relevance to subjective evaluations. In addition, it is an attractive choice for a quality index due to its relative simplicity.

We note that while the frequency-weighted segmental-SNR achieves an excellent correlation measure, it is significantly more complicated to compute. Similarly, the composite measures proposed in table 5.2 are relevant to an even broader class of distortions, but cannot be easily integrated into our models of speech quality.

5.5 Operational Rate-Distortion Results

Rate-distortion performance evaluation of secure speech communication systems is complicated by the hard-limiting nature of most encryption devices. We will move away from the purely theoretical analysis of rate-distortion performance to consider the impact of channel noise on the subjective quality of a secure speech system. Using experimentally derived segmental-SNR measures for the representative coders under a variety of channel conditions, we will present a set of operational rate-distortion curves. These curves will serve to represent the relative performance of our speech coders under secure communications conditions and will allow us to postulate models of quality as a function of redundancy.

5.5.1 Testing Methodology

To determine operational rate-distortion curves for the set of representative coders we performed a set of experiments on a large volume of speech data. The testing methodology is represented in the block diagram of figure 5.4. A database of speech samples was subjected to a variety of controlled distortions. The distorted signal was then compared to the undistorted original according to an objective quality evaluator. For the reasons discussed in section 5.4 we chose the segmental-SNR measure to evaluate the distortion.

The testing methodology outlined in figure 5.4 was applied to evaluate the relative quality of the various speech coders under noiseless conditions, and also under a wide variety of non-ideal conditions relevant to encoding in a cryptographically secured environment. These non-ideal channel characteristics were controlled by the channel simulator also indicated in the figure.

We have noted the difficulty of applying SNR-type measures to signals affected by non-waveform-type distortions as defined by Quackenbush, and consequently recognized the potential difficulty of obtaining relevant measures of CELP coder quality. To compensate for time alignment difficulties in segmental-SNR measurements of CELP coded speech,

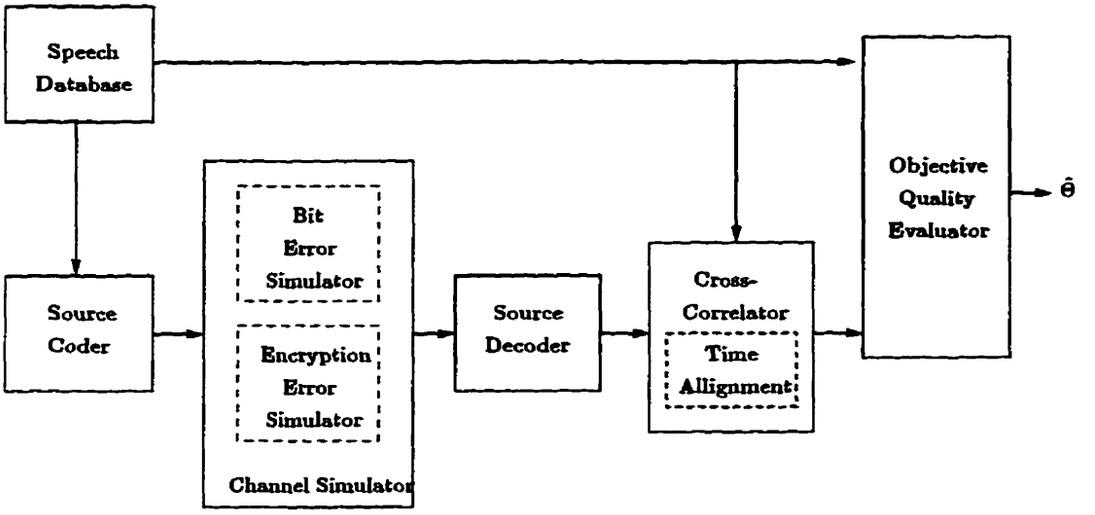


Figure 5.4: Block Diagram of Segmental-SNR Testing Methodology

the testing methodology included a provision for automatic re-alignment of the original and distorted waveforms. The testing algorithms were designed to perform periodic cross-correlation measures of the original and distorted speech waveforms and to re-align the two signals within a narrow band of permitted offsets.

The test system was monitored and compared to subjective listening results in order to verify its accuracy. The band of permitted offsets was intentionally limited to ensure a fair comparison of the speech coders, and a report of re-alignments was generated for each source file to ensure reasonable results. In general, CELP coders were found to be stable in noiseless environments, and to degrade gracefully under noisy conditions. The waveform coders were also evaluated using adaptive re-alignment, but were found to be stable under all but the noisiest channel conditions.

The objective quality evaluator was implemented as the segmental-SNR measure defined in equation (5.8). A segment length of 128 samples was used for all experimental measures. This segmental-SNR evaluator assigned a maximum SegSNR measurement of 35 dB to noiseless reproductions of the speech signal. Measurements of $(35 - \Delta)$ dB indicate a Δ dB

distortion of the original signal.

5.5.2 Quality Model Source Database

The source database for these operational quality measurements was a subset of the SWITCHBOARD database, which was defined in section 3.2 to represent the conversational English speech process. The subset selected included only those files judged by the SWITCHBOARD transcribers to exhibit maximum transmission quality, and the absence of non-speech sounds such as line noise, background noise, and echo. Examples of the transcriber records are reproduced in Appendix C. In addition, the files were reviewed to ensure a high subjective quality level. This selection procedure ensured that the uncoded source database closely resembled the pure μ-law PCM encoded English speech process defined to be our source process. The difficulty of this selection procedure was justified by the need for a nearly-uncorrupted source database with which to test the speech coders. The procedure was feasible only due to the relatively small size of the subset, and could not be repeated for the enormous database used in chapter 3 to measure the entropy rate of the speech process.

A total of 45 data files were selected, each containing speech recorded for two speakers. This database provided a total of approximately 450 minutes of conversational speech data.

5.5.3 Noiseless Rate-Distortion Results

The operational rate-distortion performance of our set of representative speech coders was determined under noiseless channel conditions. The entire quality model database was subjected to encoding, decoding, and segmental-SNR comparison according to the testing methodology outlined in figure 5.4. In this study, as in each of the following studies, quality measures were computed for each of the 45 data files independently, and then the means of these measures, and associated 95% confidence intervals were computed. The results of this study are presented in figure 5.5.

The operational rate-distortion curves of figure 5.5 are clearly defined by an exponential

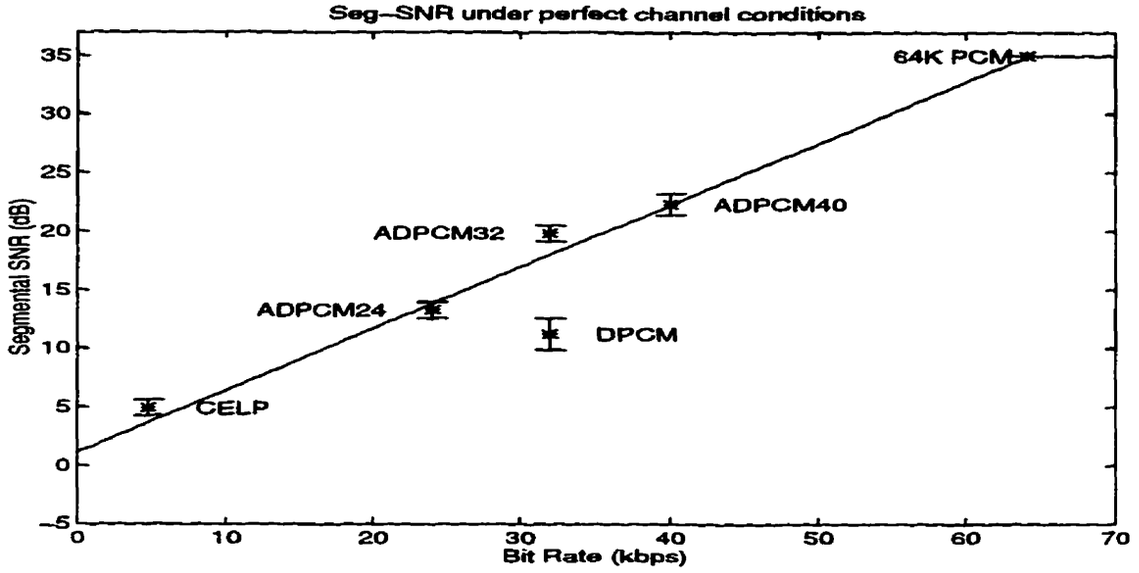


Figure 5.5: SegSNR Measures for Representative Coders

relationship between rate and distortion, which results in the linear relationship between segmental-SNR and rate observed in the figure.

A weighted linear regression analysis was performed on the experimental results, yielding a model of the form,

$$SegSNR(R) = \begin{cases} 1.142 + .528R \text{ (dB)}, & 0 < R < 64. \\ 35 \text{ (dB)}, & R \geq 64, \end{cases} \tag{5.9}$$

for rates of R kbps. This linear model is also included in the figure. An upper bound on quality of 35 dB was imposed to indicate that μ-law PCM signals can always be perfectly represented at rates greater than 64 kbps.

Referring to the theoretic quality-rate bounds of the speech process presented in section 5.3, figure 5.3, and the SNR-rate curves in Appendix H, figure H.1 we note a close correspondence to the operational results of figure 5.5.

5.5.4 Noisy Channel Condition Results

It is important to observe the behaviour of the representative coders under their expected operational conditions, and this admits the possibility of noise being introduced to the ciphertext during transmission.

Noise Types

To evaluate the effect of noise on secured speech transmissions we simulated a variety of single-bit noise conditions and applied them to the coded speech signal as indicated in figure 5.4. In anticipation of a variety of channel conditions, we first simulated bit errors with probability distributions modelled by Gaussian, Uniform, and Exponential functions. For a chosen bit error rate, the channel simulator imposed bit errors distributed with the desired characteristics.

To determine the effect of these different noise types on the coded speech quality we performed a set of quality measurements on all of the waveform coders using a small subset of the quality database. The results of this study are presented in figures 5.6, 5.7, and 5.8. Analysis of the results indicates that the probability distribution of the noise process made little difference to the objective measurements for this subset of noise types.

Given the results of this study, we selected the exponential noise distribution to generate more comprehensive results for all coders in our representative set. Presented in figure 5.9 are the means and 95% confidence intervals computed for the entire quality database of 450 minutes of coded speech subjected to exponential noise conditions.

Of particular interest in this figure are the indications that the more complex coders become preferable to simpler coders under increasingly noisy channel conditions. There are two explanations for this behaviour. First, the more sophisticated coders, particularly the CELP coder, implement better models of the speech production process and hence naturally tend to synthesize more natural speech sounds in the presence of channel errors. Second,

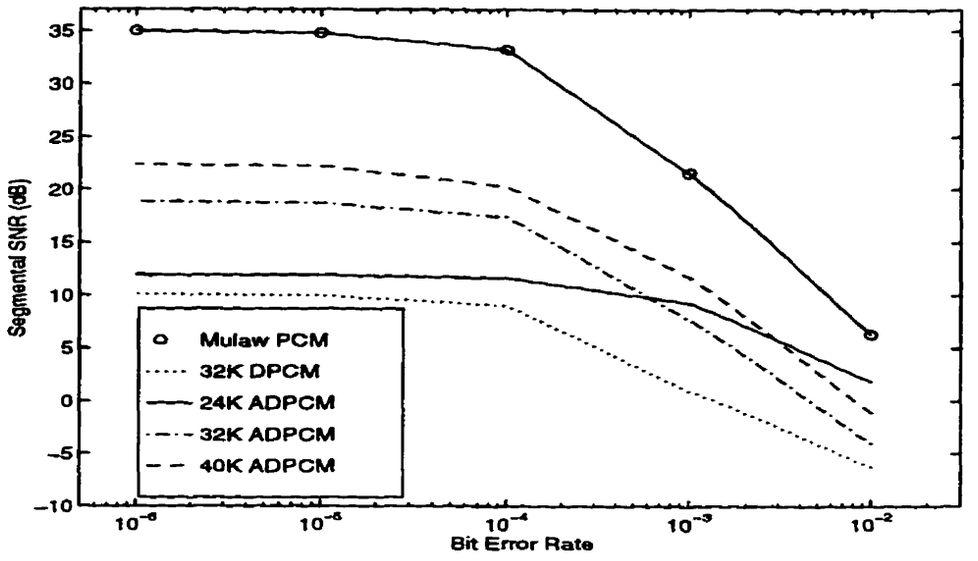


Figure 5.6: SegSNR Measures Under Gaussian Noise Conditions, σ_x^2=BER

higher levels of compression achieved by the more sophisticated coders means that a smaller number of bits will be subject to channel errors. While the impact of an error is greater when redundancy is removed from a message, bit rate reduction can compensate to some extent. In addition, all of the more complex coders have some error-recovery capability. The DPCM and ADPCM coders "forget" bad inputs over a period of time, and the CELP coder includes some forward error correcting ability.

The impact of burst errors was also measured and found to have a greater effect on objective speech quality, as might be expected. Rather than present the results of these studies here, we note that a burst error may be viewed differently in the context of a secure communication session. This leads us to the development of two models for errors under cryptographic communication requirements.

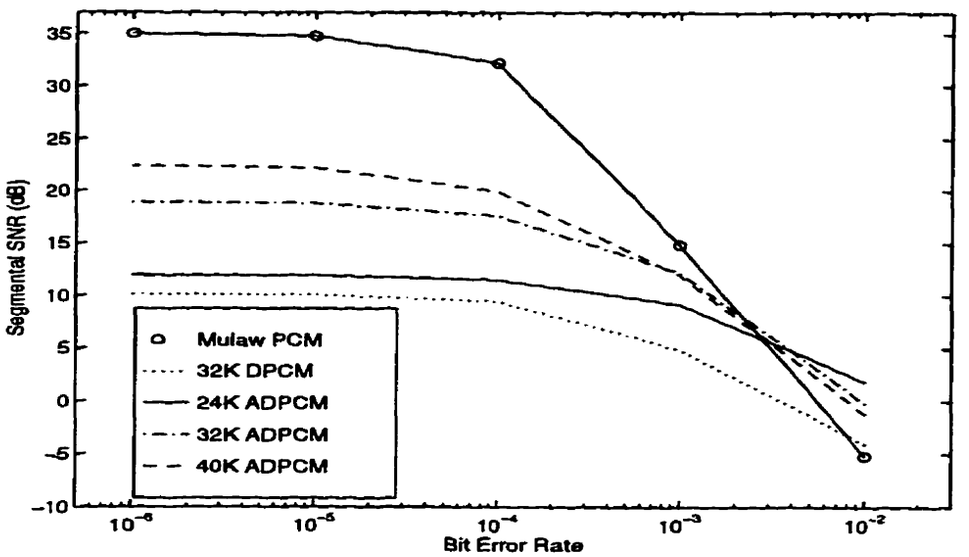


Figure 5.7: SegSNR Measures Under Uniform Noise Conditions

5.5.5 Cryptographic Impacts on Noisy Channels

Under a cryptographically secured communication session, channel errors can have a significant impact on the decrypted speech signal. The type of encryption algorithm will determine the extent of this impact. We will assume that cryptographic synchronization is maintained, and that we do not experience the effects of deliberate interference caused by an active attacker. If this is not the case, measurement of quality during the unsynchronized segment is trivial: well designed encryption algorithms will produce a signal entirely uncorrelated with the intended signal and the quality will be minimal. If synchronization is maintained, there are three possible modes of behaviour in response to a channel bit error:

1. If the encryptor is a synchronous stream cipher, a channel bit error will result in a single error in the coded speech signal.

2. If the encryptor is a self-synchronizing stream cipher, a channel bit error will propagate to cause n subsequent and sequential bit errors in the coded speech signal, where n is the number of states in the cipher.

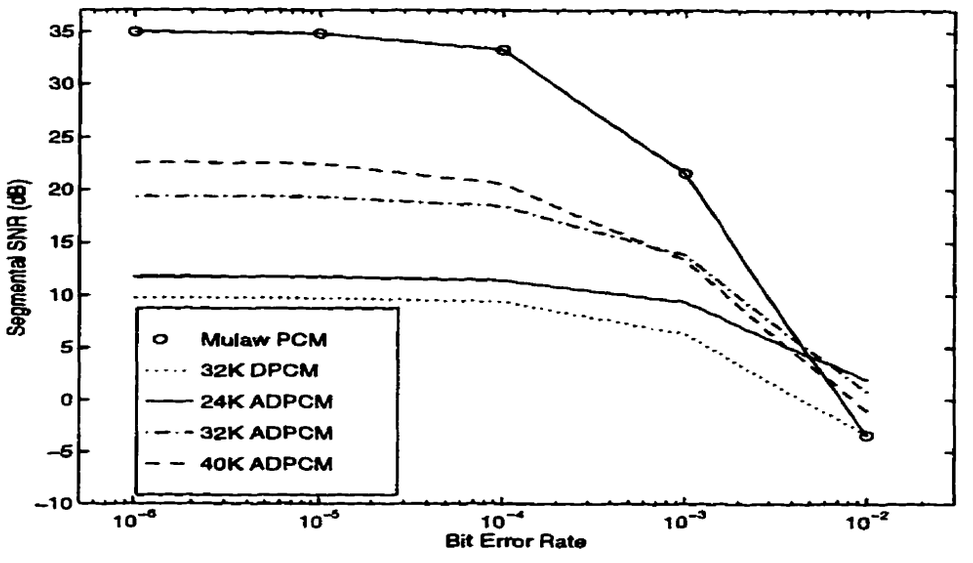


Figure 5.8: SegSNR Measures Under Exponential Noise Conditions

3. If the encryptor is a block cipher, then a channel bit error will result in a spurious decryption of the entire cipher block of n characters. This error will effect all coded speech signal bits falling within the decryption block length, even those transmitted before the occurrence of the channel error.

The cause of these effects is discussed in detail in section 4.1. In order to study the effect of these cryptographic characteristics on the quality of coded speech transmission we identify two models of cryptographic impact. The first model applies to synchronous stream ciphers, where channel errors correspond directly to errors in the coded speech signal. These cryptographic impacts can therefore be modelled by the bit error generators used in section 5.5.4. The second model can be applied to both self-synchronizing stream ciphers and block ciphers of a given complexity or block size, n. In both cases a single bit error results in an error in n coded speech signal bits. Assuming the noise source to be a stationary process, the forward error propagation effect of the self-synchronizing cipher and the block error effect of the block cipher will have the same average impact on quality. In the second model

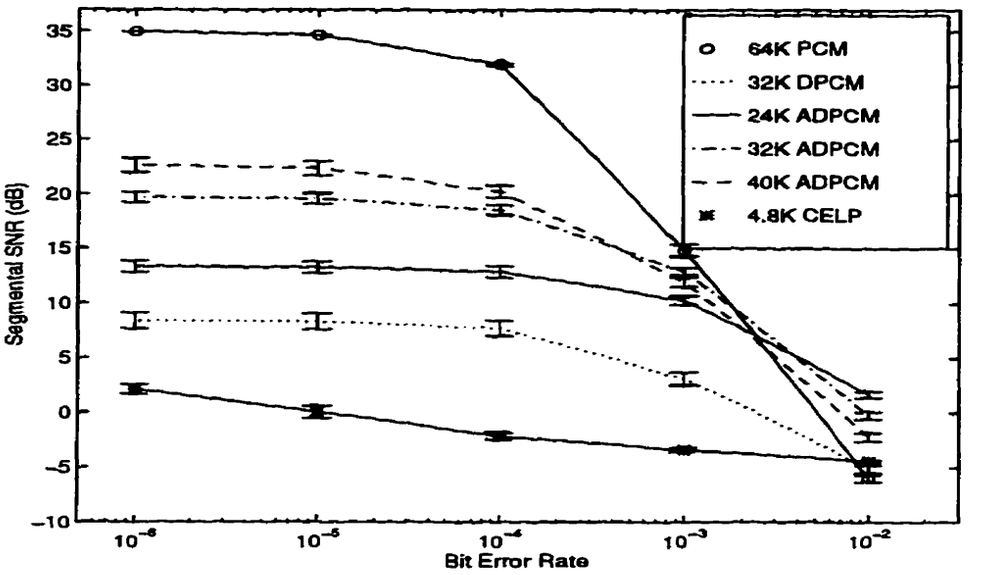


Figure 5.9: SegSNR Measures and 95% Confidence Intervals Under Exponential Noise Conditions

we observe that a channel error will correspond to a fixed-duration burst of noise on the channel.

The models used to describe the effect of error on cryptographic communication systems are summarized in table 5.3.

Encryption Type	Error Model
Synchronous stream ciphers	Bit error generator
Self-synchronous stream ciphers (n states)	n-bit block error generator
Block cipher (n bits)	n-bit block error generator

Table 5.3: Cryptographic Error Simulation Models

The first encryption error model can be adequately represented by the results of figure 5.9, while the second encryption error model may be computed using simulated block errors of a fixed duration. A block encryption error simulator was developed using an exponentially distributed block error model and a chosen constant block error length. This error simulator was applied to the waveform coders at a variety of block error rates, for a fixed block error

size of 64 bits. This would be a typical block size for encryption coders such as DES. The results of this study are presented in figure 5.10.

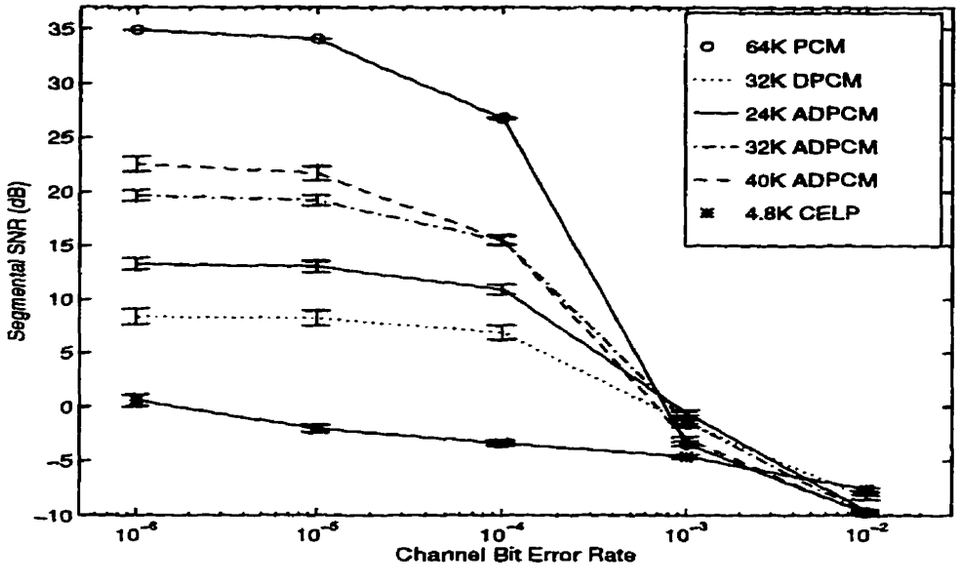


Figure 5.10: SegSNR Measures for Encryption Under 64 Bit Block Error Conditions

These results indicate an inversion of the objective quality evaluations for our waveform coders beyond a given channel error rate. We note that reduction in quality is more rapid than that experienced under single channel bit error conditions. This is obviously due to the much greater signal disruption caused by block errors. It is quite interesting to note, however, that below the threshold of a 10^{-4} bit error rate, coder performance is mostly similar to the results of figure 5.9.

5.6 Bit Rate as a Function of Redundancy

We will conclude this chapter by presenting a model of source coded bit rate as a function of residual signal redundancy.

Definition 7 (Residual signal redundancy) *We define the residual signal redundancy as the difference between the minimum rate required to transmit the coded speech signal and the actual bit rate achieved by the source coder.*

This definition corresponds exactly to the per-letter redundancy measurement used for the unicity distance calculation in section 4.4. The entropy rate, H, of the speech signal represents the limiting bit rate for lossless compression of the speech signal. If we represent the entropy rate of the conversational speech process by H bits/sample, and the actual source coded bit rate by R bits/sample, then the minimum per-sample redundancy that may be achieved by a source coder operating at rate R is,

$$D_{\min} = R - H \quad \geq 0. \tag{5.10}$$

In our experimental work we will find it more practical to scale these measures by the μ-law PCM sampling rate of 8000 samples/second in order to present models of bit rate and redundancy in units of bits/second.

The rate-redundancy relationship of equation (5.10) represents the minimum redundancy that may be obtained by a noiseless source coder. If the source coder is not noiseless, then some of the source process information is destroyed by the coding operation and the rate of real source information transmission is some $H' \leq H$.

An upper bound on the redundancy of a coded speech transmission is defined by the rate of that transmission. In cases where the speech signal information has been completely destroyed by the source coding operation, then all of the coded signal may be considered to be redundant as it does not convey the desired information. The maximum redundancy at a given transmission rate, R, is simply,

$$D_{\max} = R \quad \geq 0. \tag{5.11}$$

These simple linear bounds define a region for the rate-redundancy performance of practical speech coders that is identified by the shaded portion of figure 5.11. We can improve the accuracy of this rate-redundancy model for speech coders by quantifying the extent to which the coding operations destroy speech process information.

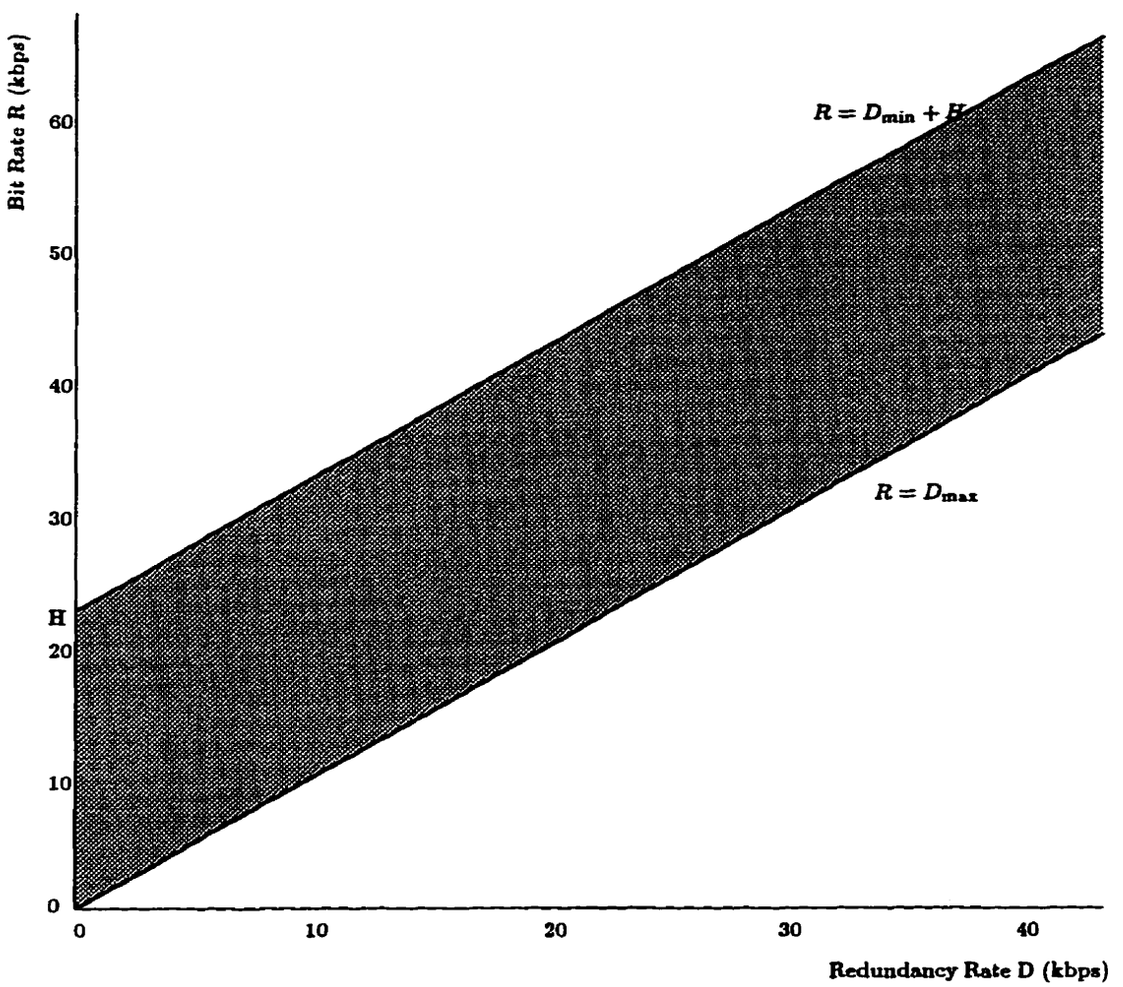


Figure 5.11: Linear Bounds on Rate-Redundancy Performance of Speech Coders

5.6.1 Divergence Measures for Coded Speech

The rate of true information content in the coded speech signal was described above as $H' \leq H$. The actual rate of desired speech information in the coded speech signal will be lower than that of the original in any coder allowing distortion of the signal. When information is destroyed in coding the signal for transmission it cannot be recovered upon decoding. We would perform entropy rate measurements of the coded speech process to obtain the desired redundancy results, but it is not possible to distinguish between true speech information and coding distortions in the recovered speech signal. An entropy rate measurement of distorted speech could be entirely uncorrelated with the original entropy rate as a result of additive noise.

To determine the amount of distortion added to the speech signal by the coding operation we can compute the Kullback-Leibler distance measure [76]. If we have two source processes, $\mathbf{P}$, and $\mathbf{Q}$, with probability mass functions $p(x)$, and $q(x)$ respectively, then the Kullback-Leibler distance measure between $\mathbf{P}$, and $\mathbf{Q}$ is defined by

$$D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)} \text{ (bits)}, \tag{5.12}$$

where we define $0 \log \frac{0}{q} = 0$, and $p \log \frac{p}{0} = \infty$.

The Kullback-Leibler distance, which is also called the relative entropy, is not a true metric in that it is not symmetric, but it does provide a useful measure of the distance between two distributions. Consider a coding operation where we wish to transmit the source process $\mathbf{P}$, but use a coding operation designed for process $\mathbf{Q}$: it can be shown that to reconstruct $\mathbf{P}$ exactly, we must send additional information at a rate of $D(p||q)$ [77].

More accurate divergence measures for processes with memory can be obtained by extending the Kullback-Leibler distance measures to higher model orders. For kth order probability mass functions, $p(x_1^k)$ and $q(x_1^k)$, we compute the generalized per-symbol divergence

measure,

$$D^k(p||q) = \frac{1}{k} \sum_{x_1^k \in \mathcal{X}^k} p(x_1^k) \log_2 \frac{p(x_1^k)}{q(x_1^k)} \text{ (bits)}. \tag{5.13}$$

By applying the Kullback-Leibler distance measure to the probability distributions of the original speech process and a distorted version of that process we can determine the amount of speech information destroyed by the coding process. We can therefore use this measure to estimate the extent of the redundancy introduced by each of our representative speech coders. The lower bound of equation (5.10) indicates the minimum level of redundancy present in any coded speech signal. The Kullback-Leibler distance measure indicates the absolute divergence of the transmitted signal from the real signal. Hence, if the transmitted signal diverges by $D^k(p||q)$, then the decoded message contains only,

$$H - D^k(p||q) = H', \tag{5.14}$$

bits of information.

5.6.2 Relevance of the Divergence Measures

The Kullback-Leibler divergence measure will not yield relevant distortion measures for all types of coding distortion. To illustrate this problem consider the case of a coder which simply introduces a time-varying delay to a stationary input process. The input and output of such a coder would have identical statistics and so achieve a zero distortion measure, despite the presence of perceptually relevant and extreme distortions.

The purpose of all practical speech coders is to reproduce a reasonable facsimile of the original process, so the Kullback-Leibler measure is appropriate for our model. However, it is clearly necessary to confirm the relevance of the Kullback-Leibler distortion measures if they are to be applied in this manner to new source processes or source coders.

5.6.3 Divergence Measures for Representative Speech Coders

A series of experiments were performed to determine the average distortion introduced to the speech signal by the coding operations of our representative coders. The testing methodology used was similar to the methodology used to compute segmental-SNR measures. A database of speech process files was subjected to encoding and decoding by means of each of the representative source coders. First, second, and third order probability mass functions of original and distorted signals were computed by maximum likelihood estimation, and then Kullback-Leibler distance measures were computed according to equations (5.12) and (5.13). The testing methodology for information divergence measures is summarized by the block diagram in figure 5.12.

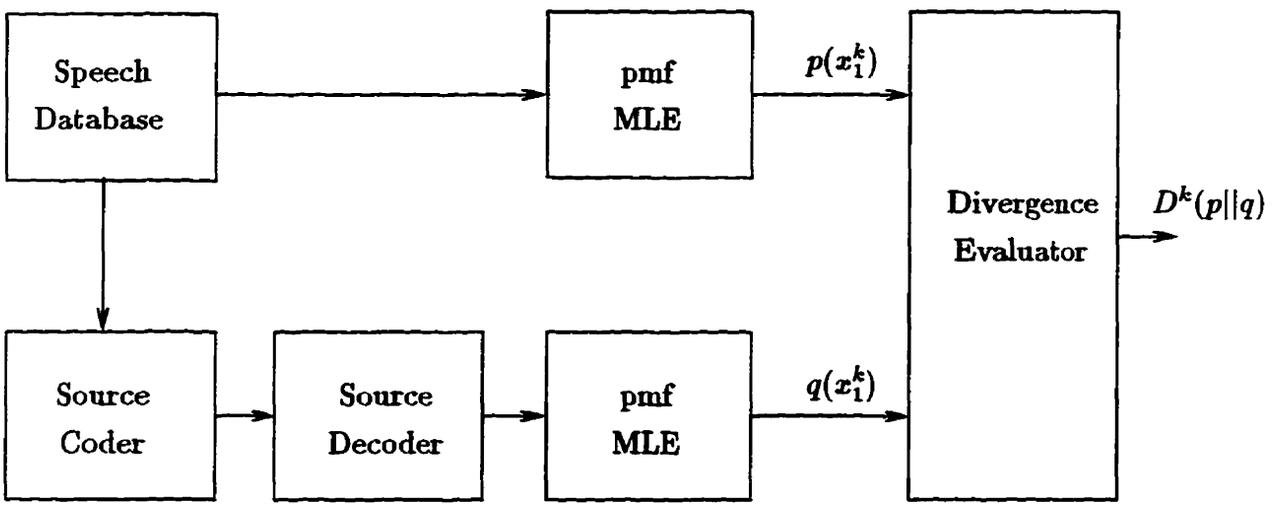


Figure 5.12: Block Diagram of Divergence Testing Methodology

Divergence Model Source Database

A source database of 127 conversations, totaling 1269 minutes of speech data, was compiled from the SWITCHBOARD database. This database contained the entire SWITCHBOARD subset of 45 files used to compute segmental SNR measurements in section 5.5 and an

additional 82 conversations. The additional conversations were selected on the basis of the subjective evaluations of the SWITCHBOARD transcribers, but were not independently evaluated for high subjective quality. The additional conversations admitted to the database were those labeled as exhibiting maximum transmission quality and the absence of non-speech sounds such as line noise, background noise and echo. The selection procedure was designed to ensure that the maximum likelihood estimations of probability distributions computed for the divergence measurements closely resembled those of the speech process.

First Order Divergence Results

The entire source database of 127 files was distorted by each of the representative source coders as indicated in figure 5.12 to produce a large set of divergence measures. The means and 95% confidence intervals were computed for each of the coders in our representative set. The results, presented in figure 5.13, indicate the divergence of the distorted speech from the original μ-law PCM speech in bits/sample.

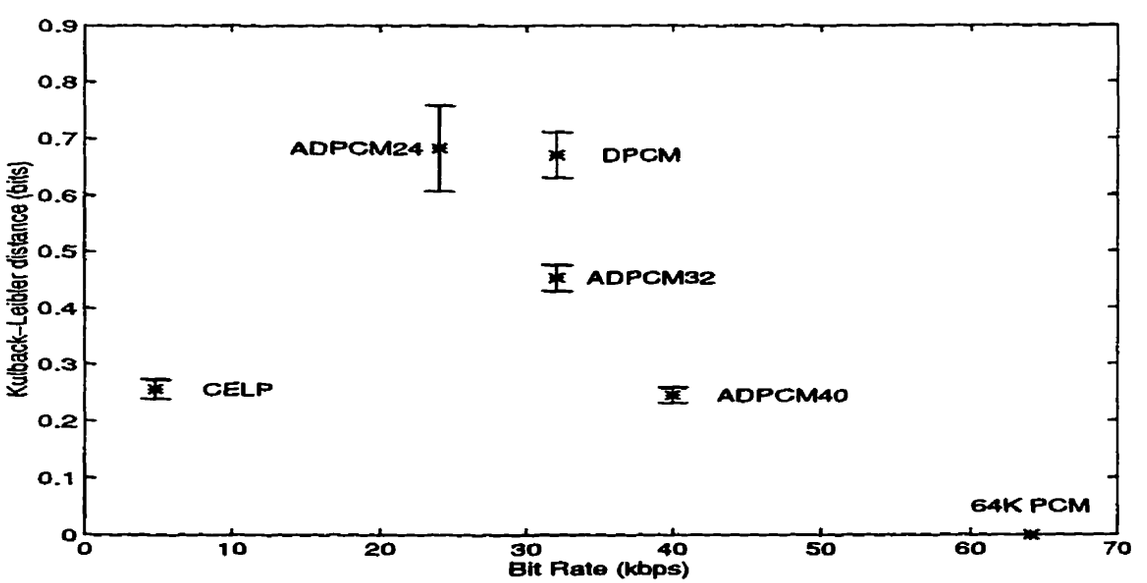


Figure 5.13: First Order Divergence Means and 95% Confidence Intervals

The results plotted in figure 5.13 indicate the relative first-order divergence of the representative coders. The low divergence of the CELP coder is surprising, considering its relatively low objective quality as measured by the segmental SNR study presented in section 5.5. Low first-order divergence is a result of the CELP coder's vocoding algorithm, which simulates the vocal excitation process of speech. The output of CELP-synthesised speech will therefore approximate the distribution of speech in low-order measures, even when the higher order characteristics contain significant distortions as a result of the CELP coding process.

As we have discussed in detail in chapters 2 and 3, the speech process should be described by a model of much higher dimensionality. Hence, while the results obtained from a first-order approximation grant a useful insight to the divergence properties of our various coders, we should apply a higher order study to obtain definitive results.

Higher Order Divergence Results

Additional Kullback-Leibler distance measures were computed for models of order 2 and 3, according to equation (5.13). These measures were computed for all representative coders using the original 45 source conversations. The entire divergence database was not studied mainly due to the high computational cost. The results of the first, second, and third order divergence measures and their 95% confidence intervals are presented in figure 5.14. To improve the clarity of the figure, data points at a given bit rate were separated slightly according to model order, and labelled by source coder type. The real data consisted of points measured at exactly 4.8, 24, 32, 40, and 64 kbps.

These results indicate a generally increasing divergence measure with bit rate reduction for the waveform coders. The exception to this rule, the CELP coder, continues to demonstrate low divergence for low order modelling. As discussed above, the CELP coder synthesizes speech with a model of the vocal tract and will generally approximate speech at low model orders. The segmental-SNR measurements presented in section 5.5 clearly

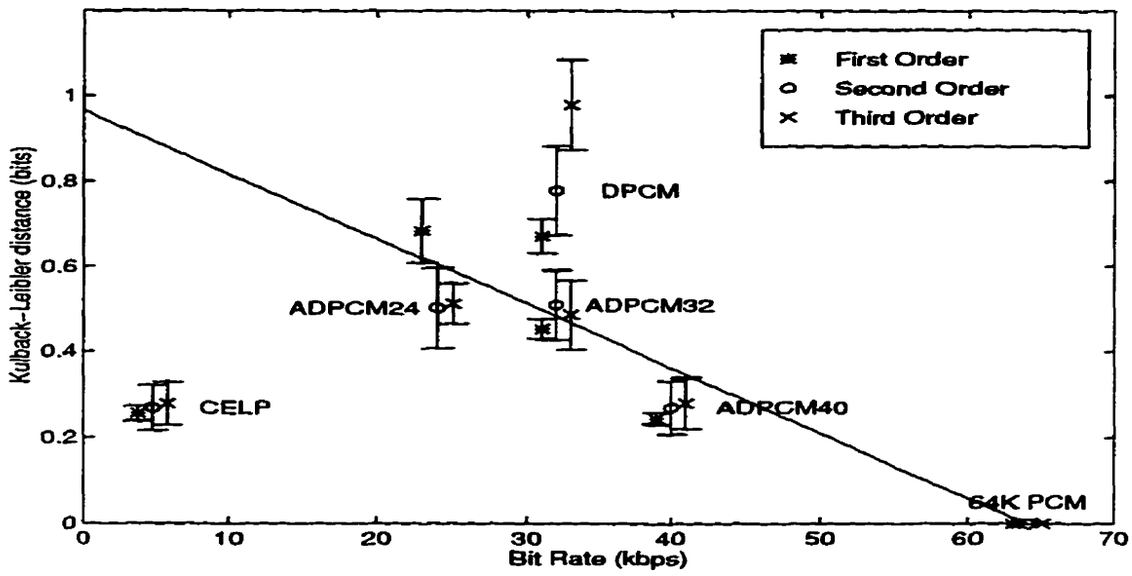


Figure 5.14: First to Third Order Divergence Means and 95% Confidence Intervals

indicate a large objective distortion in CELP speech, but this distortion is not evident in the first to third order divergence measures presented here.

Additionally, the complexity of the coding algorithm has a significant impact on divergence, as indicated by the difference between 32 kbps ADPCM and 32 kbps DPCM performance.

It would be desirable to obtain fourth, fifth, and sixth order divergence measures in order to verify the lower-order results computed here and more accurately measure the divergence of the CELP coder, but as we noted in chapter 3, high order speech modelling is a difficult task. These higher order measures were not computed here due to computational and time constraints, however we would like to point out that the entropy estimation technique presented in chapter 2 could be modified to allow a low complexity estimation of high order divergence measures. The estimation procedure could be modified by partitioning the generalized Kullback-Leibler distance measure into independent partial divergence

measures,

$$d^{(n,j)}(p||q) \triangleq \sum_{x_{j+1}^n \in \mathcal{X}^{n-j}} p(x_1^n) \log_2 \frac{p(x_1^n)}{q(x_1^n)} \quad \text{for } n > j \geq 1. \tag{5.15}$$

Selecting the vectors $x_{j+1}^n \in \mathcal{X}^{n-j}$ by a Monte Carlo procedure, we can compile a set of partial divergence measures with which to interpolate the complete nth order divergence surface. The resulting divergence surface could then be analysed and interpolated as appropriate to obtain reasonable nth order divergence estimates,

$$\tilde{D}^n(p||q) = \sum_{x_1^j \in \mathcal{X}^j} \tilde{d}^{(n,j)}(p||q). \tag{5.16}$$

5.6.4 A Rate-Redundancy Model from the Divergence Data

The first, second, and third order divergence measures do capture a sufficient amount of information to yield a relevant measure of the rate-redundancy performance of typical speech coders. We applied the experimental data of all three Kullback-Leibler model orders to estimate a model of divergence as a function of coder rate.

As we have discussed earlier, the low order divergence measures of the CELP coder are misleading and the true CELP coder divergence is expected to be much larger than indicated by our experimental measures. For this reason the CELP coder measures were omitted when we applied a weighted linear regression analysis to the data in figure 5.14.

The divergence measures of first, second, and third model orders for the waveform coders were fitted by a regression analysis that weighted each data point by the inverse of its confidence interval. The inclusion of data from all three model orders was beneficial to modelling the divergence-rate function as each divergence measure contained information unique to its model order.

The regressed model of divergence plotted in figure 5.14 is defined by,

$$D^*_{(p||q)}(R) = \begin{cases} 0.967 - 0.015R \text{ (bits)}, & 0 < R < 64. \\ 0 \text{ (bits)}, & R \geq 64, \end{cases} \tag{5.17}$$

The linear model seems to satisfactorily describe the behaviour of the representative waveform coders. We can further justify the use of a linear model by returning to the experimental segmental-SNR results presented in section 5.5.

We note that the segmental-SNR defined by equation (5.8) is a measure of distortion that is based upon the time-series of the original and distorted signals. The segmental SNR is therefore sensitive to correlation effects in the signal that could only be captured by higher order probability models. The measurements obtained by the segmental-SNR are not directly comparable to the Kullback-Leibler distance measure, but they do indicate distortions that would be captured by a higher-order divergence measure.

The segmental-SNR results for our representative speech coders in figure 5.5 were modelled by the linear function of equation (5.9). This linear model matched very well with the experimental data, and furthermore was supported by a pair of linear theoretic quality-rate bounds in section 5.3 and a set of linear analytic models of SNR performance as indicated in Appendix H. It is reasonable, given these prior observations, to propose the linear model of divergence described by equation 5.17

The predicted model of divergence indicates the relative distance between the original μ-law PCM signal and the decoded signal, which is also in μ-law PCM format. This divergence measure therefore compares two signals that would be transmitted at a rate of 64 kbps. In order to estimate the effective divergence rate of the coded signal as it is transmitted on the communications channel we scale the divergence estimates of equation (5.17) by the coder rate. For source coders using r bits/sample for a rate of R kbps we have the scaling

relationship,

$$\begin{aligned} D^*_{(p||q)\text{eff}}(R) &= D^*_{(p||q)}(R) \cdot \frac{r \text{ coded bits}}{8\ \mu\text{-law PCM bits}} \cdot \frac{8000 \text{ samples}}{\text{second}} \\ &= D^*_{(p||q)}(R) \cdot \frac{R}{64} \quad \text{(kbps)} \end{aligned} \tag{5.18}$$

Scaling yields an effective divergence rate model for speech coders described by,

$$D^*_{(p||q)\text{eff}}(R) = \begin{cases} 0.01525R - 0.000234R^2 \text{ (bits)}, & 0 < R < 64. \\ 0 \text{ (bits)}, & R \geq 64, \end{cases} \tag{5.19}$$

This function expresses the effective rate of divergence inherent in the coded bit stream produced by a typical speech coder. Its principal purpose is to describe the rate at which speech information is being destroyed by the coding operation. The effective divergence rate function, presented in figure 5.15, yields some interesting observations. While the absolute divergence measure of equation (5.17) increases linearly with decreasing bit rate, the fraction of the total coded bit stream consumed by that distortion follows the more complex curve of equation (5.19). It is apparent that the most sophisticated coders sacrifice a relatively small amount of scarce bandwidth to distorted speech components. The medium complexity waveform coders contain a relatively high proportion of distorted signal, likely as a result of their lower complexity, but achieve a lower absolute distortion. Finally, coders operating at high bit rates need introduce little absolute distortion, and so achieve a relatively low proportion of effective distortion.

We can combine the result of equation (5.19) with the theoretical rate-redundancy function of equation (5.10) to propose a model of rate-redundancy performance for typical speech coders. For ideal distortionless coders, equation (5.10) describes the relationship between bit rate and redundancy: above the entropy rate H, any additional signaling, $R - H$, is redundant, while below the entropy rate we have not enough bandwidth to communicate

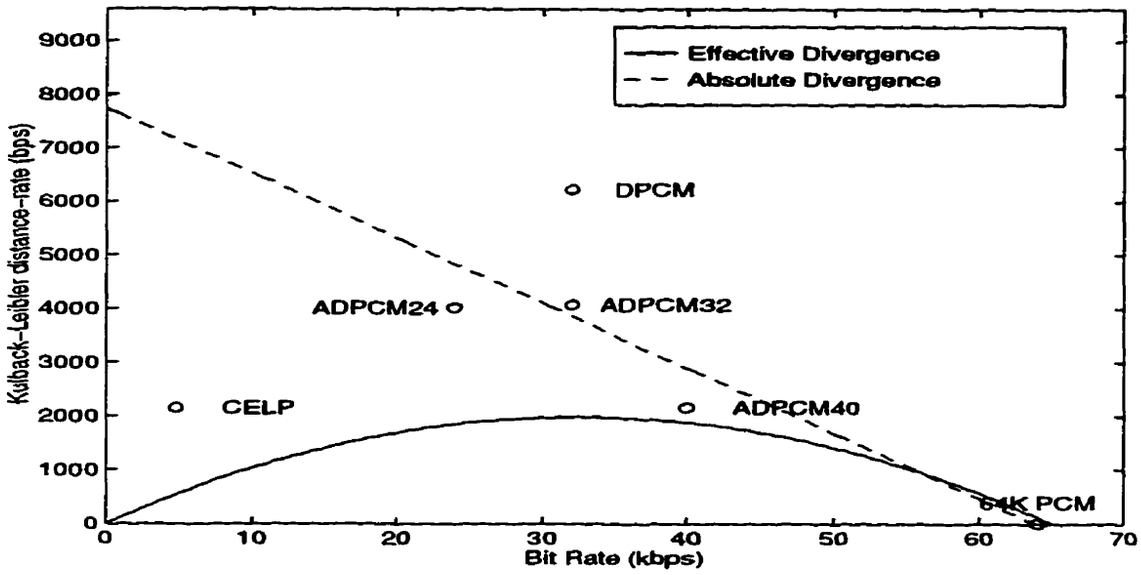


Figure 5.15: Effective and Absolute Divergence Rate Models

the entire information stream so all bits $R < H$ are essential. Non-ideal coders introduce distortion to the signal that destroys speech information, as indicated by equation (5.14). To express the rate of divergence information contained in the coded speech signal we developed the model of effective divergence expressed in equation (5.19).

Definition 8 (Operational Rate-Redundancy Function) *Combining equations (5.10), (5.14) and (5.19) yields a measure of the redundancy of the typical coded speech signal. We express the general operational rate-redundancy function for speech coders as follows,*

$$D_{operational}(R) = \begin{cases} R - H + D^*_{(p||q)eff}(R) & R \geq H, \\ D^*_{(p||q)eff}(R) & R < H. \end{cases} \tag{5.20}$$

Presented in figure 5.16 is the proposed operational rate-redundancy function of equation (5.20), and the theoretical upper and lower bounds for rate-redundancy defined by equations (5.10) and (5.11).

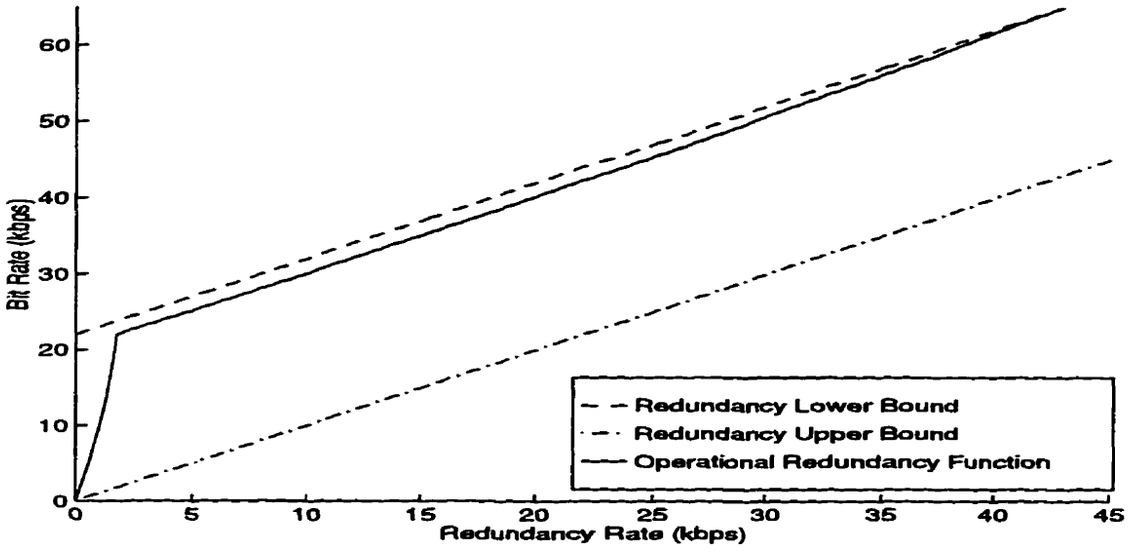


Figure 5.16: Operational Rate-Redundancy Function and Theoretic Bounds

In table 5.4 we summarize the third order Kullback-Leibler measures computed for the speech coders and used to define the linear divergence model of equation (5.17). The table includes the set of operational redundancy measures predicted by equation (5.20) using the linear divergence model. Also presented in the table are the actual redundancy measures, $D_{\text{actual}}(R)$, that would be obtained using the actual third order divergence measures, $D^3(p||q)$, rather that the approximate linear model of equation (5.17). This table confirms the close correspondence between the actual and modelled results at all points except the outlying DPCM and CELP measures. As we have noted earlier, the CELP divergence measures were not found to be indicative of the true higher order divergence characteristics. The high DPCM divergence is expected for a non-adaptive waveform coder.

The shape of the operational rate-redundancy function is determined by the effective divergence rate model computed from our experimental divergence measures for the representative coders, and by the theoretical lower bound for redundancy. The curve plotted in figure 5.16 displays a distinct discontinuity in the slope of the function at the entropy

Coders	Rate (kbps)	$D^3(p\|\|q)$	$D_{\text{operational}}(R)$ (kbps)	$D_{\text{actual}}(R)$ (kbps)
μ-law PCM	64	0.000	42.06	41.68
DPCM	32	0.979	11.95	13.62
24K ADPCM	24	0.512	3.82	3.24
32K ADPCM	32	0.486	11.95	11.64
40K ADPCM	40	0.280	19.84	19.10
CELP	4.8	0.250	0.537	0.150

Table 5.4: Actual and Modelled Operational Redundancy Measures

rate of the speech process. This is due to the discontinuity in the theoretical redundancy lower bound at the entropy rate. It may be reasonable to relax the requirement for a sharp transition between sub-entropy and super-entropy coding characteristics. If we allow for a continuous curve to describe the rate-redundancy lower bound near the entropy rate, then the operational redundancy function of figure 5.16 could be continuous. Such an adjustment to the rate-redundancy lower bound can not be supported by a theoretical development, and would rely on assumptions about the performance of near-optimal coders at the entropy rate. For this reason we prefer to use the lower bound predicted by theory.

We have noted how the Kullback-Leibler distance measures used to develop the operational rate-redundancy function could not be computed to as high a model order as might be desired. As a result, significant differences in the original and distorted waveforms may not have been detected by our measures, and the divergence rate measures of figure 5.15 may be quite conservative. We examined the effect of higher divergence measures by computing operational rate-redundancy functions for a set of divergence functions characterized by Kullback-Leibler distance function y-intercepts of 16, 24, 40, and 64 kbps. This set of potential operational rate-redundancy functions is presented in figure 5.17.

Figure 5.17 serves to validate the form of the operational rate-redundancy function of equation (5.20). The proposed model of rate-redundancy performance for typical speech coders is evidently sensitive to increases in divergence between coders, and remains within the theoretical rate-redundancy bounds under all conditions. The model indicates that

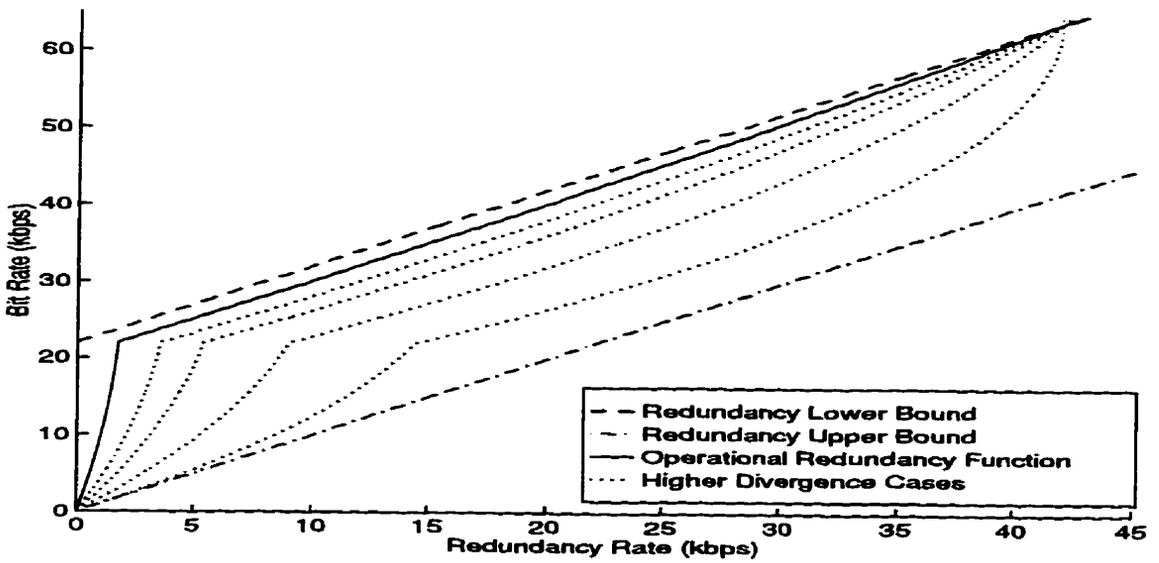


Figure 5.17: Rate-Redundancy Functions for Higher Divergence Models

coders operating at a higher bit rate may more easily approach the lower bound on redundancy, while low bit bit rate coders may tend more towards the upper bound. It is interesting to note how the discontinuity in the rate-redundancy function is much less pronounced for models assuming a higher divergence function.

Chapter 6

Experimental Measures for Security Models

The true use of speech is not so much to express our wants as to conceal them.

Oliver Goldsmith. Oct. 20, 1759.

6.1 Introduction

In this chapter we will develop a set of experimental entropy measurements with which to evaluate the 3 theoretic security indices defined in section 4.7.

These indices require measures of the message entropy, key entropy, ciphertext entropy, and a measure of message redundancy. Most of these measurements have been derived, computed, and presented in prior chapters. In chapter 3 we computed an estimate of the speech process entropy using the techniques for entropy estimation developed in chapter 2. Our definition of the speech process has remained constant throughout this research so the computed entropy estimate of 2.79 bits/sample constitutes our measure for the message entropy in a secure speech communication system. This entropy measure is independent of

any source coding or encryption operations subsequently applied to the speech process.

As we discussed in chapter 4, the entropy of the key is under the control of the cryptographer. A maximum entropy of $H(\mathbf{K}) = K \log L_K$ can be obtained by selecting keys with uniform probability over the entire keyspace. We will assume that the key entropy is always K bits when keys are K bits long. Measures of key entropy are therefore dependent only on the choice of encryption algorithm and are equivalent to the key length.

In chapter 5 we developed a relationship between the redundancy of a coded message and the coder bit rate for a class of representative source coding algorithms. We will refer to the rate-redundancy model for speech coders developed in section 5.6 and summarized by the operational rate-redundancy function of equation (5.20) for measures of redundancy.

The remaining measure required for a security model is the ciphertext entropy, $H(\mathbf{C})$. The ciphertext entropy is dependent on both the coded speech statistics and the encrypting properties of the chosen cipher, so it must be computed for the ciphertext processes resulting from each combination of unique source coding and encryption algorithms. In section 6.2 we will discuss the broad set of encryption algorithms that may be used for enciphering speech data and select a subset of generally representative encryption methods. This subset of representative encryption algorithms is small enough to allow experimental measures of ciphertext entropy to be computed for speech encoded by our set of representative speech coders.

The testing methodology and the database used for measuring the entropy of source coded and encrypted speech will be defined in section 6.3. In this section we will also discuss how the techniques derived to compute conditional entropies of the speech process in chapter 2 were modified to suit the unique characteristics of source coded and encrypted speech.

In section 6.4 we will present the experimental measures of conditional entropy that will comprise the data for our security models. This section will include a detailed study of the convergence properties of direct measures as a function of the size of the speech database.

Characteristics relevant to the computation of entropy for encrypted speech were found to differ significantly from those of unencrypted speech. As a result of a very large state-space and slow convergence characteristics, we found it significantly more difficult to compute fourth order entropy for encrypted processes than for unencrypted speech. We will present the results of our attempts to extend our conditional entropy measures for source coded and encrypted speech in section 6.5.

Finally, in section 6.6 we will combine the experimental ciphertext entropy results determined in this chapter with the message entropy computed in chapter 3 and the rate-redundancy model developed in chapter 5 to evaluate the security indices proposed for the secure speech system.

6.2 Representative Encryption Algorithms

We must define a set of encryption algorithms that represent the general performance of all encryption techniques and yet is small enough to allow experimental measures of ciphertext entropy to be computed for each representative speech coding algorithm.

There are many possible encryption algorithms available to serve as examples of the main characteristics of encrypting transformations. Algorithms based on a block cipher design include the following examples,

- **Data Encryption Standard** [78] [79]. This 64 bit block cipher has been a worldwide standard for 20 years. Developed by IBM for the U.S. National Bureau of Standards, this cipher has successfully resisted concerted efforts at cryptanalysis.

- **Lucifer** [80], [81]. Lucifer was a precursor of DES, also designed at IBM. Although Lucifer uses a 128 bit block design, it was shown to be weaker than DES when subjected to differential cryptanalysis by Biham and Shamir [82].

- **IDEA** [83]. This 128 bit key block cipher is considerably stronger than DES and has

shown itself to be resistant to known forms of cryptanalysis. Lai has suggested it may be immune to differential cryptanalysis [84].

- **NewDES** [85]. Despite the name, this algorithm is not directly related to DES. It uses a 120 bit key on 64 bit message blocks.

- **FEAL** [86]. This DES-like algorithm was intended to achieve security through a stronger round function and fewer total rounds. The original and variations on the algorithm have all been vulnerable to differential cryptanalysis [87].

- **REDOC II** [88]. This 120 bit key, 80 bit block cipher has been resistant to differential cryptanalysis [82].

- **Skipjack** [89]. This 80 bit key, 64 bit block size algorithm has been classified as Secret by the U.S. Government. It was intended for use in the Clipper key-escrow system. Due to the Secret classification little is known about the algorithm, but it is purportedly a strong iterative block cipher [90].

Algorithms based on a stream cipher methodology include these examples,

- **Vernam Cipher** [36]. The classical one-time-pad system can attain perfect secrecy when implemented with a running key of length equal to the message length. There is no passive cryptanalytic attack that can compromise a properly implemented Vernam cipher system [51], [58].

- **A5** [91]. This is the stream cipher used to encrypt Group Special Mobile (GSM) traffic. The algorithm is based on 3 linear feedback shift registers (LFSR's) and is vulnerable to an exhaustive attack of 2^{40} encryptions [90].

- **RC4** . This variable key size stream cipher was developed by Rivest for RSA Data Security Inc. The encryption algorithm is proprietary, but a compatible algorithm was published on the Internet [90].

It would be infeasible to perform experimental entropy measures on the ciphertext produced by all of these algorithms for each of our representative speech coders, so we have selected the DES algorithm as an elemental encryption engine for finite key systems and the Vernam stream cipher as an example of the behaviour of a perfect secrecy system. As discussed in section 4.1.2, asymmetric cryptosystems are not generally appropriate for real-time speech encryption and will not be included in our representative set.

6.2.1 The DES Encryption System

The DES encryption engine will be applied in four different modes to provide encryption operations with a variety of interesting characteristics. The four standard modes of DES were defined in [92] to provide a range of encryption operations. In Appendix L we provide a description of the electronic codebook (ECB), cipher feedback (CFB), cipher block chaining (CBC), and output feedback (OFB) modes of DES chosen as representative cryptosystems.

The use of a single encryption engine for our experiments allows a fair comparison of the effect that different types of cipher design have on the entropy of the encrypted stream. This fairness is particularly evident as all ciphers use the same key space and have virtually equivalent complexity. The Data Encryption Standard design has experienced perhaps the closest scrutiny of any conventional private key algorithm and has shown itself to be resistant to known forms of cryptanalysis. The DES cipher satisfies the fundamental design criteria for ciphers suggested in section 4.3, Proposition 2, and so is an excellent candidate for study by our information-theoretic security measures.

6.2.2 The Vernam Cipher System

The Vernam cipher was described in chapter 4 as a theoretically perfect stream cipher. The ciphertext is created by this system by combining each message bit with a running key bit from an unpredictable source. The output of a Vernam cipher was shown in section 4.5.1 to be statistically random if the key stream is also random. This cipher serves as an upper

bound on achievable security and so is an interesting cipher for inclusion in our security models. Stream ciphers do not introduce the characteristics of diffusion or use the mixing transformations outlined in Proposition 2, but satisfy all other requirements. The Vernam cipher achieves perfect secrecy by adding a sufficient amount of confusion to each message bit to completely obscure the message information.

6.2.3 Encryption System Implementations

The four DES encryption algorithms were implemented in software. It was not necessary to obtain real-time performance and we were not concerned with the potential security risks of performing encryption on a general use multi-user computer system. In the CFB and OFB implementations a block size of $k = 64$ bits was used to improve performance.

A perfect Vernam cipher was simulated in software using the output of a strong pseudorandom number generator. Due to the known statistical characteristics of this cipher it was unnecessary to read any real speech data when creating simulated ciphertext, allowing a significant savings in processing time.

6.3 Testing Methodology

To determine a model of security we must compute the entropy of ciphertext produced by encrypting the output of each representative speech coder with each of the representative encryption algorithms.

The general testing methodology outlined in the block diagram of figure 6.1 was applied to each combination of source and encryption coder to obtain these measures. Conditional entropy measures of maximum feasible complexity were computed to obtain the best possible security models.

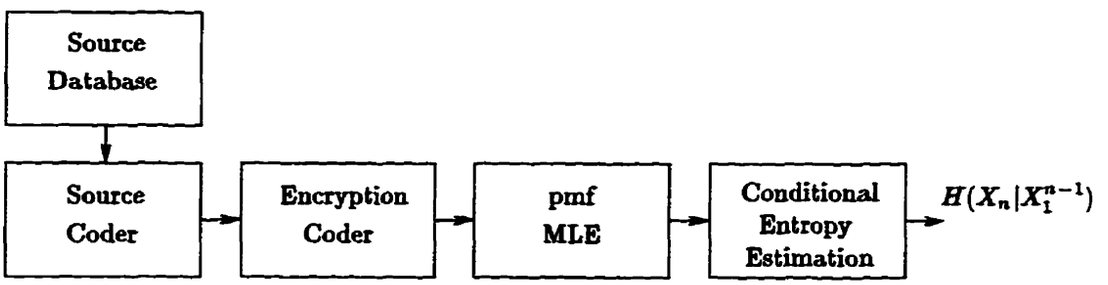


Figure 6.1: Block Diagram of Ciphertext Entropy Measurement Methodology

6.3.1 Security Model Source Database

In order to obtain convergent measures of higher order conditional entropies it was necessary to expand the database used in chapter 5 for Kullback-Leibler distance measures. The speech data used for the security model experiments included the 127 conversations compiled for the divergence measures and an additional 429 conversations selected from the SWITCHBOARD database. The additional conversations were selected according to the transcriber's subjective evaluations of quality to maximize the similarity to high quality μ-law PCM speech. Due to the large volume of data, it was not possible to confirm the quality of each conversation by means of an independent subjective quality evaluation. The complete database yielded at total of 556 files and approximately 5500 minutes of conversational English speech data.

6.3.2 Representative Speech Coders

The same set of representative speech coders defined in section 5.2.8 and used for the development of quality and redundancy models was applied to this experimental work. Experimental measures of ciphertext entropy were performed on speech encoded in μ-law PCM, DPCM, 3,4, and 5-bit ADPCM, and CELP formats.

6.3.3 Conditional Entropy Estimation Methods

Direct Measures

Conditional entropy measurements were obtained from first to third order, $H(X)$, $H(X_2|X_1)$, $H(X_3|X_1^2)$, by the direct calculation of equation (2.1). We found that encrypted speech exhibited some characteristics that were not encountered when we computed fourth order conditional entropy measurements in chapter 3. These characteristics made it infeasible to compute fourth order conditional entropy measures of the encrypted speech data by the direct approach.

In computing ciphertext entropy we noted a reduction in the rate of convergence for encrypted speech measures from the rates experienced for the unencrypted speech entropy measures of figure 3.5. Effective encryption algorithms tend to result in more uniform probability distributions in the ciphertext than were observed in the original message. As our procedure for computing conditional entropy measures requires a maximum likelihood estimation of the process pmf, randomization tends to increase the number of observations required for an accurate pmf measurement. It was necessary to increase the amount of speech data processed by the direct entropy calculation routines from the 2725 minutes used for unencrypted speech measurements in section 3.4 to a sample of approximately 5500 minutes.

Another effect of randomization was an enormous increase in the effective state-space required to record higher order pmfs. When we computed fourth order conditional entropy measurements of the original speech process in chapter 3 we employed a dynamic memory allocation scheme for the state space, as discussed in Appendix D. This allocation scheme could take advantage of the high redundancy of the speech process and minimize the number of memory locations required for the model. We discovered that encryption removed such significant amounts of temporal correlation in the samples that a fourth order probability model would require a significant fraction of its nominal 256^4 states. It was not feasible to allocate a model of this size on our 32 bit architecture Sparc 20 computing platform.

Indirect Measures

Higher order entropy measures reveal more information about a process so it was desirable to obtain the highest possible order of entropy measure for our security models. The randomizing property of encryption increased the state-space required for direct fourth order calculations beyond feasible levels so we applied the the entropy estimation algorithm of chapter 2 to obtaining fourth order estimates.

A fourth order conditional entropy estimate, $\tilde{H}(X_4|X_1^3)$ was computed from a set of PCE measures of third order. Let $\Upsilon_{p,1} \in \mathcal{X}$ be the set of p points at which we computed PCE measures. We define the set of p PCE vectors computed from a data set of size $S_{\max}$ minutes to be,

$$\mathcal{H}_{S_{\max},4,\Upsilon_{p,1}} = \psi(S_{\max}, 4, \Upsilon_{p,1}), \tag{6.1}$$

as defined in equation (2.7).

From this set of vectors we could interpolate the remainder of the fourth order entropy surface, $\tilde{\mathcal{H}}_{S_{\max},4,\mathcal{X}}$, and then compute the conditional entropy estimate, $\tilde{H}(X_4|X_1^3)$ according to equation (2.12).

The operations cost for each PCE vector was considerably higher for encrypted speech than experienced for the original unencrypted μ-law process. In Appendix E we present a detailed study of the operations required for fifth and sixth order conditional entropy measures of the speech process. For encrypted PCE calculations we would add to this estimate the significant number of operations required by the source and encryption coding stages.

In computing third order PCE vectors for data encrypted by any of our representative encryption coders we discovered the variance between the measures to be extremely small, implying a near-uniform fourth order conditional entropy surface. Taking advantage of this uniformity, we found we could reduce our calculation to a single randomly chosen PCE

vector, $(\mathfrak{h}(4,x),x)$, and compute the fourth order estimate as,

$$\hat{H}(X_4|X_1^3) = 256 * \mathfrak{h}(4,x). \tag{6.2}$$

To confirm this observation we computed complete conditional entropy surfaces for a feasible model size of third order. Theses surfaces were computed for μ-law PCM speech encrypted by DES ECB, CFB, CBC, and OFB modes and are presented in Appendix M. The four third order conditional entropy surfaces presented in figures M.1, M.2, M.3, and M.4 were found to be distinctly uniform. As the property of uniformity was found to be so strongly evident in the third order surfaces we considered the fourth order data computed to be indicative of a similar trend.

This simplification was prone to error but we found it sufficient to indicate the rate of convergence of the fourth order measure. The indirect fourth order measures were pursued no further when it was found that we could not provide a sufficient volume of encrypted speech data to achieve model convergence for our entropy estimates.

6.4 Conditional Entropy Measures for Encrypted Speech

We examined the effect of encryption on first, second, and third order conditional entropy measurements in detail for encrypted μ-law speech in order to gain insight into its convergence characteristics. These results are presented in section 6.4.1. We then performed third order conditional entropy measurements for every combination of the remaining representative source coders and encryption coders. These results are presented in section 6.4.2. The third order measures comprised the best feasible estimates of ciphertext entropy and are summarized for use in the security models in table 6.1.

6.4.1 Detailed Entropy Results for μ-law Coded Speech

The entire database of μ-law encoded speech was encrypted by each of the representative ciphers, and first to third order conditional entropy measurements were computed at a variety of sample sizes. The results of these measures are presented in figures 6.2, 6.3, 6.4, and 6.5.

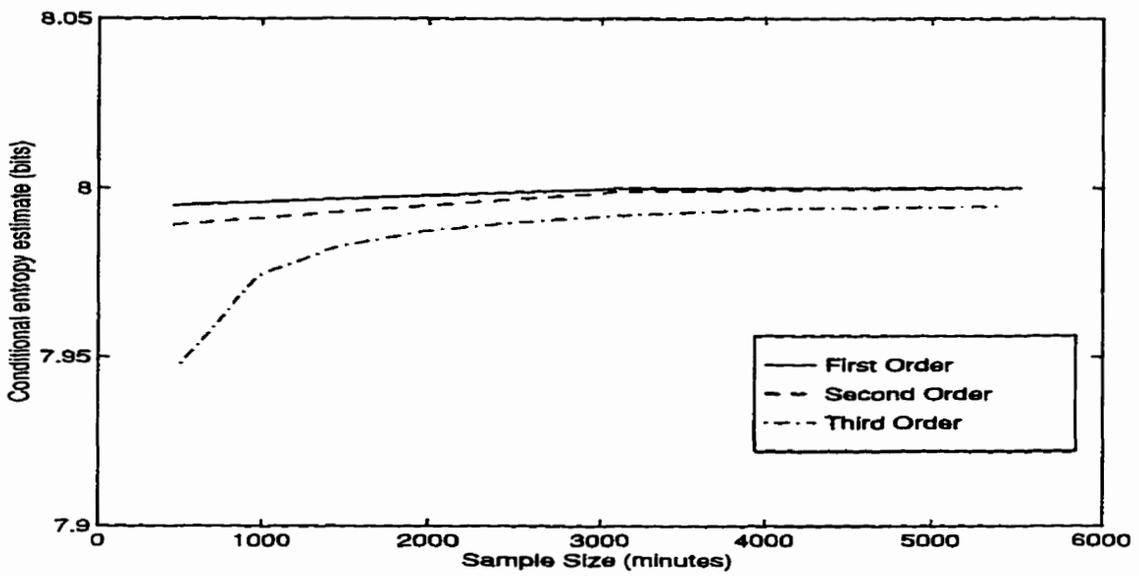


Figure 6.2: ECB Conditional Entropy Convergence of μ-law PCM

Observations on the Detailed μ-law Convergence Results

The figures indicate that the ECB, CFB, CBC, and OFB ciphers all achieve good randomization of ciphertext for first, second and third order measurements. The rates of convergence varied slightly, with the ECB cipher appearing to converge more slowly than the other ciphers. In addition, it was found that none of the block ciphers had attained an ideal entropy measure of 8 bits/ciphertext byte for the volume of source data applied to the experiments. This slow rate of convergence was confirmed with a study of the Vernam cipher using a much larger volume of simulated speech data. This rate of convergence was

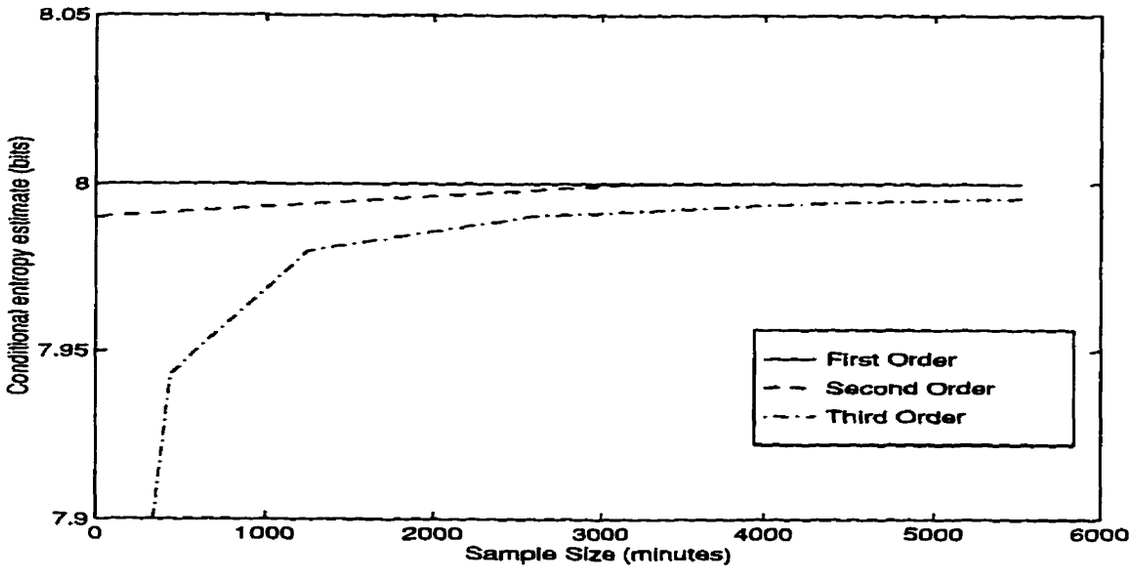


Figure 6.3: CFB Conditional Entropy Convergence of μ-law PCM

determined to be a characteristic of the maximum likelihood estimation procedure for the process pmfs. These characteristics will be discussed in section 6.4.2 and used to normalize all subsequent measures to obtain stable entropy estimates.

The results of the first and second order entropy calculations were found to be largely reflected in the third order measures. As higher order measures yield the most information about the processes, the remainder of our direct conditional entropy measures were performed using the third order measure.

6.4.2 Third Order Entropy Results for all Coder Combinations

The remaining DPCM, ADPCM, and CELP coders were applied according to the testing methodology to generate compressed speech for encryption by each of the representative encryption algorithms. Third order conditional entropy measurements were performed at various intermediate sample sizes to determine the convergence characteristics of each source type.

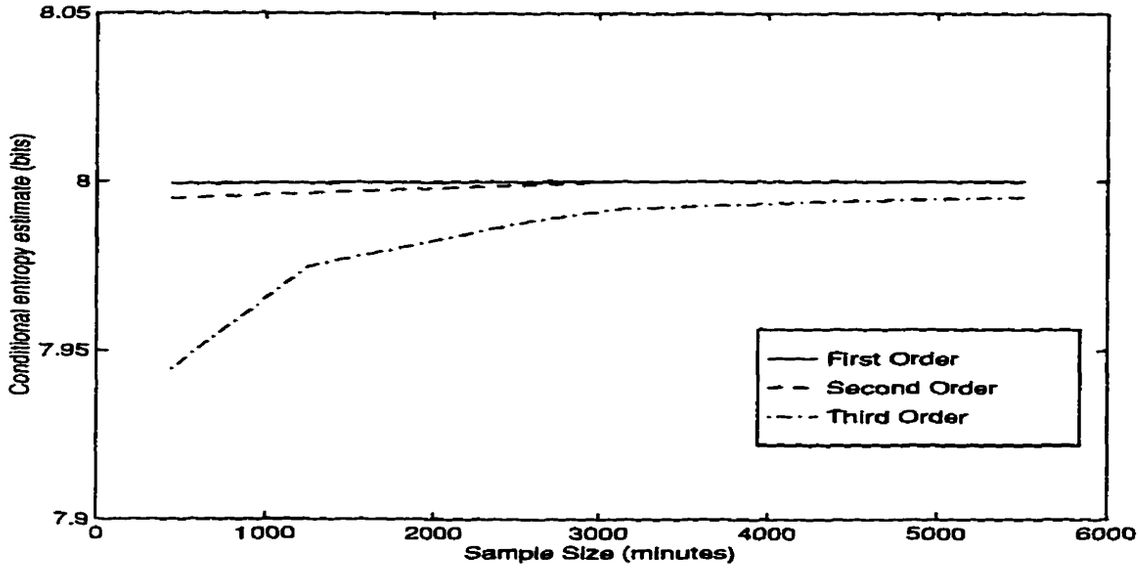


Figure 6.4: CBC Conditional Entropy Convergence of μ-law PCM

Computational Considerations

Extending the ciphertext entropy measurements to include all combinations of the representative source and encryption coders was computationally expensive. The least computationally intensive approach to the task would have been to create DPCM, 3, 4, and 5 bit ADPCM and CELP coded versions of the complete source database of 5500 minutes of μ-law speech and then perform encryption on each version with each of the representative encryption algorithms. This minimization of computational work was, unfortunately, coupled with a requirement for sufficient disk storage space for each of the source coded versions of the speech data. The original μ-law data required in excess of 2.6 Gbytes of storage space, and the coded versions would require an additional 5.4 Gbytes in total. These resources were not available so it was necessary to compress the entire speech database with each source coder for each type of encryption algorithm. The additional computational cost of this procedure added considerably to the effort required for our conditional entropy measures.

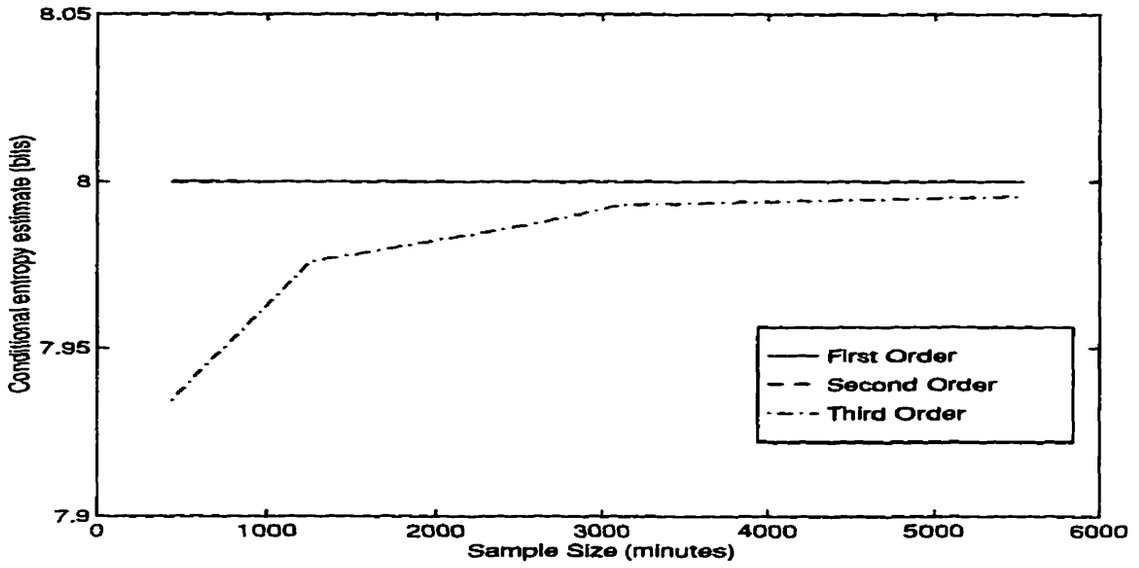


Figure 6.5: OFB Conditional Entropy Convergence of μ-law PCM

Convergence Characteristics by Encryption Type

The ciphertext entropy data can be presented in a variety of formats for insight into the relationships between source coders and encryption coders. We found it interesting to compare the ciphertext entropy measures for different source encodings under a single mode of encryption. Presented in figures 6.6 and 6.7 are convergence measures for the source coders under ECB and CFB encryption, respectively. The sample size coordinate in the figures was computed for each coder to correspond to the volume of the original μ-law database processed by the speech coder.

It is important to note that the volume of coded speech data processed by the encryption routines, and hence the volume of data available to the MLE pmf modelling routine was a function of the source coder rate. This characteristic was found to be a factor in interpreting the ciphertext entropy results. The data in figures 6.6 and 6.7 suggests the unexpected result that speech compression results in a lower entropy measure and a correspondingly lower security measure. This conclusion would be inaccurate. The discrepancy is a caused

by a reduction in the volume of encrypted data available to the pmf modelling routines as a result of speech compression.

To quantify this convergence problem we present the ciphertext entropy measurements for various speech encodings under a Vernam cipher encryption in figure 6.8. This figure indicates the relatively slow convergence expectations for a perfect cipher. A perfect cipher randomizes the ciphertext statistics so well that the pmf modelling routines require the maximum number of observations to obtain stable entropy measures. The figure indicates how lower coder rates simply increase the number of observations required for convergence.

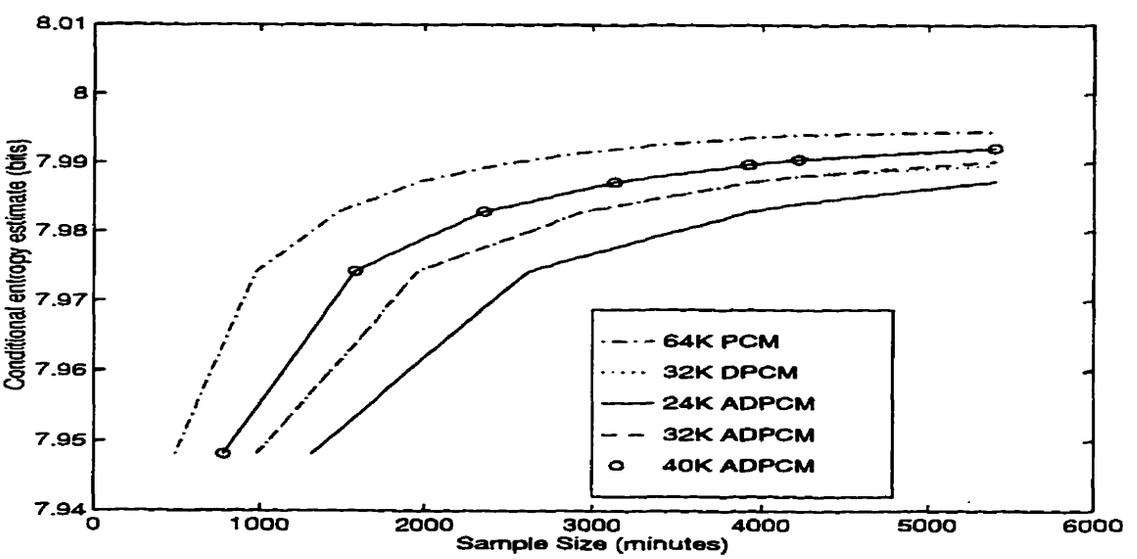


Figure 6.6: Entropy Measures of ECB Encrypted Speech for Various Source Coders

Convergence Characteristics by Source Coder

The results of comparing different speech coders under a single mode of encryption in figures 6.6 and 6.7 and the relative convergence characteristics of the Vernam cipher in figure 6.8 reveal the need to compare the accumulated entropy data on a basis relative to the rate of the source coder.

We found it preferable to present the convergence data for all encryptions of a source

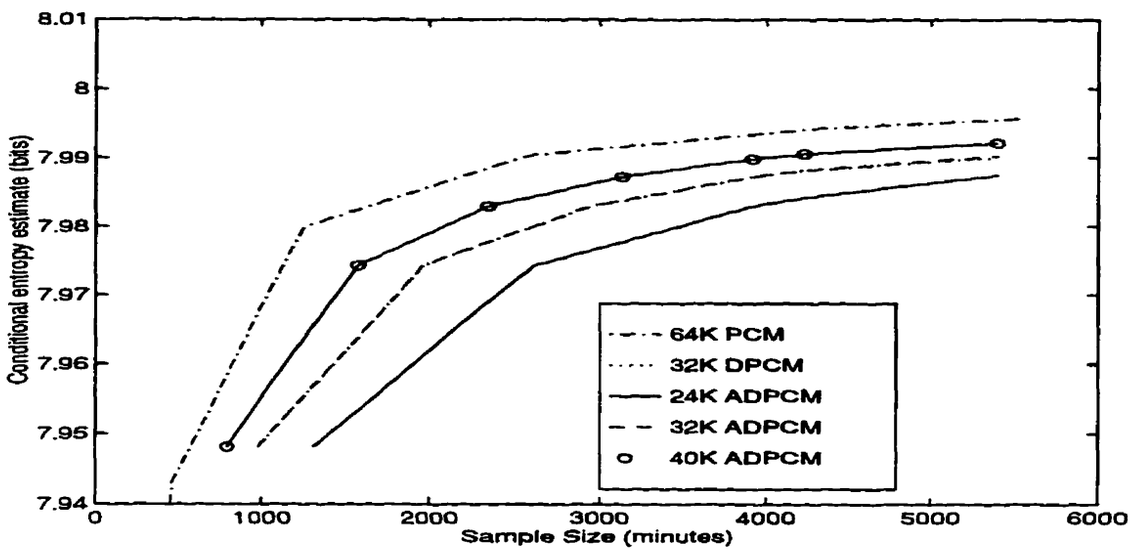


Figure 6.7: Entropy Measures of CFB Encrypted Speech for Various Source Coders

coder type in a single figure. In the interest of space, these results are presented in Appendix N, figures N.1, N.2, N.3, N.4, and N.5. Each figure is a summary of the complete set of third order conditional entropy calculations for a particular waveform coder under all encryption coder combinations.

The CELP coder results are not included in figures 6.6 and 6.7 because our CELP results consist of single measurements at the maximum database sample size. The CELP coder produces data at such a low rate that the entire database, when CELP coded, did not produce a large enough volume of ciphertext to trigger an intermediate entropy calculation in the automated testing software.

Observations on the Complete Third Order Entropy Measures

The conditional entropy data presented in Appendix N allows a fair comparison of the convergence rates for each type of encryption, given the relative convergence rate of a perfect Vernam cipher operating on the same source-coded data. The results of these

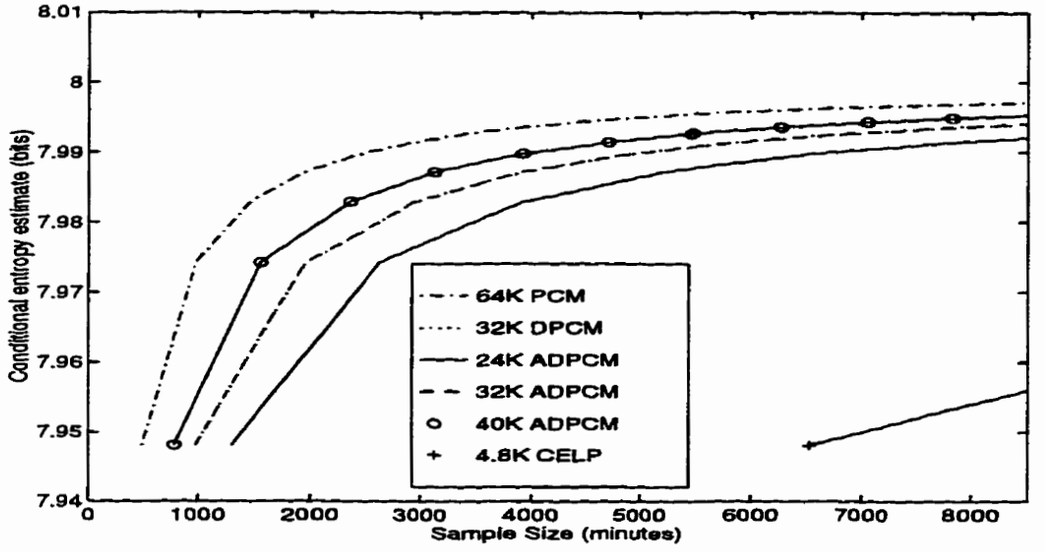


Figure 6.8: Entropy Measures of Vernam Encrypted Speech for Various Source Coders

figures indicate that the DES based ciphers achieve a performance that is similar to the Vernam cipher at this order of entropy computation.

It is clear the conditional entropy measures in Appendix N have not converged to their final values. To predict the eventual entropy rate for each coder-encryptor combination we propose to scale the measure obtained at the maximum sample size in relation to the convergence characteristics of the corresponding Vernam cipher results. This technique for scaling conditional entropy measures according to measures obtained from larger sample sets was described in detail for PCE measures in section 2.6. We propose a much simplified scaling function of the form,

$$\hat{H}_{S^\infty}(X_3|X_1^2)_{\text{coder,encryptor}} = \min\left(8, \left[\frac{H_{S_{\max}}(X_3|X_1^2)_{\text{coder,encryptor}}}{H_{S_{\max}}(X_3|X_1^2)_{\text{coder,Vernam}}} \cdot 8\right]\right), \tag{6.3}$$

where $H_{S_{\max}}(X_3|X_1^2)_{\text{coder,Vernam}}$ represents the entropy measure obtained for the Vernam cipher at the maximum sample size of $S_{\max} = 5500$ minutes. By definition, the perfect

Vernam cipher is known to attain a ciphertext entropy of 8 bits/ciphertext byte. This trivial scaling relationship assumes the rate of convergence of the various ciphertext streams remains constant relative to the Vernam cipher beyond the maximum sample set size. This is a reasonable assumption for strong ciphers such as the DES variants studied here, and it is further strengthened by having computed at a sample size S_{max} that is beyond the knee of the convergence curves.

A close observation of the ciphertext convergence curves indicates that the measures based on μ-law PCM data in figure N.1, and to a lesser extent, the DPCM results in figure N.2 diverge slightly from the ideal Vernam cipher convergence characteristics under ECB mode encryption. The divergence is slight, but may indicate that the third order statistics of ECB mode ciphertext are non-ideal. Higher order entropy measures would confirm the presence of some redundancy in ECB enciphered speech data. In section 6.5 we will present evidence that a fourth order conditional entropy measure of μ-law speech is distinctly smaller than that of the other encryption modes.

The evidence of figures N.3, N.4, and N.5 indicates that the more sophisticated forms of source coding result in ciphertext convergence curves that are almost identical to Vernam cipher convergence. We conclude that this result is a measurable effect of the removal of redundancy from the input process to the encryption device. Despite the known deficiencies of ECB encryption, source coding resulted in a ciphertext process that could not be distinguished from an ideal ciphertext process with a third order entropy calculation.

6.4.3 Entropy Measure Summary for Coder Combinations

The third order conditional entropy measures computed at the maximal sample size in sections 6.4.1 and 6.4.2 constitute our best estimates. Using the simple scaling relationship of equation (6.3) we predicted the third order measure that would be obtained at a model size sufficiently large to guarantee model convergence. These estimates are summarized for all combinations of source and encryption coders in table 6.1.

Coders	Encryption Mode				
	ECB	CFB	CBC	OFB	Vernam
μ-law PCM	7.99805	7.99919	7.99919	7.99918	8.00000
DPCM	7.99804	7.99933	7.99934	7.99933	8.00000
40K ADPCM	7.99954	7.99954	7.99954	7.99955	8.00000
32K ADPCM	7.99977	7.99987	7.99988	7.99987	8.00000
24K ADPCM	7.99954	7.99994	7.99995	7.99994	8.00000
CELP	8.00000	8.00000	8.00000	8.00000	8.00000

Table 6.1: Predicted 3rd Order Entropy Measures for Encrypted Speech

A general pattern in the results is made more evident by the graphic presentation of a surface consisting of the ciphertext entropy measurements as a function of source and encryption coder combinations in figure 6.9.

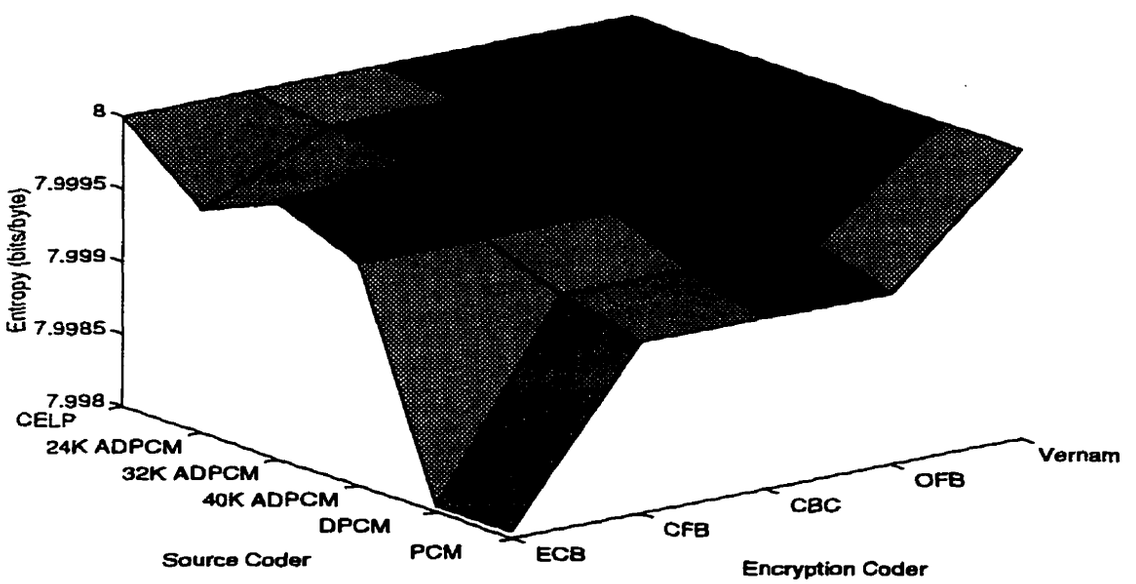


Figure 6.9: Ciphertext Entropy Surface

6.5 Fourth Order Entropy Calculations

Attempts to estimate fourth order conditional entropy for encrypted speech from a set of partial conditional entropy vectors were thwarted by the randomizing characteristics of the

encryption algorithms.

To determine the feasibility of fourth order entropy calculation, the simplified entropy measure of equation (6.2) was applied according to the test methodology of figure 6.1 for the Vernam cipher. A large volume of Vernam-encrypted data was simulated and the convergence characteristics of the fourth order conditional entropy calculation on this process were computed and presented in figure 6.10.

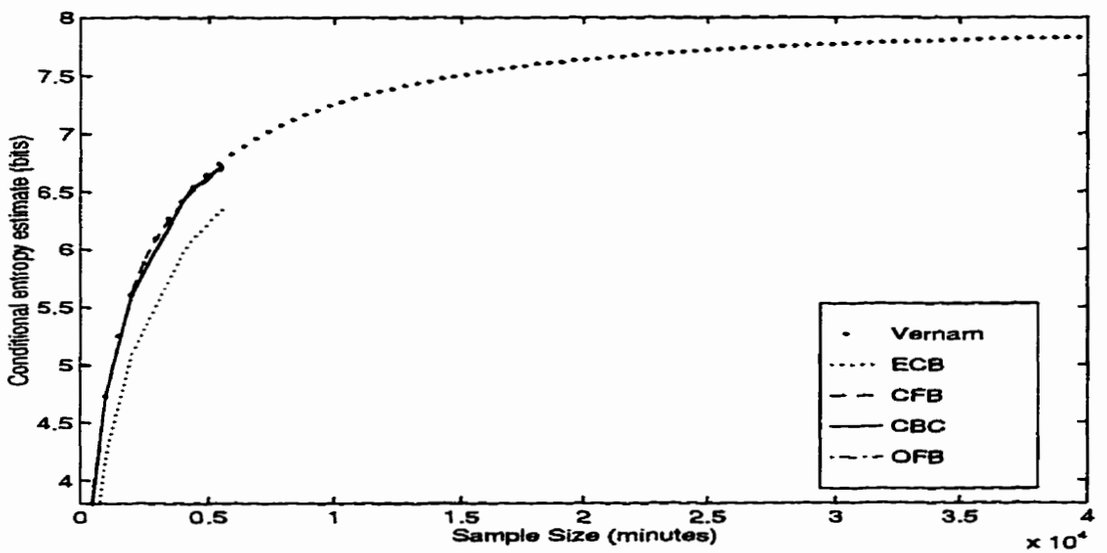


Figure 6.10: 4th Order PCM Coder Results for All Encryption Modes

This figure indicates that the entropy estimation converges to the expected level for a perfect cipher of 8 bits/ ciphertext byte, but does so at a very slow rate compared to the unencrypted speech data in figure 3.6. Approximately 40000 minutes of simulated encrypted μ-law PCM speech data were required for convergence. Speech coders at lower data rates will converge proportionately slower but will always yield the same result of 8 bits/ ciphertext byte since this is a perfect cipher.

The Vernam cipher achieves the highest possible randomization of ciphertext data and hence will display the slowest rate of convergence. To determine the rate of convergence for

speech encrypted by the DES cipher we performed a fourth order estimation on real μ-law PCM data under the four DES encryption modes. The convergence characteristics of these four measures were computed up to the full database size of 5500 minutes of speech and are also presented in figure 6.10

It was not possible to distinguish the converging entropy measures of these representative ciphers from these experiments due to the limited size of the database. We computed these measures to the maximum sample size that was feasible given finite space and computational constraints but were unable to reasonably predict fourth order conditional entropies. The randomizing properties of the representative encryption coders slowed the rate of convergence and the addition of source and encryption coding added computational costs that were not encountered when the method was applied to the original unencrypted speech process.

Referring to figure 6.10, we observe that ECB encrypted speech appears to possess a lower fourth order entropy than CFB, CBC, and OFB encrypted speech up to the maximum observable measurement. There is also evidence in the figure that the CFB, CBC, and OFB measures diverge slightly from the Vernam cipher results at maximum model size. To confirm these observations we would have to extend the measures to a much larger model size. The Vernam cipher convergence curve is nearly logarithmic, suggesting that doubling the size of the database will yield a significant improvement in the accuracy of the entropy estimates.

While we were able to demonstrate that the entropy estimation technique could be used for higher order measures, we found it was not feasible to compute for this research. We concluded that third order measures would have to suffice for our security models.

6.6 Security Index Evaluations

In this section we will evaluate the three proposed theoretic security indices using the experimental data derived in chapter 3, chapter 5, and earlier in this chapter.

6.6.1 Unicity Index

The unicity index developed in chapter 4, $N_o = \frac{H(\mathbf{K})}{D}$, yields an indication of the minimum amount of ciphertext an attacker needs to intercept in order to break a cipher. This index can provide a simple measure of the relative strength of a cipher in terms of its key space and the redundancy of the message process. As we have already discussed in section 4.6.1, the unicity index is insensitive to the encryption algorithm, so we do not require the measures of ciphertext entropy summarized in table 6.1 to evaluate it.

For the set of representative ciphers there are only two values of $H(\mathbf{K})$. The DES variant algorithms have a key entropy of 56 bits, and the Vernam cipher has a key length equal to the message length. For the conversational speech process defined for this research, the Vernam key entropy may be considered to be effectively infinite. The resulting unicity distance for Vernam ciphers is therefore infinite for all values of redundancy.

Presented in figure 6.11 is an evaluation of the unicity index model over the range of all per-symbol redundancies, $0 \leq D \leq 8$ bits/ciphertext byte, and a range of key entropies, $0 \leq H(\mathbf{K}) \leq 60$ bits. Also represented in the figure is a line indicating the range of the unicity index for realizations of the DES ciphers.

6.6.2 Efficiency Index

The encryption efficiency index proposed in chapter 4 was defined by the equation $\mathcal{E} = \frac{H(\mathbf{C})-H(\mathbf{M})}{H(\mathbf{K})}$. This index facilitates the comparison of encryption algorithms on the basis of the amount of ciphertext uncertainty added in proportion to the key cost.

The efficiency index requires the ciphertext entropy measures for the representative

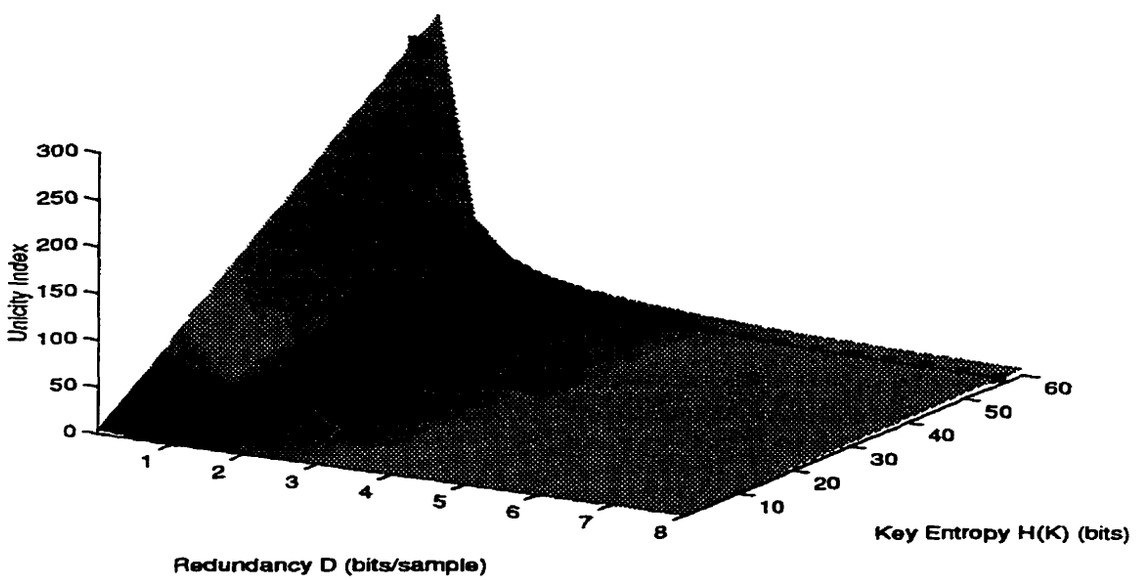


Figure 6.11: Unicity Index Model

source and encryption coders presented in this chapter and summarized in table 6.1. The index also depends on the source process entropy estimated in chapter 3 as $H(\mathbf{M})$=2.79 bits/sample, and the encryption algorithm's key entropy.

Presented in figure 6.12 is an evaluation of the efficiency index surface as a function of source and encryption coder combinations.

The form of this simple efficiency surface is similar to that of the ciphertext entropy surface in figure 6.9, except in the vicinity of the Vernam cipher measures. The key entropy $H(\mathbf{K})$ for each of the DES mode ciphers was a constant 56 bits for these efficiency measures. For the conversational speech process studied here, the key entropy of the Vernam cipher was effectively infinite. This resulted in an efficiency measurement of zero for each of the Vernam cipher results. It should be noted that the Vernam cipher efficiency points are not plotted to scale in figure 6.12 in order to allow a detailed view of the remainder of the surface.

In developing a general model of the secure speech communication system we will find it

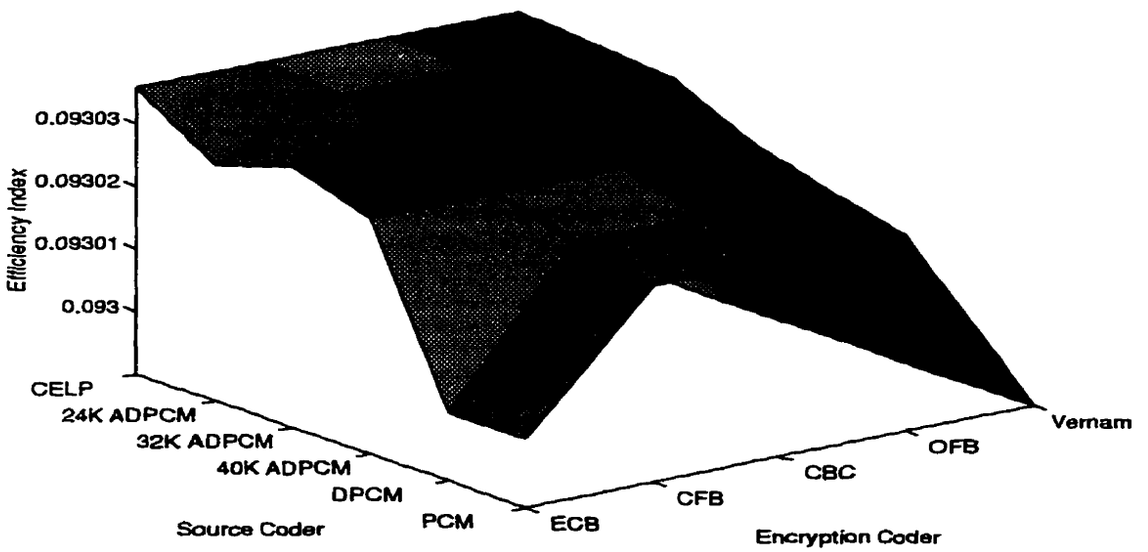


Figure 6.12: Efficiency Surface as a Function of Source and Encryption Coders

useful to compare each of our system models on the basis of a single independent variable. We found the redundancy rate of the source coded message stream to be an ideal independent variable for this comparison. To present the efficiency surface of figure 6.12 in terms of the redundancy rate we mapped the set of source coder variables to the corresponding redundancy rates for each coder according to the experimental redundancy measures performed in chapter 5.

Using the actual operational redundancy rate measures, $D_{\text{actual}}(R)$, summarized in table 5.4 we mapped the set of waveform source coders to a corresponding set of redundancy rates to produce the efficiency surfaces in figure 6.13. An exception was made for the CELP coder redundancy measures, where we used the predicted measure, $D_{\text{operational}}(R)$, rather than the experimental measure. As discussed in chapter 5 the CELP coder results were found to be unreliable and shown to be better represented by the model $D_{\text{operational}}(R)$.

The mapping to redundancy measures scales the efficiency measures in a manner more suitable to a general model of security. Four views of the surface are presented in the figure

to allow the efficiency surface to be examined in more detail. To reduce clutter in the figure, we simplified the efficiency index scale to represent the actual efficiency index measures of $0.0930x$ by the last digit, x. Once again, in this figure the Vernam cipher efficiency measures of zero are not shown to scale in order to reveal more detail in the remaining measures.

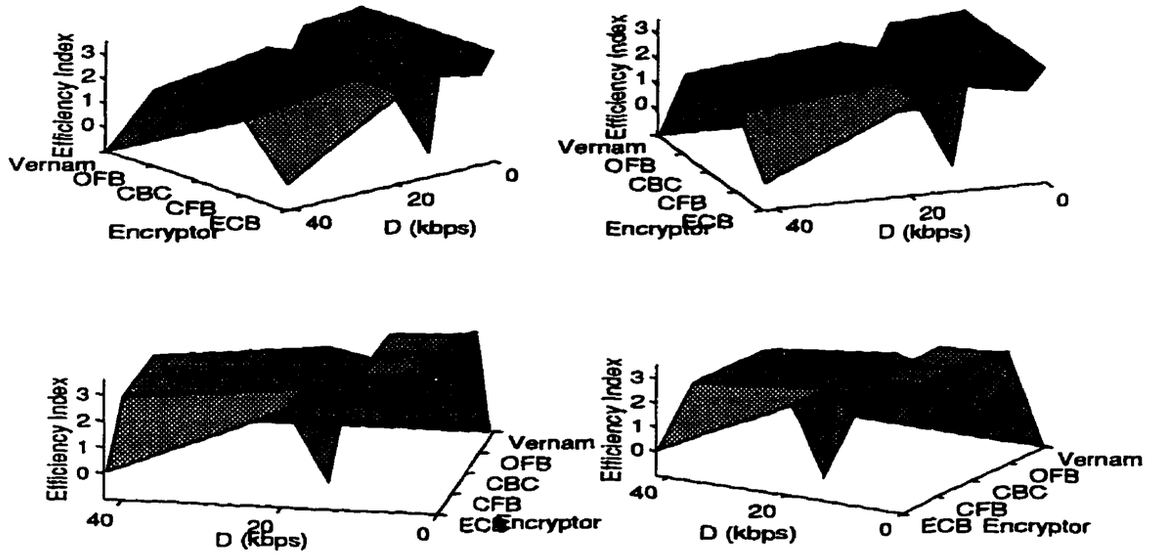


Figure 6.13: Efficiency Surface as a Function of Operational Redundancy Experiments

Presented in figure 6.14 is the same set of efficiency index measures mapped using the operational rate redundancy function of equation (5.20). We were interested in comparing the effect that our approximate operational rate redundancy function may have on the efficiency index. The results presented in the figure indicate little change from the first model, except in the vicinity of the DPCM and 32K ADPCM points. These points are mapped to the same redundancy level by the operational rate redundancy function because they operate at the same coder rate, despite the experimental evidence indicating a higher redundancy in the DPCM-coded process. This is an unavoidable consequence of using the simplified model for redundancy. Fortunately, we can avoid this problem as we have the actual redundancy rate measures for these coders, $D_{\text{actual}}(R)$. We will refer to the results

of figure 6.13 in our future evaluations of the efficiency index.

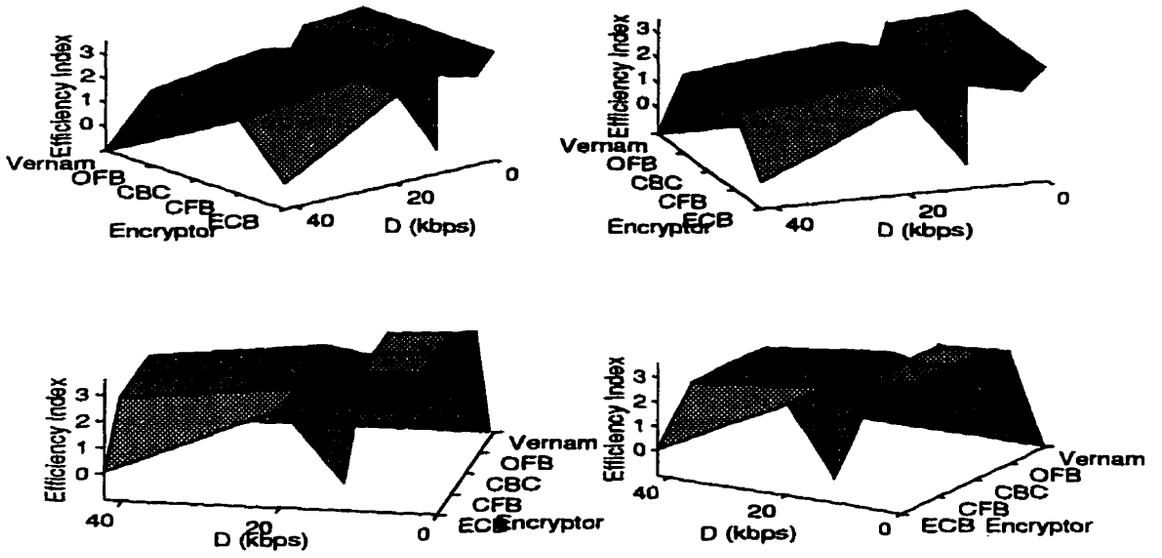


Figure 6.14: Efficiency Surface as a Function of Operational Redundancy Function

6.6.3 Quality Index

The encryption quality index proposed in chapter 4, $\mathcal{Q} = \frac{H(C)-H(M)}{D}$, was designed to facilitate the comparison of encryption algorithms on the basis of their ability to mask a redundant source process. The index measures the encryption algorithm's ability to add uncertainty to the ciphertext as a proportion of the per-symbol redundancy of the source process. The information required for this measure includes the ciphertext entropy data, message entropy, and operational redundancy measures used above in section 6.6.2.

Presented in figure 6.15 is an evaluation of the quality index as a function of the encryption coder and operational redundancy parameters. We applied the same set of operational redundancy measures for the quality index as were applied to obtaining the efficiency index in figure 6.13, using $D_{\text{actual}}(R)$ for the waveform coders, and $D_{\text{operational}}(R)$ for an accurate measure of CELP-coded redundancy.

The quality index surface is strongly determined by the redundancy parameter over the

experimental data set. The features of this surface are difficult to observe in this format, so we have presented a set of normalized cross-sectional views of the quality surface in figure 6.16. Each line in the figure represents a set of quality index measures corresponding to a source coder operating at the indicated redundancy rate, D (kbps). To allow a comparison of the measures, each quality function is normalized by the quality index computed for that redundancy rate under DES ECB encryption.

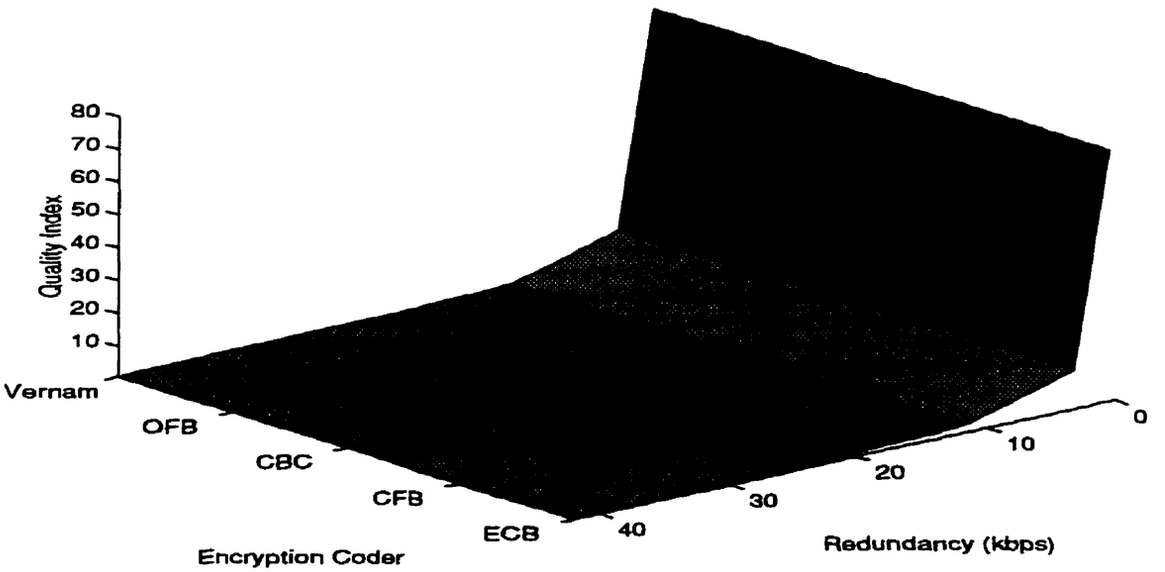


Figure 6.15: Quality Surface as a Function of Operational Redundancy Experiments

The cross-sectional functions of figure 6.16 indicate lowest quality for DES ECB encryption, a higher quality for DES CFB, CBC, and OFB ciphers, and the highest quality for Vernam encryption.

6.7 Observations on the Security Indices

We can make some general observations on the characteristics of the three security indices based on the results in figures 6.11, 6.13, 6.15, and 6.16:

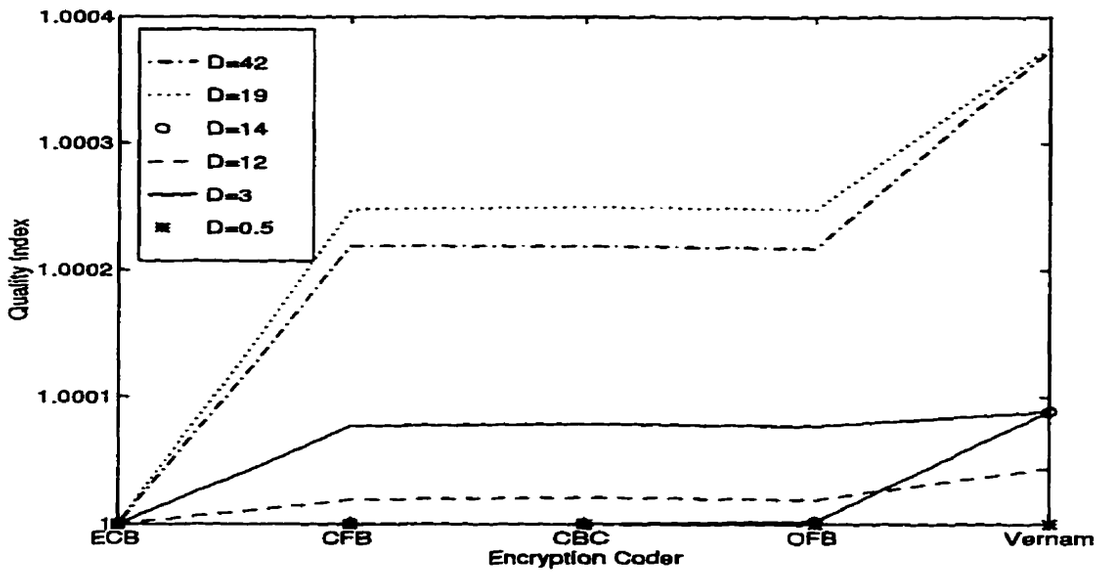


Figure 6.16: Quality Curves Normalized to ECB Quality Measure

1. The unicity index model allows comparison of encryption coders on the basis of the size of the key space only. The unicity index confirms the value of performing source coding to reduce redundancy prior to encryption.

2. Redundancy plays a strong role in determining the unicity distance and quality indices. The efficiency index is responsive to redundancy, but is also determined by other factors.

3. The efficiency index generally increases with reductions in redundancy, but the source coder complexity also has a significant impact. A comparison of 32 kbps ADPCM and DPCM coders revealed that the lower complexity DPCM coder resulted in a lower efficiency index, despite a similar redundancy level.

4. The perfect Vernam cipher reflected the expected theoretic security indices: an infinite unicity distance, maximal quality for a given level of redundancy, and a minimal efficiency measure of zero due to the large key size.

5. **The four DES cipher modes studied were found to be marginally distinguishable by the quality and efficiency indices. The ECB mode cipher was found to exhibit the lowest quality and efficiency over the range of operational redundancies. The CFB, CBC, and OFB modes of DES exhibited almost identical indices over the range studied.**

We have evaluated the three proposed security indices over a sample space defined by a set of representative source coders and a set of representative encryption coders. Evaluating the ciphertext entropy components of the indices was revealed to be a challenging task, due to the effects of encryption on the ciphertext statistics. Despite a limitation in the maximum feasible calculation order, the security index results are consistent with expectations. The observations summarized above indicate that the security indices are capable of representing the relative strength, efficiency, and quality of encryption coders. Higher order ciphertext entropy measures would improve the ability to distinguish between encryption methods.

Chapter 7

Formulation of a General Model

When we mean to build,
We first survey the plot, then draw the model;
And when we see the figure of the house,
Then must we rate the cost of the erection.

William Shakespeare. King Henry IV. Act II, Sc. 3.

7.1 Introduction

In section 7.2 of this chapter we will define rudimentary models for the complexity of source, encryption, and channel coders. These models will provide an essential dimension to the analysis of secure communication system designs. We will then combine the models of objective speech quality, bit rate, security, and complexity to form an optimization problem for the integrated secure speech communication system.

As the general model comprises a multidimensional optimization problem, it is difficult to simply summarize all of the characteristics of the integrated secure speech communication system. Specific solutions will depend on the objectives and constraints on the secure system

design. In section 7.3 we will present an analysis of the interrelationships among the model parameters and define the functional notation for framing the optimization problem.

In section 7.4 we will formulate the objective function for optimization of the general model of the secure speech communication system. The general model will serve to give insight into what we have found to be a complex problem with many interdependent parameters.

Finally, section 7.5 will present the solutions to a variety of constrained and unconstrained optimizations with varying objectives in order to demonstrate the general behaviour of the secure speech communication system as represented by our experiments. We will include two examples of practical system design problems and demonstrate the effectiveness of the general model for developing secure systems.

7.2 Complexity Models

In this section we will define rudimentary models for the complexity of source, encryption, and channel coders. These models are best considered as indicators of general trends rather than as exact measurements.

The inclusion of complexity models adds a useful dimension to the study of secure communications systems. In addition to the objective parameters of quality, bit rate, and security, the complexity of a system realization is often a deciding factor in the design of an integrated system. Often a system is constrained by complexity considerations such as the maximum execution speed of a digital signal processor, the cost of implementing a design in hardware, a limit on the area available to the speech processing units in a integrated circuit design, or the need for real-time execution of an algorithm in software. An acceptable solution to the secure system design problem must be sensitive to the issues of cost represented by a complexity measure.

7.2.1 Definition of Complexity

In many instances it is important to develop a precise definition of complexity before attempting to evaluate it for a system implementation. Standard measures of complexity include gate counts in integrated circuit realizations, operation counts such as the order of operations estimates for entropy estimation presented in Appendix E, or measures of the rate of operations required during execution such as MIPS counts (million instructions per second). Less accurate may be estimates of complexity based on the time required to execute a required function or component counts in a discrete physical implementation.

While a precise measurement of complexity is most desirable, for defining a complexity function for the general system model we require only a measure of the relative performance. We wish to develop estimates of relative complexity of the coders in each class of system component. As a result of classifying the source, encryption, and channel coder complexities separately, the measures need not be based on the same definition of complexity and we are free to choose measures that are appropriate to each class of system component. This simplifies the development of complexity estimates and allows the solution of the general secure system model to be performed over three independent models of complexity.

In the following three sections we will present simple complexity models for the source, encryption and channel coder components of the system.

7.2.2 Source Coder Complexities

Presented in table 7.1 is an estimate of the relative complexities of the 6 representative source coders. These results are very approximate, and they are based on an evaluation of three different types of complexity measures for the set of speech coders. The first type of measure, presented in [19], was based on the relative number of gates required for hardware implementations of the speech coders. The second measure was an estimate of the MIPS count for each of the source coders as presented in [93]. Neither of the measures in [19]

or [93] contained a reference to the simple DPCM coder included in our representative set. For this reason, the third measurement type was based on our observations of the execution times and sizes for the software implementations of the 6 source coders. The third measure allowed an estimate of the relative complexity of the DPCM coder.

It should be stated that the results in [19] and [93] were considered to be very approximate, and hence our tabulated complexities must be viewed accordingly. Also included in table 7.1 is a measure of the effective operational redundancy rate for each source coder.

Source Coder	Relative Complexity	Redundancy (kbps)
64 kbps μ-law PCM	1	41.7
32 kbps DPCM	2	13.6
24 kbps ADPCM	10	3.24
32 kbps ADPCM	10	11.6
40 kbps ADPCM	10	19.1
4.8 kbps CELP	1000	0.54

Table 7.1: Approximate Relative Complexity of Representative Speech Coders

7.2.3 Encryption Coder Complexities

The development of a model for the representative encryption coder complexities is uncomplicated because it includes only evaluations of the relative complexity of the Vernam cipher and the four modes of the DES cipher.

The Vernam cipher has minimal complexity, and we will consider its only operation to be the execution of an XOR function for each ciphertext bit produced. The four DES modes have almost identical complexity, which is considerably higher than that of the Vernam cipher. We can estimate the relative complexity of the basic DES block encryption by observing the operations performed in a single round of the cipher. A single DES round consists of 1 32 bit permutation and 1 32 to 48 bit expansion, a 48 bit shift operation for the round key, 8 table substitutions to convert 6 bit inputs to 4 bit outputs, 1 32 bit XOR and 1 48 bit XOR. If we assume an efficient hardware implementation of the DES algorithm,

such as those discussed in [94], [95], [96], [97], and [98], we can consider the permutation, expansion, and bit shifting operations to be of minimal complexity. We conservatively estimate each of these operations to have the same complexity as an XOR operation. The table substitutions pose a more difficult problem for determining operations equivalence as we must consider the implementation cost for the tables. We note that each substitution box could be replaced with a combinational logic equivalent mapping 6 inputs to 4 outputs. Such a circuit would contain roughly $6 \times 4 = 24$ gates, requiring a total of 192 gates to implement the set of 8 substitution boxes. Based on these approximations, it is estimated that a total of $3 + 192 + 32 + 48 = 275$ XOR-equivalent operations are required per round. The DES algorithm performs 16 rounds of these operations, leading to an estimated total complexity of 4400 XOR equivalent operations to produce 64 bits of ciphertext in ECB mode. Referring to the block diagrams of the CFB, CBC, and OFB ciphers in figures L.2, L.3 and L.4, it can be seen that each of these algorithms requires an additional 64 bit XOR operation to produce a ciphertext block.

Based on these estimates, and allowing for 64 XOR operations in the Vernam cipher to encrypt a block of the same size as the DES modes, we obtained the estimates of relative complexity presented in table 7.2.

Encryption Coder	Relative Complexity
Vernam Cipher	1
DES ECB	69
DES CFB	70
DES CBC	70
DES OFB	70

Table 7.2: Approximate Relative Complexity of Representative Encryption Coders

In future work, it would be desirable to include a measure of the complexity required to maintain cryptographic synchronization. As discussed in chapter 4 and in Appendix L, a temporary loss of cryptographic synchronization causes complete distortion in the decoded signal over a time period that is determined by the type of encryption algorithm. For highest

security, it may be intended that there is no recovery in the event of suspected interference. More robust implementations may be designed to re-synchronize whenever possible. The cost of maintaining cryptographic synchronization is therefore dependent on the type of encryption algorithm and the security objectives of the system designer.

7.2.4 Channel Coder Complexities

It is beyond the scope of this work to explore the characteristics of channel coders in detail. This rich and complex subject requires considerable expertise and could not be adequately represented by a cursory survey. In order to achieve a model of channel coder complexity we must make many simplifications in our analysis of the relevant factors.

The most significant simplification is to conceive a model of source coder complexity as a function of the bit error rate in the received transmission. We assume that an arbitrary source coder operates under given channel conditions to produce a message stream at the input to the encryption coder with a particular bit error rate. It is recognized that this simplification does not consider the effects that the channel type may have on relative channel coder performance. The interference statistics of a channel can determine the optimal channel coder type, but these important characteristics will not be addressed in this research.

We propose to represent the relative complexity of source coders as a non-increasing function of the bit error rate in the message stream reaching the encryption coder. This simple model identifies the likelihood that a channel coder operating on a given channel must perform more complex operations to achieve reductions in the bit error rate.

As we have performed no detailed study of channel coder complexities, we can not justify the choice of one model type over another, but for simplicity we will represent the channel coder complexity as a linear function of BER.

$$C_{\text{C.C.}}(\text{BER}) = C_{\max} - C_{\max} \cdot \text{BER}, \tag{7.1}$$

where $C_{\max} > 0$, and $0 \leq \text{BER} \leq 1$.

Despite its lack of quantitative background, this simple model serves its purpose in the formulation of a general model by indicating the requirement for an increase in complexity with decreasing bit error rates. More specific data about source coder complexities under the expected channel conditions, and an appropriate choice of weighting in the objective function will improve the accuracy of this model.

7.3 Model Summary

Development of the quality, bit rate, security, and complexity models has suggested a degree of interaction between these parameters that is dependent on the combination of specific source, encryption, and channel coder characteristics. It is our goal to refine our observations of interdependence between the parameters to create a clear and quantifiable relationship that may be controlled to the benefit of the designer of the secure communication system.

7.3.1 Terminology

We must first present some common definitions for describing our terminology:

Definition 9 (Objective Parameters) *The objective parameters are those system parameters of direct interest to the designer of an integrated secure communication system. In this model of the secure system the objective parameters consist of the quality, bit rate, security, and complexity measures of the system. Measurement of these parameters has been the primary objective of our modelling work.*

Definition 10 (Intermediate Parameters) *The intermediate parameters are those quantities measured in the development of the objective parameter models that were found to affect the outcome of the objective parameter measurements. The intermediate parameters found to have a significant effect in this research were redundancy, ciphertext entropy, keyspace, cipher mode, and bit error rate.*

Definition 11 (Controlling Parameters) *The controlling parameters for our system have been idealized to the choice of source, encryption, and channel coders. Through the selection of a particular combination of these three system components the designer may directly determine the system complexity, and indirectly affect the remainder of the objective parameters.*

7.3.2 Intermediate Parameters

In figure 7.1 we present a diagram indicating how the controlling parameters determine the complexity measurements directly by the choice of particular source, encryption, and channel coders, and indirectly affect the set of intermediate parameters.

The dotted arrows in figure 7.1 indicate the data path through the coder components, while the solid arrows indicate the set of interrelationships noted among the objective and intermediate parameters. These interrelationships were discovered in the development of the objective parameter models. Here we will summarize our observations on the intermediate parameters:

- **Redundancy.** The redundancy parameter was defined in equation (5.20) to be determined by the rate of non-essential information remaining in the source coded bit stream. As we will show in figure 7.2, this parameter was found to affect nearly all of the objective parameters.

- **Ciphertext Entropy.** The ciphertext statistics are determined by both the source and encryption coder characteristics. These statistics determined the level of ciphertext entropy computed in chapter 6.

- **Keyspace.** The size of the keyspace is determined entirely by the design of the encryption coder. This parameter was necessary for evaluating two of the theoretic security models.

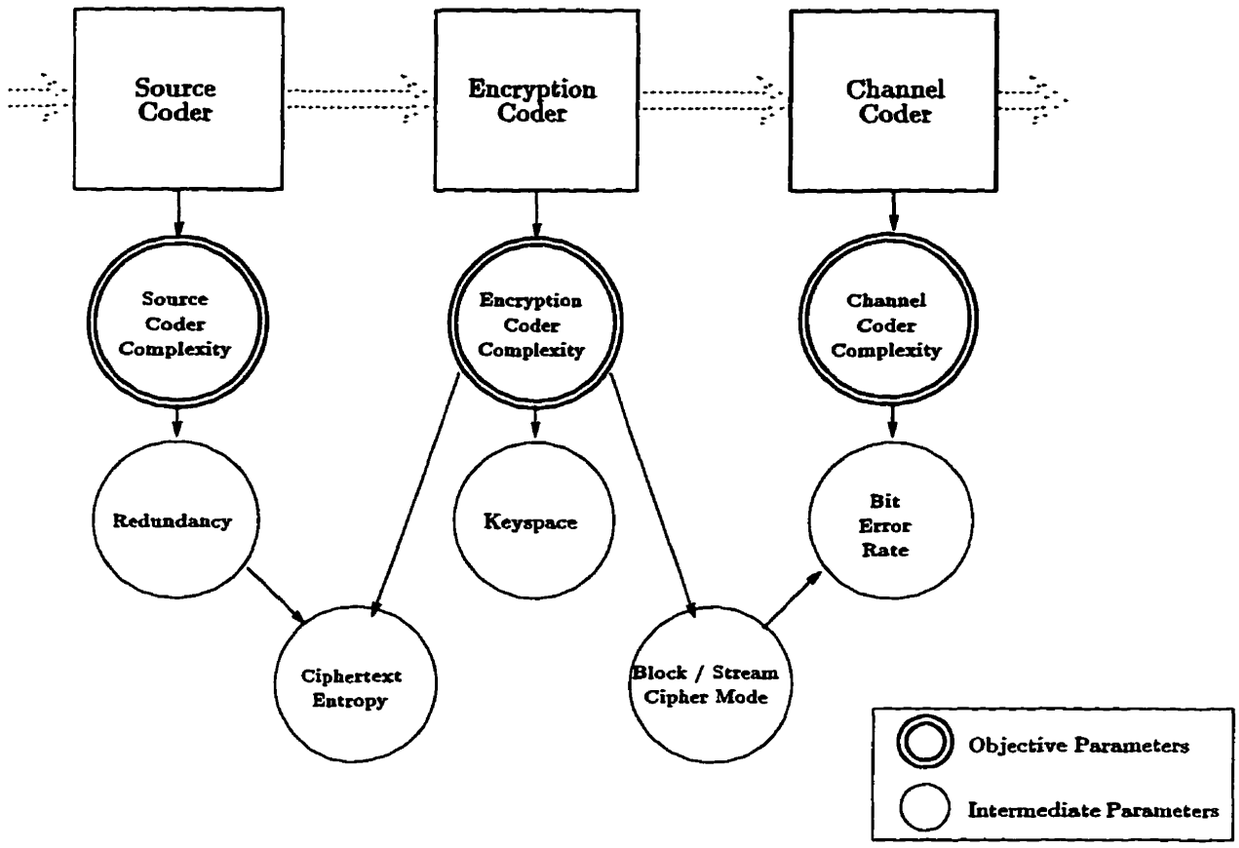


Figure 7.1: Intermediate Parameter Relationships in the Secure System

- **Block/Stream Cipher Mode.** The type of encryption coder, whether block or stream cipher design, affects the propagation of errors in the decrypted message process and so affects the effective bit error rate percieved by the source coder.

- **Bit Error Rate.** In our idealized model of channel coders, the bit error rate is determined by the error correcting capabilities of the channel coder. The encryption device can transform single bit errors into block errors when in block cipher mode, affecting the effective BER perceived by the source coder.

7.3.3 Objective Parameters

In this section we will formally define all of the objective parameters in terms of the appropriate intermediate parameters and the independent controlling parameters. This will allow the formulation of an objective function for solution as an optimization problem.

The diagram of figure 7.2 indicates the significant relationships noted between the objective parameters as a function of the intermediate parameters. These relationships were discovered in developing models of the objective parameters, and were described in detail in chapters 4, 5, and 6. It is clear that the intermediate parameter of redundancy has a significant impact on the performance of the entire secure communication system. We note that 6 of the 8 objective parameters are in some way determined by the redundancy parameter. Only the encryption and source coder complexities are insensitive to redundancy. This figure may be supplemented with the diagram of figure 7.1 for an overview of the interdependencies noted in the integrated system.

In formulating the objective parameter functions we will use the intermediate parameter of redundancy wherever possible as a replacement for the controlling source coder parameter. This will make no significant change in our results as we may apply a direct mapping from source coder to redundancy measure according to table 7.1. The substitution of redundancy for source coder parameter allows a slightly simplified objective function where the objective parameters of quality, bit rate, and security have been developed as functions of redundancy. Similarly, we will use the intermediate BER parameter instead of a reference to the channel coder as our research has not extended so far as to define an appropriate mapping function.

Summarized below are our observations on the objective parameters needed to define an objective function for the general optimization problem. We will review the intermediate parameters controlling each measure and define the functional notation for each objective parameter as a function of the redundancy and BER parameters and the encryption coder type.

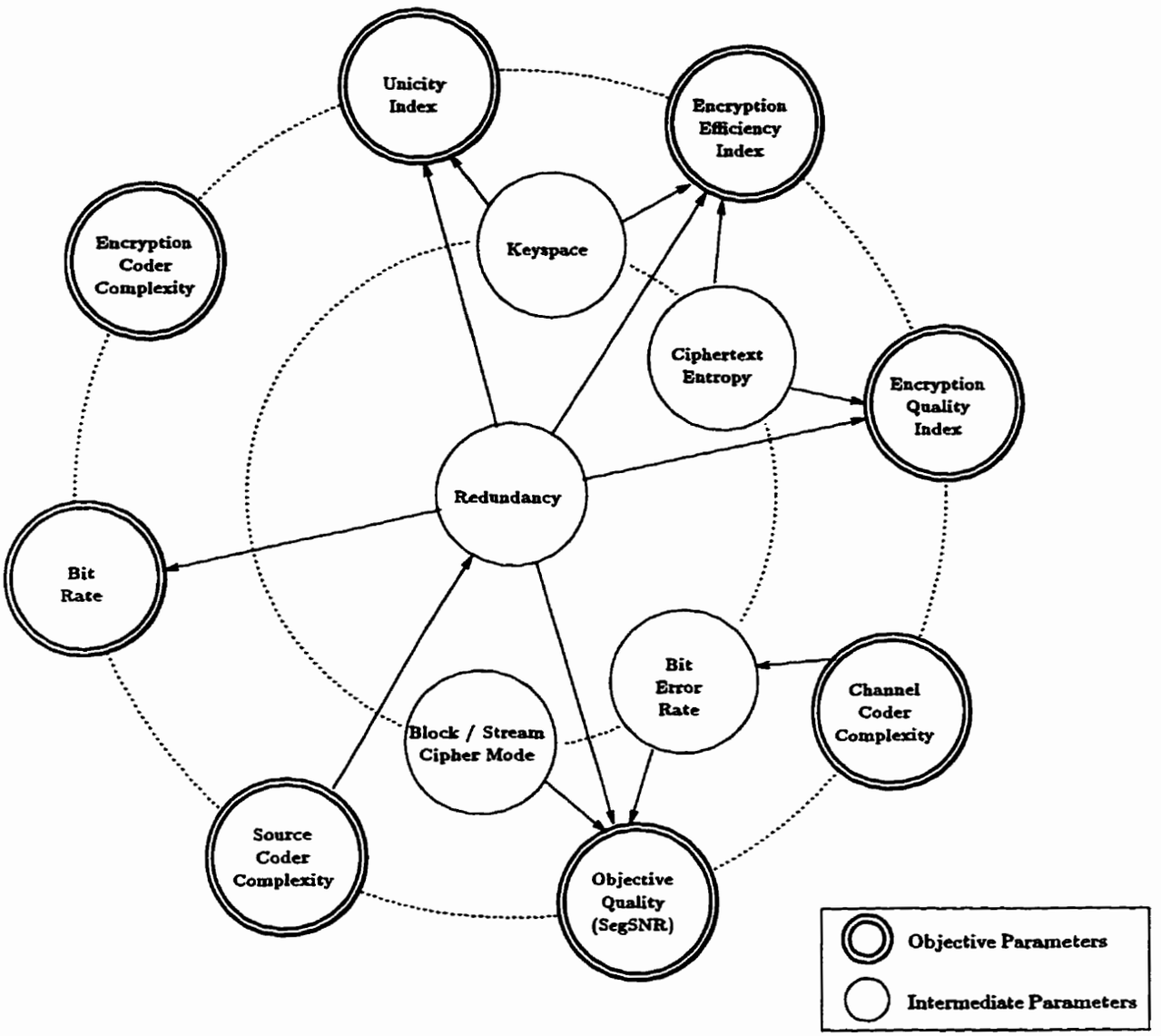


Figure 7.2: Major Parameter Relationships in the Secure System

Unicity Model

The unicity index model of equation (4.2) is defined in terms of the redundancy parameter and the key entropy. As the key entropy is determined by the choice of encryption coder,

we can define the unicity distance model for the objective function as, $N_o(D, \text{E.C.})$, where D is the redundancy rate in kbps and E.C. determines the encryption coder and the entropy of the keyspace. This model has the form indicated in figure 6.11.

Encryption Efficiency Model

The efficiency index model of equation (4.7) is defined in terms of the intermediate parameters of ciphertext entropy and key entropy. It also contains a constant term representing the source process entropy. The key entropy is determined by the choice of encryption coder, while the ciphertext entropy is determined by both the encryption coder and the redundancy, as indicated in figures 7.1 and 7.2. This leads to the definition of the efficiency index model for the objective function as, $\mathcal{E}(D, \text{E.C.})$, where D is the redundancy rate and E.C. indicates the encryption coder type. This model is represented by the experimental data presented in figure 6.13.

Encryption Quality Model

The encryption quality index model is defined in equation (4.8) in terms of the intermediate parameters of ciphertext entropy and redundancy. It also contains a constant term representing the source process entropy. Similar to the efficiency index, we define the quality index model for the objective function as, $\mathcal{Q}(D, \text{E.C.})$, where D is the redundancy rate and E.C. indicates the encryption coder type. This model is represented by the experimental data presented in figures 6.15 and 6.16.

Bit Rate Models

A model of the bit rate as a function of redundancy was derived in chapter 5. The model was based on a linear model of divergence computed from experimental measures, and a theoretical bound for redundancy. Redundancy as a function of bit rate is defined by equation (5.20). We will define the bit rate model for the objective function as $R(D)$, where

D is the redundancy rate. The corresponding bit rate model is represented in figure 5.16 as a function of redundancy.

Quality Models

The data necessary for defining a quality model for the objective function was summarized in figures 5.5, 5.9 and 5.10. These models present objective quality measures as a function of the source coder type, encryption coder mode, and bit error rate. We can perform a mapping from the source coder types to redundancy measures using the observed redundancies summarized in table 7.1. This allows a similar presentation of all security, bit rate and quality models in terms of the intermediate redundancy parameter.

Figures 7.3 and 7.4 summarize the segmental-SNR measures computed in chapter 5 as a function of the source coder and bit error rate. The additional intermediate parameter of encryption mode differentiates the results observed under 64 bit block encryption in figure 7.3 from the results obtained under stream cipher encryption in figure 7.4.

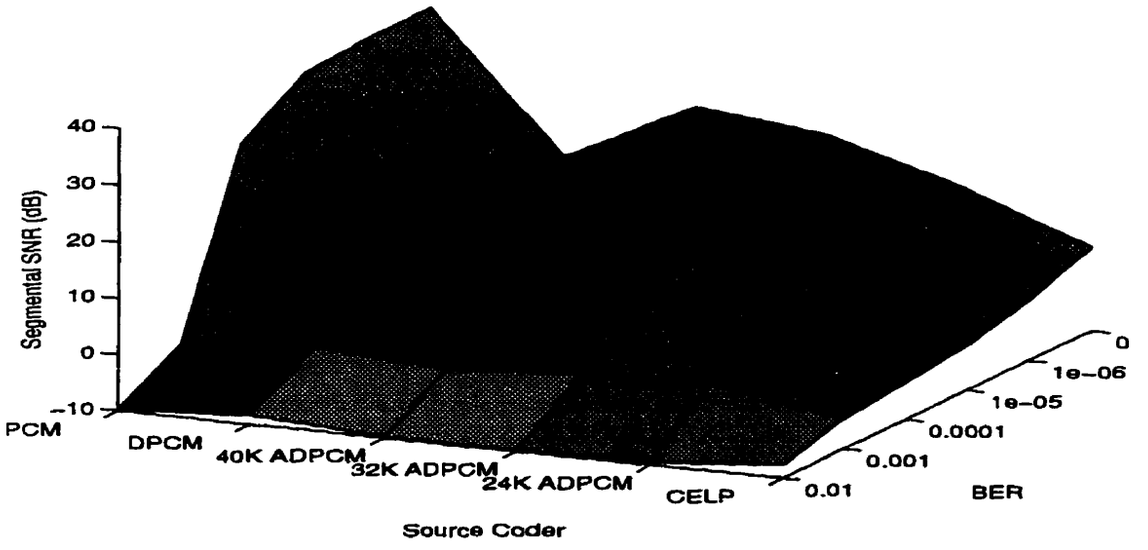


Figure 7.3: Objective Quality Measures under 64 Bit Block Encryption

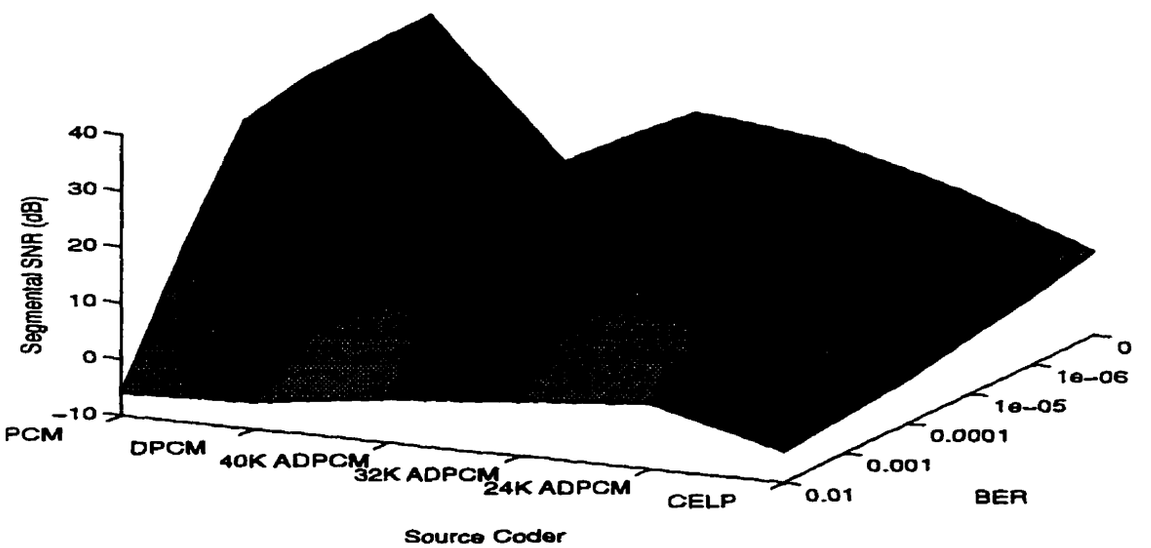


Figure 7.4: Objective Quality Measures under Stream Encryption

In figure 7.5 we present a summary of the objective quality data as a function of redundancy and bit error rates under 64 bit block encryption. In figure 7.6 we present a similar summary under stream cipher encryption conditions. Four different views of the models are presented in each figure to allow a more detailed examination of the surface features.

Figures 7.5 and 7.6 will comprise our models of objective speech quality as a function of three parameters: redundancy, encryption mode, and bit error rate. We define the quality model for the objective function as $SegSNR(D, \text{E.C.}, \text{BER})$, where D is the redundancy rate, E.C. indicates the encryption coder type, and BER indicates the bit error rate at the input to the encryption coder.

Source Coder Complexity

The source coder complexity function can be represented by the approximate relative complexities for the representative source coders tabulated in table 7.1. We choose to map the set of source coders to the appropriate redundancy measures using the redundancy mea-

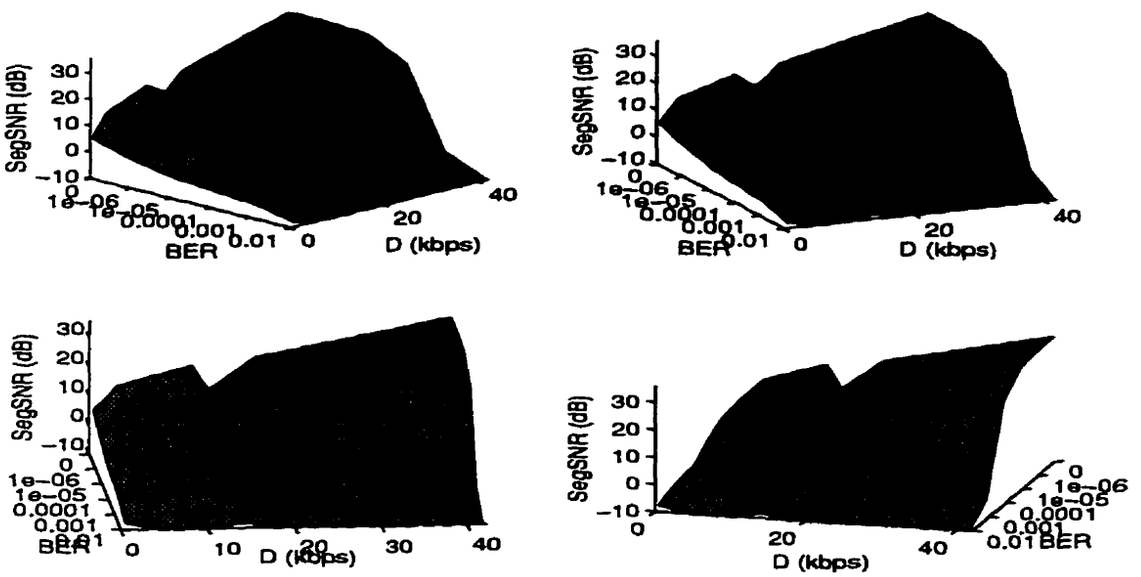


Figure 7.5: Redundancy Dependent Objective Quality Measures under 64 Bit Block Encryption

sures also summarized in table 7.1. We define the source coder complexity for the objective function as $C_{S.C.}(D)$, where D is the redundancy rate.

Encryption Coder Complexity

The encryption coder complexity was found to be purely dependent on the choice of encryption coder. Tabulated in table 7.2 are the relative complexities of the representative encryption coders used in this research. The encryption coder complexity will be defined for the objective function as $C_{E.C.}(\text{E.C.})$, to indicate the dependence on the controlling parameter.

Channel Coder Complexity

Our simplified model of channel coder complexity was defined by equation (7.1) to be a linear function of the bit error rate experienced at the encryption coder. This function was defined as $C_{C.C.}(\text{BER})$, where BER represents the bit error rate.

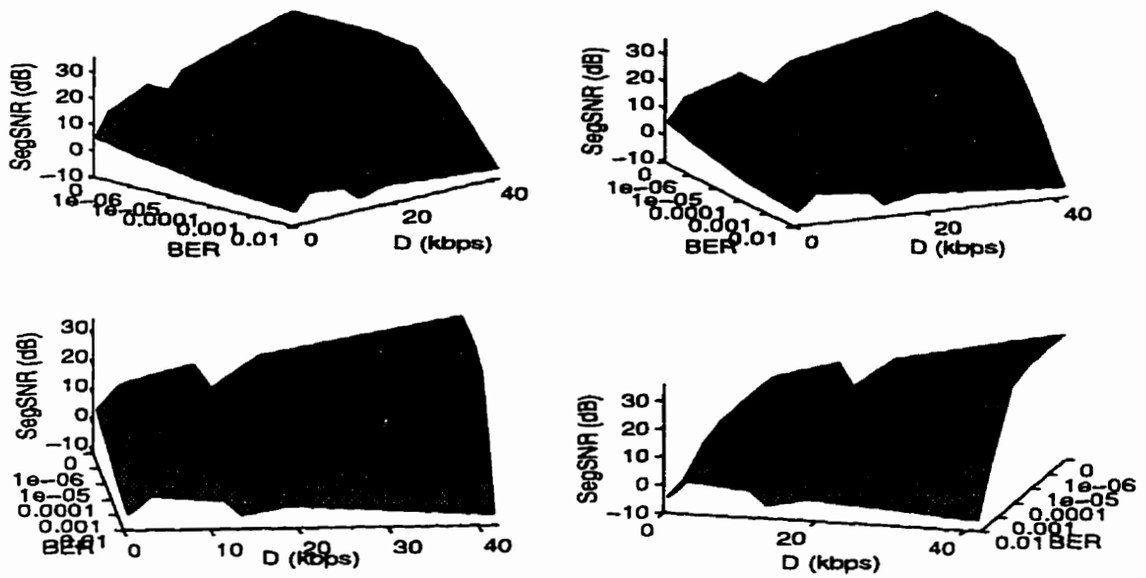


Figure 7.6: Redundancy Dependent Objective Quality Measures under Stream Encryption

7.4 Optimization Problem Formulation

The functional definitions presented in section 7.3.3 lead to the development of an objective function for the integrated secure speech communication system. We define the optimization problem over the parameters of redundancy, encryption coder, and bit error rate. We wish to perform a maximization of the objective function, I, as follows,

$$\begin{aligned} \max_{D,\text{E.C.},\text{BER}} \; & I = \gamma_1 \mathcal{E}(D, \text{E.C.}) + \gamma_2 \mathcal{Q}(D, \text{E.C.}) \\ & - \gamma_3 R(D) \\ & + \gamma_4 SegSNR(D, \text{E.C.}, \text{BER}) \\ & - \gamma_5 \mathcal{C}_{\text{S.C.}}(D) - \gamma_6 \mathcal{C}_{\text{E.C.}}(\text{E.C.}) - \gamma_7 \mathcal{C}_{\text{C.C.}}(BER), \\ & \text{s.t.} \quad \Phi(D, \text{E.C.}, \text{BER}) \quad \text{satisfied}, \end{aligned} \tag{7.2}$$

where $\gamma_i \geq 0, \;\; i = 1 \ldots 7$. The expression $\Phi(D, \text{E.C.}, \text{BER})$ denotes a set of constraints

on the system parameters. The unicity index has been omitted from the objective function because, as demonstrated in equation (4.9), it may be described in terms of the encryption quality and efficiency indices. We have chosen to describe the optimization problem using the distinct security indices of quality and efficiency, so it would not be appropriate to include another security function that could be derived from these indices. It is felt that these two measures are relevant to describing the security characteristics of a system, and have the added advantage of having been derived from experimental measures on the representative coders.

The objective function positively weights the security indices and the segmental-SNR quality model and negatively weights the bit rate and complexity models. This functional description acknowledges the general design goals for a secure communication system, which are to achieve high security and objective quality measures at a low cost in transmitted bit rate and coder complexity. Defining the objective function in this manner allows all of the weighting factors, γ_i, to be positive.

The optimal solution to this problem depends on the constraints, $\Phi(D, \text{E.C.}, \text{BER})$, that may exist on each of the system parameters, and the secure system designer's choice of weighting factors, γ_i. The weighting factors indicate the relative importance of each component in the secure system model and also serve to normalize the magnitude of the underlying measures so that parameters expressed in units of large magnitude do not "swamp" the features of smaller scale parameters. The general model has been defined as a linear combination of objective models. This requires that each parameter behave in a linear fashion as a function of the independent parameters. For models exhibiting non-linear behaviour it is necessary to include a linearizing function to ensure that normalization will be effective.

The constraints on the system design may consist of factors beyond the designer's control or may indicate desired bounds on performance. Examples of constraints on system parameters include the following,

1. **Bit Rate Limit** $R(D) \leq R_{\max}$. The secure communication system may be designed for a limited bandwidth environment. Given a maximum sustainable bit rate for the channel, the designer may need to constrain the source coder bit rate to a lower rate. One factor not considered in the model is the bit rate expansion caused by the channel coding operations. Allowance for this expansion must be considered in determining a maximum coder bit rate. In the future, it would be desirable to include a parameter describing channel coder bit rate expansion in the objective function.

2. **Objective Quality Bounds** $S_{\min} \leq SegSNR(D, \mathbf{E.C.}, \mathbf{BER}) \leq S_{\max}$. The designer may have a requirement for a minimum level of objective quality, say $S_{\min}$ dB. It may also be considered acceptable to provide service below some maximum quality level so the designer may seek to obtain more flexibility in the design by limiting the requirement on objective quality to $S_{\max}$ dB.

3. **Security Requirements** $\mathcal{E}(D, \mathbf{E.C.}) \geq e, \quad \mathcal{Q}(D, \mathbf{E.C.}) \geq q$. The designer may define certain minimum encryption security requirements. An example may be to provide at least the level of encryption quality afforded by the DES ECB mode cipher and a higher level of encryption efficiency. It would be desirable to extend the number of representative encryption algorithms to include the ciphers discussed in section 6.2. We may also wish to indicate the desired type of cryptographic synchronization protocol. These options remain objectives for future research.

4. **Complexity Limits** $C_{\mathbf{S.C.}}(D) + C_{\mathbf{E.C.}}(\mathbf{E.C.}) + C_{\mathbf{C.C.}}(\mathbf{BER}) \leq C_{\max}$. The combination of source, encryption, and channel coders may be subject to a maximum feasible complexity, or individual coders may be constrained separately.

A set of constraints, $\Phi(D, \text{E.C.}, \text{BER})$, which must be satisfied in solving the optimization problem will consist of some subset of these examples.

7.5 Optimization Problem Solutions

In this section we will evaluate the optimization problem for a variety of design constraints and objectives. We will first examine the effects that the relative weighting assigned to each parameter may have on the unconstrained system. We will then present examples of the same effects under some chosen constraints. Finally, we will define two design problems that are representative of the sort faced by secure speech system designers and present solutions to the design problems based on the general model.

To solve the optimization problem we must convert it to a simpler single-mode form. This may be accomplished by evaluating the objective function, I, over the space defined by all of our representative source, encryption, and channel coders. Specifically, we compute each of the 7 parameters measures in the objective function at specific points in the space consisting of 6 source coders, 5 encryption coders, and 6 bit error rate channel conditions, or over a subset of that space as defined by the constraint condition $\Phi(D, \text{E.C.}, \text{BER})$. In development of the objective parameter models we have either compiled experimental measurements at each of the required points, or derived an algebraic model that may be evaluated at the points. In this simplification the optimal system is considered to be the source, encryption, and channel coder combination which yields the largest value of I.

7.5.1 Objective Model Weightings

In defining the objective function, I, of equation (7.2) we described the quantities, γ_i, as factors for normalizing the models and for indicating the relative importance of each model to the system solution. We can distinguish these two factors by defining,

$$\gamma_i = w_i \alpha_i, \quad i = 1 \ldots 7,$$

where α_i is the normalizing factor and w_i is the weighting factor for the parameter.

The weighting factors, w_i, are determined by the system designer's objectives, while the

normalizing factors are uniquely determined by the average value of the objective parameter over the subset of points defined for the optimization problem.

For an objective parameter, $P_i(D, \text{E.C.}, \text{BER})$, the normalizing factor is,

$$\alpha_i = \frac{1}{\underset{\{D', \text{E.C.}', \text{BER}'\}}{E} [P_i(D, \text{E.C.}, \text{BER})]}, \tag{7.3}$$

where $\{D', \text{E.C.}', \text{BER}'\}$ represents the subset of points permitted in the optimization under the constraints indicated by $\Phi(D, \text{E.C.}, \text{BER})$.

We note that the method defined for parameter normalization is adaptive, depending on the design constraints. Normalizing over a subset may result in a greater sensitivity to variations in the objective parameters. This is likely to occur for objective parameters where some of the points in the unconstrained set are significantly higher or lower than the remainder of the set. Normalization in these unconstrained cases is correct in that the average measure in a normalized objective parameter set is comparable to that of the all other parameter sets, but small-scale variations in a set may become insignificant to the objective function. Normalization in a constrained case may yield significant variations in optimal results when the constraints exclude points that vary significantly from the remainder of the parameter observations. This is a desirable characteristic for an accurate optimization solution.

Normalization was performed according to equation (7.3) for all optimizations performed on the general model.

7.5.2 Unconstrained General Optimization Problem Solutions

Due to the multidimensional problem definition and variable weightings that may be applied to each parameter in the model, there are an enormous number of possible solutions to the optimization problem.

To characterize the behaviour of the general model we present here a summary of the

solutions obtained in an unconstrained problem definition when one choice of parameter weighting is varied while the rest remain constant. Presented in table 7.3 is a summary indicating the parameter studied, and the optimal problem solution obtained for various ranges of the parameter weighting, w_i. Weights were assigned relative to a unit weight for the fixed parameters.

Objective Parameter	Relative Weight w_i	Optimal System
Objective Quality	[0.00, 0.71]	{24K ADPCM, ECB, 10^{-6}}
	[0.72, 10.0]	{PCM, ECB, 0}
Bit Rate	[0.00, 1.56]	{PCM, ECB, 0}
	[1.57, 4.92]	{24K ADPCM, ECB, 10^{-6}}
	[4.94, 10.0]	{CELP, ECB, 0}
Encryption Efficiency	[0.00, 0.04]	{24K ADPCM, Vernam, 10^{-6}}
	[0.04, 10.0]	{PCM, ECB, 0}
Encryption Quality	[0.00, 1.65]	{PCM, ECB, 0}
	[1.67, 10.0]	{CELP, ECB, 0}
Source Coder Complexity	[0.00, 0.47]	{CELP, ECB, 0}
	[0.49, 99.9]	{PCM, ECB, 0}
Encryption Coder Complexity	[0.00, 0.02]	{PCM, CFB, 0}
	[0.03, 1.90]	{PCM, ECB, 0}
	[1.91, 10.0]	{24K ADPCM, Vernam, 10^{-6}}
Channel Coder Complexity	[0.0, 210]	{PCM, ECB, 0}
	[211, 500]	{24K ADPCM, Vernam, 10^{-2}}

Table 7.3: Optimal Solutions for Unconstrained Problem Definitions

The optimal solutions generated by the model consist of a choice of source coder, encryption coder, and channel bit error rate. The existence of a unique mapping from redundancy to source coder in table 7.1 allows us to interpret the optimal solution for redundancy as a unique source coder. These solutions are consistent with the definition of the objective function in equation (7.2) where optimization is performed over the space defined by the representative source and encryption coders and simulated channel conditions.

The results of table 7.3 reveal many of the expected characteristics for the general optimization problem. We can summarize our observations on the effect of each objective parameter in the general model as follows,

- **Objective Quality.** There are two modes of operation recommended by the unconstrained model. When quality is considered relatively unimportant, the best system includes the lower quality 24 kbps ADPCM coder and allows a BER of 10^{-6}. Higher relative quality objectives yield an optimal system recommendation of highest quality 64 kbps μ-law PCM and no bit errors.

- **Bit Rate.** The unconstrained model indicated three solutions over a range of weightings. At a low relative bit rate weighting, when the cost of bandwidth is not important to the designer, the 64 kbps μ-law PCM coder was recommended. Higher relative weightings reduced the system bandwidth, first to the 24 kbps ADPCM coder, and finally to 4.8 kbps CELP coding.

- **Encryption Efficiency.** These results indicated the effect that the relative importance attached to encryption efficiency had on the recommended system. The low weight range recommended a 24 kbps ADPCM coder and the perfect, but inefficient, cryptographic protection of the Vernam cipher. Higher weightings quickly discounted Vernam ciphers and recommended the more efficient ECB mode DES cipher.

- **Encryption Quality.** When the level of cryptographic protection required in the system was unimportant, the model allowed PCM encoding and an ECB mode cipher. It is interesting to note that higher encryption quality weightings resulted in a transition in the source coder type, but not in the encryption coder type. In this case we can explain the transition by noting that the ciphertext entropy results measured for CELP coded speech were equally good for all encryption coders, and considerably better than the results for PCM coded speech. There were additional factors in the general model that contributed to the choice of ECB mode rather than another cipher type.

- **Source Coder Complexity.** Optimal solutions included CELP coding when the

cost of coder implementation was deemed unimportant. A higher cost sensitivity causes the transition to the low complexity PCM format.

- **Encryption Coder Complexity.** The DES CFB mode attains preferred status when encryption coder complexity is deemed unimportant by the system designer. Higher cost sensitivities result first in a recommendation for the slightly less complex ECB mode, and finally require the minimal complexity Vernam cipher. It is interesting that the transition to Vernam encryption is accompanied with a recommendation to use 24 kbps ADPCM source coding instead of PCM. In this case, the lower redundancy in 24 kbps ADPCM contributes sufficiently to higher evaluations of the bit rate and encryption quality models to compensate for the increase in source coder complexity.

- **Channel Coder Complexity.** As may be expected, an increased weighting on the cost function for the channel coder results in an increase from a recommended zero bit error rate to a rate of 10^{-2} for an optimal system implementation. Another interesting result is revealed by the recommended reduction in bit rate to 24 kbps ADPCM and a change to the Vernam stream cipher from the 64 bit DES ECB block cipher. This transition was motivated by the higher objective quality measures obtained by stream ciphers, compared to block ciphers, under noisy channel conditions.

Clearly, the dependencies in the model are complex and non-linear and result in optimal system solutions that could not be predicted by simple observations. We note that often factors other than the primary weighting contributed to the choice of one coder over another in table 7.3.

The solutions to the unconstrained optimization problem include only 3 of the 6 possible source coders, 3 of the 5 encryption coders, and 3 of the 6 possible channel coders represented by bit error rates. The results should not imply that these are the only solutions to the problem. These are simply the most attractive solutions over the unconstrained set of coders. In section 7.5.3 we will observe the presence of many other coders in the optimal

solution when constraints are added to the objective function.

7.5.3 Constrained General Optimization Problem Solutions

Under constrained operational conditions we observed coders in the optimal solutions that were not selected under unconstrained conditions. This was caused by the exclusion of some previously attractive systems by the constraint conditions, and the effect of new normalization factors on the measures remaining in the subset of possible solutions.

We wish to demonstrate how the addition of constraints affects the set of optimal solutions presented in table 7.3. We define a sample constraint to be,

$$\Phi(D, \text{E.C.}, \text{BER}) = \{R(D) \leq 40 \text{ kbps}, \quad \text{E.C.} \neq \text{ECB}\}.$$

This immediate effect of this constraint is to exclude the μ-law PCM source coder and the DES ECB mode cipher from the solution set.

To characterize the behaviour of the general model under this constraint we computed optimal solutions over the same range of relative parameter weightings used for the unconstrained results of table 7.3. These constrained results are presented in table 7.4.

We note that the addition of constraints to the problem produced a variety of new solutions. Adding constraints that excluded two of the more common optimal solution components, PCM and ECB coding, increased the number of alternative systems suggested by the model. The constrained system recommended CELP, DPCM, 24K, 32K, and 40K ADPCM source coders, and each of the CFB, CBC, OFB and Vernam ciphers.

Given the large number of variables and potential constraint conditions, it is clear that any of the source, encryption, and channel coding solutions may be obtained by solution of the appropriate optimization problem. Even the simple DPCM coder, which is clearly inferior in quality for a given bit rate, was obtained in an optimal solution.

Objective Parameter	Relative Weight w_i	Optimal System
Objective Quality	[0.00, 0.73]	{24K ADPCM, CBC, 10^{-6}}
	[0.74, 0.83]	{32K ADPCM, CBC, 0}
	[0.84, 10.0]	{40K ADPCM, OFB, 10^{-6}}
Bit Rate	[0.00, 1.23]	{40K ADPCM, OFB, 10^{-6}}
	[1.24, 1.93]	{32K ADPCM, CBC, 0}
	[1.94, 4.00]	{24K ADPCM, CBC, 10^{-6}}
	[4.01, 10.0]	{CELP, CFB, 0}
Encryption Efficiency	[0.00, 0.15]	{24K ADPCM, Vernam, 10^{-6}}
	[0.16, 10.0]	{40K ADPCM, OFB, 10^{-6}}
Encryption Quality	[0.00, 1.65]	{40K ADPCM, OFB, 10^{-6}}
	[1.67, 10.0]	{CELP, CFB, 0}
Source Coder Complexity	[0.00, 0.47]	{CELP, CFB, 0}
	[0.49, 37.0]	{40K ADPCM, CFB, 10^{-6}}
	[38.0, 99.9]	{DPCM, CBC, 0}
Encryption Coder Complexity	[0.00, 1.85]	{40K ADPCM, OFB, 10^{-6}}
	[1.87, 10.0]	{24K ADPCM, Vernam, 10^{-6}}
Channel Coder Complexity	[0.0, 230]	{40K ADPCM, OFB, 10^{-6}}
	[231, 500]	{24K ADPCM, Vernam, 10^{-2}}

Table 7.4: Optimal Solutions for Constrained Problem Definitions

7.5.4 Practical Secure System Design Examples

We wish to demonstrate the use of the general model and objective function for designing optimal secure communication systems. We present here two examples where we utilize the general model to obtain an optimal solution to a constrained system design problem.

Example 1

A system designer has a requirement to build a secure communication system in a constrained environment. Specifically, the designer is concerned with the complexity of the integrated system as it must be implemented in a very small area as part of a larger IC design. The communications channel available to the system is capable of supporting up to a maximum of 40 kbps. The designer, being aware of the risks inherent in electronic codebook mode encryption, requires an encryption quality greater than that offered by ECB mode

DES. In addition, key management costs are a consideration in the overall design. The key management system can tolerate relatively long keys, but is not sufficient for supplying the bandwidth required for Vernam ciphers.

This problem statement defines the constraints required for objective function formulation, and implies a set of weighting criteria for the objective parameters. The statements regarding bit rate, encryption quality and efficiency describe the following constraints,

$$\begin{aligned}\Phi(D, \text{E.C.}, \text{BER}) = \{&R(D) \leq 40 \text{ kbps}, \\ &\mathcal{Q}(D, \text{E.C.}) > \mathcal{Q}(D, \text{ECB}), \\ &\mathcal{E}(D, \text{E.C.}) > 0\}.\end{aligned}$$

The problem description implies that the complexity parameters are of the greatest importance in the design. No mention is made of the level of quality expected in the design, so we will assume this is of only medium importance. It also appears that as long as the constraints are satisfied, the designer will be content with any level of bit rate, encryption efficiency and encryption quality. We can assign these remaining parameters relatively low weightings.

Weights were assigned to the objective parameters according to the preferences indicated above. Two schemes for the assignment of relative weightings were explored. The weightings of the high, medium, and low importance groupings in the two schemes were assigned as shown in table 7.5,

Relative Importance	Relative Weight	
	Scheme 1	Scheme 2
High	5	10
Medium	1	1
Low	0.5	0.1

Table 7.5: Alternate Weighting Schemes

Under both weighting schemes the optimal system recommendation was for a 40 kbps

ADPCM source coder, an output feedback mode DES cipher, and a channel coder yielding a bit error rate of 10^{-6}.

The sensitivity of both schemes to variations in the relative parameter weightings was examined to confirm the choice of system. Under the first weighting scheme, variations in the relative weightings of each importance grouping indicated a very stable optimal solution. The results, summarized in table 7.6, indicate no recommended change in optimal system, except when the low-importance parameters were disregarded completely.

Importance Group	Relative Weight w_i	Optimal System
Low	[0.00, 0.00]	{40K ADPCM, CFB, 10^{-6}}
	[0.01, 1.00]	{40K ADPCM, OFB, 10^{-6}}
Medium	[0.50, 5.00]	{40K ADPCM, OFB, 10^{-6}}
High	[1.00, 10.0]	{40K ADPCM, OFB, 10^{-6}}

Table 7.6: Sensitivity Analysis of Example 1 with Scheme 1 Weightings

The second weighting scheme was found to be almost equally stable, except for a transition to DPCM coders when the weighting on the medium importance parameters was reduced to the level of the low importance parameters. These results are summarized in table 7.7.

Importance Group	Relative Weight w_i	Optimal System
Low	[0.00, 0.00]	{40K ADPCM, CFB, 10^{-6}}
	[0.01, 1.00]	{40K ADPCM, OFB, 10^{-6}}
Medium	[0.00, 0.22]	{DPCM, CBC, 0}
	[0.23, 10.0]	{40K ADPCM, OFB, 10^{-6}}
High	[1.00, 20.0]	{40K ADPCM, OFB, 10^{-6}}

Table 7.7: Sensitivity Analysis of Example 1 with Scheme 2 Weightings

We conclude that the constraints and weightings defined for this problem lead to a relatively stable optimal solution. The designer may implement the recommended system with confidence that it will achieve the optimal performance for the stated objectives and conditions.

Example 2

The secure system designer's first project was so successful that he was asked to produce another integrated system, this time as a discrete IC device. Given the entire chip area to work with, the complexity of the secure system is no longer important to the designer and he believes he can support any of the possible coder combinations contained in the general model. The new design is intended to support communications quality speech at a maximum rate of 32 kbps, but would be more favourably received if it could operate with a lower bandwidth. The designer's security objectives and the key management structure remain the same as in the first design, so he will apply the same constraints to the new system formulation. Allowing a segmental-SNR measurement of 20 dB to define the minimum acceptable speech quality level, we may describe the set of constraints as follows,

$$\begin{aligned}\Phi(D, \text{E.C.}, \text{BER}) = \{&R(D) \leq 32 \text{ kbps}, \\ &\mathcal{Q}(D, \text{E.C.}) > \mathcal{Q}(D, \text{ECB}), \\ &\mathcal{E}(D, \text{E.C.}) > 0, \\ &SegSNR(D, \text{E.C.}, \text{BER}) \geq 20 \text{ dB}\}.\end{aligned}$$

The designer's objective being to provide high quality speech, we will consider this parameter to be of relatively high importance and weight it accordingly. The bit rate required for the system is a consideration in this design so we will also give the bit rate parameter high importance in our weighting schemes. The large chip surface available to the designer suggests that complexity is not an issue so we will give the complexity parameters low relative weightings. The remaining encryption quality and security parameters are considered of medium importance.

Unfortunately for the system designer, there were no solutions to the general model under the indicated constraints. This definition of the problem yielded no valid solutions

to the model for either of the two relative weighting schemes summarized in table 7.5. Closer examination of the problem revealed that the primary conflict was between the bit rate constraint and the minimum quality constraint. Since both of these parameters were considered of high importance by the designer, the choice of an alternate relative weighting scheme could not alleviate the conflict.

This analysis resulted in the designer's decision to re-evaluate and refine his design goal using results from the general model. The designer found he had two options, the first was to reduce the minimum acceptable segmental-SNR level, and the second was to increase the allowable bandwidth of the system. The designer found that valid solutions could be obtained by reducing the minimum acceptable segmental-SNR level from 20 dB to 19.8 dB, or by increasing the bandwidth of the system to 40 kbps. In consultation with the project manager it was decided that the small degradation in signal quality would be preferable to the relatively large increase in bandwidth required for the other solution.

A sensitivity analysis was performed on the model under the refined constraints for both of the relative weighting schemes. The results obtained under weighting scheme 1, presented in table 7.8, and the results under scheme 2, presented in table 7.9, reveal a highly stable optimal solution consisting of a 32 kbps ADPCM source coder, a cipher block chaining mode DES cipher, and a channel coder capable of providing perfect transmission.

Importance Group	Relative Weight w_i	Optimal System
Low	[0.00, 1.00]	{32K ADPCM, CBC, 0}
Medium	[0.50, 5.00]	{32K ADPCM, CBC, 0}
High	[1.00, 10.0]	{32K ADPCM, CBC, 0}

Table 7.8: Sensitivity Analysis of Example 2 with Scheme 1 Weightings

Importance Group	Relative Weight w_i	Optimal System
Low	[0.00, 1.00]	{32K ADPCM, CBC, 0}
Medium	[0.00, 10.0]	{32K ADPCM, CBC, 0}
High	[1.00, 20.0]	{32K ADPCM, CBC, 0}

Table 7.9: Sensitivity Analysis of Example 2 with Scheme 2 Weightings

This second example demonstrates how the general model can be a valuable tool for developing secure communications systems. It allows decisions on the refinement of system objectives and constraints to be based on the quantifiable results provided by the general model.

7.6 Observations on the General Model

Most of the parameters comprising the general model were developed by means of experimental measures performed on representative sets of coders. An exception, the complexity parameters, were developed using a variety of approximation techniques and general observations. The accuracy of the general model could be improved by several means. The first improvement would be to increase the representative coder sets to include a wider variety of coders. This would reduce the granularity of observations in the regions of interest to the system designer.

If, for instance, the designer could obtain quality and redundancy measures for a source coder not included in the representative set, these could supplement the quality and bit rate models. If the suite of ciphertext entropy measures could not be performed for this coder, the designer could still obtain approximate measures by interpolating the security models at the level of redundancy measured for the new source coder.

An additional improvement would be to increase the accuracy of each of the parameter models contained in the general model. The manner in which this could be accomplished is dependent on the nature of the individual measurements. The objective quality measures may be improved slightly by additional measurements, but would benefit most from a more specific description of the channel conditions. As was noted in chapters 5 and 6, the bit rate and security models could be improved by higher order probability measures on the underlying processes. As we have already noted in section 7.2, the complexity models, particularly the channel coder model, would all benefit from a more sophisticated

development.

The general model is an effective tool for exploring the design alternatives in a constrained secure communication environment, despite the opportunity for increasing its accuracy by additional experiments. The data contained in the models is as accurate as it was feasible to compute, and is truly indicative of the behaviour of the alternatives subjected to study. Our choice of representative coders spanned a broad range of interest, so the results contained in the model will represent approximate solutions that are relevant to a variety of specific system implementations.

Chapter 8

Contributions, Conclusions, and Future Research

8.1 Contributions

In chapter 2 we presented a method for high order conditional entropy estimation. This technique allows the calculation of conditional entropy measures that are infeasibly complex for the conventional direct calculation method. The method has the additional advantage of being partitionable in time and across multiple processing units.

In addition to the estimation algorithm, we identified opportunities for increased efficiencies in its application to real source processes. An adaptive sequential Monte Carlo sampling algorithm was presented as an efficient method of obtaining estimates of various information measures for a variety of source processes.

While the description of the entropy estimation algorithm is, in our view, quite straightforward, application of the algorithm to a large entropy estimation problem was found to be a challenging task. In chapter 3 we presented the results of our experimental work towards an estimate of the entropy rate of the speech process. The results of chapter 3 included

measures of the first to sixth order conditional entropy, studies of the convergence properties of these measures, and the development of conditional entropy surfaces for all model orders. All of the experimental PCE data computed in this study has been retained and can be applied in future studies to increase the accuracy and reduce the computational cost of additional conditional entropy measurements.

Applying non-linear regression analysis to fit an appropriate model to the first to sixth order conditional entropy measurements, we obtained an estimate of the entropy rate of the speech process to be 22.3 kbps $\pm$ 7 kbps with 95% confidence.

In chapter 4 we presented an argument for the development of theoretic measures of the security of encryption systems and proposed two unique indices for use in the general model of the secure speech system. We also defined in Proposition 2 a set of design criteria for cryptosystems to ensure the relevance of the theoretic measures.

Chapter 5 presented models of objective quality and bit rate as a function of parameters under the control of a secure system designer. We examined the segmental-SNR performance of a class of representative source coders under a variety of channel conditions. Of particular importance to the general model was the development of measures of the impact of channel errors under cryptographic conditions.

In developing a model of bit rate we computed measures based on the informational divergence in the coded speech process. These studies resulted in a well-supported analytic model relating bit rate to the level of redundancy in the source coded signal.

In chapter 6 we defined a small set of representative encryption techniques meeting Proposition 2 and performed experiments to evaluate the theoretic security indices. This work combined the results of our entropy estimation work and redundancy modelling to produce experimental models of cryptographic security.

In chapter 7 we defined rudimentary models for the complexity of source, encryption, and channel coders. These simple models added an essential dimension in the analysis of secure communications systems.

Finally, we combined the models of objective speech quality, bit rate, security, and complexity to define a general model of the secure communication system. Chapter 7 outlined the sets of objective, intermediate, and controlling parameters in the system and indicated the general interrelationships between them. We then formulated an optimization problem for secure system evaluation based on a general constrained objective function over the parameters of redundancy, encryption, and bit error rate.

Due to its non-linearity and high dimensionality, it is difficult to observe all facets of the resulting model. We performed studies of the sensitivity of the model to changes in the design objectives and constraints in order to demonstrate its general characteristics. We then demonstrated the use of the model with two examples where the model was applied to determine an optimal secure communication solution under typical design constraints.

8.2 Conclusions

The entropy estimation work presented in chapters 2 and 3 revealed some general characteristics of the estimation technique. Our experimental work found the high order conditional entropy surfaces to be relatively smooth and highly peaked around the mean of the process. This form of entropy surface was particularly suitable to efficient sampling and interpolation by the Monte Carlo selection procedure, allowing a good estimate of the entropy rate to be computed from a relatively small number of PCE measures.

In general, we found characterizing the shape of the entropy surface to be a critical step in efficient entropy estimation. A clear understanding of the entropy surface of a process would allow a better *a priori* sampling distribution and increase the accuracy and efficiency of the entropy estimate.

The experimental work performed in chapter 5 revealed a complex and non-linear relationship between the objective quality measure and the parameters of redundancy, bit error rate, and encryption coder type. The experimental results determined under noiseless con-

ditions indicated a linear relationship between bit rate and quality. Under noisy encrypted conditions, however, the relative quality obtained by the representative coders was found to exhibit more complex characteristics that were best represented by the sample observations alone. This research clearly indicated the non-trivial nature of secure system design as it related to the quality measure.

The model of bit rate as a function of redundancy, also presented in chapter 5, was found to conform to the theoretic bounds on performance and to meet our intuitive expectations. We noted that the data supporting the operational rate-redundancy function of equation (5.20) could be improved by computing higher order divergence measures.

In developing the experimental models of cryptographic security in chapter 6 we discovered the characteristics of encrypted speech to differ significantly from unencrypted speech. Due to the randomizing effects of encryption, modelling ciphertext entropy was found to be a difficult activity. A significant increase in the required state-space for a ciphertext entropy calculation made it necessary to employ the entropy estimation technique at a lower model order and to supply much larger volumes of sample data than were required for unencrypted speech. We noted that these difficulties could be addressed with an increase in the time and storage available for the experiments.

It was difficult to distinguish the security results of the representative encryption coders on the basis of third order entropy measures. Higher order measures would reveal larger variations between the coders and improve the accuracy of the security components in the general model.

Most of the parameters comprising the general model were developed by means of experimental measures performed on representative sets of coders. An exception, the complexity parameters, were developed using a variety of approximation techniques and general observations. We could improve the accuracy and relevance of the general model by increasing the representative coder sets to include a wider variety of coders. This would reduce the granularity of observations in the regions of interest to the system designer.

The general model is an effective tool for exploring the design alternatives in a constrained secure communication environment, despite the opportunity for increasing its accuracy by additional experiments. The data contained in the models is as accurate as it was feasible to compute, and is truly indicative of the behaviour of the alternatives subjected to study. Our choice of representative coders spanned a broad range of interest, so the results contained in the model will represent approximate solutions that are relevant to a variety of specific system implementations.

While the modelling methodology and entropy estimation algorithms were applied in this research to the specific case of a secure speech communication system, they may also be applied to a wider class of sources possessing high levels of redundancy, including video and facsimile signals.

8.3 Future Research Topics

There were a number of topics identified as being of interest for further research. Some of the major topics are noted below,

- In developing estimates of the probability distributions of the speech process we encountered a variety of difficulties due to the large volume of speech data being processed and the long execution times required of our experiments. These problems precluded a constant and maximal sample size for the speech process under study. The following are a few areas in our experimental work where we noted the opportunity for improvements that would enhance the accuracy and confidence interval of the predicted entropy rate,

    1. More accurate measures of the conditional entropy of the speech process could be obtained by computing all PCE vectors from a single large sample set. A larger sample set would improve the convergence of PCE vectors outside the

more probable regions, and a constant sample set size would eliminate the need to normalize the PCE measures with the set of computed convergence surfaces. This procedure would require the use of larger, and more reliable, computing facilities than were available for our initial prediction.

2. Were this not feasible, we could still increase the accuracy of our results by improving the resolution of the set of convergence surfaces. This could be achieved by expanding the set of convergence measures to include a larger number of points over the calculation space. This process is feasible with the available computing resources and simply requires the expenditure of a greater amount of time to collect the convergence data.

3. The accuracy of the entropy measure could also be improved by including more PCE vectors on the entropy surface. A larger number of PCE vectors would directly improve accuracy by providing a higher resolution set of measures for interpolating the complete entropy surface.

- We have noted that the Kullback-Leibler distance measures used to develop the operational rate-redundancy function could not be computed to as high a model order as might be desired. As a result, significant differences in the original and distorted waveforms may not have been detected by our measures, and the divergence rate measures of figure 5.15 may be quite conservative.

  It would be desirable to obtain fourth, fifth, and sixth order divergence measures in order to verify the lower-order results and more accurately measure the divergence of the CELP coder. In future work it should be possible to extend the divergence measure to higher model orders using a modified form of the entropy estimation technique presented in chapter 2. The estimation procedure could be modified by partitioning the generalized Kullback-Leibler distance measure into independent partial divergence

measures,

$$d^{(n,j)}(p||q) \triangleq \sum_{x_{j+1}^n \in \mathcal{X}^{n-j}} p(x_1^n) \log_2 \frac{p(x_1^n)}{q(x_1^n)} \quad \text{for } n > j \geq 1.$$

Selecting the vectors $x_{j+1}^n \in \mathcal{X}^{n-j}$ by a Monte Carlo procedure, we can compile a set of partial divergence measures with which to interpolate the complete nth order divergence surface.

- In the future it would be desirable to extend the order of the conditional entropy calculation applied to the security models. Higher order measures could be obtained by applying the entropy estimation technique to a much larger source coded database. There is no theoretical impediment to these calculations, but many of the practical problems of managing this large experimental effort would have to be addressed.

- In future work, it would be desirable to examine in detail the effects of maintaining cryptographic synchronization on the model parameters. The cryptographic synchronization protocol may be designed with a variety of security objectives, ranging from a non-recovering high-security model to a robust self-synchronizing type. These choices affect the parameters of complexity, objective quality, bit rate, and security.

- Finally, a more sophisticated treatment of the complexity models, particularly for channel coders, would expand the abilities of the general model. Many of the characteristics related to the channel were idealized in our work. More accurate models of channel error rates and distributions, message expansion, and delay would contribute greatly to the general system model.

Appendix A

Theoretic Entropy Rate of the Speech Process

Theorem 1 (Shannon-McMillan-Breiman theorem) *For stationary ergodic processes,*

$$\frac{1}{n}\log p(X_0,\ldots,X_{n-1}) \rightarrow \lim_{n\rightarrow\infty} E[-\log p(X_n|X_0^{n-1})] \rightarrow H \quad \textit{with probability 1.}$$

Proof: Algoet and Cover demonstrated in [99] a proof of the theorem by sandwiching the non-ergodic sequence $p(X_i|X_0^{i-1})$ between the closely related ergodic quantities $p(X_i|X_{i-l}^{i-1})$ and $p(X_i|X_{-\infty}^{i-1})$. These more tractable quantities define two measures of interest,

$$\begin{aligned} H^l &= E[-\log p(X_l|X_{l-1},\ldots,X_0)] \\ &= E[-\log p(X_0|X_{-1},\ldots,X_{-l})] \text{ by stationarity,} \end{aligned}$$

an upper bound on the entropy rate and,

$$H^\infty = E[-\log p(X_0|X_{-1},\ldots)],$$

a lower bound.

Using Levy's martingale convergence theorem for conditional probabilities it was shown that

$$\begin{aligned}\lim_{l\to\infty} H^l &= \lim_{l\to\infty} E[-\sum_{x_0\in\mathcal{X}} p(x_0|X_{-l}^{-1})\log p(x_0|X_{-l}^{-1})] \\ &= E[-\sum_{x_0\in\mathcal{X}} p(x_0|X_{-\infty}^{-1})\log p(x_0|X_{-\infty}^{-1})] \\ &= H^\infty.\end{aligned}$$

Since $H^l \to H^\infty = H$, the sandwich is closed around the sequence of interest and it is proven to be asymptotically equal to the entropy rate. □

We can apply a minor modification to the Shannon-McMillan-Breiman theorem to derive the entropy rate of the speech process.

Theorem 2 (Entropy rate of the speech process) *Assuming the speech process to be ergodic as well as stationary with limited memory as defined in equation (3.1),*

$$H^l \to H^k \to = H^\infty = H \quad \textit{with probability 1.}$$

Proof: By equation (3.1) we have,

$$p(x_0|X_{-\infty}^{-1}) = p(x_0|X_{-k}^{-1}) \;\; \forall \;\; x_0 \in \mathcal{X}.$$

Therefore,

$$\begin{aligned}\lim_{l\to\infty} H^l &= \lim_{l\to k} E[-\sum_{x_0\in\mathcal{X}} p(x_0|X_{-l}^{-1})\log p(x_0|X_{-l}^{-1})] \\ &= E[-\sum_{x_0\in\mathcal{X}} p(x_0|X_{-k}^{-1})\log p(x_0|X_{-k}^{-1})] \\ &= H^k,\end{aligned}$$

and,

$$\begin{aligned} H^{\infty} &= E[-\sum_{x_0 \in \mathcal{X}} p(x_0|X_{-k}^{-1}) \log p(x_0|X_{-k}^{-1})] \\ &= H^k. \end{aligned}$$

So $H^l \rightarrow H^k = H^{\infty} = H$. Applying the characteristic of limited signal memory, we have shown that the entropy rate of the stationary ergodic speech process is bounded by the two tractable quantities and is equivalent to the $k+1$th order conditional entropy, $H(X_k|X_{k-1}, \ldots, X_0)$. For ease of use, we can apply the principle of stationarity to express the jth order conditional entropy in the form, $H(X_j|X_{j-1}, \ldots, X_1)$. □

Appendix B

The SWITCHBOARD Database

The NIST SWITCHBOARD database [15] was designed to address the need for a large multi-speaker database of telephone bandwidth speech. It consists of 2430 spontaneous conversations spoken by 542 different speakers from every major dialect region in the United States. The average conversation has a duration of 6 minutes, resulting in over 480 hours of recorded speech. The collection process was automated in order to minimize any effect of experimenter bias and to ensure uniformity throughout a long testing period. The initiation protocol was designed to encourage a natural and spontaneous conversation from the participants.

The data was collected directly from the telephone network's T1 lines and stored in 8 bit μ-law PCM format at 8 kHz for each of the two speech channels. This digital collection technique avoided any degradation due to the collection system. The speech quality of these recordings was determined solely by the speaker's speech patterns and the local telephone equipment.

We consider a conversation to be the vocal audio signal processed by a single speech coder at the network end-point. A single SWITCHBOARD file therefore consists of two conversations. Ideally we would screen out any conversations exhibiting non-speech sounds

or post-quantization noise added by the telephone network, but to do so would be infeasible for this experiment. The transcriber records included in the database provide subjective assessments of noise, echo and clarity, but unfortunately they were found to be unreliable because they were based upon an analog recording of the conversation and not the digital data itself. It is infeasible to reassess the database on a case-by-case basis so it was decided to admit all conversations to the model.

It is reasonable to assume that where these additional noises, σ, do occur in the process, they are uncorrelated with the signal. Their impact on the entropy rate measurement will therefore be additive such that our estimate will include the signal and the noise, $H(X + \sigma) = H(X) + H(\sigma)$. It should be stressed that these artifacts are rare and will contribute little error to our estimate of the entropy rate. In future work it may be possible to estimate and bound the magnitude of $H(\sigma)$ by computing the entropy of the recorded database during non-speaking segments.

Appendix C

SWITCHBOARD Database Transcription Examples

A wide variety of subjective responses were recorded for the SWITCHBOARD database [15] by the transcribers responsible for producing a time-alligned transcript of each conversation. The transcribers recorded subjective evaluations including the intelligibility of the conversation, the apparent naturalness of the conversation, echos on either communication channel, and noise in the form of static or non-speech contributions in the background. The degree of each characteristic was recorded as an integer measure between 1 and 4, with 1 being ideal and 4 being the worst performance. The transcribers made additional comments as they deemed necessary to describe additional information about each conversation.

As we have discussed in Appendix B, the subjective measures provided by the transcribers were found to be unreliable due in part to an inadequate testing methodology. The problem with the subjective evaluations arose because the transcribers were supplied with an analog recording of the conversations, and not the exact μ-law data supplied in the database. As a result of poor analog reproduction, the subjective evaluations of some recordings do not correspond to the actual quality levels. The second problem arose from

inconsistencies in subjective evaluations between individual transcribers. The results were not normalized in any way, so the conclusions reached by the various transcribers are not generally consistent.

We reproduce here several examples of the subjective evaluations accompanying the transcription records of speech samples used in our model of the speech process.

```
================================================================================

==> sw3805.txt <==
FILENAME: 3805_1490_1263
TOPIC#: 340
DATE: 920120
TRANSCRIBER: glp
DIFFICULTY: 1
TOPICALITY: 1
NATURALNESS: 2
ECHO_FROM_B: 1
ECHO_FROM_A: 2
STATIC_ON_A: 1
STATIC_ON_B: 1
BACKGROUND_A: 1
BACKGROUND_B: 3
REMARKS:        Speaker A dominated the conversation.  Speaker B
had a TV on in the background.

================================================================================

==> sw3806.txt <==
FILENAME: 3806_1477_1462
TOPIC#: 358
DATE: 920120
TRANSCRIBER: GLP
DIFFICULTY: 2
TOPICALITY: 1
NATURALNESS: 4
ECHO_FROM_B: 2
ECHO_FROM_A: 1
STATIC_ON_A: 1
STATIC_ON_B: 2
```

```
BACKGROUND_A: 4
BACKGROUND_B: 2
REMARKS:        Speakers (especially A) had a hard time keeping the
conversation going.  Speaker B seemed distracted toward the end of
the conversation.

================================================================================

==> sw3808.txt <==
FILENAME: 3808_1372_1023
TOPIC#: 354
DATE: 920120
TRANSCRIBER: glp
DIFFICULTY: 2
TOPICALITY: 1
NATURALNESS: 3
ECHO_FROM_B: 1
ECHO_FROM_A: 2
STATIC_ON_A: 2
STATIC_ON_B: 1
BACKGROUND_A: 2
BACKGROUND_B: 2
REMARKS:        Speaker B sounded as if he was using a speaker phone.

================================================================================

==> sw3810.txt <==
FILENAME: 3810_1253_1307
TOPIC#: 308
DATE: 920121
TRANSCRIBER: glp
DIFFICULTY: 1
TOPICALITY: 1
NATURALNESS: 2
ECHO_FROM_B: 2
ECHO_FROM_A: 2
STATIC_ON_A: 1
STATIC_ON_B: 1
BACKGROUND_A: 1
BACKGROUND_B: 1
REMARKS:        None.
```

```
==========================================================================

==> sw3811.txt <==
FILENAME: 3811_1477_1490
TOPIC#: 323
DATE: 920121
TRANSCRIBER: glp
DIFFICULTY: 3
TOPICALITY: 3
NATURALNESS: 3
ECHO_FROM_B: 2
ECHO_FROM_A: 1
STATIC_ON_A: 1
STATIC_ON_B: 2
BACKGROUND_A: 4
BACKGROUND_B: 2
REMARKS:        Speaker A had excessive background noise (i. e.
dishes, child, doors)

==========================================================================
```

Appendix D

Implementation Details of the Speech Models

The memory requirements for calculation of the PCE measurements were much too large for the computing resources at our disposal. In fact, full specification of any $O(256^4)$ calculation exceeds the practical capacity of a 32 bit address computing architecture. We used a group of 8 networked Sparc II workstations with real memory spaces ranging from 24 to 64 MBytes. Sample data was distributed to the set of modelling and entropy calculating processes on each machine by means of the network. The probabilistic models were maintained independently in the local memory of each machine and were not distributed across multiple processors. This separate modelling activity was made possible by the independent nature of PCE measure.

Computing each $O(256^4)$ PCE vector required a careful implementation of the data structure for the probabilistic model. Naturally, it was essential to take advantage of any redundancies in the source process to reduce the state-space of the model to a tenable size. This was accomplished with a dynamic storage routine that recorded data for only those 4-tuples observed in the processed data set.

We approached the implementation of the data structure with a dynamically linked list structure, indexed by a hash table computed from the stored data values. Originally, the

hashing function was designed to ensure roughly equal length in each of the linked lists. This type of balance was expected to minimize the time required to search and update the data structure, but was discovered to be very inefficient for storing the speech model. The poor performance of the original hashing function was caused by the combination of the stochastic nature of the speech process with the dynamic storage allocation scheme.

We found our original approach achieved a highly homogeneous distribution of probability data throughout virtual system memory. This resulted in poor performance when the data structure was large, as it was necessary to perform frequent swaps in and out of active memory to perform the entropy calculation. For large models, a significant fraction of execution time could be spent handling page faults.

These observations led to an improved hashing function. The new function was designed to maintain locality among consecutive data values, meaning that 4-tuples that differed only in the last data value were hashed to the same value. Our intent was to increase the probability that when performing the entropy calculation, consecutive data values would be found in the same page of memory. This design sacrificed some of the uniformity among list lengths, but resulted in a 25% reduction in the entropy computation time.

A second improvement was found to increase the entropy computation and modelling efficiency by an additional 25%. We found that despite the improvement in the logical locality of the linked list structure, many of the supposedly adjacent elements were still widely distributed across virtual memory. It became clear that an improved hashing function alone would not compensate fully for the combination of a stochastic input process with another essentially stochastic dynamic storage routine. The computers on which these processes were running had multi-user operating systems which support a variety of simultaneous tasks of varying priority. Our jobs, being long-running and computationally intensive, were run at a lower priority and consequently were vulnerable to preemption and frequent swapping out of active memory. As a result, while data elements may have been logically adjacent in the list structure, if they did not appear in the sample set within the

same time period they could be allocated to widely separated locations in virtual memory. Such separation would incur the same swapping time penalty that was encountered prior to increasing the logical locality of the list structure.

Our solution was to periodically store the entire list structure to disk and then retrieve it. By storing the data structure in logical order on disk we could increase the real memory locality of the list elements by a significant margin upon retrieval. Each time the data structure was retrieved it was re-allocated in logical order, performing an implicit sorting operation on the list elements. By balancing the time required to perform the store and retrieve the list structure with the improvement in data storage and retrieval operations, we achieved a 25% improvement in the time required for PCE calculations.

In total, the improvements in the hashing function and logical list order doubled the rate of the entropy estimation routines.

Appendix E

Work Estimate for Entropy Rate Prediction

In this section we will estimate the total number of operations required to produce the set of six conditional entropy measures used to predict the entropy rate. The normalization, interpolation, and non-linear regression procedures all contributed to the total work required for the calculation, but the dominant cost was represented by the modelling and entropy computation phases as these were repeated for every measurement.

In table E.1 we summarize the notation that will be used to denote the operations performed in computing the entropy estimates. We will differentiate our count of the total operations performed in computing the entropy estimates according to these operation types. Also in this table we include an estimate of the relative cost of each operation [100]. These relative weightings allow us to form a comprehensive estimate of the computation cost of our measurements. Throughout this section we will determine very conservative estimates of the operational complexity of our algorithms. Our final summation will provide a lower bound on the real computational cost of our work in predicting the entropy rate of the speech process.

Symbol	Operation	Relative Cost
L/S	Load,Store word	1 (Int)
<<	Bit shift	1 (Int)
==	Comparison	1 (Int)
=	Assignment	1 (Int)
+	Integer addition,subtraction	1 (Int)
×	Integer multiplication	5 (Int)
%	Modulus	10 (Int)
++	F.P. addition	1 (F.P.)
*	F.P. multiplication	1 (F.P.)
/	F.P. division	4 (F.P.)
log	Logarithm	10 (F.P.)

Table E.1: Symbols, Operations, and Relative Operation Costs

E.1 Modelling Cost

Each entropy calculation, whether the exact measure for first to fourth order, or a single $O(256^4)$ PCE measure, required the two steps of model estimation followed by entropy computation. The modelling process required an amount of work that was linear in relation to the length of the data set being analysed.

E.1.1 Direct Algorithms

Storage of the first and second order joint probability models required no more than a static array in memory. For each byte of input data we performed one integer addition and one load/store operation to update the data structure. The third and fourth order models were also implemented in static arrays, utilizing a complete set of 256 second and third order PCE models, respectively, to obtain exact entropy measures. For the third order measure we managed to compute 64 PCE models simultaneously on a single pass through the data set, while the fourth order model required 256 passes through the data. For complete PCE modelling, we performed comparison operations to identify the elements belonging in the PCE models being computed on a given pass, a single integer addition and 2 load/stores for every byte in the input sample set. Summarized in table E.2 are

the estimated computation costs for building the probability models required for the direct conditional entropy measures. These direct calculations were computed on a range of sample set sizes and were repeated on several disjoint sample sets to obtain confidence intervals. The number of these direct calculations is summarized in the convergence data in figure 3.5 and the observation counts in table 3.1. The product of the sample set sizes and the number of observations at each size yielded a measure of the total volume of sample data that was processed by the direct modelling routines.

Model Order	Obs. (min.)	Obs. (Bytes)	No. Passes	Operations per Byte (a_i)	Total Operations
1	18044	8.7×10^9	1	1 (+,L/S)	$8.7\times10^9(a_1)$
2	18044	8.7×10^9	1	1 (+,L/S)	$8.7\times10^9(a_2)$
3	15319	7.4×10^9	4	$\frac{1}{4}$ (1+,2 L/S),1(==)	$3.0\times10^{10}(a_3)$
4	12594	6.1×10^9	256	$\frac{1}{256}$ (1+,2 L/S),1(==)	$1.6\times10^{12}(a_4)$

Table E.2: Operations Count for Direct Modelling

E.1.2 Indirect Algorithms

The models used for fifth and sixth order PCE measures were implemented in dynamically allocated linked lists in memory. This structure required a hash value to be computed, a list to be traversed and one addition to be performed for every input byte. We maintained lists with an average length of 10 elements, so we can assume an average of under 5 traversal operations per list search. The hashing function was a permutation of a 3 byte word, reduced by the modulus of the hash table size. We can conservatively estimate the hash value calculation to require 24 bit shifting operations, and one modulus operation. Omitting the cost of initializing new list elements, and estimating a single comparison and 4 load/stores per traversal operation, we can estimate a total cost of 5 comparisons, 20 load/stores, 24 bit shifts, one integer addition, and one modulus operation for each access operation to the dynamic memory structure.

To further complicate the work estimate for the dynamic structure we note that only

those 4-tuples prefaced with the PCE vector of interest were stored in the data structure. The Rabin-Karp search routine was used to screen the input data stream, requiring several operations per sample set byte [101]. In the pseudocode excerpt shown in figure E.1 we detail the operations performed by the search routine. We note that a simple optimization will convert the multiplication operation, $x * 256$, to an 8 bit upshift of the integer x. We will conservatively estimate 1 comparison, 2 load/stores, 2 assignments, 2 integer additions, 1 integer multiplication, 1 bit shift, and two modulus operations per input byte.

```
search_pattern_hash = Compute_Search_Pattern_Hash(Search_Pattern);
data_hash = Compute_Initial_data_hash();

/* Slide data window to end of file */
while ( More_Data() ) {
    if (data_hash==search_pattern_hash) {
        if Check_For_Match(Search_Pattern,Data)
            Add_Data_Point(HashTable,Data);
    }

    /* Advance by one element and re-compute window value */
    Data_next = Read_Data();
    data_hash = (data_hash-(Data[last]*h %q))*256+Data_next)%q;

    /* Shift data window by one element */
    Shift_Data(Data,Data_next);

} /* End of while */
```

Figure E.1: Pseudocode Excerpt from Rabin-Karp Search Algorithm

As discussed earlier, the search algorithm served to screen out significant quantities of the input data set. In some cases as little as .01 % of the sample set was admitted to the storage routine. The dominant component in the computation cost is therefore the Rabin-Karp search algorithm. We will use this measure exclusively to estimate the cost of building the dynamic models.

Summarized in table E.3 is an estimate of the number of operations required to build the fifth and sixth order probability models for the entropy calculations. Convergence statistics were computed for these indirect measurements at a variety of sample set sizes and sample points as is summarized in figures 3.6 and 3.7. Also included is an estimate of the volume of data processed to generate the sets of K_1 and K_2 PCE vectors. For this calculation we used an average data set size of 3800 minutes for the 5th order calculation and 10000 minutes for the 6th order calculation. Summing the number of convergence observations with the product of the K_i's and their respective average data set sizes yielded an estimate of the total number of bytes processed by the indirect modelling routines. This allowed a conservative estimate of the number of indirect modelling operations.

Model Order	Converg. Obs. (min.)	PCE Obs. (min.)	Total Obs. (Bytes)	Operations per. Byte (b)	Total Operations
5	75950	155800	1.1×10^{11}	1(==,×,<<),	1.1×10^{11}(b)
6	185220	1010000	5.7×10^{11}	2(=,+,%,L/S)	5.7×10^{11}(b)

Table E.3: Operations Count for Indirect Modelling

E.2 Entropy Calculation Cost

The work performed in the entropy computation phase was largely independent of the size of the data sample sets and primarily determined by the order of the entropy calculation being performed. There were two algorithms used for the entropy calculations, one for direct measures which were stored in static arrays of memory, and one for measures that had to be stored in a dynamic structure.

The probabilistic models stored in both types of structure consisted of integer counts of all events observed in the sample space. Dividing each count by the total number of observations in the model would yield the probability measure, but it was more efficient to perform this operation at the end of the entropy calculation rather than for every element. For both types of algorithm, the probabilistic model was pre-processed in order to

compute conditional entropy measures from the joint probability distributions developed by the modelling process. This was accomplished by computing the marginal distributions of the model, in terms of integer counts, and dividing each joint probability count by that marginal count.

E.2.1 Direct Algorithms

In the pseudocode excerpt of figure E.2 we detail the operations required to compute the conditional probability measures and the direct 3rd order conditional entropy measure. We note that the static array design of the direct modelling approach allowed us to tabulate the marginal distributions, P[i][j], without significant cost. A very similar algorithm was used for all other direct entropy measures. From this code segment we estimate a total of 2 load/stores, 1 floating point (F.P.) addition, 1 F.P. multiply, 1 F.P. divide, and 1 logarithm operation for each iteration of the entropy calculation.

```
for ( all_i ){
  for ( all_j ){
    for ( all_k ){
      entropy -= P[i][j][k] * log( P[i][j][k] / P[i][j] );
    }
  }
}
entropy /= ( Total_count * log(2) );
```

Figure E.2: Direct Entropy Calculation Excerpt

Using these estimates we have summarized the number of operations required for all direct entropy measures in table E.4. We note again that the direct calculations were performed on a range of sample set sizes and were repeated on several disjoint sample sets to obtain confidence intervals, as is summarized in figure 3.5 and table 3.1.

Model Order	No. Iterations	Ops. per Computation (c_i)	Total Ops.
1	64	256*(1(++,*,/,log) 2(L/S))	64(c_1)
2	64	256^2*(1(++,*,/,log) 2(L/S))	64(c_2)
3	63	256^3*(1(++,*,/,log) 2(L/S))	63(c_3)
4	62	256^4*(1(++,*,/,log) 2(L/S))	62(c_4)

Table E.4: Operations Count for Direct Entropy Calculations

E.2.2 Indirect Algorithms

In the pseudocode excerpt of figure E.3 we detail the operations required to compute the conditional probability measures and a PCE measure of complexity $O(256^4)$. We must explicitly calculate the marginal distributions from the observations in the dynamic structure, resulting in an increase in the number of calculations performed per loop iteration. Assuming the same characteristics for list searches noted in the modelling cost section, we estimate the following operations being performed for each iteration of the indirect entropy calculation: 6 assignments, 10 comparisons, 42 load/stores, 48 bit shifts, 1 integer addition, 1 F.P. addition, 1 F.P. multiplication, 1 F.P. division, and 1 logarithm .

Using these estimates we have summarized the number of operations required for all indirect entropy measures in table E.5. We note again that convergence statistics were computed for the indirect calculations at a variety of sample set sizes and sample points as summarized in figures 3.6 and 3.7. We also computed $K_1 = 41$ and $K_2 = 101$ PCE vectors to form the entropy surfaces. Summing the number of convergence calculations with the number of data PCE calculations yielded the total number of entropy calculations.

Model Order	Converg. Points	Total PCE Calcs.	Operations per Computation (d)	Total Ops.
5	25	66	256^4*(6(=),10(==),42(L/S),	66(d)
6	36	137	48(>>),1(+,++,*,/,log))	137(d)

Table E.5: Operations Count for Indirect Entropy Calculations

By combining the dominant components of the modelling and entropy calculation rou-

```
for( all_i ) {
  for( all_j ) {
    for( all_k ) {

      /* Compute C(1) /*
      C(1) = 0;
      for( all_l ) {
        data_value = (i,j,k,l);
        data_node =
      SearchNodes(HashTable[Compute_hash(data_value)],data_value);
        C(1) += (*data_node).count;
      }

      /* Compute Sum_(i,j,k,l) C(i,j,k,l) log C(i,j,k,l)/C(1) */
      for( all_l ) {
        data_value = (i,j,k,l);
        data_node =
      SearchNodes(HashTable[Compute_hash(data_value)],data_value);
        PCE -= (*data_node).count * log((*data_node).count/C(1));
      }
    }
  }
}
PCE /= ( Total_count * log(2) );
```

Figure E.3: Indirect Entropy Calculation Excerpt

tines we can now summarize the total number of operations required to predict the entropy rate of the speech process. In table E.6 we summarize the number of operations by model order and provide a total measure for all six calculations.

Recognizing that some of these operations require more execution time than others, we use the set of relative operation costs detailed in table E.1 to weight each operation by its approximate cost. This weighted sum gives us a comprehensive lower bound estimate of 1.2×10^{14} integer and 1.8×10^{13} floating point operations for the complete entropy rate prediction procedure. As noted before, this is a very conservative lower bound based upon

Order	Total Modelling and Entropy Operation Counts ($\times 10^{10}$)										
	L/S	<<	==	=	+	×	%	++	*	/	log
1	.870	-	-	-	.870	-	-	.000	.000	.000	.000
2	.878	-	-	-	.870	-	-	.004	.004	.004	.004
3	1.69	-	2.96	-	.740	-	-	1.05	1.05	1.05	1.05
4	65.2	-	160	-	.610	-	-	26.2	26.2	26.2	26.2
5	1213	1372	295	192	50	11	22	28	28	28	28
6	2585	2881	645	467	173	57	114	59	59	59	59
Total	3867	4253	1103	659	226	68	136	114	114	114	114

Table E.6: Total Modelling and Entropy Operations Counts

operation counts taken from simplified pseudocode excerpts of the real algorithms. It does serve to indicate the magnitude of the entropy calculation and outline the computational requirements for repeating this research.

Appendix F

Development of the Unicity Distance and Related Measures

F.1 Equivocation

To define the theoretic security of a cryptosystem we make use of the properties of entropy and conditional entropy defined in [57]. We may consider a cipher to be compromised by the attacker when he is able to determine the secret key, or the message, from observations of the encrypted message.

The natural information-theoretic measure to describe the attacker's knowledge of the key, or message, is defined by the conditional entropy function. We will denote a series of sequential observations of a random variable, for instance j ciphertext elements, by C_1^j. Shannon termed the uncertainty about the key given j observations of the ciphertext the *key equivocation* and defined it by,

$$H(\mathbf{K}|C_1^j) = \sum_{\mathbf{K},C_1^j} p(\mathbf{K}, C_1^j) \log p(\mathbf{K}|C_1^j). \tag{F.1}$$

Similarly, the *message equivocation* is,

$$H(\mathbf{M}|C_1^j) = \sum_{\mathbf{M},C_1^j} p(\mathbf{M}, C_1^j) \log p(\mathbf{M}|C_1^j). \tag{F.2}$$

F.2 Unicity Distance

If the attacker is to compromise the secrecy system he will utilize the information gained about the key, or the message, from observations of the ciphertext. The unicity distance, N_o, is defined as the minimum number of ciphertext characters that the attacker must observe before he may deduce a unique key or message corresponding to the ciphertext. This is equivalent to reducing the equivocation to approximately zero.

The unicity distance of a cryptogram will be determined by the minimum of the key or message equivocation, since the attacker will focus on whichever calculation is easier. In general, it is more efficient for an attacker to attempt to recover the key to a cryptosystem than a particular message. This is because a successful key attack will reveal all subsequent messages, while a message attack will have to be repeated to decrypt each subsequent message. For this reason a message equivocation solution is generally not a meaningful measure of cipher security. We will assume a key attack and use the key equivocation measure to derive a security index.

The key equivocation can be expressed in terms of quantities under the cryptographer's control. Since the key and message are chosen independently,

$$H(\mathbf{M},\mathbf{K}) = H(\mathbf{M}) + H(\mathbf{K}), \tag{F.3}$$

and since $\mathbf{K}$ and $\mathbf{C}$ uniquely determine $\mathbf{M}$,

$$H(\mathbf{M},\mathbf{K}) = H(\mathbf{C},\mathbf{K}) = H(\mathbf{C}) + H(\mathbf{K}|\mathbf{C}). \tag{F.4}$$

Combining equations (F.3) and (F.4) we obtain,

$$H(\mathbf{K}|\mathbf{C}) = H(\mathbf{K}) - (H(\mathbf{C}) - H(\mathbf{M})). \tag{F.5}$$

We can upper bound the term $H(\mathbf{C}) - H(\mathbf{M})$ by observing that an ideal cipher would produce an output uniformly distributed over the possible set of ciphertexts. Thus the

entropy of j consecutive observations of the ciphertext is upper bounded by,

$$H(C_1^j) \leq j \log L_C, \tag{F.6}$$

where L_C is the size of the ciphertext alphabet. The quantity $H(\mathbf{M})$ is defined precisely by the entropy rate of the source process, $R_o \log L_M$, where L_M is the size of the message alphabet. Therefore the entropy of j source elements is,

$$H(M_1^j) = j R_o \log L_M, \tag{F.7}$$

and the quantity $H(C_1^j) - H(M_1^j)$ is upper bounded by the redundancy of j observations of the source process. Using equations (F.6) and (F.7) we define the total redundancy of j message elements,

$$D_j = j(\log L_C - R_o \log L_M). \tag{F.8}$$

Therefore, using equations (F.5) and (F.8) we can obtain a lower bound for the key equivocation, given that j ciphertext characters have been transmitted,

$$\begin{aligned} H(\mathbf{K}|C_1^j) &= H(\mathbf{K}) - \left(H(C_1^j) - H(M_1^j)\right) \\ &\geq H(\mathbf{K}) - j(\log L_C - R_o \log L_M) \\ &\geq H(\mathbf{K}) - D_j. \end{aligned}$$

The unicity distance is the minimum number, N_o, of ciphertext elements that the attacker needs to intercept in order to reduce the uncertainty about the key to approximately zero. Hence,

$$\begin{aligned} N_o &\triangleq \min\{j: \ H(\mathbf{K}|C_1^j) = 0)\} \\ &\approx \min\{j: \ H(\mathbf{K}) - D_j = 0\}. \end{aligned} \tag{F.9}$$

Maximum key equivocation is achieved by maximizing $H(\mathbf{K})$. We shall assume therefore

that the cryptographer designs the key source process to select keys with uniform probability over the key space. This yields a maximal key entropy of,

$$H(\mathbf{K}) = K \log L_K,$$

where the K digits in each key are chosen from an alphabet of size L_K.

Solving for the unicity distance yields,

$$N_o = \frac{K \log L_K}{\log L_C - R_o \log L_M}. \tag{F.10}$$

Thus, in a cryptosystem where $L_K = L_C = L_M$ the unicity distance is simply expressed as,

$$N_o = \frac{K}{D}, \tag{F.11}$$

where $D = 1 - R_o$ is the percentage redundancy of the source process.

F.3 Unicity Distance with a Randomizer

Ciphers utilizing a randomizer as defined in figure 4.4 can attain a higher unicity distance than those without. If the randomizing stream, $\mathbf{R}$, is chosen independently of the message and key, then the key equivocation for the cryptosystem can be developed as follows,

$$\begin{aligned} H(\mathbf{M},\mathbf{K},\mathbf{R}) &= H(\mathbf{M}) + H(\mathbf{K}) + H(\mathbf{R}) \\ &= H(\mathbf{C},\mathbf{K},\mathbf{R}) \\ &= H(\mathbf{C}) + H(\mathbf{K}|\mathbf{C}) + H(\mathbf{R}|\mathbf{C},\mathbf{K}). \end{aligned} \tag{F.12}$$

Rearranging the terms yields the key equivocation,

$$\begin{aligned} H(\mathbf{K}|\mathbf{C}) &= H(\mathbf{K}) - (H(\mathbf{C}) - H(\mathbf{M})) + (H(\mathbf{R}) - H(\mathbf{R}|\mathbf{C},\mathbf{K})) \\ &= H(\mathbf{K}) + H(\mathbf{R}) - (H(\mathbf{C}) - H(\mathbf{M})). \end{aligned} \tag{F.13}$$

Knowledge of the key and the ciphertext reveals the true message, so it must implicitly reveal the randomizer stream, $H(\mathbf{R}|\mathbf{C},\mathbf{K}) = 0$.

This development yields a measure of the unicity distance for cryptosystems employing a randomizer,

$$N_o = \frac{K + H(\mathbf{R})}{D}. \tag{F.14}$$

Appendix G

Speech Coder Implementations

The following are brief descriptions of the software algorithms used to simulate the μ-law PCM, DPCM, ADPCM, and CELP coders used in our experiments.

G.1 PCM Coder

The basic μ-law PCM signals required only an elementary pre-processing in order to remove a header and de-interleave μ-law PCM samples contained in the SWITCHBOARD source files. No additional coding or decoding was required for the PCM samples. The SWITCHBOARD database is discussed in more detail in Appendix B.

The remainder of the coders were implemented in the C programming language and were run on Sun SparcStation II and SparcStation 20 computing platforms. There was no need for real-time performance from the algorithms as the input data rate could be controlled by file I/O operations and the quality evaluation routine was entirely unaffected by delay.

G.2 DPCM Coder

The DPCM coder was a simple non-adaptive 10 pole differential predictive filter design of the type described in the block diagram of figure G.1.

The set of optimal non-adaptive predictor filter parameters, $\{a_i\}$, were computed by

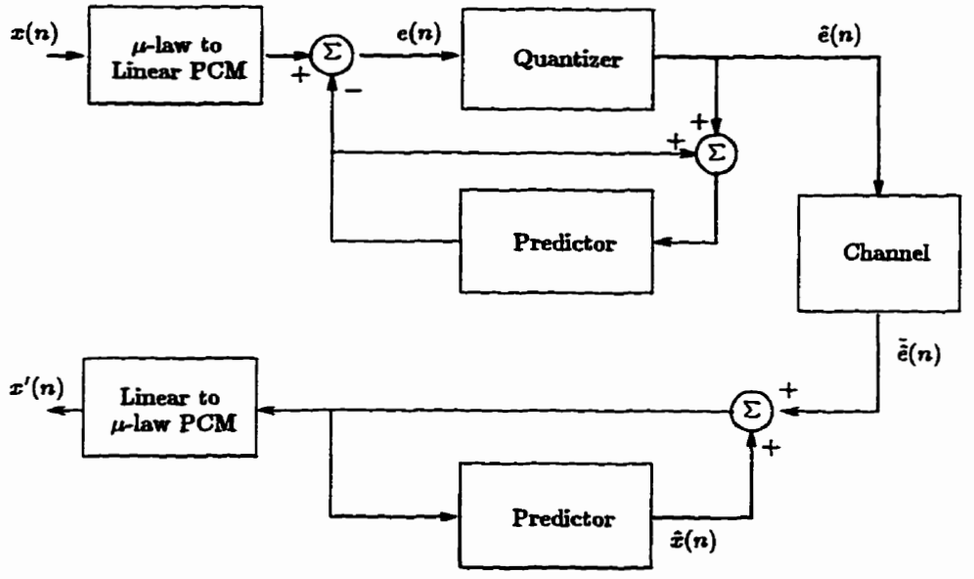


Figure G.1: Block Diagram of DPCM Coder/Decoder

solving the set of equations represented by,

$$\sum_{i=1}^{M} a_i \rho_{n-i} = \rho_n \quad \forall \; n = 1, 2, \ldots, M,$$

where the model order is $M = 10$, and the terms ρ_i are the correlation coefficients for the speech process. These equations are known as normal equations, Yule-Walker prediction equations, and Weiner-Hopf equations [22].

The correlation coefficients used to determine the filter parameters for this speech coder were computed from a long-term study of the conversational speech process as presented in figure 3.1. The 16 level quantizer was optimized for a gamma distributed input signal according to the results of [102], resulting in a data rate of 32 kbps. The software implementation included a module for converting a 64 kbps μ-law PCM input to 12 bit linear PCM format according to the CCITT G.711 definition [14].

G.3 ADPCM Coder

The ADPCM coder used for this research was implemented in C by Sun Microsystems Inc. as a variable rate coder, providing 3, 4, and 5 bit quantizations of the residual signal. This implementation was designed and tested to satisfy the CCITT G.721 standard for 32 kbps ADPCM transmission [66], and the G.723 standard for ADPCM transmission at 24 and 40 kbps. The software implementation included a module for converting a 64 kbps μ-law PCM input to 12 bit linear PCM format according to the CCITT G.711 definition [14].

The CCITT ADPCM standard utilizes both predictor and step-size adaptation procedures, as discussed in section 5.2. A block diagram of the ADPCM coder and decoder is provided in figure G.2.

G.4 CELP Coder

The CELP coder used for these experiments consisted of an algorithm satisfying the standard for the Federal Standard 1016 4.8 kbps CELP voice coder as described in [67]. This algorithm was implemented in C and distributed by the U.S. Federal government for public use.

Our experiments were performed using the default values for all parameters of the CELP coder, including a Hamming $(15, 11)$ error control code on some of the more sensitive bits in the encoded frames. The error control feature of the coder is specified by the CELP standard to protect the most significant bits of the pitch delay and pitch gain portions of the data frame and so was left functional for our experiments. The default parameters specified a code book size of 512, and an LPC filter of 10th order. The remainder of the CELP coder parameters are summarized in table G.1.

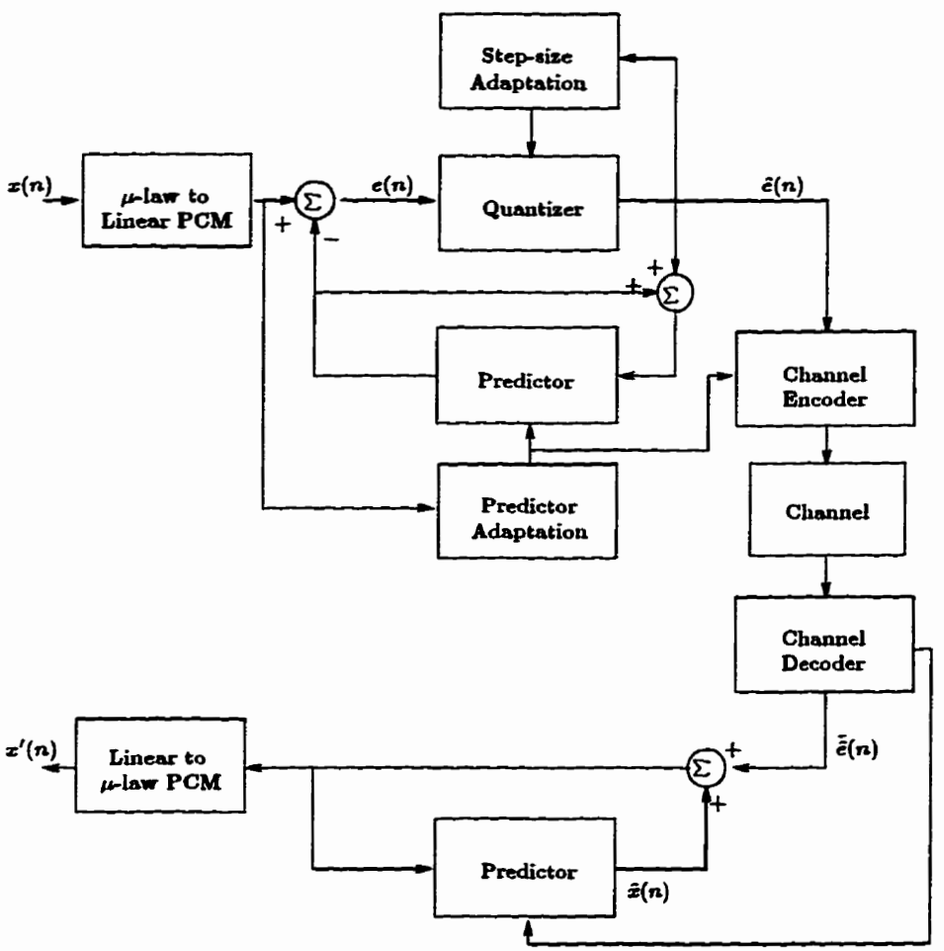


Figure G.2: Block Diagram of ADPCM Coder/Decoder

Parameter	Value
Code book size	512
Code word length	60
LPC analysis frame size	240
LPC filter order	10
Pitch analysis frame size	60
Pitch order	1
Noise weighting factor	0.8
Input speech scaling factor	1.0
Output speech scaling factor	1.0
% bit error rate	0.0
modified excitation logical switch	1
prefilter logical switch	0.0
type of fractional pitch search	hier
codebook gain quantization	log
pitch quantization	max2

Table G.1: CELP Coder Parameters

Appendix H

SNR Models for Representative Speech Coders

It is possible to derive operational rate-distortion functions for the simpler coders in our representative subset in terms of the simple SNR measurement. The models derived below for linear, μ-law and DPCM coders tend to support the experimental evidence compiled in section 5.5 for the segmental-SNR measurement. In addition, this study will reveal a correspondence between the performance of the representative speech coders and the exponential behaviour of the theoretical rate-distortion curves for speech in figure 5.3.

H.1 μ-law PCM

The μ-law compression characteristic is

$$c(x) = x_{\max} \frac{log_e\left(1 + \frac{\mu|x|}{x_{\max}}\right)}{log_e(1+\mu)} sgn(x).$$

This characteristic is linear for small x values, but for large signals such that $\mu|x| >> x_{\max}$ it is logarithmic.

The signal to noise ratio for nonlinear quantization is

$$SNR_{\text{nonlinear PCM}} = \frac{\sigma_x^2}{\sigma_q^2} = \frac{\int_{-\infty}^{\infty} x^2 f_X(x) dx}{\sigma_q^2}$$

If we assume the input pdf is approximately flat in each of our non-uniform intervals we may express the quantization error variance as

$$\sigma_q^2 = \sum_{k=1}^{L} f_X(y_k) \int_{x_k}^{x_{k+1}} (x - y_k)^2 dx,$$

where y_k is the representation level of our input signal x. As a consequence of our assumption of a flat pdf within each interval we must have $y_k = \frac{1}{2}(x_k + x_{k+1})$. This leads to the result

$$\sigma_q^2 = \frac{1}{12} \sum_{k=1}^{L} p_k \Delta_k^2, \tag{H.1}$$

where $p_k = f_X(y_k)\Delta_k$.

Following the result of Bennett as presented in [103] we note that for large L, the slope of the compressor characteristic in each interval is inversely proportional to the size of that interval

$$\frac{dc(x)}{dx} \approx \frac{2x_{\max}}{L\Delta_k}. \tag{H.2}$$

Substituting H.2 into H.1 yields

$$\sigma_q^2 \approx \frac{x_{\max}^2}{3L^2} \sum_{k=1}^{L} p_k \left(\frac{dc(x)}{dx} \right)^{-2} \quad \text{for large L.}$$

Setting $p_k \approx f_X(x)dx$ yields a useful approximation

$$\sigma_q^2 \approx \frac{x_{\max}^2}{3L^2} \int_{-x_{\max}}^{x_{\max}} f_X(x) \left(\frac{dc(x)}{dx} \right)^{-2} dx \quad \text{for large L.}$$

Computing the SNR for this nonlinear quantization yields

$$\begin{aligned} SNR_{\text{nonlinear PCM}} &= \frac{3L^2}{x_{\max}^2} \frac{\int_{-\infty}^{\infty} x^2 f_X(x) dx}{\int_{-x_{\max}}^{x_{\max}} f_X(x) \left(\frac{dc(x)}{dx} \right)^{-2} dx} \\ &= \frac{3L^2}{x_{\max}^2} \frac{\int_{-\infty}^{\infty} x^2 f_X(x) dx}{\int_{-x_{\max}}^{x_{\max}} f_X(x) \left(\frac{|x| \log_e(1+\mu)}{x_{\max}} \right)^{-2} dx} \\ &= \frac{3L^2}{\log_e(1+\mu)} \quad \text{for } \mu|x| >> x_{\max} \end{aligned}$$

The North American PCM standard [14] sets $\mu = 255$ and so we have an SNR measure independent of the signal characteristics

$$SNR_{\mu\text{-law PCM}} = 6.02R - 10.1 \quad dB.$$

H.2 DPCM

The input to the quantizer in a DPCM coder is the difference signal,

$$d(n) = x(n) - \hat{x}(n),$$

where $\hat{x}(n)$ is a prediction of $x(n)$. The ratio,

$$G_p = \frac{\sigma_x^2}{\sigma_d^2},$$

is the prediction gain indicating reduction in error variance that is achieved by linear prediction of the signal.

It is possible to define the SNR performance of the DPCM coder in relation to that of the PCM coder, subject to some minor constraints on the signal probability distributions. We will not concern ourselves with these details, but rather present an approximate relationship as developed in [22],

$$SNR_{\text{DPCM}} = SNR_{\text{linear PCM}} + 10 \log G_p.$$

In general, a DPCM system can achieve better SNR performance than PCM by way of attaining a prediction gain $G_p > 1$. It may be shown that the maximum prediction gain attainable by a linear predictor is equivalent to the reciprocal of the spectral flatness measure of equation 5.5, hence,

$$\max\{G_p\} = \gamma^{-2}.$$

The SNR performance of an optimal DPCM coder is therefore bounded by

$$SNR_{\text{DPCM}} = SNR_{\text{linear PCM}} - 10 \log \gamma^2,$$

which becomes,

$$SNR_{\text{DPCM}} = SNR_{\text{linear PCM}} + 7 \quad dB,$$

for the real speech signal with a computed spectral flatness measure of $\gamma^2 = 0.2$.

H.3 ADPCM

The combination of adaptive prediction and adaptive quantization make calculation of general ADPCM SNR performance very difficult.

Experimental work by Noll [104], and Jayant [105] demonstrated that adaptive quantization could achieve a SNR gain of 5-6 dB over μ-law PCM performance. A study of adaptive and fixed prediction by Noll in [106] has indicated an upper bound for adaptive prediction to be a gain of about 14 dB over μ-law PCM. Fixed prediction was found to achieve at most a 10.5 dB gain over μ-law PCM, so an adaptive characteristic in a linear predictor can be expected to exhibit a gain of approximately 3.5 dB over non-adaptive systems.

A corresponding gain of 10-12 dB over μ-law PCM has been observed in studies of the CCITT G.721 ADPCM coder [21], which uses both types of adaptation, and similar ADPCM coders operating at bit rates ranging from 16 kbps to 40 kbps [106], [107].

H.4 CELP

As we noted in section 5.4, the SNR measure is generally not a reliable indicator of subjective quality for non-waveform type coders. The Federal Standard 1016 4.8 kbps CELP coder [67] is a non-waveform coder, and as such introduces distortions to the original signal that are less noticeable from a perceptual perspective than they are from a purely objective measurement.

A subjective evaluation of our standard CELP coder was provided in [67]. This study found the Diagnostic Acceptability Measure (DAM) [71] to range between 55 and 67 in a variety of acoustic environments. For comparison, the input speech of μ-law PCM demon-

strated a DAM measurement between 60 and 84 in the same acoustic environments, and a 32 kbps ADPCM coder ranged from 54 to 68.

There would be little justification for associating these measures with SNR measures since the SNR has little relevance to perceived quality for this type of distortion. However, this is not to say that SNR measures, or in particular the segmental-SNR measure selected for our objective quality measure, cannot indicate the relative quality of CELP coded speech under a variety of channel conditions.

H.5 SNR-Rate Performance

Plotted in figure H.1 are the SNR-rate curves developed for the representative coders. In this figure we have estimated the SNR performance of the CELP coder on the basis of the subjective performance results discussed above to be similar to an ADPCM coder at low bit rates. This figure, plotted on a logarithmic scale in dB, indicates rate-distortion performance for real coders that corresponds to the exponential nature of the theoretical rate-distortion developments of section 5.3.

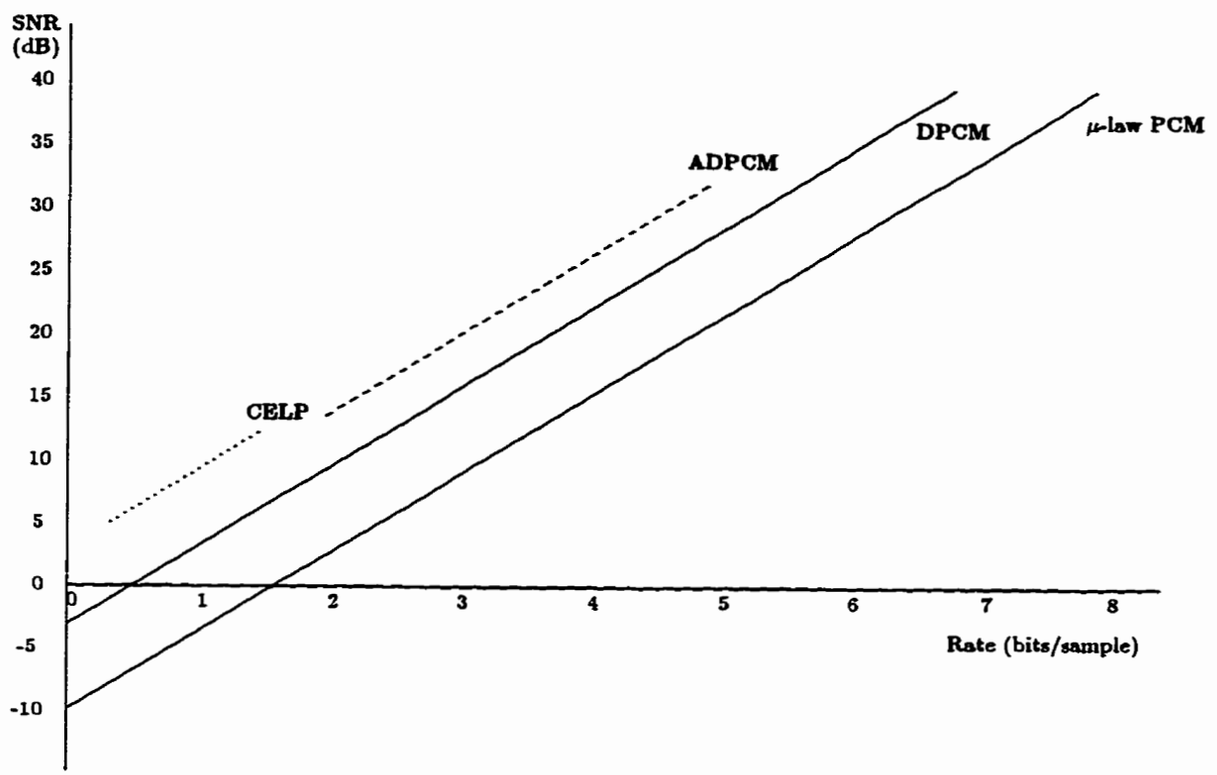


Figure H.1: SNR-Rate Performance of Representative Coders

Appendix I

Entropy-maximizing Property of the Gaussian Distribution

The following theorem proves that for a fixed second moment, the zero-mean Gaussian process yields the maximum entropy of any process.

Theorem 3 (Maximum Entropy of the Gaussian Process) *The entropy of a process, X, defined to have a fixed second moment of σ_x^2,*

$$\int_{-\infty}^{\infty} x^2 f_X(x)\, dx = \sigma_x^2,$$

is maximized when $f_X(x) = \phi_X(x)$, the pdf of the Gaussian process.

Proof: We compute the difference between the entropy of any process, $H(X)$, and the entropy of the Gaussian process, $H_G(X)$, for a fixed second moment, σ_x^2,

$$\begin{aligned} H(X) - H_G(X) &= H(X) - \int_{-\infty}^{\infty} \phi_X(x) \log\left[\frac{1}{\phi_X(x)}\right] dx \\ &= H(X) - \int_{-\infty}^{\infty} f_X(x) \log\left[\frac{1}{\phi_X(x)}\right] dx \\ &= \int_{-\infty}^{\infty} f_X(x) \log\left[\frac{\phi_X(x)}{f_X(x)}\right] dx \\ &\leq \int_{-\infty}^{\infty} f_X(x) \left[\frac{\phi_X(x)}{f_X(x)} - 1\right] dx \qquad \text{(I.1)} \\ &= \int_{-\infty}^{\infty} [\phi_X(x) - f_X(x)]\, dx. \\ &= 0. \end{aligned}$$

Hence, $H(X) \leq H_G(X)$ for all probability distributions, $f_X(x)$, with equality only when $f_X(x) = \phi_X(x)$. We have shown then, that maximum entropy is obtained by the Gaussian distribution when the process has a fixed second moment. Note that the inequality of equation I.1 follows the relation, $\log z \leq z - 1 \ \forall \ z > 0$.

Appendix J

Analytic Models of the Speech Process

A large number of studies have been performed over the years to determine an analytic model of the speech signal. Early papers by Sivian [108], Dunn and White [109], Purton [110], and Richards [111], Paez and Glisson [102] all contributed to the development of first order probability models for a signal widely held to be the result of a stationary and ergodic stochastic process. All studies are agreed that the long term distribution of speech amplitude is highly peaked around a zero mean. We refer to the experimental work performed in [102] for the model of the analog speech signal pdf and proposed analytic functions for the signal presented in figure J.1.

Here we will present a few of the models proposed to describe the first-order characteristics of the signal. The short-term probability density function (pdf) of the speech waveform is often characterized by a Gaussian distribution, as reported in [112], and [22]. This short term model is useful for some applications, as it serves to simplify calculations for the speech signal, but it is not considered to be adequate to express the long-term characteristics of the process. Better approximations have been obtained from studies of larger speech samples. McDonald [113] proposed an approximation of the long term speech waveform pdf to be a

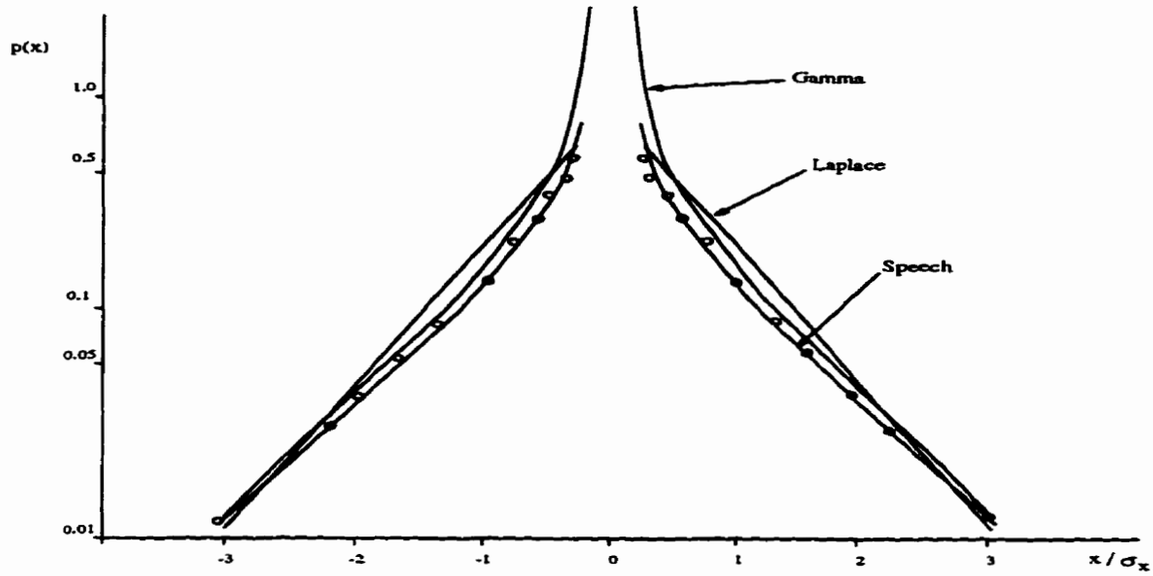


Figure J.1: Normalized Probability Density of Speech with Analytic Approximations

special form of the gamma distribution,

$$p_g(x) = \frac{\sqrt{k}\ e^{-k|x|}}{2\sqrt{\pi}\sqrt{|x|}},$$

where $k = \frac{\sqrt{0.75}}{\sigma_x}$. And a simpler approximation with the Laplacian distribution,

$$p_l(x) = \frac{\alpha}{2} e^{-\alpha|x|},$$

where $\alpha = \frac{\sqrt{2}}{\sigma_x}$. Drawing on the work of Williams [114], he found the gamma distribution to yield a better fit to the true signal. Another model was proposed by Davenport in [112]. He determined a more specific model to fit the results of speech recorded in an anechoic chamber. Combining the behaviour of the exponential distribution with the Gaussian distribution he obtained,

$$p_D(x) = \frac{0.6}{\sqrt{2}\sigma_1} e^{\frac{-\sqrt{2}|x|}{\sigma_1}} + \frac{0.4}{\sqrt{2\pi}\sigma_2} e^{\frac{-x^2}{2\sigma_2^2}},$$

where $\sigma_1 = 1.23$, and $\sigma_2 = 0.188$.

We note that these models are equally relevant to PCM quantized speech, subject to the granularity imposed by quantization into discrete steps. Jayant and Noll noted in [22] that the residual signal in discrete differential-type speech coders such as DPCM and ADPCM were also satisfactorily modelled by gamma and Laplacian distributions.

Appendix K

Converting Discrete Entropy to Differential Entropy

Where we have used discrete entropy instead of differential entropy in our development of rate-distortion measures for the real speech process we must justify our results with the following theorem due to Cover and Thomas [1],

Theorem 4 *If the probability density function, $f(x)$, of a continuous random variable, X, is Reimann integrable, then the entropy of an n-bit quantization of that variable is approximately,*

$$H(X) \approx h(X) + n.$$

Proof: The proof of this theorem follows from defining the discrete probability distribution corresponding to the continuous distribution in terms of a uniform quantization over the range of the random variable, X.

We define the values of the quantized random variable to be,

$$X = x_i \quad \text{if} \quad i\Delta \leq X \leq (i+1)\Delta,$$

and the discrete probability distribution to be,

$$p(x_i) = \int_{i\Delta}^{(i+1)\Delta} f(x)dx = f(x_i)\Delta.$$

Hence the discrete entropy of the process is,

$$\begin{aligned} H(X) &= -\sum_{-\infty}^{\infty} p(x_i) \log p(x_i) \\ &= -\sum_{-\infty}^{\infty} f(x_i)\Delta \log f(x_i)\Delta \\ &= -\sum_{-\infty}^{\infty} \Delta f(x_i) \log f(x_i) - \log \Delta. \end{aligned}$$

Therefore,

$$H(X) \to h(X) - \log \Delta, \quad \text{as } \Delta \to 0.$$

Appendix L

DES Encryption Modes

Electronic Codebook Mode DES

The electronic codebook (ECB) mode of DES presented in figure L.1 represents a pure block cipher design. Message blocks of 64 bits each are enciphered sequentially to produce a corresponding sequence of 64 bit ciphertext blocks. This is the simplest implementation of the DES cryptosystem and is also the most vulnerable to cryptanalysis as the statistical characteristics of the input blocks are reproduced in the corresponding ciphertext blocks. In addition, this cipher design is inherently vulnerable to insertion and deletion of ciphertext blocks unless an additional synchronizing protocol is used to verify the message sequencing.

Cipher Feedback Mode DES

The statistical weaknesses of ECB mode can be countered by modifying the direction of data flow through the elemental DES encryption algorithm. The Cipher Feedback (CFB) mode arrangement denoted in figure L.2 achieves a self-synchronizing encryption implementation. The CFB mode cipher of figure L.2 begins with a 64 bit initialization vector (not shown) being inputed to the encryption device. The message is then processed in blocks of k bits, and after each encryption a block of k ciphertext bits is shifted into the input buffer of the encryption device. This arrangement is self-synchronous as the decrypting device can recover from any errors occurring on the channel after receiving k correct ciphertext

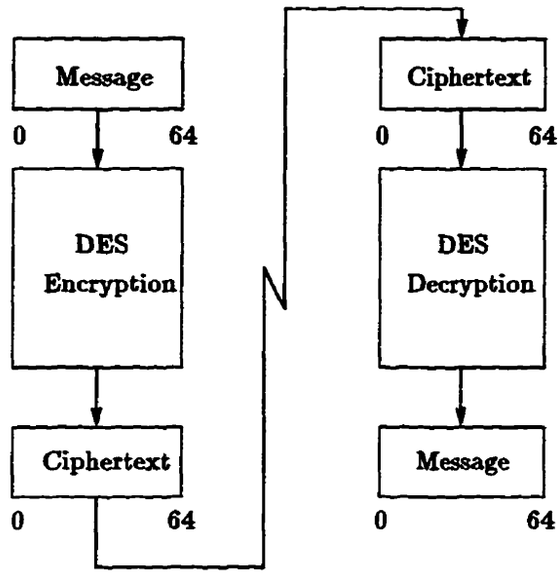


Figure L.1: Electronic Codebook Mode DES

characters. The statistical characteristics of the message blocks are concealed in this arrangement by the addition of encrypted ciphertext information with each step. Repeated blocks of message will result in a sequence of uncorrelated ciphertext blocks.

The self-synchronizing design of the CFB cipher leaves it vulnerable to ciphertext being inserted or deleted undetectably.

Cipher Block Chaining Mode DES

Another implementation, the cipher block chaining (CBC) mode denoted in figure L.3, can also conceal the statistical characteristics of the message blocks. The CBC mode cipher of figure L.3 begins by combining a 64 bit block of message data with a 64 bit initialization vector and enciphering the product with the elemental ECB mode DES cipher. The ciphertext is then combined with the next message block for all subsequent blocks to ensure that identical message blocks produce non-identical ciphertext blocks. Any errors occurring in transmission will propagate to at most 2 sequential blocks of 64 bits.

The CBC mode cipher solves the problem of message insertion and deletion by mak-

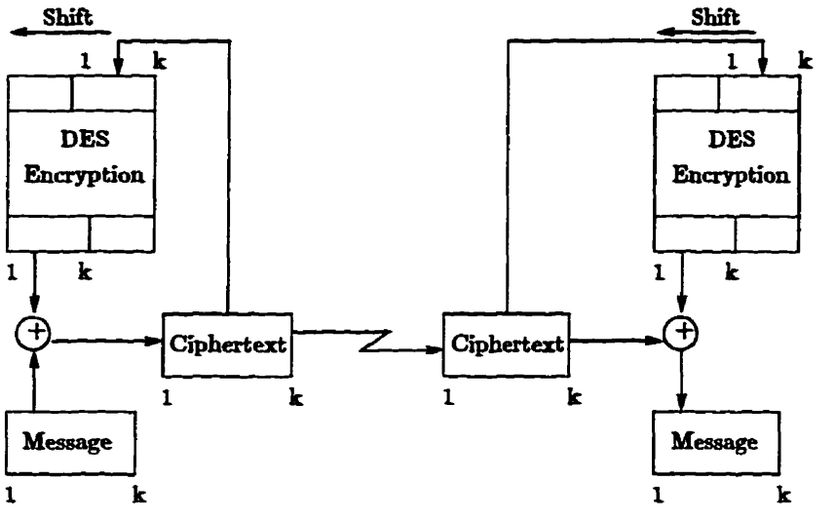


Figure L.2: Cipher Feedback Mode DES

ing the ciphertext blocks dependent on the adjacent blocks. Any insertion or deletion of ciphertext can be detected by the errors caused in the subsequent block.

Output Feedback Mode DES

The output feedback (OFB) arrangement creates a running key generator [115] for a stream cipher based on the ECB DES encryption algorithm. The OFB arrangement indicated in figure L.4 combines blocks of k message bits with blocks of k running key bits created by the DES algorithm. The running key stream is obtained by feeding k bits of key stream into the ECB cipher recursively. The result is a pseudorandom key stream with a high linear complexity. Errors occurring in the ciphertext do not propagate in the OFB cipher; they are limited to the affected bits only.

A purely synchronous stream cipher such as OFB DES is immune to insertion and deletion attacks as it requires perfect synchronization between sender and receiver.

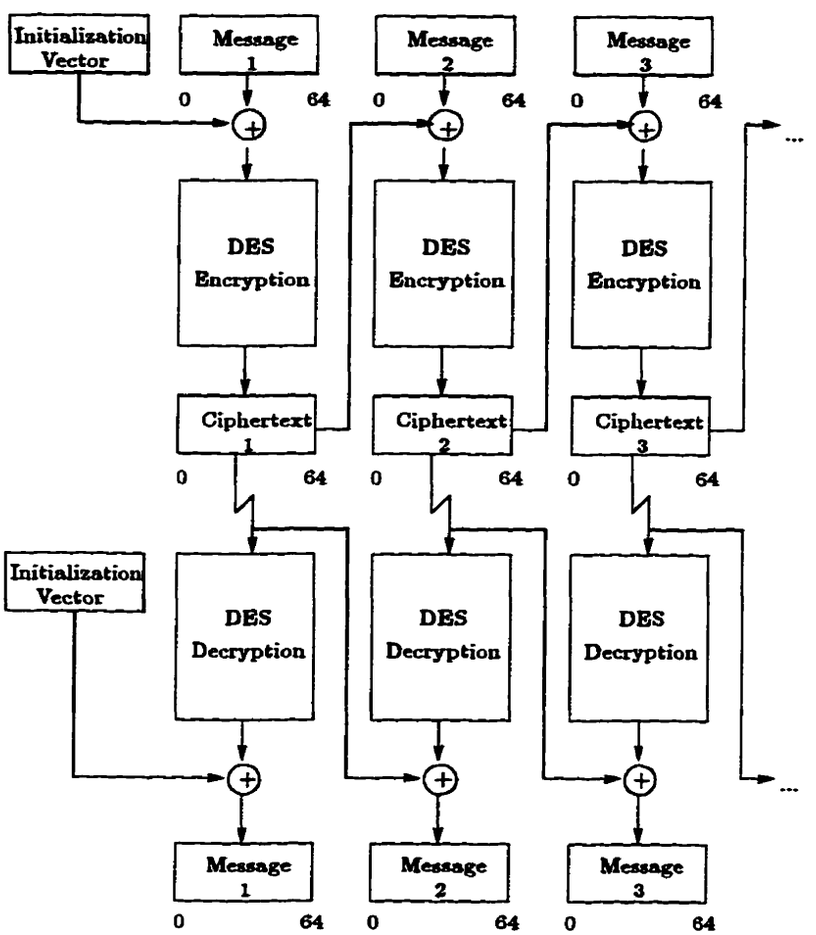


Figure L.3: Cipher Block Chaining Mode DES

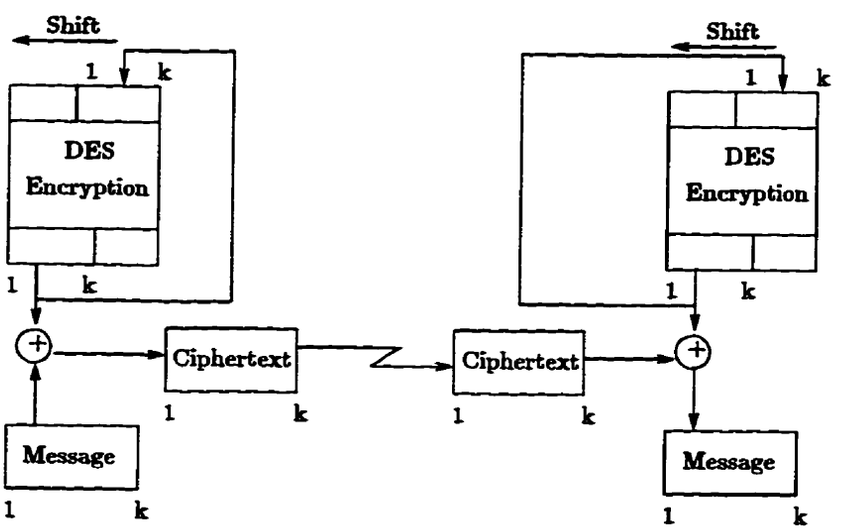


Figure L.4: Output Feedback Mode DES

Appendix M

Third Order Entropy Surfaces for Encrypted μ-law Data

Third order conditional entropy surfaces were computed at maximum sample size $S_{\max} =$ 5500 minutes for the μ-law speech database used for security measures in chapter 6 under four modes of DES encryption. These surfaces are defined by the equation $\mathcal{H}_{3,\mathcal{X}}$ as derived in section 2.4. The PCE axis scale in these figures is highly magnified to reveal small variations in the conditional entropy surfaces. Deviations from the mean surface value of these figures was found to be a maximum of 0.16 %. The average excursion from the mean was approximately 0.08 %. All four conditional entropy surfaces can be reasonably approximated with uniform distributions.

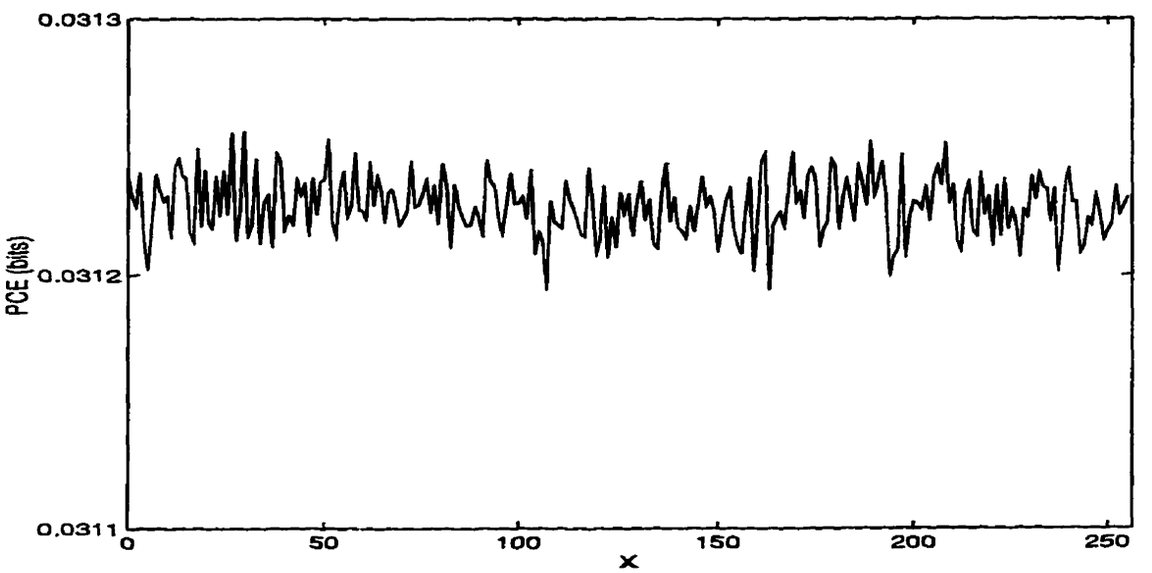


Figure M.1: Third Order Conditional Entropy Surface for ECB Mode

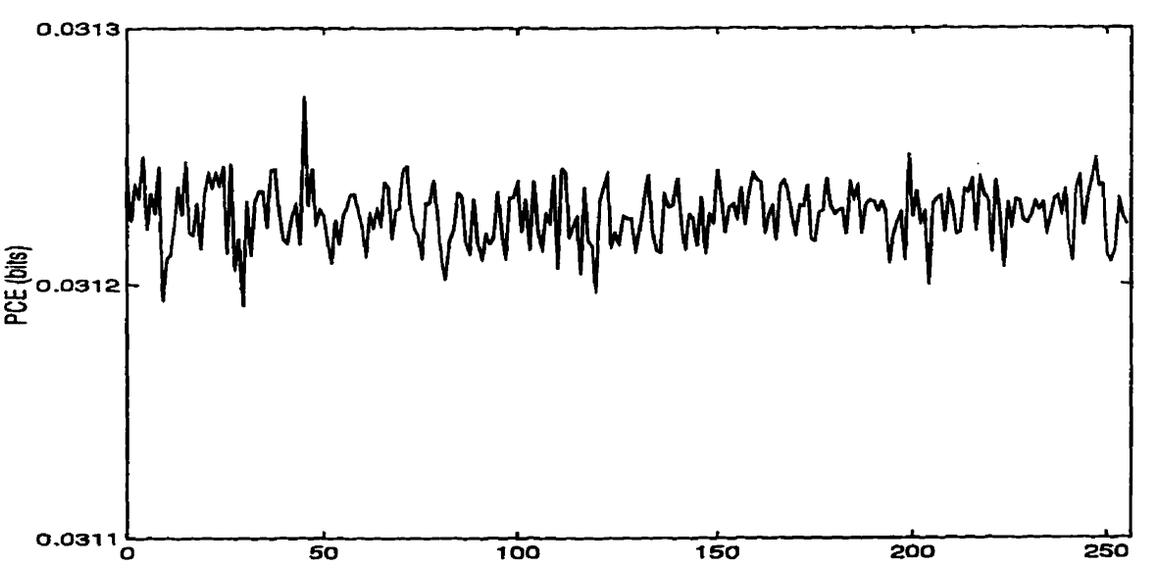


Figure M.2: Third Order Conditional Entropy Surface for CFB Mode

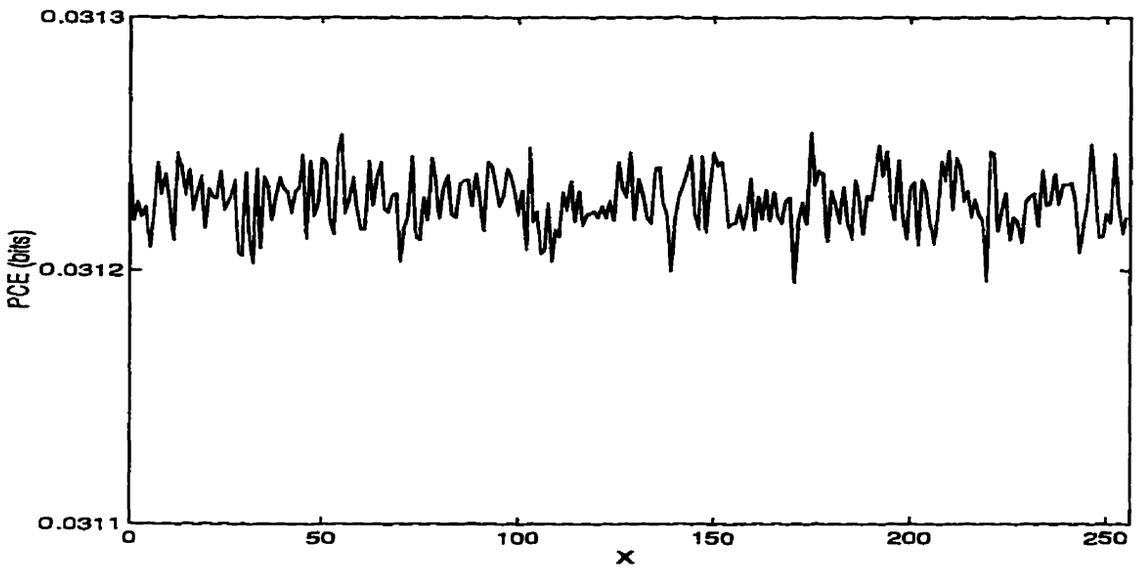


Figure M.3: Third Order Conditional Entropy Surface for CBC Mode

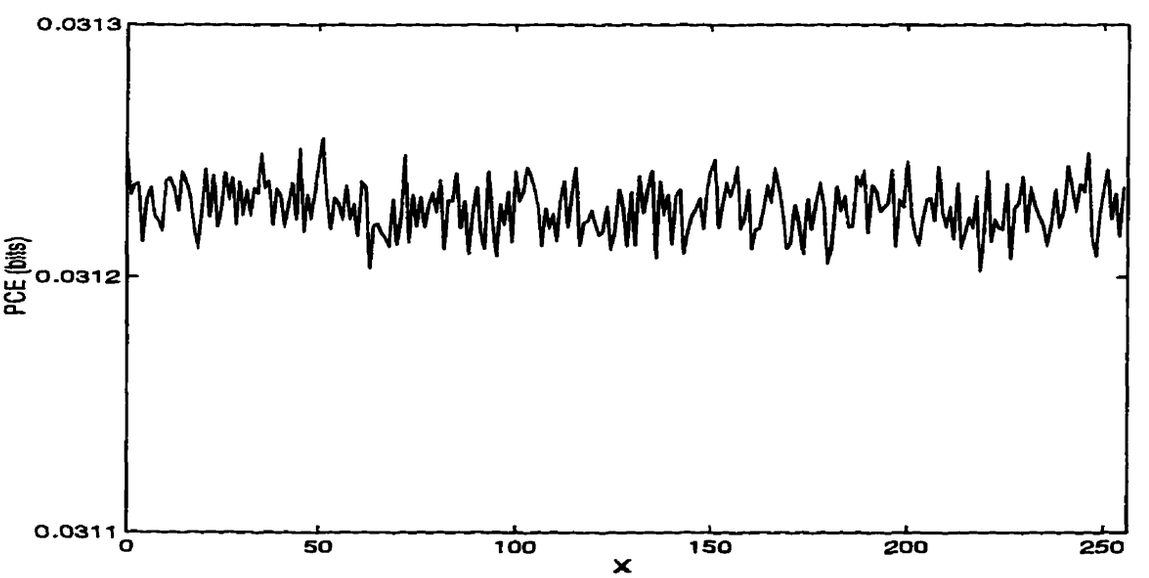


Figure M.4: Third Order Conditional Entropy Surface for OFB Mode

Appendix N

Ciphertext Entropy Convergence by Source Coder

Presented in figures N.1, N.2, N.3, N.4, and N.5 is a summary of the complete set of third order conditional entropy calculations for all waveform speech coders, as required in section 6.4.2. Each graph compares the convergence measures obtained for a single type of source coding over the range of representative encryption coders.

CELP coder results are not included here because the CELP coder produces data at such a low rate that the entire database, when CELP coded, did not trigger an intermediate entropy calculation in the automated testing software. The computed results consist of single points at the maximum sample size.

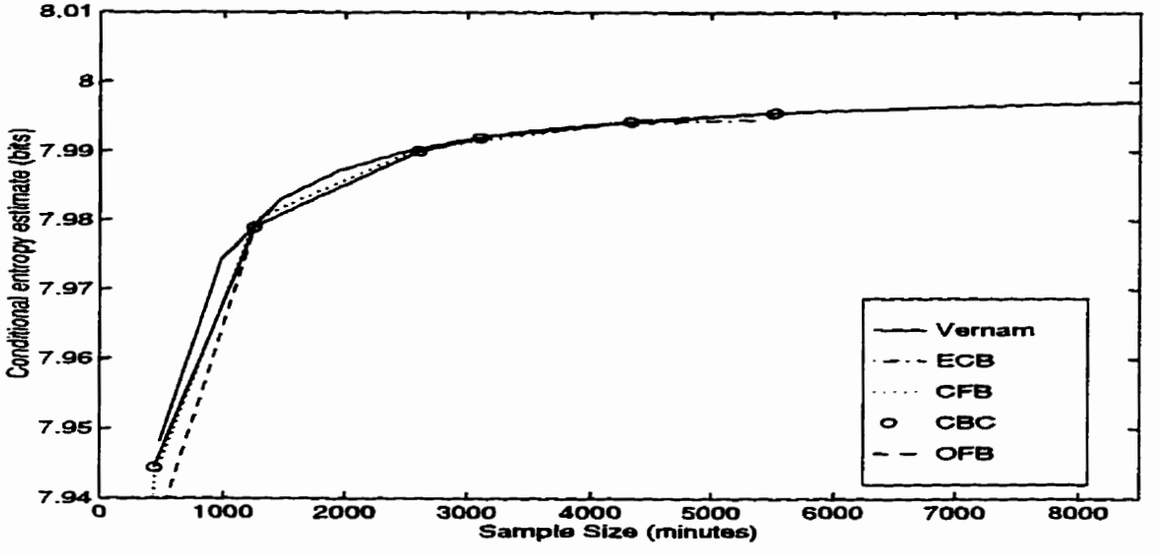


Figure N.1: Convergence Rate Comparison for Encrypted 64 kbps μ-law PCM speech

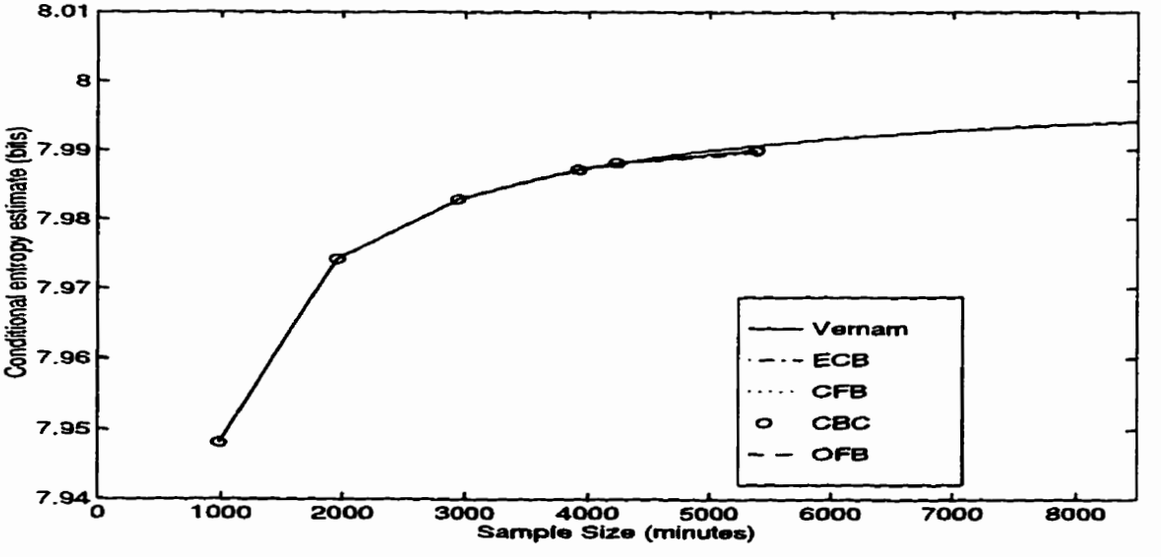


Figure N.2: Convergence Rate Comparison for Encrypted 32 kbps DPCM speech

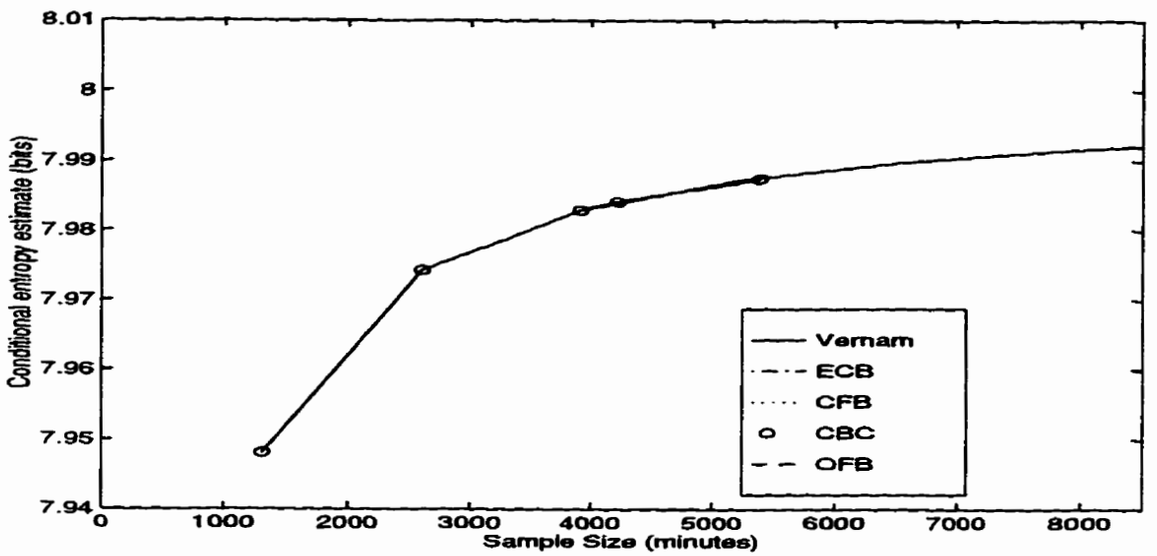


Figure N.3: Convergence Rate Comparison for Encrypted 24 kbps ADPCM speech

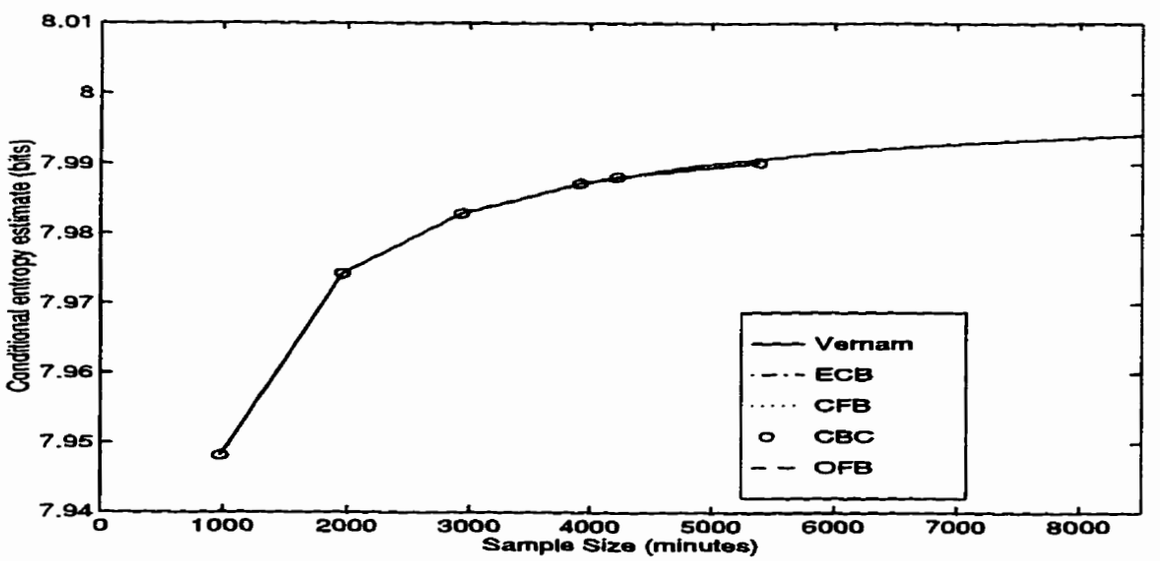


Figure N.4: Convergence Rate Comparison for Encrypted 32 kbps ADPCM speech

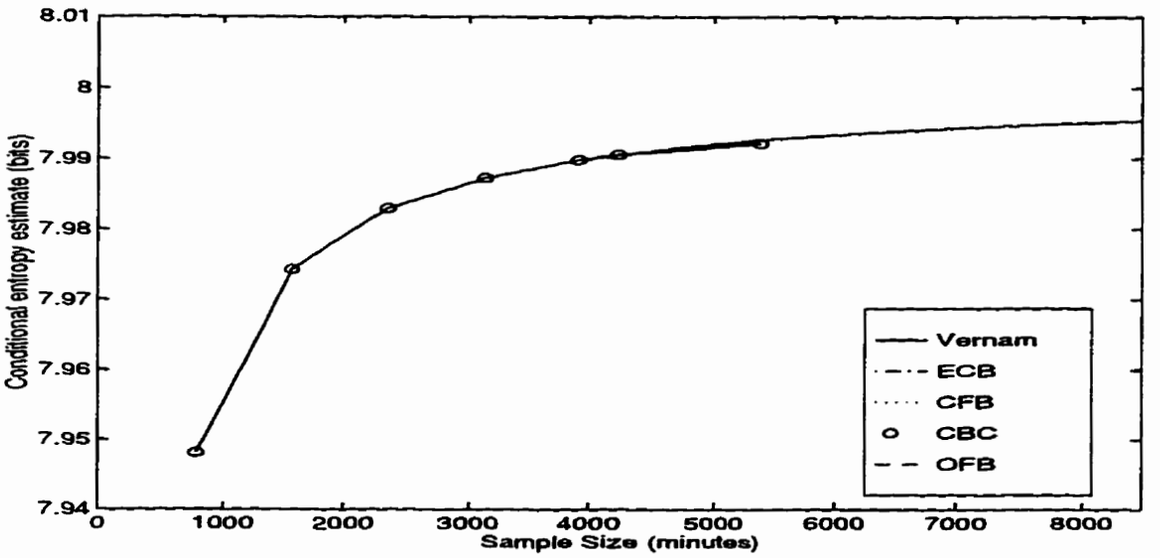


Figure N.5: Convergence Rate Comparison for Encrypted 40 kbps ADPCM speech

Appendix O

Information Theory Background

One of the main contributions of Shannon's 1948 paper [57] was the development of a mathematical theory, called information theory, for describing the fundamental characteristics of communications systems. The theory requires a measure of the event probabilities that make up the set of all possible outcomes. This is simply the probability mass function in discrete space, or the probability density function in continuous space: $p_X(x)$.

Shannon defined a measure of the amount of information contained in a set of events $x_1, x_2, ..., x_n$ with the function,

$$H(x_1, x_2, ..., x_n) = -\sum_{i=1}^{n} p_X(x_i) \log p_X(x_i)$$

where the base of the logarithm corresponds to the base of the message alphabet. This formula is commonly abbreviated to:

$$H(X) = -\sum_{i=1}^{n} p_X(x_i) \log p_X(x_i)$$

in the discrete case, and

$$h(X) = -\int_{-\infty}^{\infty} p_X(x) \log p_X(x)\, dx$$

in the continuous case. The quantity $H(X)$ is called the *entropy* of the function. This measure gives us some idea of how much "choice" is involved in the selection of an event,

or the *uncertainty* regarding the outcome. For example, if we have a set of all possible k-digit binary vectors, uniformly distributed over the sample space of size $n = 2^k$ then their entropy is:

$$H(X) = -\sum_{i=1}^{n} p_X(x_i) \log p_X(x_i) = -\sum_{i=1}^{n} \frac{1}{n} \log_2 \frac{1}{n} = k.$$

In this example the measure of uncertainty tells us that, on average, each vector contains k bits of information. Since X is only k bits long, we see that this is an example of a distribution that yields maximum entropy.

Shannon also developed a measure of the amount of information that is provided about one event given that another event has occurred. He called this a measure of *mutual information* and defined it, as one would expect, on the basis of a measure of conditional probability. The average mutual information that one random variable, Y, provides about another random variable, X, is:

$$I(X;Y) = \sum_{k=1}^{K} \sum_{j=1}^{J} p_{XY}(x_k, y_j) \log \frac{p_{X|Y}(x_k|y_j)}{p_X(x_k)}$$

where X and Y are discrete distributions with K elements and J elements, respectively. This expression is commonly simplified to:

$$I(X;Y) = \sum_{x} \sum_{y} p(x,y) \log \frac{p(x|y)}{p(x)}$$

in the discrete case, and to:

$$I(X;Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \, dx \, dy$$

in the continuous case.

There is one important distinction between continuous and discrete entropies. In the discrete case, the entropy is an absolute measure of the randomness of a random variable. In the continuous case the measurement of entropy is relative to the coordinate system. If

we change coordinates from $x_1 \ldots x_n$ to $y_1 \ldots y_n$, the new entropy would be given by:

$$h(Y) = h(X) - \int \cdots \int p(x_1, \ldots, x_n) \log J\left(\frac{x}{y}\right) dx_1 \cdots dx_n$$

where $J\left(\frac{x}{y}\right)$ is the Jacobian of the coordinate transformation.

One variation on the definition of entropy is of particular interest to cryptographic applications. This is the *conditional entropy*, or *equivocation* of a random variable X when the value of another random variable Y is known. The equivocation of X when Y is known is:

$$H(X|Y) = H_Y(X) = -\sum_x \sum_y p(x,y) \log p(x|y).$$

The preceding definitions allow us to develop some interesting relations. For two random processes X and Y:

$$H(X,Y) \leq H(X) + H(Y)$$

with equality only when the events are independent. This tells us that the uncertainty of any joint event is less than or equal to the sum of the individual uncertainties. We can also obtain:

$$H(X,Y) = H(X) + H_X(Y)$$

$$H(X) + H(Y) \geq H(X,Y) = H(X) + H_X(Y)$$

and hence,

$$H(Y) \geq H_X(Y).$$

This tells us that knowledge of X can only lessen our uncertainty about Y and that in order for X to reveal no information about Y they must be independent.

Finally, we may obtain another expression for the mutual information between two processes:

$$I(X;Y) = H(X) - H_Y(X).$$

From a cryptographic perspective, it is desirable to limit the amount of information

that leaks between the plaintext, X, and the ciphertext, Y. From above, we see that if X and Y are statistically independent then $H_Y(X) = H(X)$ and the information leakage $I(X;Y)$ is zero. Conversely, if X and Y are completely dependent then $H_Y(X) = 0$ and the information leakage is a maximum, $I(X;Y) = H(X)$.

Bibliography

[1] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley Interscience, 1991.

[2] J.R. Deller, J.G. Proakis, and J.H.L Hansen. *Discrete-Time Processing of Speech Signals.* Macmillan Publishing Company, 1993.

[3] J.M. Hammersley and D.C. Handscomb. *Monte Carlo Methods.* Methuen & Co. Ltd., London, 1964.

[4] C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(8):656–715, 1949.

[5] J. L. Massey. The relevance of Information Theory to modern cryptography. In E. Arikan, editor, *Communication, Control, and Signal Processing*, pages 176–182, 1990.

[6] S. B. Davis and P. Mermelstein. Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-28(4):357–366, 1980.

[7] L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi. Recognition of isolated digits using hidden Markov models with continuous mixture densities. *AT&T Technical Journal*, 64(6):1211–1233, 1985.

[8] S. Seneff. A joint synchrony/mean-rate model of auditory speech processing. *Journal of Phonetics*, 16(1):55–76, 1988.

[9] J. R. Cohen. Application of an auditory model to speech recognition. *Journal of the Acoustical Society of America*, 85(6):2623–2629, 1989.

[10] K.-F. Lee and F. Alleva. *Continuous Speech Recognition.* Marcel Dekker Inc., 1991. In Advances in Speech Signal Processing, S. Furui and M.M. Sondhi (Eds.).

[11] W. R. Anderson and G. B. Agnew. Towards an accurate entropy rate measurement of the speech process. *TRIO/ITRC Researcher Retreat, Kingston, Ontario, Canada*, 1995.

[12] W. R. Anderson and G. B. Agnew. A Monte Carlo approach to high order entropy estimation of the speech process. *Submitted to IEEE Transactions on Information Theory*, 1995.

[13] W. R. Anderson and G. B. Agnew. A measure of the minimum rate for distortionless speech coders. *TRIO/ITRC Researcher Retreat, Kingston, Ontario, Canada*, 1996.

[14] M. Villeret, P. Deschenes, and H. Stephenne. A new digital technique for implementation of any continuous PCM companding law. *IEEE International Conference on Communications*, Vol. 1:pp.11.12–11.17, 1973.

[15] NIST. *SWITCHBOARD: A User's Manual.* Transcription disk, NIST SWITCHBOARD CORPUS, October 1992.

[16] N. S. Jayant. High quality coding of telephone speech and wideband audio. In *IEEE Globecom '90 Conference Proceedings*, pages 927–931, 1990.

[17] N. S. Jayant. High quality coding of telephone speech and wideband audio. *IEEE Communications Magazine*, pages 10–20, January 1990.

[18] N. S. Jayant. Signal compression: Technology targets and research directions. *IEEE Journal on Selected Areas in Communications*, 10(5):796–818, June 1992.

[19] J. L. Flanagan, M. R. Schroeder, B. A. Atal, R. E. Crochiere, N. S. Jayant, and J. M. Tribolet. Speech coding. *IEEE Transactions on Communications*, COM-27(4):710–736, April 1979.

[20] S. M. Ross. *Introduction to Probability Models.* Academic Press Inc., 1989.

[21] W. R. Daumer, P. Mermelstein, X. Maitre, and I. Tokizawa. Overview of the ADPCM coding algorithm. In *IEEE Globecom '84 Conference Proceedings*, pages 774–777, 1984.

[22] N. S. Jayant and P. Noll. *Digital Coding of Waveforms.* Prentice-Hall, Inc., 1984.

[23] D. O'Shaughnessy. *Speech Communication - Human and Machine.* Addison-Wesley Publishing Company, 1987.

[24] J. S. Garafolo, T. Robinson, and J. G. Fiscus. The development of file formats for very large speech corpora: Sphere and Shorten. *Proceedings of the ICASSP*, 1994.

[25] R. F. Rice. Some practical noiseless coding techniques, Part II, Module PSI14,K+. *JPL Publication 91-3*, November 1991.

[26] W. J. Hess. *Pitch and Voicing Determination.* Marcel Dekker Inc., 1991. In Advances in Speech Signal Processing, S. Furui and M.M. Sondhi (Eds.).

[27] A. E. Rosenberg and F. K. Soong. *Recent Research in Automatic Speaker Recognition.* Marcel Dekker Inc., 1991. In Advances in Speech Signal Processing, S. Furui and M.M. Sondhi (Eds.).

[28] S. Fortune. Voronoi diagrams and Delaunay triangulations. *Computing in Euclidean Geometry*, pages 193–234, 1992.

[29] F. Aurenhammer. Voronoi diagrams: a survey of a fundamental geometric data structure. *ACM Comput. Surv. 23*, pages 345–405, 1991.

[30] J. O'Rourke. *Computational geometry in C.* Cambridge University Press, Cambridge, 1994.

[31] H. Feistel. Cryptography and computer privacy. *Scientific American*, 228(5):15–23, 1973.

[32] H. Feistel, W. A. Notz, and J. L. Smith. Some cryptographic techniques for machine-to-machine data communications. *Proceedings of the IEEE*, 63(11):1545–1554, 1975.

[33] A. F. Webster and S. E. Tavares. On the design of S-boxes. In *Advances in Cryptology, Proceedings of CRYPTO '85*, pages 523–534. Springer Verlag, 1986.

[34] C. M. Adams. On immunity against Biham and Shamir's differential cryptanalysis. *Information Processing Letters*, 41(2):77–80, 1992.

[35] R. Forré. The strict avalanche criterion: Spectral properties of Boolean functions and an extended definition. In *Advances in Cryptology, Proceedings of CRYPTO '88*, pages 450–468. Springer Verlag, 1988.

[36] G. S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Journal of the American Institute of Electrical Engineers*, 55:109–115, 1926.

[37] D. Kahn. Cryptology and the origins of spread spectrum. *IEEE Spectrum*, pages 70–80, Sept. 1984.

[38] J. L. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, IT-15, January 1969.

[39] W. Meier and O. Staffelbach. Nonlinearity criteria for cryptographic functions. In *Advances in Cryptology, Proceedings of EUROCRYPT '88*. Springer Verlag, 1989.

[40] R. A. Rueppel. *Analysis and Design of Stream Ciphers*. Springer-Verlag, 1986.

[41] K. Zeng, C.-H. Yang, D.-Y. Wei, and T. R. N. Rao. Pseudorandom bit generators in stream-cipher cryptography. *IEEE Computer Magazine*, pages 8–17, Feb. 1991.

[42] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22:644–654, November 1976.

[43] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[44] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, July 1985.

[45] R. J. McEleice. A public key cryptosystem based on algebraic coding theory. *DSN Progress Report 42-44, Jet Propulsion Laboratory*, pages 114–116, 1978.

[46] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.

[47] S. G. Akl. Digital signatures: A tutorial survey. *Computer*, 16(2):15–24, Feb. 1983.

[48] D. W. Davies. Applying RSA digital signature to electronic mail. *Computer*, 16(2):55–62, Feb. 1983.

[49] D. Kahn. *The Codebreakers: The Story of Secret Writing*. Macmillan Publishing Co., 1967.

[50] G. B. Agnew. *Computer Network Security Course Notes*. Department of Electrical & Computer Engineering, University of Waterloo, 1991.

[51] R. A. Rueppel. Security models and notions for stream ciphers. *IMA Conference on Cryptography and Coding*, pages 213–230, 1992.

[52] M. E. Hellman. An extension of the Shannon Theory approach to cryptography. *IEEE Transactions on Information Theory*, IT-23(3):289–294, 1977.

[53] M. Sivabalan, S. E. Tavares, and L. E. Peppard. On the design of SP networks from an Information Theoretic point of view. In *Advances in Cryptology, Proceedings of CRYPTO '92*, pages 260–279. Springer-Verlag, 1992.

[54] M. H. Dawson and S. E. Tavares. An expanded set of S-box design criteria based on information theory and its relation to differential-like attacks. In *Advances in Cryptology, Proceedings of EUROCRYPT '91*, pages 352–367. Springer-Verlag, 1991.

[55] M. Zhang, S. E. Tavares, and L. L. Campbell. Information leakage of Boolean functions and its relationship to other cryptographic criteria. Technical report, Department of Electrical & Computer Engineering, Queens University, Kingston Ontario, 1994.

[56] U. M. Maurer. Fast generation of prime numbers and secure public-key cryptographic parameters. *Journal of Cryptology*, 8:123–155, 1995.

[57] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–657, 1948.

[58] J. L. Massey. An introduction to contemporary cryptology. *Proceedings of the IEEE*, 76(5):533–549, May 1988.

[59] A. R. Prieto and J. Garcia Tomas. A model to order the encryption algorithms according to their quality. *Computer Communication Review*, 17(3):30–47, July/August 1987.

[60] U. Maurer. A statistical test for random bit generators. In *Advances in Cryptology, Proceedings of Crypto '90.* Springer Verlag, 1991.

[61] S. R. Quackenbush, T. P. Barnwell III, and M. A. Clements. *Objective Measures of Speech Quality.* Prentice-Hall, Inc., 1988.

[62] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Waveforms.* Prentice-Hall, Inc., 1978.

[63] H. Nyquist. Certain topics in telegraph transmission theory. *AIEE Transactions*, pages 617–644, 1928.

[64] W. R. Daumer. Subjective evaluation of several efficient speech coders. *IEEE Transactions on Communications*, pages 655–662, April 1982.

[65] N. Kitawaki and N. Nagabuchi. Quality assessment of speech coding and speech synthesis systems. *IEEE Communications Magazine*, pages 36–44, Oct. 1988.

[66] N. Benvenuto, G. Bertocci, and W. R. Daumer. The 32-kbs ADPCM coding standard. *AT&T Technical Journal*, 65:12–22, Sept.-Oct. 1986.

[67] J. Campbell, T. Tremain, and V. Welch. *The DoD 4.8 kbps standard (Proposed federal standard 1016).* Kluwer Academic Publishers, 1990. In Advances in Speech Coding, B. S. Atal, V. Cuperman, and A. Gersho (Eds.).

[68] T. Berger. *Rate Distortion Theory.* Prentice-Hall Inc., 1971.

[69] A. N. Kolmogorov. On the Shannon theory of information transmission in the case of continuous signals. *IRE Transactions on Information Theory*, pages 102–103, September 1956.

[70] J.I. Makhoul and J.J. Wolf. *Linear Prediction and the Spectral Analysis of Speech.* Bolt, Beranek and Newman, Inc. Technical Report, 1972.

[71] W. D. Voiers. Diagnostic acceptability measure for speech communication systems. *Proceedings 1977 IEEE ICASSP*, pages 204–207, May 1977.

[72] A. Gray Jr and J. D. Markel. Distance measures for speech processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-24(5):380–391, October 1976.

[73] T. P. Barnwell III. Objective measures for speech quality testing. *Journal of the Acoustical Society of America*, 66(6):1658–1663, December 1979.

[74] R. Billi and C. Scagliola. Artificial signals and identification methods to evaluate the quality of speech coders. *IEEE Transactions on Communications*, 30(2):325–335, February 1982.

[75] R. M. Gray, A. Buzo, A. H. Gray, and Y. Matsuyama. Distortion measures for speech processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):367–376, August 1980.

[76] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[77] S. Amari. *Differential-Geometrical Methods in Statistics*. Springer-Verlag, 1985.

[78] National Bureau of Standards (U.S.). Data encryption standard (DES). *Federal Information Processing Standards Publication 46*, April 1977.

[79] M. E. Smid and D. K. Branstad. The data encryption standard, past and future. *Proceedings of the IEEE*, 76(5):550–559, May 1988.

[80] J. L. Smith. The design of Lucifer, a cryptographic device for data communications. *IBM Research Report RC3326*, 1971.

[81] A. Sorkin. Lucifer, a cryptographic algorithm. *Cryptologia*, 8(1):22–41, Jan 1984.

[82] E. Biham and A. Shamir. Differential cryptanalysis of Snefru, Khafre, REDOC-II, LOKI, and Lucifer. In *Advances in Cryptology, Proceedings of Crypto '91*, pages 156–171, 1991.

[83] X. Lai and J. Massey. A proposal for a new block encryption standard. In *Advances in Cryptology, Proceedings of Eurocrypt '90*, pages 157–165. Springer-Verlag, 1991.

[84] X. Lai. On the design and security of block ciphers. *ETH Series in Information Processing*, 1, 1992.

[85] C. Connell. An analysis of NewDES: A modified version os DES. *Cryptologia*, 14(3):217–223, Jul 1990.

[86] A. Shimizu and S. Miyaguchi. Fast data encipherment algorithm FEAL. In *Advances in Cryptology, Proceedings of Eurocrypt '87*, pages 267–278. Springer-Verlag, 1988.

[87] E. Biham and A. Shamir. Differential analysis of FEAL and N-Hash. In *Advances in Cryptology, Proceedings of Eurocrypt '91*, pages 1–16. Springer-Verlag, 1991.

[88] T. W. Cusick and M. C. Wood. The REDOC-II cryptosystem. In *Advances in Cryptology, Proceedings of Crypto '90*, pages 545–563. Springer-Verlag, 1991.

[89] National Institute of Standards and Technology. Clipper chip technology. 1993.

[90] B. Schneier. *Applied Cryptography, Second Edition*. John Wiley & Sons, 1996.

[91] S. B. Xu, D. K. He, and X. M. Wang. An implementation of the GSM general data encryption algorithm A5. *Chinacrypt '94 Proceedings*, pages 287–291, Nov 1994.

[92] National Bureau of Standards (U.S.). DES modes of operation. *Federal Information Processing Standards Publication 81*, Dec. 1980.

[93] S. Furui. *Digital Speech Processing, Synthesis, and Recognition*. Marcel Dekker Inc., 1989.

[94] D. MacMillan. Single chip encrypts data at 14Mb/s. *Electronics*, 54(12):161–165, June 1981.

[95] R.C. Fairfield, A. Matusevich, and J. Plany. An LSI digital encryption processor (DEP). In *Advances in Cryptology, Proceedings of Crypto '84*, pages 115–143. Springer-Verlag, 1985.

[96] R.C. Fairfield, A. Matusevich, and J. Plany. An LSI digital encryption processor (DEP). *IEEE Communications*, 23(7):30–41, July 1985.

[97] F. Hoornaert, J. Goubert, and Y. Desmedt. Efficient hardware implementation of the DES. In *Advances in Cryptology, Proceedings of Crypto '84*, pages 147–173. Springer-Verlag, 1985.

[98] A. G. Broscius and J. M. Smith. Exploiting paralellism in hardware implementation of the DES. In *Advances in Cryptology, Proceedings of Crypto '91*, pages 367–376. Springer-Verlag, 1992.

[99] P. Algoet and T. M. Cover. A sandwich proof of the Shannon-McMillan-Breiman theorem. *Annals of Probability*, 16:899–909, 1988.

[100] J.L. Hennessy and D.A. Patterson. *Computer Architecture - A Quantitative Approach.* Morgan Kaufmann Publishers, Inc., San Mateo, California, 1990.

[101] R. Sedgewick. *Algorithms.* Addison-Wesley Publishing Co. Inc., Don Mills, Ontario, 1983.

[102] M. D. Paez and T. H. Glisson. Minimum mean-squared-error quantization in speech PCM and DPCM systems. *IEEE Transactions on Communications*, pages 225–230, April 1972.

[103] W. R. Bennett. Spectra of quantized signals. *Bell System Technical Journal*, pages 446–472, July 1948.

[104] P. Noll. Adaptive quantizing in speech coding systems. *Proceedings 1974 Zurich Seminar on Digital Communications*, March 1974.

[105] N. S. Jayant. Adaptive quantization with a one word memory. *Bell System Technical Journal*, pages 1119–1144, September 1973.

[106] P. Noll. A comparative study of various schemes for speech encoding. *Bell System Technical Journal*, 54(9):1597–1614, November 1975.

[107] P. Cummiskey, N. S. Jayant, and J. L Flanagan. Adaptive quantization in differential PCM coding of speech. *Bell System Technical Journal*, 52(7):1105–1118, September 1973.

[108] L. J. Sivian. Speech power and its measurement. *Bell System Technical Journal*, 8:646–657, 1929.

[109] H. K. Dunn and S. D. White. Statistical measurements on conversational speech. *Journal of the Acoustical Society of America*, 11(1), January 1940.

[110] R. E. Purton. A survey of telephone speech signal statistics and their significance in the choice of PCM companding law. *IEEE Paper No. 3773E*, January 1962.

[111] D. L. Richards. Statistical properties of speech signals. *Proceedings of the IEE*, 111(5):941–949, 1964.

[112] W. B. Davenport. An experimental study of speech-wave probability distributions. *Journal of the Acoustical Society of America*, 24(4):390–399, 1952.

[113] R. A. McDonald. Signal-to-noise and idle channel performance of differential pulse code modulation systems - particular applications to voice signals. *Bell System Technical Journal*, 38:1123–1151, 1959.

[114] G. Williams. Quantizing for minimum error with particular reference to speech. *Electronics Letters*, 3:134–135, April 1967.

[115] R. A. Rueppel. *Stream Ciphers*, chapter 2, pages 65–134. IEEE Press, 1991. In Contemporary Cryptology - The Science of Information Integrity, G. J. Simmons (Ed.).

Glossary

Chapters 2 and 3

PCE: Partial Conditional Entropy.

$\mathcal{X}$: The sample space of a process.

$\mathfrak{h}(n, x_1^j)$: The $O(\mathcal{X}^{n-j})$ partial conditional entropy (PCE) measure.

$\mathfrak{h}(S, n, x_1^j)$: The $O(\mathcal{X}^{n-j})$ partial conditional entropy measure from a sample set of size S.

$\psi(\cdot)$: The mapping from PCE measure to PCE vector.

$\Upsilon_{\kappa_i,j}$: The set of κ_i $(j+1)$ dimensional PCE vectors.

$\mathcal{H}_{S_l,n,\Upsilon_{\kappa_i,j}}$: The set of κ_i PCE vectors computed at sample size S_l.

$\mathcal{H}_{n,\Upsilon_{K,j}}$: The set of K PCE vectors computed at sample sizes in S^*.

$\widehat{\mathcal{H}}_{n,\Upsilon_{K,j}}$: The unique set of PCE vectors obtained from the largest available sample set for each element in $\Upsilon_{K,j}$. This gives our best estimate of a set of points on the nth order conditional entropy surface.

$\mathcal{H}_{n,\mathcal{X}^j}$: The set of all PCE vectors in $\mathcal{X}^j$ computed from exact models of the probability distribution of the process. This set is an exact representation of the nth order conditional entropy surface.

$\tilde{\mathcal{H}}_{n,\mathcal{X}^j}$: The approximate nth order conditional entropy surface obtained by interpolating the set of known PCE vectors.

$\varphi(\cdot)$: The mapping from PCE vector to scalar PCE measure.

$\tilde{H}(X_n|X_1^{n-1})$: The approximation of the nth order conditional entropy computed from the interpolated conditional entropy surface.

$\widehat{\mathcal{H}}_{S_m,n,\Upsilon_{K,j}}$: The unique set of PCE vectors normalized to a single, maximal sample size, S_m.

$\tilde{\mathcal{H}}_{S_m,n,\mathcal{X}^j}$: The approximate nth order conditional entropy surface obtained by interpolating the set of normalized PCE vectors.

$\tilde{H}_{S_m}(X_n|X_1^{n-1})$: The approximation of the nth order conditional entropy computed from the normalized conditional entropy surface.

$\mathcal{H}_{S^C,n,\Upsilon_{r,j}}$: The set of $r * |S^C|$ PCE measures representing the convergence characteristics of the source process.

$\tilde{\mathcal{H}}_{S^C,n,\mathcal{X}^j}$: The set of $|S^C|$ surfaces interpolated from the convergence characteristics.

Chapter 4

N_o: The unicity distance index of a cipher.

D: The per-letter redundancy or the redundancy rate in kbps.

$\mathcal{E}$: The encryption efficiency index of a cipher.

$\mathcal{Q}$: The encryption quality index of a cipher.

Chapter 5

γ^2: **The spectral flatness measure of a source.**

$SegSNR$: **The segmental-SNR, an objective measure of quality.**

$D^k(p||q)$: **The kth order Kullback-Leibler distance measure.**

$D^*_{(p||q)}(R)$: **A linearly regressed model of information divergence.**

$D^*_{(p||q)\text{eff}}(R)$: **The effective rate of information divergence in a source coded bit stream.**

$D_{\text{operational}}(R)$: **The operational rate-redundancy function for speech coders.**

Chapter 7

$C_{\text{S.C.}}(D)$: **The complexity model for source coders.**

$C_{\text{E.C.}}(\text{E.C.})$: **The complexity model for encryption coders.**

$C_{\text{C.C.}}(\text{BER})$: **The complexity model for channel coders.**

$\Phi(D, \text{E.C.}, \text{BER})$: **A set of constraints on the system parameters.**

α_i: **Model parameter normalizing factors.**

w_i: **Model parameter weighting factors.**