VLSI SOURCE CODING FOR EFFICIENT ANALOGUE-TO-DIGITAL CONVERSION

Transfer-of-Status Report

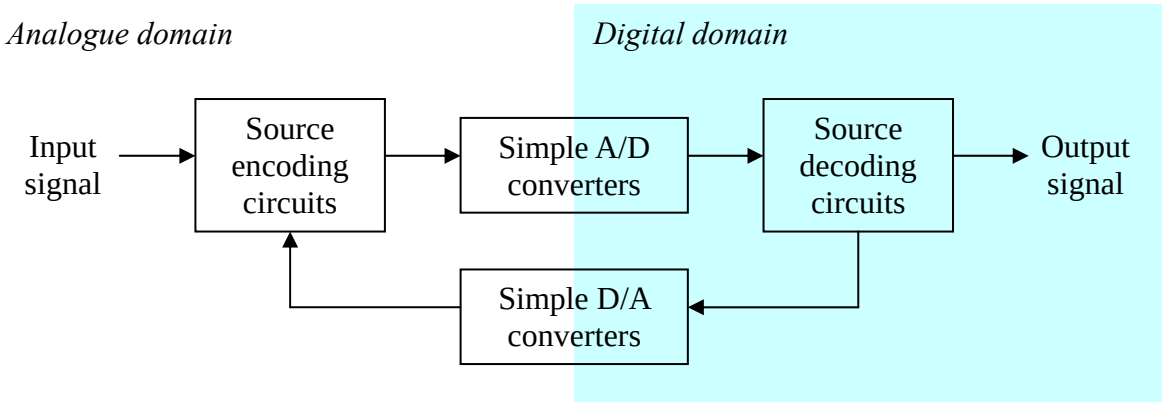
Department of Engineering Science

University of Oxford

Dileepan Joseph, Keble College

December 7, 1998

Supervised by Professor Lionel Tarassenko


Analogue domain
Digital domain
Input signal
Source encoding circuits
Simple A/D converters
Source decoding circuits
Output signal
Simple D/A converters

Executive Summary

Digital technology has been gradually taking over the operations of information processing, storage, retrieval and, lately, transmission from analogue technology. However, because real world information is fundamentally analogue, information acquisition and generation will always require analogue technology. Analogue VLSI designers should therefore focus on analogue-to-digital (A/D) and digital-to-analogue (D/A) conversion, designing converters that efficiently use the huge number of transistors that can be integrated on a silicon chip.

Source coding is often employed to transmit digital signals over digital channels of limited capacity. By exploiting structure in the information, source coding allocates more resources to important features in the signal and sometimes allocates no resources to insignificant features. This report proposes that the total A/D interface on an integrated circuit, which will include both A/D and feedback D/A converters, is like a channel of limited capacity. Since real world information often has considerable structure, analogue and digital circuitry should be used to source code the signals passing through this A/D interface.

Digital coding of speech was investigated to explore source coding, since research in this area has been intensive. Linear pulse code modulation (PCM), using uniform quantisation, is the simplest technique to code speech. However, logarithmic PCM is more efficient because it exploits the non-uniform probability distribution of speech through non-uniform quantisation. A formula is derived for non-uniform quantisation and the K-means clustering algorithm is used to find a locally optimal quantiser, which consistently improves the signal-to-noise ratio (SNR). Two heuristics were designed to produce good results in a fraction of the time required for optimisation. Because speech is non-stationary, adaptive non-uniform quantisation, where optimisation is applied to speech frames, was also devised to enhance the segmented SNR and improve the perceptual quality. Vector quantisation, of pairs of consecutive samples, improves the results further by exploiting simple correlations in voiced speech.

Linear predictive coding, used in modern coders, subtracts a linear prediction from the speech and encodes the result, which has a lower dynamic range than the original speech. Code-excited linear prediction models this difference signal by a sequence of Gaussian random vectors. Such an assumption is found to be very good except when the vectors are analysed contextually, opening the door for a better stochastic model and better coding results.

To develop source coding of signals by analogue and digital circuitry, modulation of oversampled signals was examined next. Digitising a signal at many times the Nyquist rate spreads the quantisation noise power thinly over a wide band. Decimation then down-samples the signal, eliminating noise at high frequencies. Delta modulation reduces the dynamic range of the signal before quantisation by first subtracting a simple prediction. Delta-sigma modulation shapes the quantisation noise spectrum so that more power lies at high frequencies. A cascaded modulator was proposed, where a delta-sigma modulator resolves the error of a delta modulator. These circuits were tested using signals derived from an artificial random process. When the process equalled the sum of a high dynamic range narrow-band process and a low dynamic range wide-band process, the proposed modulator outperformed the others.

This report establishes the usefulness of source coding, some principles behind it, and shows that it is possible to source code signals for A/D conversion with simulated circuitry. Further research must prove the feasibility and usefulness of source coding for efficient A/D conversion with real circuitry. Such research should focus on a specific application where the A/D demands exceed the capabilities of current low cost technology and where the input information is not arbitrary. An application that meets these criteria is the development of a high resolution, high frame rate, and high dynamic range, colour digital video camera. This report proposes a two-year plan for this work, aiming to submit a chip layout for fabrication after one year and to test the chip, and document the design and results, in the second year.

Table of Contents

List of Figures

I. Introduction

A. Purpose

This report proposes that source coding of signals, using VLSI circuitry, may be used to improve the conversion of analogue information to digital information.

B. Context

The public often requires electronic systems to acquire, process, store, retrieve, transmit and generate real world information, the physical information found in nature. These operations may be implemented in an analogue, a digital, or a hybrid analogue-digital paradigm. The analogue paradigm operates with analogue information only, using quantities that may change continuously with time and that may take on any value in a continuous interval, whereas the digital paradigm operates with digital information only, using quantities that may change at discrete moments in time and that may take on any value within a discrete set of values. The hybrid paradigm operates with a combination of analogue and digital information.

Historically, all six information operations were first implemented in the analogue paradigm. However, the invention and refinement of the microprocessor (μP), analogue-to-digital (A/D) converter, digital-to-analogue (D/A) converter, and digital signal processor (DSP) have developed the digital paradigm. At present, three of the six operations, namely processing, storage, and retrieval, are implemented predominantly in the digital paradigm because of the predictability and repeatability of digital circuits and the availability of general purpose computing and high-level specification. Information transmission is currently moving to the digital paradigm. However, the acquisition and generation of real world information may never be implemented exclusively in the digital paradigm because real world information is fundamentally analogue. Relativistic and quantum effects aside, measurable quantities in the real world may change continuously with time and may take on any value in a continuous interval.

A digital circuit is predictable because, given the same inputs and initial state, the same outputs are always generated. Noise such as thermal drift and background electromagnetic radiation typically do not affect the operation of the circuit. A digital circuit is repeatable because every copy of the circuit performs exactly the same function (providing the fabrication is not faulty). Variation in device parameters from one copy to the next does not really affect functionality. Analogue circuits for computation or memory have run-time and implementation uncertainties associated with them.[1,] As a result, the precision of analogue circuitry must be estimated from empirical results whereas the precision of digital circuitry is fixed by the number of bits used in the encoding and computing of information.

Perhaps the most important advantage of the digital paradigm is the availability of a general purpose, or stored-programmable, computer such as a Von Neumann machine (e.g. a μP). These machines process information according to instructions that are read into memory, requiring only one piece of complex hardware to implement arbitrarily complex processing. While analogue computers do exist, they do not display much versatility without reconfiguration of hardware. Additionally, a digital program may be specified in a high level language that can automatically be converted to low level machine instructions. Although languages do exist for the specification of analogue computation, their usefulness is limited given the non-existence of a general purpose analogue computer.

Nonetheless, a significant research effort has been expended in the study of more general and versatile analogue information processing, particularly in artificial neural network processing.[2] Analogue neural networks delivered the promise of massively parallel non-linear computation, where tens of thousands of synaptic operations were performed simultaneously on a single VLSI chip having millions of transistors, since each synaptic operation could be implemented with relatively few transistors.[3] Digital implementations of parallel neural networks could not approach the material or power efficiency of analogue implementations.

However, massively parallel computation is easily slowed by the serial communication of input and output data.[3] Secondly, the imprecision of analogue computation makes network learning difficult.[4] Thirdly, developments in the theory of neural networks has tended to recommend the use of relatively small networks.[5] Because of these fundamental problems, analogue neural network technology has not succeeded beyond a few research applications.

As a result of these developments, the trend has been for digital technology to dominate the operations of information processing, storage, retrieval, and transmission.

C. Hypothesis

Since the operations of information acquisition and generation will always require analogue technology, analogue VLSI designers should focus on the niche of A/D and D/A conversion, designing converters that efficiently use the huge and ever-growing number of transistors that can be integrated on a silicon chip. This report focuses on A/D conversion.

Information Theory provides a framework for studying information representation and transmission.[6] It introduces the concept of *source coding* whereby the statistics of a signal, or a source that emits symbols regularly over time, are used to map the signal efficiently to a binary sequence. Source coding has been successfully used to compress digital signals for transmission over digital channels of limited capacity. Digital algorithms exploit the structure in the information before transmission, allocating more resources to important features in the signal and sometimes allocating no resources to insignificant features.

An integrated circuit that interfaces with the real world includes A/D and D/A converters. The capacity of each converter is simply the product of the number of bits it converts and the frequency at which it operates. Taken together, these converters define an analogue-digital interface on the integrated circuit, a bi-directional channel with a capacity in each direction equal to the sum of the A/D or D/A converter capacities. This report hypothesises that source coding the signals that pass through this channel, using circuitry, would improve conversion.

Figure 1 shows an abstraction of this report, where a real world signal is first encoded by analogue circuitry, passed through an analogue-digital interface, and finally decoded by digital circuitry to produce a digital representation of the signal. The system is itself a complex A/D converter, which exploits the statistics of the information under conversion with hybrid circuitry to use efficiently the limited capacity of the simple embedded A/D and D/A converters (e.g. comparators and switched ladders) that make up the interface.

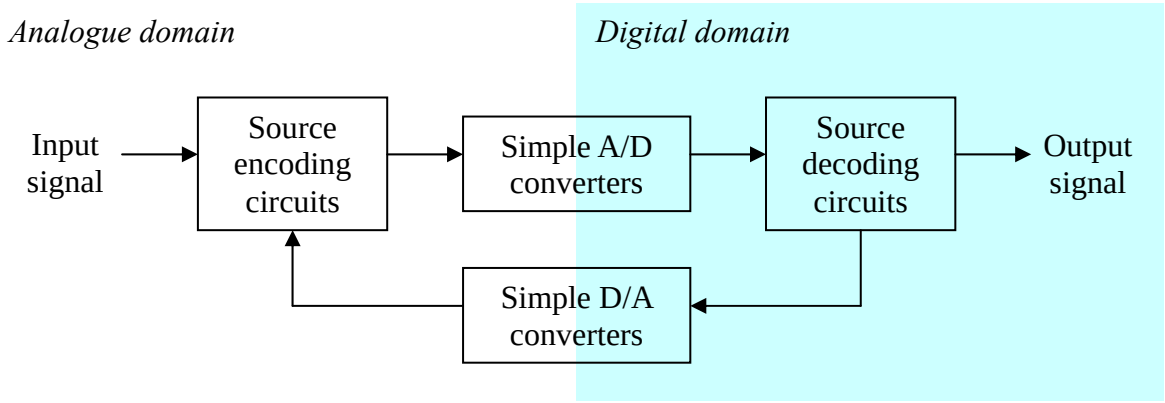


Figure 1. Source coding of signals for efficient A/D conversion.

Feedback D/A converters are useful in a source coded A/D converter because they allow the digital domain to inform the analogue domain of the features in the signal that appear to be redundant. Therefore, the analogue domain can alter the way it passes information through the feed-forward A/D converters, allocating more resources to other features. The D/A converters can also provide a reliable and accurate memory for use by the analogue circuitry.

D. Scope

The rest of this report is organised as follows. Chapter II presents background theory, referred to by other chapters, on random processes, sampling, and quantisation. To explore source coding, Chapter III reviews digital algorithms for speech compression and develops new ideas. Chapter IV explores source coding by simulated analogue and digital circuitry, extending the theory of oversampled modulation. Finally, Chapter V draws conclusions about source coding and proposes further work to complete the research introduced by this report.

II. Signal Coding Theory

A. Random processes

This chapter reviews the theory of signal coding, which underlies A/D conversion. To begin with, A/D converters operate on signals that are realisations of *random processes*, meaning that there is always some uncertainty about a signal prior to conversion. If everything about a signal were known in advance, there would be no need for conversion since the signal could be synthesised perfectly. A random process $X(t)$ defines an ensemble of signals $x_i(t)$ and a rule that assigns a probability to any event associated with the observation of one of these signals.[6] In other words, the random process is a sample space S where each sample s_i corresponds to a signal, as shown in Figure 2. Note that observing the random process at time τ results in a *random variable* $X(\tau)$ having some probability density function $f_X(x)$.

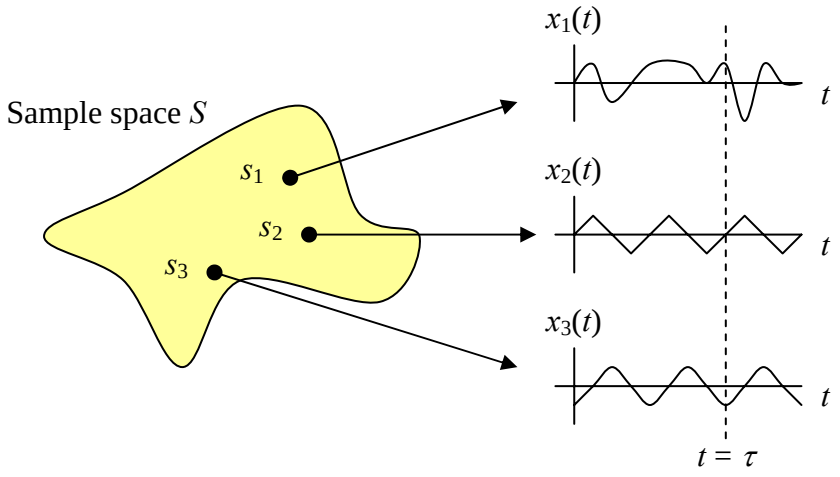


Figure 2. An example of a random process.

Random processes may be further described as *stationary* or *non-stationary*. Let $X(t_1)$, $X(t_2)$, …$X(t_k)$ denote the random variables created by observing the process $X(t)$ at times t_1, t_2, …t_k and let $P_{X(t1),X(t2),\ldots X(tk)}(x_1,x_2,\ldots x_k)$ denote their joint probability density. Shifting all the observation times by T creates k new random variables $X(t_1+T)$, $X(t_2+T)$, …$X(t_k+T)$ with a joint probability density $P_{X(t1+T),X(t2+T),\ldots X(tk+T)}(x_1,x_2,\ldots x_k)$. A random process $X(t)$ is stationary,

in the strict-sense, if $P_{X(t1),X(t2),\ldots X(tk)}(x_1,x_2,\ldots x_k)$ equals $P_{X(t1+T),X(t2+T),\ldots X(tk+T)}(x_1,x_2,\ldots x_k)$ for all T, all k, and all choice of t_1, t_2, $\ldots t_k$.[6] The significance of stationarity is that the statistical characterisation of the process is time-invariant. An example of a stationary process is a sinusoidal oscillator having a fixed frequency and amplitude but a random, uniformly distributed, phase.

Signals produced by a random process may certainly exhibit structure. The process may not use all possible amplitudes equally. Consider a process that generates binary pulses of random duration that begin at random moments in time. Over time, such a process concentrates its probability mass at two amplitudes – the pulse amplitude and the baseline amplitude. Another process may exhibit relationships between signal amplitudes separated by an interval in time. For example, a periodic signal may be predicted perfectly once its period is determined. Two important statistical measures of a process, alluded to by these examples, are its *probability density function* and its *autocorrelation function*.

The probability density function $f_X(x,t)$ of a random process $X(t)$ is the probability that the output of the process, at time t, is within a differential dx of the amplitude x.[6] The autocorrelation function $R_X(t_1,t_2)$ is the correlation $E\{X(t_1)X(t_2)\}$ between two random variables $X(t_1)$ and $X(t_2)$ defined at times t_1 and t_2 on the process $X(t)$.[6] A *white*, or uncorrelated, random process has an autocorrelation function that is zero for all $t_1 \neq t_2$. If the process is stationary, the probability density function is independent of time, simplifying to $f_X(x)$, and the autocorrelation function depends only on the time difference $\tau = t_2 - t_1$, simplifying to $R_X(\tau)$.

Random processes produce sample signals that often undergo further processing. When a time-invariant random process $X(t)$ passes through a linear time-invariant filter, the result is straightforward to compute from the *transfer function* of the filter and the *power spectral density* of the process.[6] The transfer function $H(f)$ is the Fourier Transform of the filter's response $h(t)$ to an impulse. It gives the gain and phase shift of a sinusoid with frequency f that passes through the filter. The power spectral density $S_X(f)$ is the Fourier Transform of the

autocorrelation function $R_X(\tau)$ of the process. It gives the average power delivered by the process in an infinitesimally small frequency band centred on the frequency f. Passing $X(t)$ through $H(f)$ creates a random process $Y(t)$ with a power spectral density $S_Y(f)$ given by Equation 1. The equation may be understood by noting that (1) the phase of a sinusoid does not affect its power and that (2) doubling the amplitude of a sinusoid quadruples its power.

Equation 1. $S_Y(f) = |H(f)|^2 S_X(f)$

Following a simplifying convention employed in the literature, the rest of this report will refer to both a random process $X(t)$ and a realisation $x(t)$ as signals $x(t)$. Any reference to underlying statistics implies a random process and not a realisation. For example, $x(t)$ stands for a process in the expectation $E\{x(t)\}$. The distinction will be made explicit when necessary.

B. Sampling and quantisation

An analogue signal $x(t)$ takes on values at any time t in a given interval. Similarly, each value that is taken may be any amplitude x in a given range. When the analogue signal is converted to a digital signal $y[n]$, it is *sampled* in time and *quantised* in amplitude. Sampling captures the signal at a finite number of moments, indexed by the integer n, in a given interval. Likewise, quantisation represents the continuous range of values that the signal may take with a finite set of amplitude levels y. Figure 3 gives an example of sampling and quantisation.

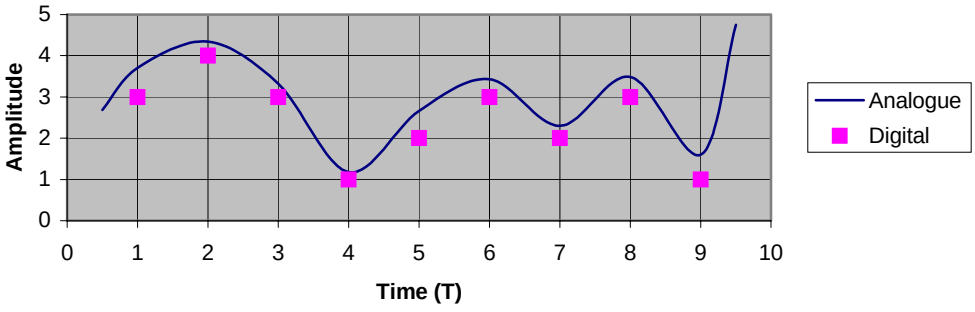


Figure 3. An example of sampling and quantisation, $y[n] = \lfloor x(nT) \rfloor$.

Sampling a signal $x(t)$ at intervals of T (denoted $x[n]$ in discrete-time) does not corrupt any information if the sampling frequency $f_S = 1/T$ is more than twice the bandwidth f_B of the signal.[6] However, quantisation of a signal does corrupt some information. Uniform quantisation of x into y can be modelled by a linear function with an error term, $y(x) = Gx+e(x)$.[7] Figure 4 gives an example of this model, where the quantisation levels are placed symmetrically about the axis with a uniform spacing of $Q = 2$, called the *quantisation step-size*, and where the quantisation process merely rounds the amplified input Gx to the nearest quantisation level. It is typical in the literature to use a gain G of unity, as is done in the figure. Note that, if the input to the quantiser does not saturate, the error term $e(x)$ is bounded by $\pm Q/2$.

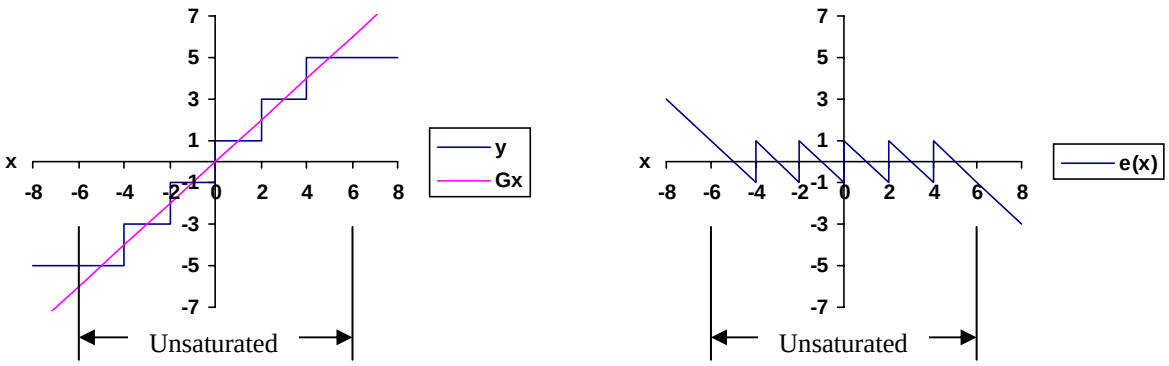


Figure 4. Linear model of quantisation, $y = Gx+e(x)$.

Using the linear model of quantisation with unity gain, the processes of sampling and quantisation may be represented by $y[n] = x(nT)+e(x(nT)) = x[n]+e[n]$. If the input signal $x[n]$ changes from sample to sample by random amounts on the order of Q, without saturating, then the error signal $e[n]$ will be white, or uncorrelated from moment to moment, and will have a uniform probability density in the interval $-Q/2$ to $+Q/2$.[7] Furthermore, if the error has statistical properties that are independent of the signal then it may be represented by a noise source.[7] Figure 5 depicts the linear discrete-time model of quantisation (which implicitly includes sampling) that results from these assumptions.

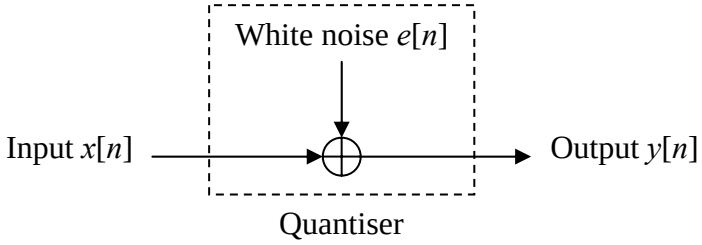


Figure 5. Linear discrete-time model of quantisation.

The noise power σ_e^2, or expected square value of $e[n]$, is calculated in Equation 2 by treating the quantisation error as uniformly distributed in an interval of size Q, centred on zero, with a probability $1/Q$.[7] Note that the expected value μ_e of $e[n]$, or $E\{e[n]\}$, is zero.

Equation 2. $$\sigma_e^2 = E\left\{e^2[n]\right\} = \frac{1}{Q}\int_{-Q/2}^{Q/2} e^2\,de = \frac{Q^2}{12}$$

Since the quantisation error is treated as white, or completely uncorrelated, noise then the power spectral density $S_e(f)$ of the noise $e[n]$ is a constant. In addition, the noise power given by Equation 2 must all fall in the band $|f| \le f_S/2$ of the power spectral density, assuming a two-sided spectral representation, since $e[n]$ is sampled at f_S.' Equation 3 calculates $S_e(f)$.

Equation 3. $$\sigma_e^2 = \int_{-f_S/2}^{f_S/2} S_e(f)\,df = S_e(f)\int_{-f_S/2}^{f_S/2} df = S_e(f) f_S \quad \rightarrow \quad S_e(f) = \frac{\sigma_e^2}{f_S}$$

A simple way to reduce the noise power and its spectral density is to reduce the value of the quantisation step-size Q. However, a reduction in Q corresponds to a reduction in the unsaturated range of the quantiser. If an input signal causes saturation, the quantisation error is no longer bounded by $\pm Q/2$ or the above equations. Therefore, to permit the same dynamic range of input signals, the number of quantisation levels have to increase if the step-size decreases. Nevertheless, increasing the number of quantisation levels increases the hardware complexity and the quantiser non-linearity (i.e. the model in Figure 5 is less accurate since the staircase function in Figure 4 is less likely to have regularly spaced corners).

C. Optimal quantisation

The previous section modelled quantisation as the addition of statistically independent white noise to the quantiser input. However, the distortion introduced by quantisation is wholly dependent on the input signal. Linear assumptions, which ignore this dependence, may hinder the use of source coding to improve quantisation.

As shown in Figure 6, a general K-level quantiser is a non-linear mapping from a set $\{Q_i\}$ of K disjoint subsets $\{Q_1,Q_2,\ldots Q_K\}$ or *partitions* of the input domain or x-axis, the union of which covers the entire axis, to a set $\{q_i\}$ of K points $\{q_1,q_2,\ldots q_K\}$ or *quanta* of the output domain or y-axis.[8] Quantisation, given in Equation 4, determines to which partition Q_i of the x-axis the input belongs and then outputs the quantum q_i on the y-axis. Note that the quantisation index i may be encoded by a binary integer that is $N = \lceil \log_2 K \rceil$ bits long.

Equation 4. $$y(x) = q_i \in \{q_1, q_2, \mathrm{K}\ q_K\} \mid x \in (Q_i \in \{Q_1, Q_2, \mathrm{K}\ Q_K\})$$

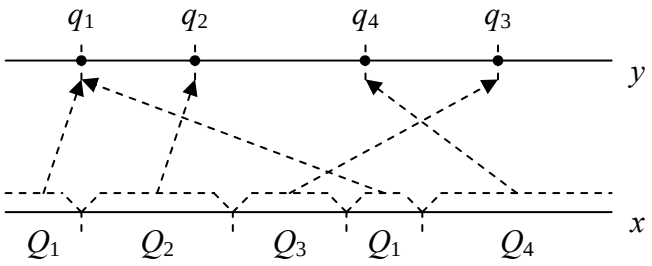


Figure 6. An example of generalised quantisation.

Assuming ideal sampling, the output sample $y[m]$ of the quantiser at time $n = m$ depends non-linearly on the single input sample $x[m]$. If the input $x[n]$ is stationary then the output $y[n]$ must also be stationary, meaning that the statistical characterisation of the input or output is time-invariant. Equation 5 gives the power σ_e^2 of the quantisation noise $e[n] = y[n]\text{-}x[n]$, where $f_x(x)$ is the time-invariant probability density function of the input.[8] The equation shows that the noise power depends on the probability distribution of the input, the choice of quanta, the choice of partitions, and the assignment of quanta to partitions.

Equation 5. $\sigma_e^2 = E\{e^2[n]\} = \int_{-\infty}^{\infty} (y(x) - x)^2 f_x(x) dx = \sum_{i=1}^{K} \int_{Q_i} (q_i - x)^2 f_x(x) dx$

If two quanta q_a and q_b are equal then it is possible to reduce the number of quantisation levels K without affecting the noise power by joining Q_a and Q_b and eliminating one of the redundant quanta. Similarly, if the probability density function is zero in a partition Q_i then K may be reduced without changing the noise by adding Q_i to any other partition and discarding the quantum q_i. If either of the previous conditions hold then the noise power may be reduced by dividing a partition into two and creating a new quantum level. Thus, any quantiser that duplicates quantum levels or that has a partition with no probability mass is not optimal.[8]

Consider a quantiser that does not have the redundancies described above. Holding everything else constant, the choice of quanta may be optimised to reduce the noise power. Because each integral in the sum of Equation 5 is independent, the power may be minimised by minimising each integral. Equation 6 gives the choice of quantum that minimises each integral.[8] This result may be derived using standard techniques in Calculus, especially by observing that the given integral resembles a moment of inertia of the region Q_i, with mass density $f_x(x)$, around the point q_i. Thus, the integral is minimised if q_i is the centre of mass.

Equation 6. $\int_{Q_i} (q_i - x)^2 f_x(x) dx$ is a minimum if $q_i = \dfrac{\int_{Q_i} x f_x(x) dx}{\int_{Q_i} f_x(x) dx}$

Now suppose that the location of quanta is fixed but the choice and assignment of partitions are optimised instead. From Equation 5, if a point x with probability mass $f_x(x)dx$ belongs to partition Q_i then it contributes $(q_i\text{-}x)^2 f_x(x)dx$ to the noise power. Therefore, the point x should be assigned to the partition Q_i with the lowest corresponding value $(q_i\text{-}x)^2$ to minimise the noise. The optimal partitions $\{Q_i\}$ are thus given by Equation 7.[8]

Equation 7. $Q_i = \{x \mid (q_i - x)^2 < (q_j - x)^2, i \neq j\}$

The K optimal partitions $\{Q_i\}$ must therefore be contiguous intervals whose endpoints $x_1, x_2, \ldots x_{K\text{-}1}$ bisect the segments between adjacent quantum levels, as in Equation 8.[8]

Equation 8.
$$\begin{aligned} Q_1 &= \{x \mid -\infty < x \le x_1\} \\ Q_2 &= \{x \mid x_1 < x \le x_2\} \\ &\text{M} \\ Q_K &= \{x \mid x_{K-1} < x < \infty\} \end{aligned} \quad \text{where} \quad x_i = \frac{q_i + q_{i+1}}{2}$$

A quantiser may therefore be specified by a $2K$-1 dimensional vector $\mathbf{p}$ of the ordered parameters $q_1 < x_1 < q_2 < \ldots x_{K\text{-}1} < q_K$. The noise power is then expressed by Equation 9.[8]

Equation 9. $$\sigma_e^2(\mathbf{p}) = \sum_{i=1}^{K} \int_{x_{i-1}}^{x_i} (q_i - x)^2 f_x(x)dx, \quad \begin{matrix} x_0 = -\infty \\ x_K = \infty \end{matrix}$$

To find the quantiser configuration that minimises the noise power, all stationary points of $\sigma_e^2(\mathbf{p})$ must be found by simultaneously solving the set of $2K$-1 equations $\nabla\mathbf{p} = \mathbf{0}$ generated by Equation 6 and Equation 8.[8] These stationary points comprise local minima of the noise power function and the global minimum will occur at one of these points. An example of optimal quantisation is given in Figure 7. Note that the quanta are not uniformly spaced.

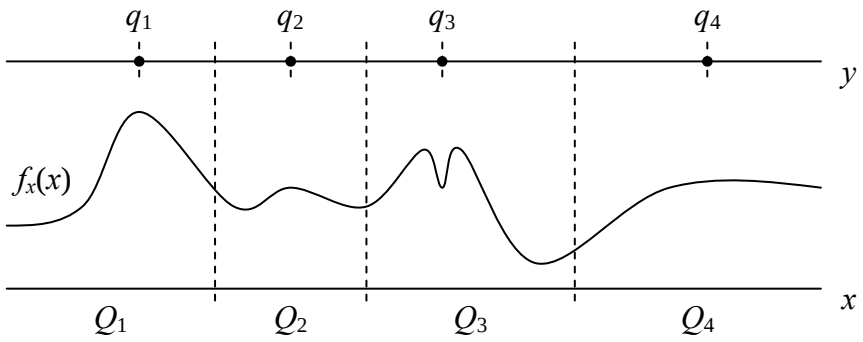


Figure 7. An example of optimised quantisation.

This chapter reviewed background theory on random processes, on sampling, and on linear and non-linear models of quantisation. Chapter III investigates digital speech coding, referring to the material on sampling and quantisation. Chapter IV relies on the theory of random processes to describe the effect of oversampled modulation on quantisation noise.

III. Source Coding of Speech Signals

A. Pulse code modulation

In the last two decades, source coding of digital speech has been extensively researched for the purpose of increasing communication capacity by speech compression. To explore source coding, this chapter reviews the theory of speech coding and develops some new ideas.

The simplest coding, termed *pulse code modulation* (PCM), represents analogue speech by a sequence of quantised pulses having integer codes. Figure 8 shows a high-quality 128 kbps speech signal $x[n]$ digitised by sampling an analogue signal $x(t)$ at 8 kHz and uniformly quantising each sample using 16 bits. To facilitate comparison, the simulation results given in this chapter refer to this test signal. The signal was chosen randomly from a commercial speech database and was not used to develop or optimise any coding algorithm.[9]

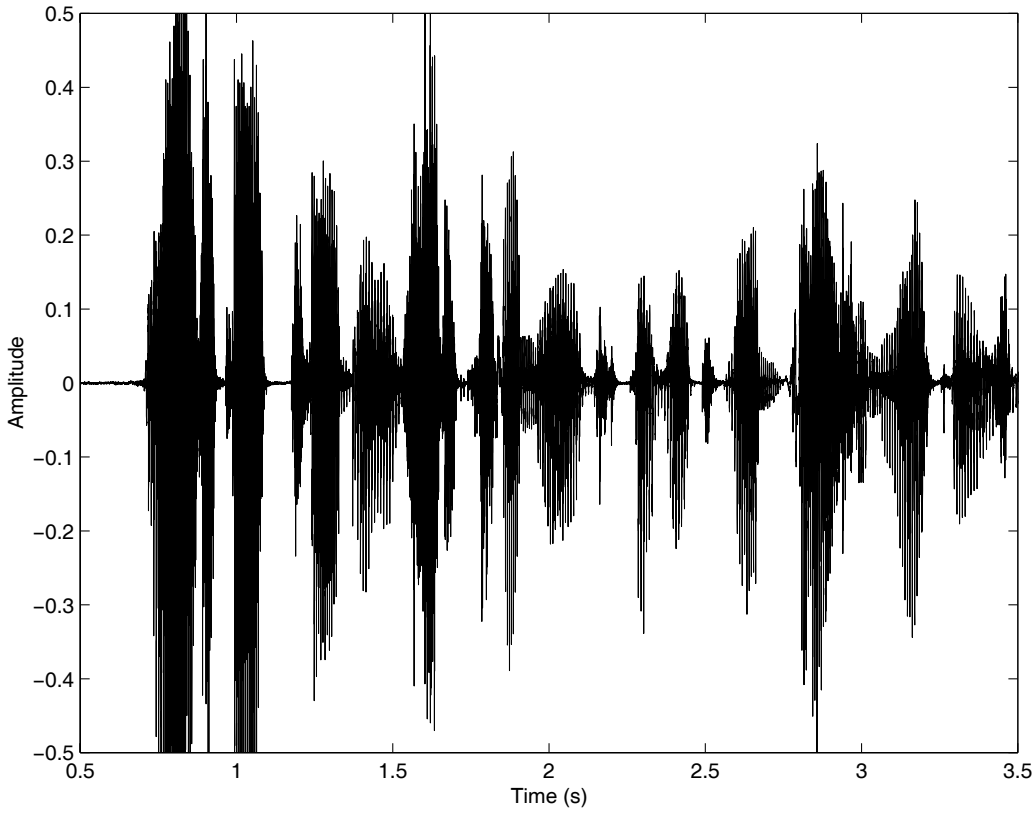


Figure 8. A 128 kbps uniformly quantised PCM speech signal.

A 128 kbps digital speech signal may be compressed by *waveform coding*, or quantising each digital sample using fewer bits than the original quantisation. Uniform quantisation (discarding the least significant bits in each sample) is optimal only if the signal amplitudes are uniformly distributed in the range of possible values. However, the left plot in Figure 9 shows that the speech amplitudes in Figure 8 are not uniformly distributed. In fact, speech samples generally have a probability distribution close to a Laplacian distribution.[10]

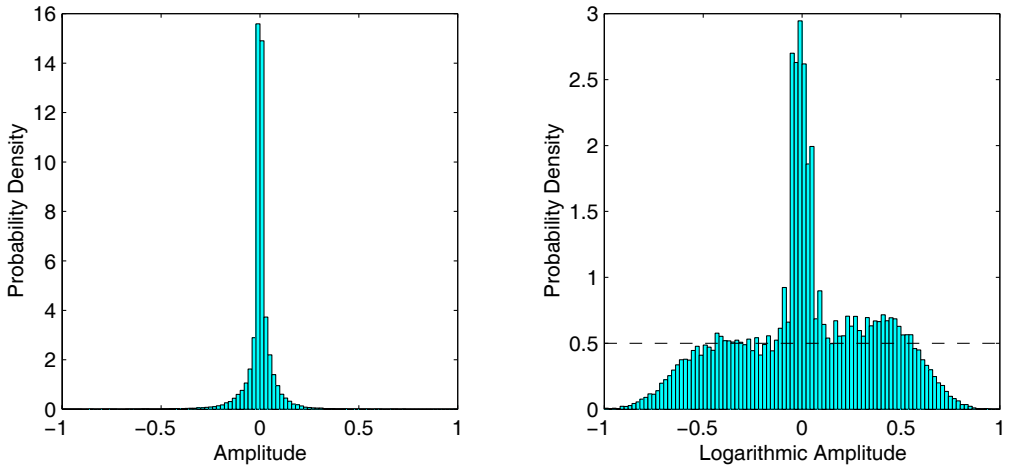
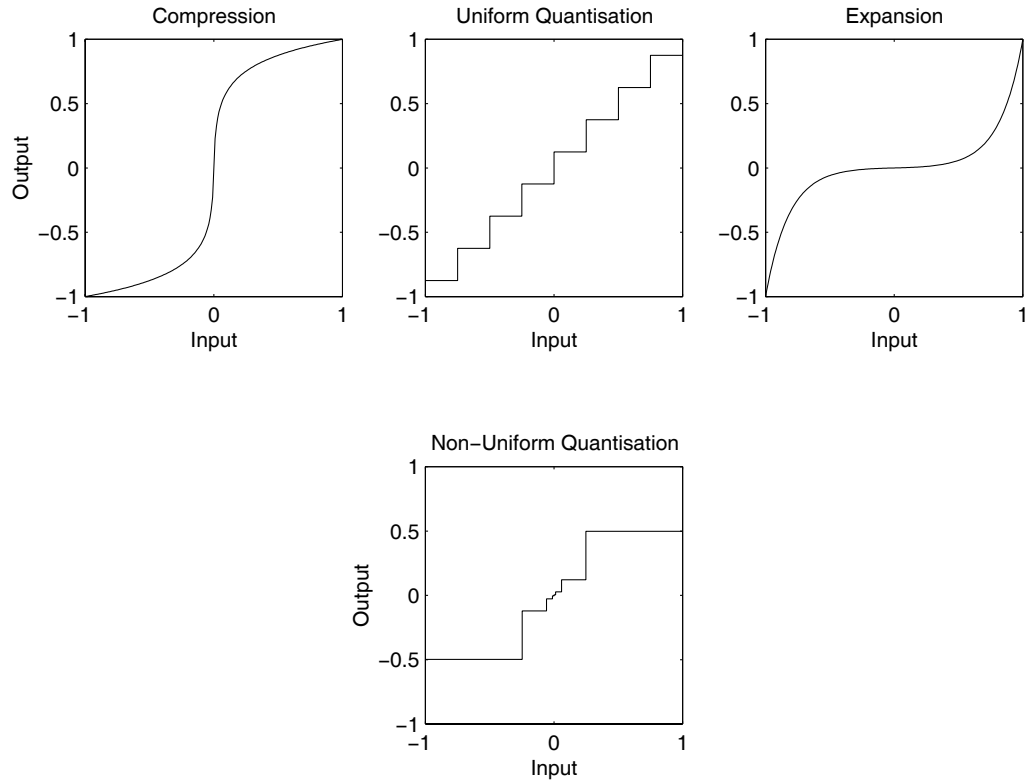


Figure 9. Probability densities of speech and μ-law compressed speech.

Given the non-uniform distribution of speech amplitudes, uniform quantisation would waste quanta in regions of low probability. *Logarithmic PCM* encoders apply a logarithmic function to the speech waveform and then quantise the result uniformly. A μ-law logarithmic encoder compresses the input waveform $x[n]$ using Equation 10 and then quantises the result $x_\mu[n]$ uniformly. The right plot in Figure 9 shows the probability distribution of the μ-law compressed waveform for the speech in Figure 8. Apart from values near zero, this distribution is nearly uniform. Following convention, the compression parameter μ equals 255.[10]

Equation 10. $$x_\mu[n] = x_{\max} \frac{\log\left(1+\mu\dfrac{|x[n]|}{x_{\max}}\right)}{\log(1+\mu)} \operatorname{sgn}(x[n]), \quad x_{\max} = \max\{|x[n]|\}$$

To decode the output of a logarithmic speech encoder, an inverse logarithmic function is applied. A μ-law decoder expands the compressed and quantised μ-law waveform $y_\mu[n]$ using Equation 11, resulting in an approximation $y[n]$ to the original signal. Such a logarithmic compander effectively implements non-uniform quantisation, as illustrated in Figure 10. PCM coding using uniform quantisation without companding is also called *linear PCM*.



Equation 11. $$y[n] = \frac{x_{\max}}{\mu}\left((1+\mu)^{\frac{|y_\mu[n]|}{x_{\max}}} - 1\right)\operatorname{sgn}(y_\mu[n]), \quad y_\mu[n] = x_\mu[n] + e[n]$$

Figure 10. Equivalence between companding and non-uniform quantisation.

There are other forms of logarithmic speech coding besides μ-law compression, an American standard. A-law compression, a European standard, uses a piecewise encoding

function, linear for small inputs but logarithmic for large ones, given in Equation 12.[10] However, a plot of the A-law function is nearly identical to the μ-law function plotted in Figure 10.

Equation 12.
$$x_A[n] = \begin{cases} \dfrac{Ax[n]}{1+\log A} & 0 \le |x[n]| \le \dfrac{x_{\max}}{A} \\ x_{\max} \dfrac{1+\log\left(A|x[n]|/x_{\max}\right)}{1+\log A}\operatorname{sgn}(x[n]) & \dfrac{x_{\max}}{A} \le |x[n]| \le x_{\max} \end{cases}$$

Given a perfect Laplacian distribution of amplitudes, m-law compression is optimal.[10] Unlike the μ-law and A-law functions, the m-law function, shown in Equation 13, requires the standard deviation of the amplitudes in addition to the maximum amplitude.

Equation 13.
$$x_m[n] = x_{\max} \frac{1-e^{-m|x[n]|/x_{\max}}}{1-e^{-m}}\operatorname{sgn}(x[n]), \quad m = \sqrt{2}\frac{x_{\max}}{3\sigma_x}$$

Figure 11 plots the signal-to-noise ratio (SNR) of linear, μ-law, A-law, and m-law PCM, for coding the signal in Figure 8, against the bit rate. The bit rate equals the product of the number of bits per sample, ranging from one to eight, and the sampling frequency of 8 kHz.

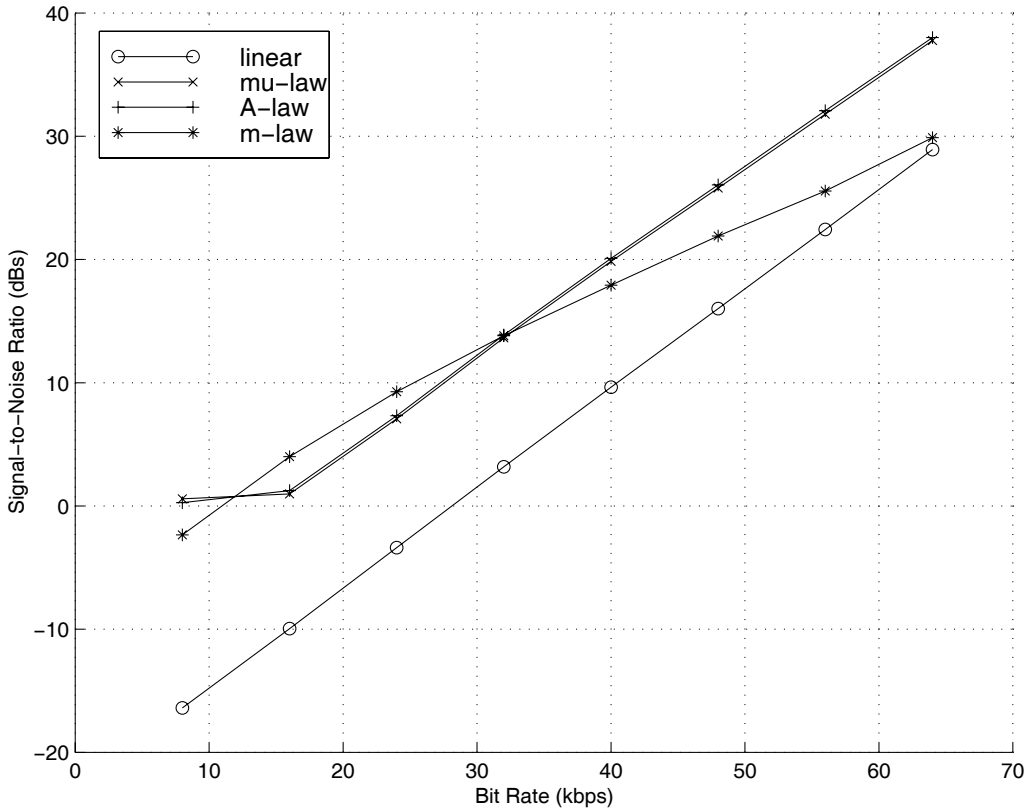


Figure 11. Signal-to-noise ratios for linear and logarithmic PCM.

Linear PCM is sub-optimal by at least 10 dB SNR for all bit rates, not surprising given the probability distribution of speech amplitudes. The μ-law and A-law coders give the best (and similar) results. Whereas linear, μ-law, and A-law typically exhibit slopes of over 6 dB per quantisation bit, m-law displays more variation in slope. Its performance lies between the linear and other logarithmic coders at high bit rates but is superior for some low bit rates.

Listening tests for linear and logarithmic coded speech do not fully support conclusions drawn from SNR analysis. This is mainly due to the aural noise masking property of the human auditory system.[10] Noise of a certain energy is less noticeable in periods of high speech energy than in periods of low energy. The *segmented signal-to-noise ratio* (SEGSNR), plotted in Figure 12, tries to account for this aural property by averaging the SNR dB values computed in sequential speech frames of 16 ms each, the approximate syllabic duration.[10]

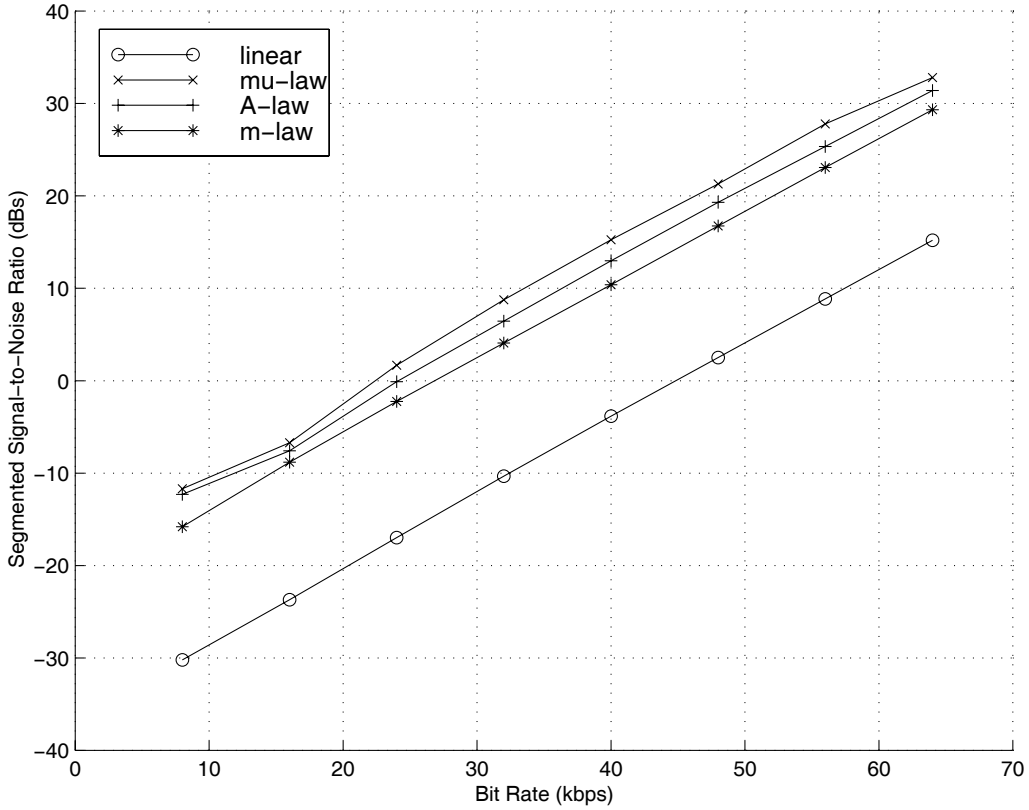


Figure 12. Segmented signal-to-noise ratios for linear and logarithmic PCM.

The SEGSNR results in Figure 12 consistently place μ-law PCM coding at the top, agreeing with listening tests, although there is generally little difference between the logarithmic coders. Linear PCM is sub-optimal at all bit rates by about 20 dB SEGSNR. Informal listening tests suggest that SEGSNR differences on the order of 1 dB are discernible and that 0 dB SEGSNR is a useful threshold for intelligibility, though not clarity. Thus, μ-law coding is intelligible at and above 24 kbps. However, it becomes clear at 32 kbps and good at 40 kbps (toll-quality or telephone-quality speech is arguably achieved at 48 kbps).

B. Optimised non-uniform quantisation

Although μ-law companding avoids sub-optimal uniform quantisation, it may still not realise optimal quantisation. Optimal quantisation $y[n]$ of a signal $x[n]$, using N bits per sample, is achieved by minimising the power σ_e^2, given by Equation 5, Section II.C, of the quantisation

noise $e[n] = y[n]-x[n]$. Minimising σ_e^2 results in two conditions: (1) the quantum q_i must be the centroid of the probability mass $f(x)dx$ over $x \in Q_i$ and (2) the domain Q_i must be a single contiguous interval with endpoints halfway between q_i and the nearest two quanta.

The K-means clustering algorithm may be used to optimise the $K = 2^N$ quanta $\{q_i\}$ for T samples of a signal $x[n]$.[11] An initial set of quanta are chosen, either randomly or uniformly. The T samples are then divided into K partitions by assigning a sample to the partition Q_i when it is nearest to the quantum q_i. This guarantees condition (2) above. The mean μ_i of each partition is computed, which equals the centroid of probability mass $E\{Q_i\}$ in the partition. Condition (1) is satisfied if $\{\mu_i\} = \{q_i\}$. Otherwise, the K means are taken as new quanta and the procedure is iterated. Assuming the initial quanta are ordered, which allows binary search, each iteration may be implemented with $O(N \cdot T)$ arithmetic operations.

Although each iteration of the K-means algorithm is quick, many iterations are often needed. To reduce the number of iterations, two novel heuristics were devised that choose better initial quanta. The T samples are first divided into two partitions, one containing all samples less than the mean and the other containing the rest. Using the same procedure, the *greedy* heuristic divides the partition having the greatest variance into two, yielding three partitions, whereas the *fair* heuristic simply divides both partitions into two, doubling the total number of partitions. The greedy and fair algorithms are iterated respectively until K partitions remain, at which point the means of the partitions are returned as quanta. The time required by the greedy heuristic depends on the data but equals, on average, $O(N \cdot T)$ arithmetic operations whereas the fair heuristic simply requires $O(N \cdot T)$ arithmetic operations.

Figure 13 plots the SNR versus the bit rate for PCM with optimised non-uniform quantisation (PCM-ONQ), using the K-means algorithm, and for PCM with the greedy and fair heuristics but no optimisation. The results of μ-law PCM are repeated for comparison.

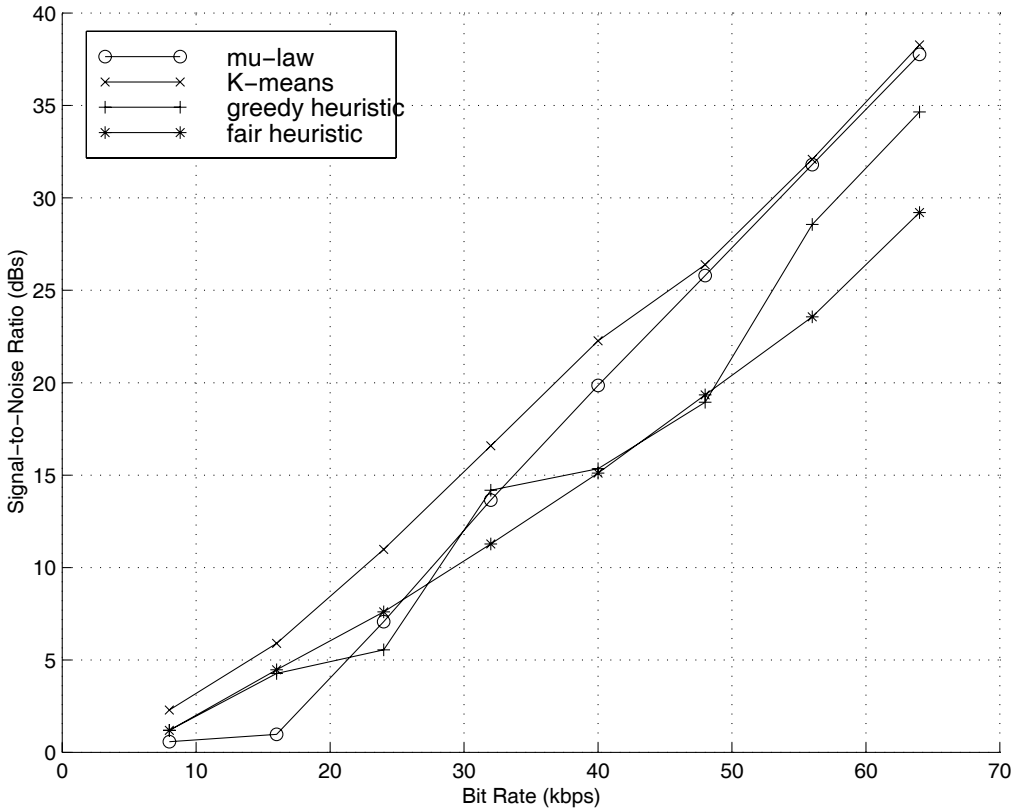


Figure 13. Signal-to-noise ratios for PCM with optimised quantisation.

As expected, PCM-ONQ gives better SNR results, at any bit rate, than μ-law PCM. Note that the heuristics, especially the greedy one, produce good results without any optimisation. The slope of the greedy heuristic changes frequently compared to that of the fair. The fact that the fair heuristic sometimes performs better than the greedy one is not surprising. Both heuristics create K partitions and use the partition means, or centroids, as the quanta. Such a scheme does not guarantee that the partition boundaries lie halfway between adjacent quanta whereas the PCM quantiser always performs optimal nearest-neighbour quantisation. The extent of boundary sub-optimality depends on the heuristic and data.

Once again, listening tests do not correlate well with the SNR results of Figure 13. Therefore, the SEGSNR results for μ-law, PCM-ONQ, and the heuristics are given in Figure 14. The duration of the speech frames for segmented calculation is 16 ms, as before.

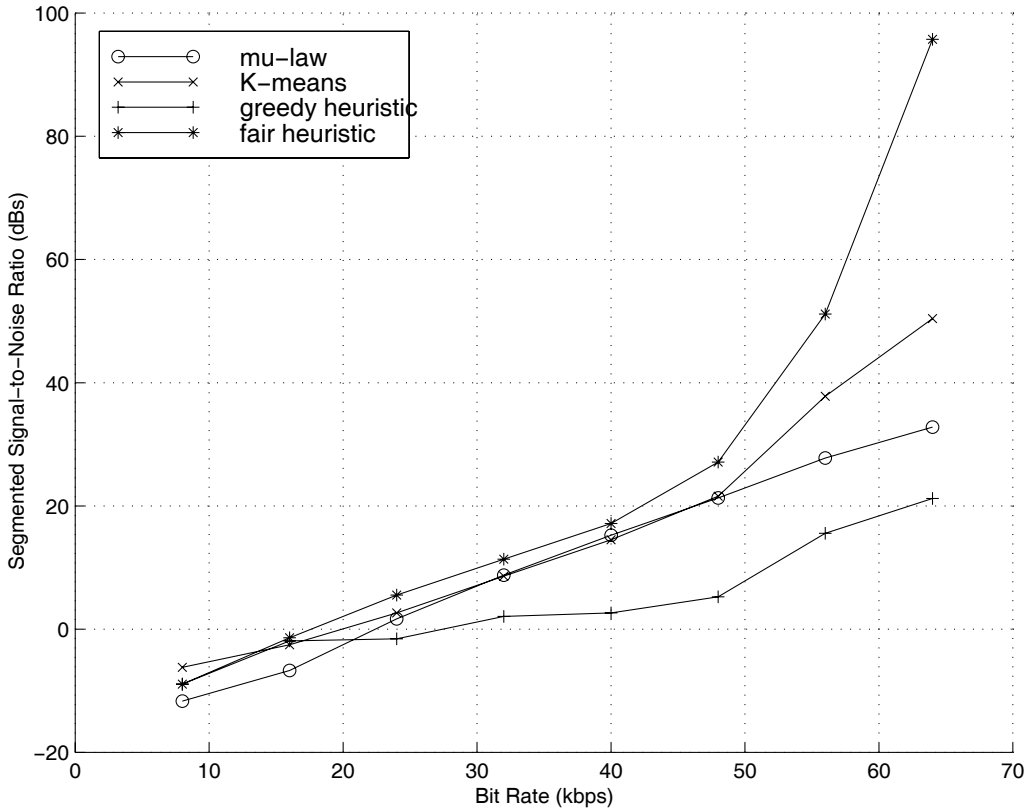


Figure 14. Segmented signal-to-noise ratios for PCM with optimised quantisation.

According to Figure 14, the fair heuristic with no optimisation yields a better PCM coding of speech than μ-law or K-means optimised PCM. The greedy heuristic gives the worst performance. A reason for this conclusion is that the K-means algorithm minimises the total noise power and hence the SNR, as shown in Figure 13, which is somewhat at odds with minimising the SEGSNR, or average SNR, although the latter is a better measure of speech quality. Since the fair heuristic partitions the amplitude space without regard to variance, partitions with extreme outliers will not steal bits away from smaller, but perceptively more important, partitions. Though it is closer than μ-law, the fair heuristic still does not achieve intelligible speech at 16 kbps. However, its 24 kbps speech quality is more than intelligible.

The heuristics and K-means optimisation may be applied to improve the PCM coding of signals other than speech, especially those that have non-uniform probability distributions.

C. Adaptive non-uniform quantisation

The assumption so far is that the statistical process underlying the signal $x[n]$ is stationary, otherwise the probability density function $f(x)$ in Equation 5 would be a function $f(x,n)$ of sample time n. Speech, however, is a non-stationary process but may be modelled as a short-time stationary process.[10] Applying the K-means algorithm to an entire speech signal produces a non-uniform quantiser that is similar to a logarithmic quantiser. To improve 16 kbps speech coding, an adaptive PCM coding scheme (PCM-ANQ) was devised, whereby the K-means algorithm is used to optimise the quantisation of sequential 16 ms speech frames.

Figure 15 shows the time-varying quanta of PCM-ANQ, when applied to the speech signal of Figure 8. The four non-uniform quanta track different energy regions of the signal and each speech sample is quantised to the nearest time-varying quantum using two bits. Coding the speech signal by this method results in clear, not just intelligible, speech at 16 kbps (plus side information). PCM-ANQ was compared to the established technique of adaptive gain quantisation (PCM-AGQ), where each frame is uniformly quantised after amplification based on the frame variance.[10] Adaptive m-law quantisation (PCM-AML) was also investigated, where Equation 13 was applied to quantise each frame, with time-varying values for $x_{\max}$ and σ_x. PCM-ANQ was noticeably better than the others. For the example in Figure 8, SEGSNRs of 8.14, 8.29, and 9.72 dB were obtained using PCM-AGQ, PCM-AML, and PCM-ANQ respectively, quantising with two bits per sample.

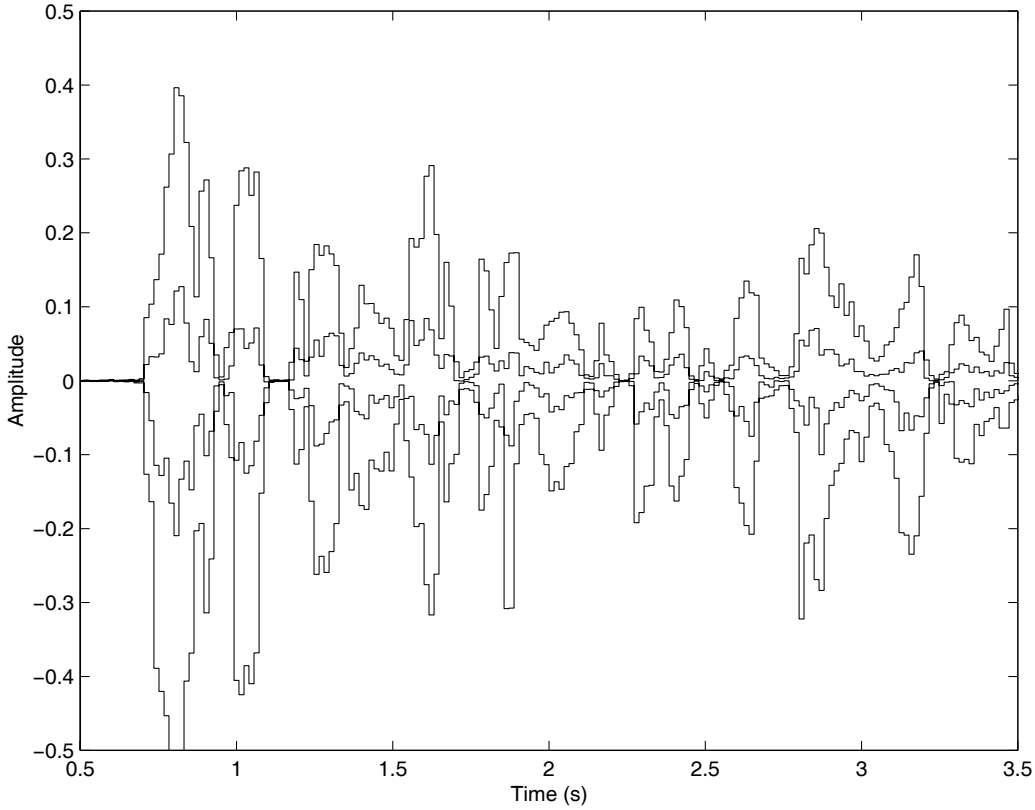


Figure 15. Time-varying quanta (side information) for 16 kbps PCM-ANQ.

One problem with PCM-ANQ is that a lot of side information is generated. Whereas PCM-AGQ needs one parameter per frame and PCM-AML needs two parameters per frame, PCM-ANQ needs one parameter per quantum per frame. This side information can be halved, using prior knowledge of speech, by equating the quanta above and below zero amplitude. Complete coders, that operate using 17 kbps, may then be implemented for all three methods. Encoding the side information of PCM-AGQ, PCM-AML, and PCM-ANQ using 1 kbps each degrades their SEGSNR results to 8.11, 8.07, and 8.73 dB respectively.

D. Scalar versus vector quantisation

Another problem with PCM-ANQ is that it does not exploit the correlations in speech. Speech is composed of voiced, or high-energy and regular pitch, sounds and unvoiced, or low-energy and noise-like, sounds.[10] Generally, voiced sounds are vowels and unvoiced ones are conso-

nants. Consecutive samples taken during voiced sounds are highly correlated. Figure 16 plots the absolute value of the correlation coefficient obtained from the set of paired consecutive samples in each 16 ms frame of the speech in Figure 8. The correlation coefficient, of two random variables X and Y, ranges from –1 to +1 and equals the covariance $E\{(X-\mu_X)(Y-\mu_Y)\}$ of the variables divided by the product $\sigma_X\sigma_Y$ of their standard deviations.

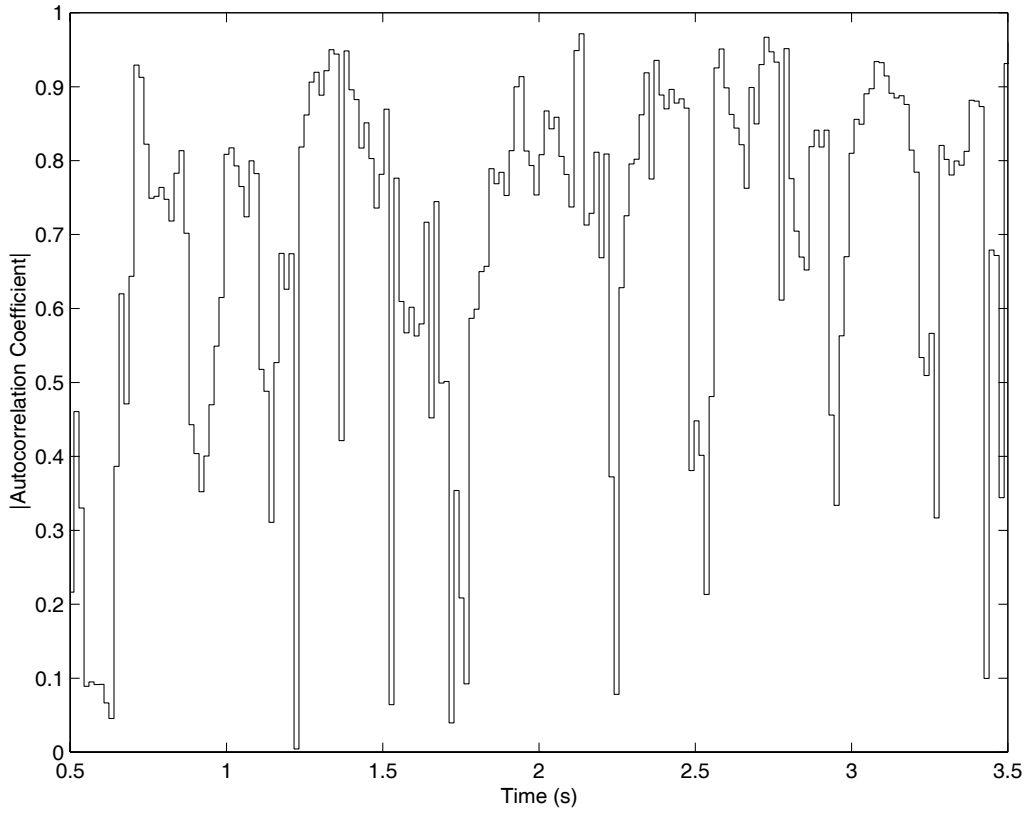


Figure 16. Time-varying correlation of consecutive speech samples.

Listening tests confirm that frames with an absolute correlation coefficient near unity are voiced and that frames with a coefficient near zero are unvoiced. Applying a threshold of 0.5 to the absolute coefficients in Figure 16 shows that at least half of speech is correlated. Adaptive non-uniform vector quantisation (VCM-ANQ) was devised to exploit the correlation of consecutive samples in voiced speech. The K-means algorithm is used to cluster the pairs of consecutive samples, that make up each 16 ms speech frame, into K two-dimensional

quanta. Figure 17 depicts the difference between scalar quantisation, at one bit per sample, and vector quantisation, at two bits per two samples, for a voiced and unvoiced speech frame.

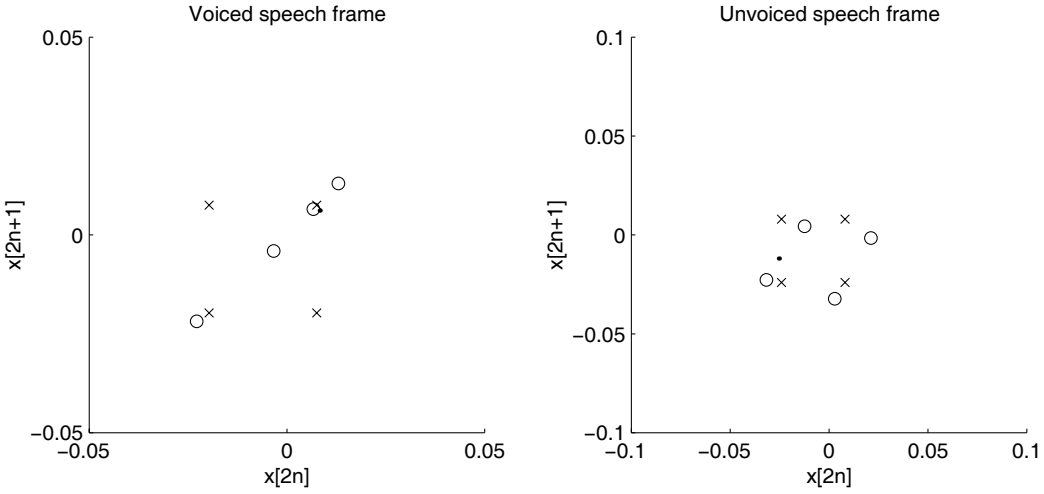


Figure 17. Scalar (X) and vector (O) quantisation of voiced and unvoiced speech.

As seen in Figure 17, vector quantisation can take advantage of simple correlations in voiced speech frames. Ignoring side information, VCM-ANQ at 12 kbps (3 bits per 2 samples) gives 10.33 dB SEGSNR for the speech in Figure 8, better than PCM-ANQ at 16 kbps. However, the side information for the former, at 16 parameters (i.e. eight two-dimensional quanta) per 16 ms frame, is much worse than the latter. The side information may be encoded in 4 kbps by using eight bits per parameter and by making the quanta symmetric about the origin, which halves the number of parameters. At 16 kbps, a complete VCM-ANQ coder results in 9.03 dB SEGSNR (a degradation from 10.33 dB above) whereas μ-law coding gives -5.16 dB SEGSNR. Better encoding of the side information may reduce the degradation.

One way to improve upon VCM-ANQ would be to model each speech frame by a few statistical parameters, such as the covariance matrix of the set of consecutive sample pairs in the frame (the matrix size can be halved due to self-similarity). The optimal frame quanta may then be estimated from these parameters alone. Rather than defining an analytical model for speech statistics, as is done in m-law quantisation, a neural network may be trained to map the

statistical parameters into quanta by minimising the quantisation error during training. More work needs to be done but tests indicate that some statistical parameters, like variance, may be well predicted from previous quantised samples, allowing a reduction of side information.

In summary, adaptive non-uniform quantisation codes speech better than logarithmic or adaptive uniform quantisation. Vector, as opposed to scalar, adaptive non-uniform quantisation can exploit simple speech correlations. These methods may be applied to short-time stationary signals other than speech to reduce quantisation noise.

E. Linear predictive coding

Although the adaptive non-uniform quantisation techniques, developed here, are better than classical waveform coding techniques for medium bit rate coding (12 to 18 kbps), techniques based on *linear predictive coding* (LPC) are capable of coding speech with an acceptable quality at low bit rates (2 to 6 kbps) and are therefore worth exploring.

The GSM standard provides toll-quality speech, indistinguishable from a good analogue telephone, at 16 kbps.[12] Toll-quality coders operating at 8 kbps are currently being developed. GSM, and other modern speech coders, use linear prediction. Put simply, these coders rely on the high correlation in speech to accurately predict the current sample from previous samples. Unfortunately, the SEGSNR measure that was described earlier is not useful for describing the performance of these coders because they use properties of the human auditory system, such as non-uniform frequency sensitivity, that the measure does not account for.

Figure 18 shows the typical architecture for an LPC encoder.[10] A linear filter $H(z)$ is constructed, by analysis of the speech autocorrelation function, that estimates the current speech sample from a finite number of previous samples. The *excitation* $d[n]$, which is the difference between the actual speech $x[n]$ and the prediction $x_{est}[n]$, is encoded using a finite number of bits and then decoded to produce an approximate excitation $c[n]$. The decoded speech $y[n]$ is produced by adding $c[n]$ to $x_{est}[n]$. Note that, irrespective of the predictive filter,

the decoded speech will equal the original if the excitation is encoded perfectly. In other words, the error $e[n]$ equals $y[n]$-$x[n]$, which simplifies to $c[n]$-$d[n]$.

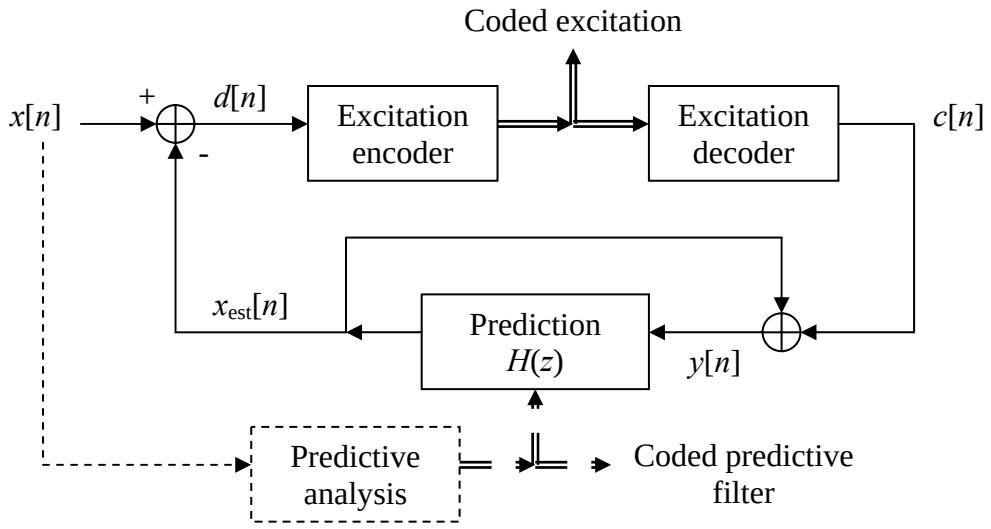


Figure 18. Speech encoding using linear prediction.

The LPC decoder, which is a subsection of the encoder, is shown in Figure 19. Both the encoder and decoder use the previous coded samples to predict the current speech sample. If the encoder were to use the previous actual samples in its prediction, which are not available to the decoder, then the error $e[n]$ would not simply depend on the excitation coding error.

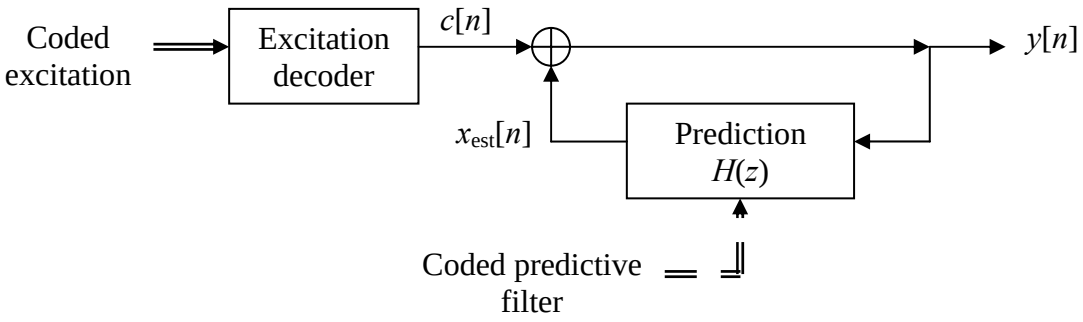


Figure 19. Speech decoding using linear prediction.

The advantage of LPC is that the dynamic range of the differential signal $d[n]$ is less than the dynamic range of the speech $x[n]$, meaning that the domain of possibilities to encode is smaller.[10] The predictive filter $H(z)$ may be kept constant, as in *differential pulse code modulation* (DPCM), but is usually updated slowly, requiring the transmission of side infor-

mation. Such a time-varying filter models the slowly changing human vocal tract, which reflects and absorbs sounds thereby filtering the excitations generated by the glottis.[10]

Following a design given in the literature, LPC was implemented for the purposes of the next section.[12] An all-pole predictive filter was used to predict the current sample by a linear combination of the ten previous samples and of three consecutive samples at a longer delay. The speech was divided into a sequence of 20 ms frames and, for each frame, autocorrelation estimates were used to generate systems of linear equations. Solving these equations provided the filter coefficients and delay parameter that minimised the prediction error in that frame. The ten short-term filter coefficients were each quantised logarithmically using 47 bits in total (fewer bits were allocated to the coefficients of higher delays). The three long-term filter coefficients and the delay were quantised uniformly with 12 and 7 bits respectively. These bit allocations permitted the time-varying LPC filter to be encoded using 3.3 kbps. Converting speech signals to excitation signals, this filter typically reduced the dynamic range by 10 dB.

Although LPC theory has changed very little in recent years, the design of the excitation encoder continues to be actively researched and many alternatives are in use.[12] The simplest excitation encoder is a uniform quantiser, resulting in *adaptive differential pulse code modulation* (ADPCM). The GSM standard encodes the excitation using *regular pulse excitation* (RPE). Speech frames of 20 ms, the update rate for LPC, are divided into four 5 ms sub-frames. The excitation in each sub-frame is modelled as a sequence of thirteen equally spaced pulses.[12] However, the amplitude of each pulse and the phase of each sequence are optimised. Including LPC and error protection, the total GSM bit rate is 16 kbps for toll-quality speech.

F. Code-excited linear prediction

Good quality speech coding may be realised below 8 kbps using *code-excited linear prediction* (CELP).[12] The CELP encoder operates on the excitation signal produced by LPC and, like GSM, typically divides a 20 ms LPC frame into four 5 ms sub-frames. These sub-frames

contain 40 samples each, given an 8 kHz sampling rate, and are called *excitation vectors*. The efficiency of CELP in encoding the excitation vectors comes from three principles: (1) CELP models the excitation vectors as random Gaussian vectors, with a mean of zero; (2) CELP maintains a stochastically-populated codebook of Gaussian vectors and exhaustively searches the codebook for the vector that best encodes each excitation vector; and (3) only the index of the selected vector is encoded since the decoder has an exact copy of the codebook.

A typical CELP codebook contains 1024 vectors, requiring 10 bits to encode each 5 ms sub-frame, and would use 2 kbps to encode the excitation signal. However, there are a few other parameters to encode. During the exhaustive search procedure, known as *analysis-by-synthesis*, the gain of the selected vector is optimised and must therefore be encoded separately. Analysis-by-synthesis coding minimises the weighted error between the reconstructed and original speech over the free parameters (which are the pulse amplitudes and phase in GSM). The free parameters in CELP are the codebook index and the gain. The error is weighted to account for the non-uniform spectral sensitivity of the human auditory system. A complete CELP encoder may be implemented at a 4.8 kbps bit rate.[12]

Since the analysis-by-synthesis procedure optimises the vector gain, normalising the Gaussian vectors to unit length loses no generality. Such a normalisation facilitates a geometric interpretation of the excitation. Random 40-dimensional Gaussian vectors, of zero mean and unit length, happen to be uniformly distributed on the surface of a 40-dimensional sphere of unit radius.[12] CELP multiplies a sequence of these vectors by a time-varying gain, which accounts for the slowly changing speech envelope, to model the excitation signal.

The fact that CELP produces good quality speech gives support to the assumption that the normalised excitation vectors are uniformly distributed on a hyper-sphere. The CELP codebook contains a finite population of vectors drawn from this distribution. Although doubling the codebook size increases the code by just one bit, and improves the speech quality, it

does double the search time. A codebook of 1024 entries reasonably balances the incremental cost of searching with the incremental gain in speech quality.[12] However, if a more accurate distribution is found, for the normalised excitation vectors, then speech quality may be improved, without increasing the codebook size, by selecting vectors from this distribution.

An experiment was set up to test the assumption of uniform distribution by comparing the distribution of angles between pairs of excitation vectors selected at random with the distribution of angles between pairs of random Gaussian vectors. The dot product of two vectors in N-dimensional Euclidean space equals the product of their lengths and the cosine of the angle between them.[13] Therefore, the angle between any two vectors $\mathbf{x_1}$ and $\mathbf{x_2}$ may be found according to Equation 14. This angle has the usual geometric interpretation since two vectors, in any number of dimensions, always lie in a two-dimensional subspace (i.e. a plane).

Equation 14. $$\angle(\mathbf{x}_1, \mathbf{x}_2) = \cos^{-1} \frac{\mathbf{x}_1 \bullet \mathbf{x}_2}{\|\mathbf{x}_1\| \cdot \|\mathbf{x}_2\|}$$

If the distribution of angles derived from the excitation vectors matches the distribution of angles from the random Gaussian vectors then the assumption of uniform distribution on the surface of a hyper-sphere is consistent with the data. Strictly speaking, other distributions are possible since transforming the distribution of excitation vectors (i.e. a 40-dimensional probability density function) into a distribution of angles (i.e. a one dimensional probability density function) is a many-to-one mapping. However, if the two angle distributions do not match then the assumption of uniform distribution must be incorrect because performing the same many-to-one transformation on two identical distributions must yield the same result.

Figure 20 shows the distribution of angles between randomly selected vectors from a database of 24,000 excitation vectors, plotted against the distribution of angles between random Gaussian vectors. The database was generated by applying LPC, as described in the

previous section, to 120 seconds of speech from a single male.[9] The closeness of the two distributions in Figure 20 suggests that the CELP assumption is very good.

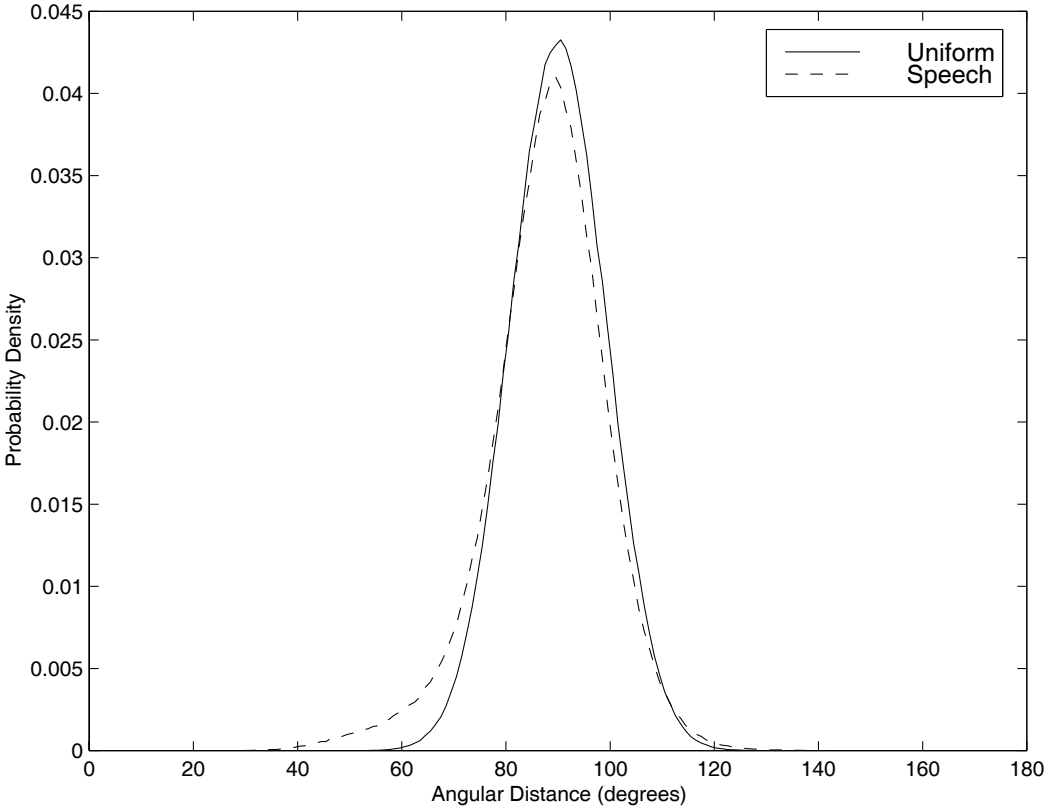


Figure 20. Distribution of angles between random excitation vectors.

The probability peak at 90 degrees is not surprising considering that, given a vector $\mathbf{x_1}$ on a three-dimensional sphere, a greater circumference may be traversed by tracing all vectors lying on the sphere at right angles to $\mathbf{x_1}$ than at any other angle to $\mathbf{x_1}$. Therefore, if the vector $\mathbf{x_2}$ is selected from a uniform distribution according to surface area, it is most likely to make a right angle with $\mathbf{x_1}$ since there is differentially more surface area at 90 degrees. A similar situation exists on the 40-dimensional hyper-sphere but the concepts of area and circumference must be replaced by 39-dimensional and 38-dimensional subspaces respectively.

LPC analysis makes use of the linear predictive properties of speech to reduce the dynamic range of the encoded signal. However, there may still be higher-order non-linear predictive properties in the excitation signal derived from LPC. Figure 21 shows the distribution

of angles between consecutive excitation vectors, as opposed to randomly selected excitation vectors, derived from the same database used in Figure 20. Since this distribution is different from the distribution of angles between random Gaussian vectors (i.e. the CELP model), the figure suggests that consecutive excitation vectors are statistically dependent.

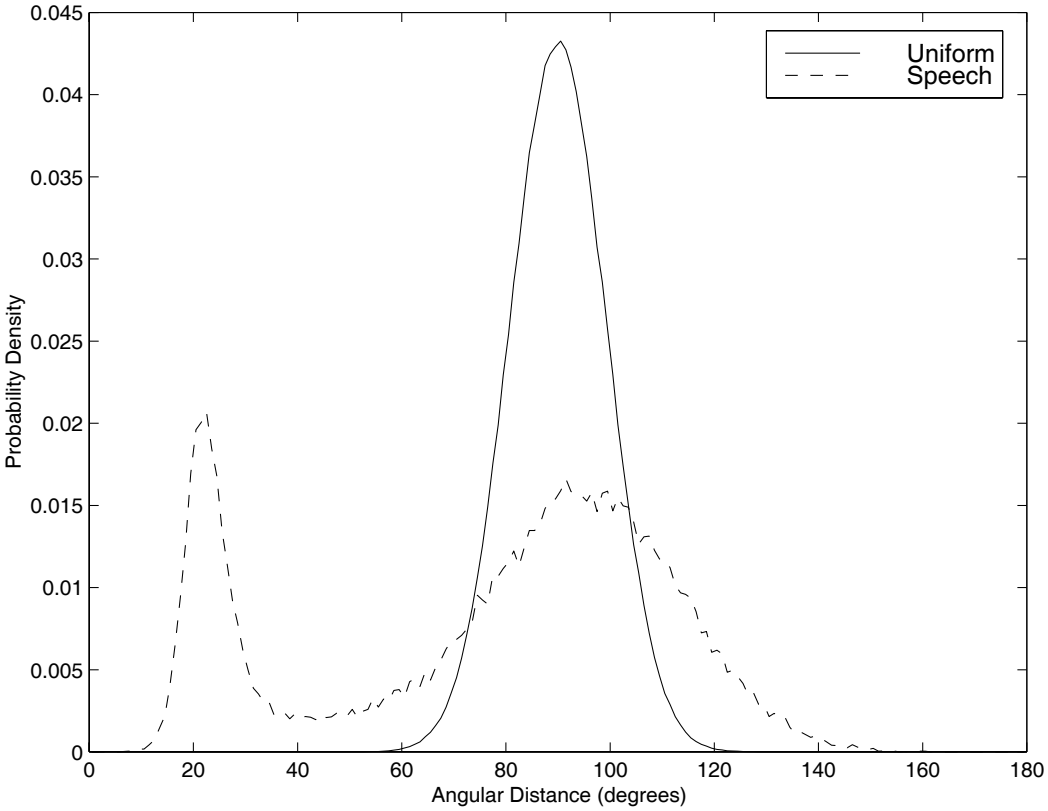


Figure 21. Distribution of angles between consecutive excitation vectors.

Figure 21 shows a bimodal distribution, with peaks at angles slightly above 20 and 90 degrees, perhaps corresponding to periods of voiced and unvoiced speech respectively. According to the figure, the current excitation vector $\mathbf{d}_{\mathbf{curr}}$ is equally likely to make a 22 degree or a 95 degree angle with the previous excitation vector $\mathbf{d}_{\mathbf{prev}}$. A CELP codebook will contain almost no vectors at a 22 degree angle from $\mathbf{d}_{\mathbf{prev}}$, since the codebook population is the same (i.e. spherically uniform) with respect to any reference vector. However, as in previous sections on waveform quantisation, more code vectors (which are essentially quanta) ought to be placed in regions of higher excitation vector probability. The codebook population should

therefore be drawn from a non-uniform bimodal distribution, according to Figure 21, on the surface of a hyper-sphere having two bands of high probability at 22 and 95 degrees to $\mathbf{d}_{\mathbf{prev}}$.

One way to exploit the non-uniform distribution of consecutive excitation vectors is suggested in Figure 22. A reference vector $\mathbf{c}_{\mathbf{ref}}$ is chosen and a codebook is created where the distribution of angles between $\mathbf{c}_{\mathbf{ref}}$ and the code vectors $\mathbf{c}_{\mathbf{i}}$ follows the bimodal distribution of Figure 21. To encode an excitation vector $\mathbf{d}_{\mathbf{curr}}$, a unitary matrix **A** is constructed such that $\mathbf{A}\mathbf{c}_{\mathbf{ref}}$ equals $\mathbf{c}_{\mathbf{prev}}$, where $\mathbf{c}_{\mathbf{prev}}$ is the previous decoded excitation vector. Note that $\mathbf{c}_{\mathbf{prev}}$ approximates the previous excitation vector $\mathbf{d}_{\mathbf{prev}}$. During analysis-by-synthesis, the matrix **A** is applied to each code vector, encoding $\mathbf{d}_{\mathbf{curr}}$ by rotated vectors $\mathbf{A}\mathbf{c}_{\mathbf{i}}$ instead of vectors $\mathbf{c}_{\mathbf{i}}$. In other words, the non-uniformly populated codebook is rotated prior to each encoding. The decoder has an identical codebook and can also generate **A** from $\mathbf{c}_{\mathbf{ref}}$ and $\mathbf{c}_{\mathbf{prev}}$.

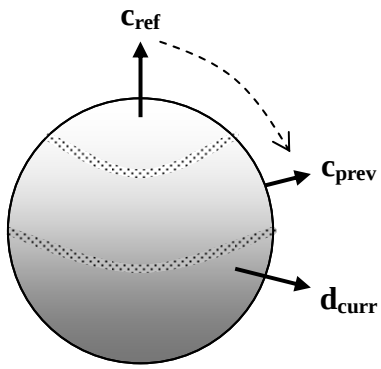


Figure 22. Exploiting the statistics of consecutive excitation vectors.

Although rotating the entire codebook to encode each sub-frame is a significant computation, preliminary results suggest that a more localised search may be possible than full analysis-by-synthesis using this method. Preliminary results also suggest an improvement in speech quality. However, further work is needed to refine these ideas.

IV. Source Coding of Oversampled Signals

A. Oversampling and decimation

The previous chapter reviewed and developed the theory of source coding of speech. Since the input signals were already digitised, source coding was used to compress the digital information for transmission over a narrow digital channel. This chapter reviews and extends the theory of oversampled signal modulation for A/D conversion. Modulation is interpreted here as source coding of an oversampled analogue input signal, by analogue and digital circuitry instead of a digital algorithm, to transmit the analogue information efficiently through a narrow A/D interface in order to produce a digital signal.

Consider an analogue signal $x(t)$ carrying information of interest in the band $|f| \leq f_B$. A conventional A/D converter would sample and quantise $x(t)$ at the Nyquist rate $f_S = 2f_B$ to produce a digital signal $y[n] = x[n]+e[n]$ with a noise spectral density $S_e(f) = \sigma_e^2/2f_B$. Such a converter must use sharp low-pass filtering before sampling to prevent out-of-band components from aliasing into the sampled signal.[6] Figure 23 depicts A/D conversion of this kind.

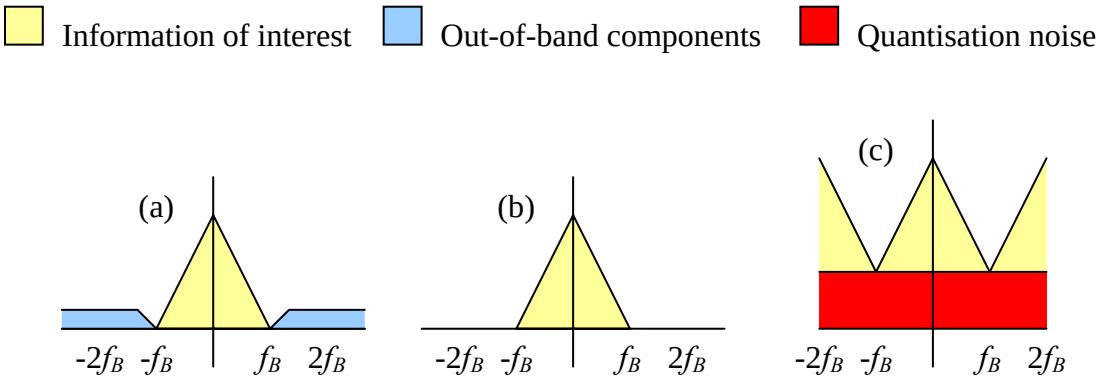


Figure 23. Conventional A/D conversion: (a) the original power spectrum, (b) after anti-alias filtering, and (c) after sampling and quantisation.

Suppose instead that $x(t)$ is digitised at a sampling frequency $f_S = 2\eta f_B$ that is η times the Nyquist rate, where the multiplier η is called the *oversampling ratio*. If the quantisation error remains uniformly distributed in $-Q/2$ to $+Q/2$ and uncorrelated across samples then the power

spectral density of the noise reduces to $S_e(f) = \sigma_e^2/2\eta f_B$. Oversampling also allows the anti-alias function to be performed gradually. Figure 24 depicts A/D conversion of this kind.

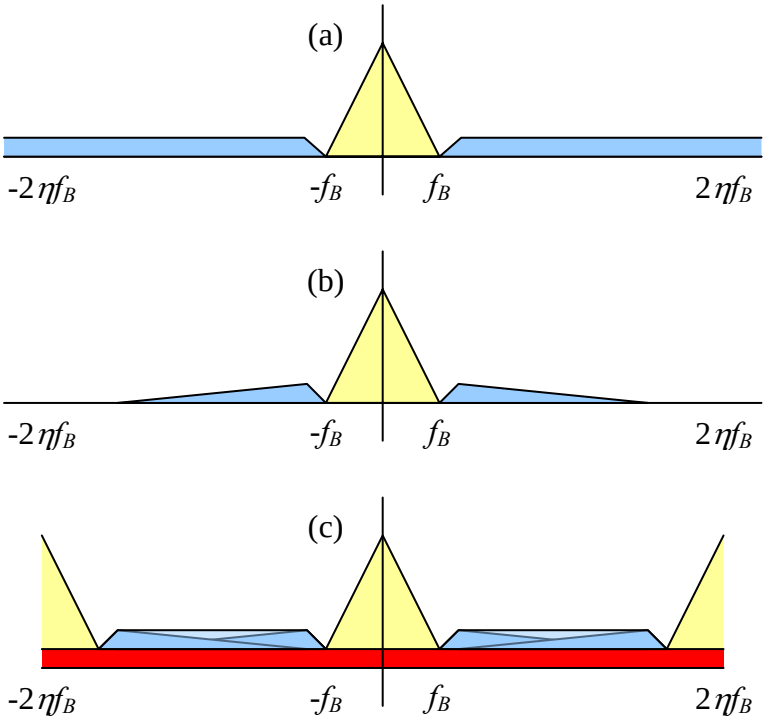


Figure 24. Oversampled A/D conversion: (a) the original power spectrum, (b) after anti-alias filtering, and (c) after sampling and quantisation.

The benefits of oversampling are realised by *decimation*. Oversampled quantisation of an analogue signal $x(t)$ yields a digital signal $y_\eta[n] = x(nT_\eta)+e(nT_\eta)$, where $T_\eta = 1/2\eta f_B$. Since the information of interest in the input signal $x(t)$ only occupies a bandwidth of $|f| \leq f_B$, the A/D converter should ultimately output a digital signal $y[n]$ at the Nyquist rate $2f_B$ such that $y[n]$ approximates $x(nT)$, where $T = 1/2f_B$. The process of decimation converts an oversampled digital signal $y_\eta[n]$ into a digital signal $y[n] = x(nT)+\varepsilon(nT)$ at the Nyquist rate.

Decimation is a two-step process of anti-aliasing an oversampled signal and then down-sampling the result to a lower frequency. The process is entirely digital because the oversampled signal is already digitised. Sharp anti-aliasing is necessary to prevent out of band components and quantisation noise, that were captured during oversampled quantisation, from folding into the signal band upon down-sampling. The process is illustrated in Figure 25.

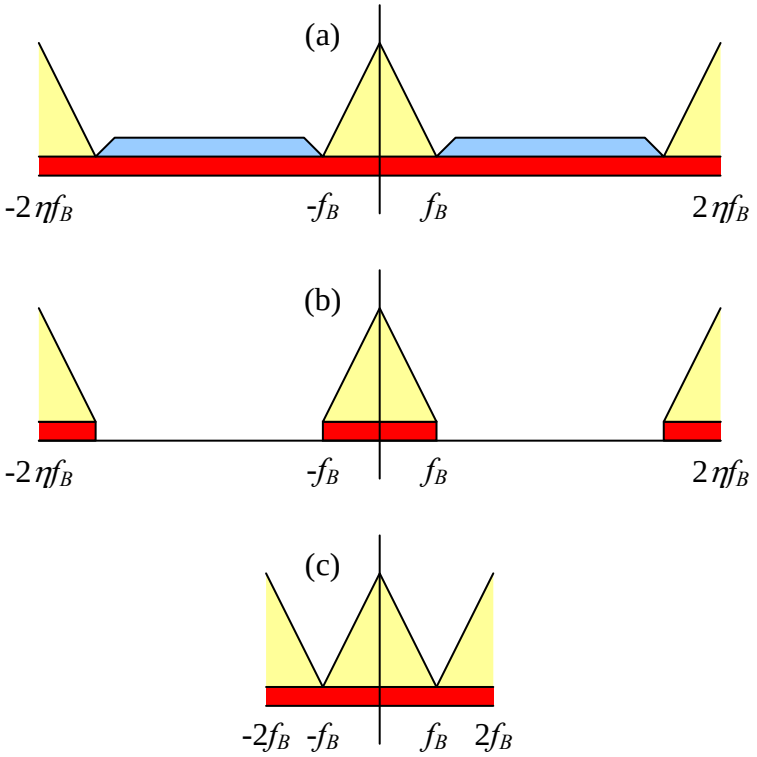


Figure 25. Decimation of an oversampled signal: (a) the original power spectrum, (b) after anti-alias filtering, and (c) after down-sampling.

Providing a sharp anti-alias filter is used, decimation does not change the power spectral density of the quantisation noise in the band $|f| \le f_B$. Equation 15 relates the noise power σ_ε^2 in the decimated signal to the noise power σ_e^2 in the oversampled signal. The equation repeats the well-known result that oversampled quantisation and decimation, compared to Nyquist rate quantisation, reduces the noise power by the oversampling ratio.[7] Each doubling of η thus adds 3 dB to the SNR, increasing the effective resolution of the converter by half a bit.

Equation 15. $$\sigma_\varepsilon^2 = \int_{-f_B}^{f_B} S_\varepsilon(f)df = \int_{-f_B}^{f_B} S_e(f)df = \frac{\sigma_e^2}{2\eta f_B}\int_{-f_B}^{f_B} df = \frac{\sigma_e^2}{\eta}$$

The advantages of oversampling and decimation are: (1) sharp anti-aliasing is moved from the analogue to digital domain, where it is easier to implement given the low analogue precision of most VLSI processes; and (2) output noise may be decreased without increasing the number of quantisation levels. The disadvantages are: (1) a huge oversampling ratio (e.g.

billion-fold) is needed to achieve high resolution (e.g. 16 bits), restricting the technology to very low bandwidth input signals compared to the VLSI process bandwidth; and (2) the digital signal processing demanded by decimation requires more silicon area.

B. Delta modulation

Sampling a signal $x(t)$ at many times the Nyquist rate decreases the time between consecutive samples $x[n]$ and $x[n\text{-}1]$ and increases their correlation (if the signal is continuous). Therefore, as the oversampling ratio increases, the dynamic range of the difference $x[n]\text{-}x[n\text{-}1]$ decreases compared to the dynamic range of $x[n]$. *Delta modulation* of an oversampled signal, shown in Figure 26, basically quantises this difference. The similarity between delta modulation and DPCM of speech is a reason to interpret the circuit operation as source coding.

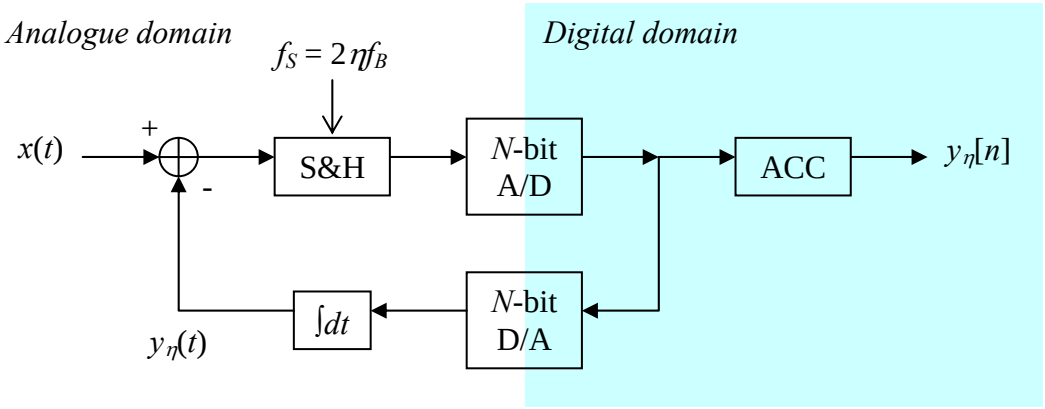


Figure 26. A delta modulator.

The feedback signal $y_\eta(t)$ in the delta modulator predicts the input $x(t)$. The error in the prediction, buffered by a sample-and-hold unit (S&H), is digitised by an embedded flash A/D converter. If the prediction is less than the input, the A/D converter (an excitation encoder) outputs a proportionate positive integer, making the D/A converter output a positive signal to increase the prediction. If the prediction is more than the input, the A/D converter outputs a negative integer causing the D/A converter to output a negative signal to decrease the prediction. The digital accumulator (ACC) is simply an up/down counter, or digital integrator, that increments or decrements the output word using the integer updates from the A/D converter.

The integration in the feedback path of Figure 26 is an analogue operation whereas the integration in the output path is a digital operation. Providing the leakage in the analogue integrator is controlled or compensated, the digital output $y_\eta[n]$ will approximate the analogue prediction $y_\eta(t)$. Figure 27 gives a linear discrete-time model of delta modulation, where quantisation is modelled as the addition of white and uncorrelated noise.

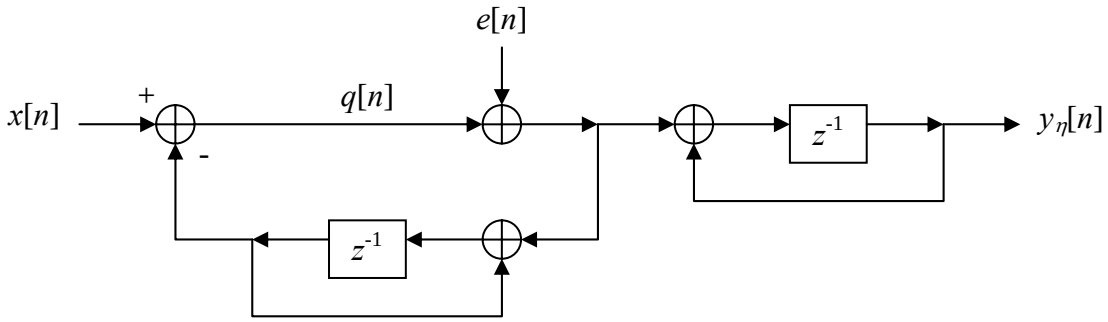


Figure 27. Linear discrete-time model of delta modulation.

The transfer function of the delta modulator, expressed as $Y_\eta(z) = H_x(z)X(z)+H_e(z)E(z)$ using superposition, is given in Equation 16. Delta modulation does not distinguish between the input and the noise, delaying both by one sample. The functions $H_x(z)$ and $H_e(z)$ are z-transforms of the system responses to separate impulses on x and e.[14] The response of the discrete-time system to a sinusoid of frequency f may be found using $z = e^{j\omega T}$ and $\omega = 2\pi f$.

Equation 16. $$Y_\eta(z) = z^{-1}X(z) + z^{-1}E(z) \quad \leftrightarrow \quad y_\eta[n] = x[n-1] + e[n-1]$$

Since the delta modulator only delays the noise, its power spectral density $S_e(f)$ remains unchanged at the modulator output. Ideal decimation of the oversampled signal $y_\eta[n]$ to the Nyquist rate signal $y[n] = x[n]+\varepsilon[n]$ eliminates the noise outside $|f| \le f_B$, resulting in a output noise power σ_ε^2 given by Equation 17, which is the same as the result of Equation 15.

Equation 17. $$\sigma_\varepsilon^2 = \int_{-f_B}^{f_B} S_\varepsilon(f)\,df = \int_{-f_B}^{f_B} |H_e(f)|^2 S_e(f)\,df = \frac{\sigma_e^2}{2\eta f_B}\int_{-f_B}^{f_B} \left|e^{-j\omega T_\eta}\right|^2 df = \frac{\sigma_e^2}{\eta}$$

Although oversampled conversion without modulation seems to have the same noise power as delta modulation, the latter is actually better. Delta modulation reduces the dynamic range of the quantiser input $q[n]$, lowering the quantiser step-size Q and the quantisation noise power $\sigma_e^2 = Q^2/12$. Denoting the maximum amplitude of the input by A and the maximum slew rate by δ, the noise powers for oversampled conversion without and with delta modulation are given in Equation 18 (N is the number of quantisation bits). For delta modulation, doubling η divides the noise power by eight and increases the SNR by 9 dB or 1½ bits. Note that the performance of delta modulation is not limited by the dynamic range A of the input.

Equation 18.

$$\begin{array}{lll|l} \text{No modulation} & Q = \dfrac{2A}{2^N} & \sigma_\varepsilon^2 = \dfrac{A^2}{3\eta \cdot 2^{2N}} & A = \max\left\{\left|x(t)\right|\right\} \\ \text{Delta modulation} & Q = \dfrac{2T_\eta \delta}{2^N - 1} & \sigma_\varepsilon^2 = \dfrac{\delta^2}{12\eta^3 f_B^2 \left(2^N - 1\right)^2} & \delta = \max\left\{\left|\dfrac{dx}{dt}\right|\right\} \end{array}$$

The choice of Q for delta modulation, in Equation 18, keeps the difference between the input and prediction within $\pm Q/2$. A smaller value for Q may mean that the prediction will not slew as quickly as the input, causing a dramatic increase in noise when the input changes too quickly. However, using a large value for Q may increase the granular noise since the prediction often alternates above and below the input during periods of slow change. Optimal choice of Q thus depends on the probability distribution of slew rates, an opportunity to source code.

C. First-order delta-sigma modulation

Delta-sigma modulation is another method to source code an oversampled input signal to reduce the noise in the output signal. Figure 28 depicts a first-order delta-sigma modulator. The modulator operates by shaping the quantisation noise spectrum so that less noise remains after decimation.[7] Oversampling creates a band $f_B \le |f| \le (2\eta\text{-}1)f_B$ that is completely erased by decimation and is thus insensitive to noise. Delta-sigma modulation source codes an oversampled input signal by placing most of the quantisation noise in this band.

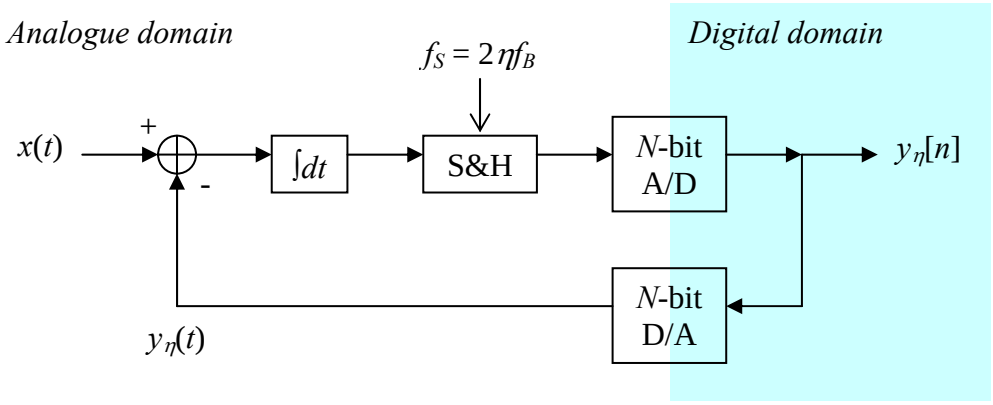


Figure 28. A first-order delta-sigma modulator.

Figure 28 shows an N-bit quantiser (the cascade combination of A/D and D/A converter) although first-order delta-sigma modulation usually involves two-level quantisation.[7] The analogue signal $y_\eta(t)$ tracks the digital signal $y_\eta[n]$ almost perfectly since D/A conversion introduces almost no error. Increasing the number of quantisation levels, however, would increase the non-linearity of the D/A converter and would cause harmonic distortion.

The integrator in the delta-sigma modulator keeps the average value of the input $x(t)$ equal to the average value of the feedback signal $y_\eta(t)$ which forces most of the quantisation noise in $y_\eta[n]$ to higher frequencies. Figure 29 shows a linear model, in discrete-time, of a first-order delta-sigma modulator. Once again, integration is modelled by accumulation.

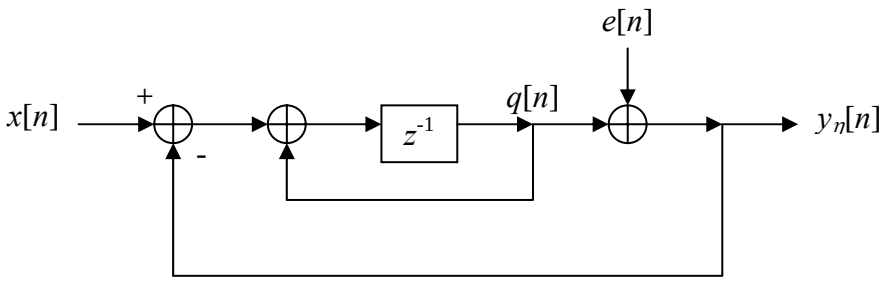


Figure 29. Linear discrete-time model of first-order delta-sigma modulation.

Equation 19 expresses the relationship, using superposition, between the input signal and quantisation noise and the output of the first-order delta-sigma modulator.

Equation 19. $Y_\eta(z) = z^{-1}X(z) + (1 - z^{-1})E(z) \quad \leftrightarrow \quad y_\eta[n] = x[n-1] + e'[n]$

According to Equation 19, the modulator adds the first difference $e'[n] = e[n]-e[n-1]$ of the quantisation noise to the delayed input signal. Equation 20 relates the power spectral density $S_{e'}(f)$ of the filtered quantisation noise to the unshaped power spectral density $S_e(f)$.

Equation 20. $$S_{e'}(f) = |H_e(f)|^2 S_e(f) = \left|1 - e^{-j\omega T_\eta}\right|^2 \frac{\sigma_e^2}{2\eta f_B} = \frac{2\sigma_e^2}{\eta f_B} \sin^2 \frac{\pi f}{2\eta f_B}$$

Figure 30 plots the power spectral densities of the original $S_e(f)$ and shaped $S_{e'}(f)$ noise, normalised to $\sigma_e^2/\eta f_B$, versus frequency, normalised to ηf_B. Delta-sigma modulation reduces the noise power in $y_\eta[n]$ at low frequencies but increases it at higher frequencies.

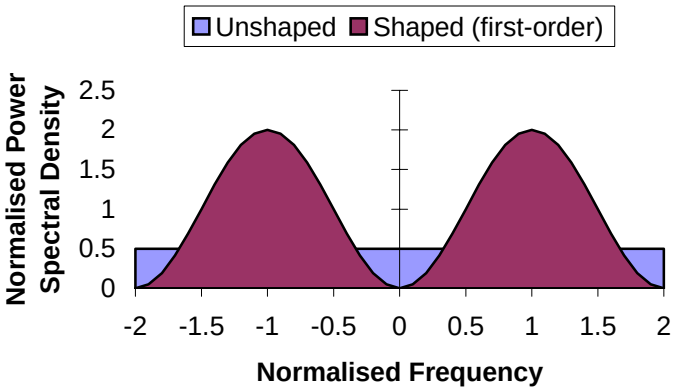


Figure 30. Noise power spectrum of first-order delta-sigma modulation.

Assuming $y_\eta[n]$ is decimated with perfect anti-aliasing, the power spectral density $S_\varepsilon(f)$ of the output noise $\varepsilon[n]$ (i.e. in the final digital signal $y[n]$) will equal $S_{e'}(f)$ in the band $|f| \le f_B$. Equation 21, therefore, calculates the noise power σ_ε^2 in $y[n]$.

Equation 21.
$$\sigma_\varepsilon^2 = \int_{-f_B}^{f_B} S_\varepsilon(f)\,df = \frac{2\sigma_e^2}{\eta f_B} \int_{-f_B}^{f_B} \sin^2 \frac{\pi f}{2\eta f_B}\,df \approx \frac{4\sigma_e^2}{\eta f_B} \int_0^{f_B} \left(\frac{\pi f}{2\eta f_B}\right)^2 df, \quad \eta >> 1$$
$$= \frac{\pi^2 \sigma_e^2}{3\eta^3} = \frac{\pi^2 A^2}{9\eta^3 \left(2^N - 1\right)^2} \quad \left(Q = \frac{2A}{2^N - 1}\right)$$

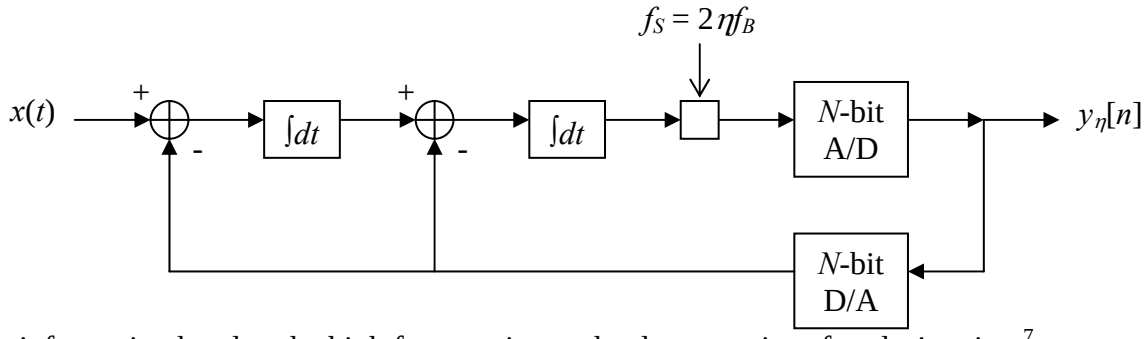
Every doubling of the oversampling ratio divides the noise power by eight, adding 9dB to the SNR and increasing the effective resolution of the converter by 1½ bits.[7] Thus, over-sampling with noise shaping requires a lower oversampling ratio to obtain a given resolution

than does oversampling without noise shaping. This improvement is achieved without an increase in the number of quantisation levels. First-order delta-sigma modulation and delta modulation have similar noise performances with respect to η and N but the former is affected by the dynamic range A whereas the latter is affected by the maximum slew rate δ.

The advantage that first-order delta-sigma modulation gains in the oversampling ratio, compared to oversampling without noise shaping, is huge (e.g. a thousand-fold oversampling ratio is sufficient to achieve 16 bit resolution). Lower oversampling ratios allow higher bandwidth input signals to be digitised for a given resolution and VLSI process speed. However, noise shaping requires more analogue precision since the band $f_B \le |f| \le (2\eta\text{-}1)f_B$ that is used for anti-alias filtering is smaller for a lower oversampling ratio. Secondly, the decimator must perform sharper anti-aliasing because noise shaping places much more noise outside the input bandwidth. Higher-order digital anti-alias filters consume more silicon area.

D. Second-order delta-sigma modulation

It is possible to perform better noise shaping than first-order delta-sigma modulation. A double loop or second-order delta-sigma modulator, shown in Figure 31, shifts more noise from



the information band to the high frequencies so that less remains after decimation.[7]

Figure 31. A second-order delta-sigma modulator.

The second-order delta-sigma modulator is essentially a first-order feedback loop within another first-order loop.[15] A linear discrete-time model of the modulator is given in Figure 32.

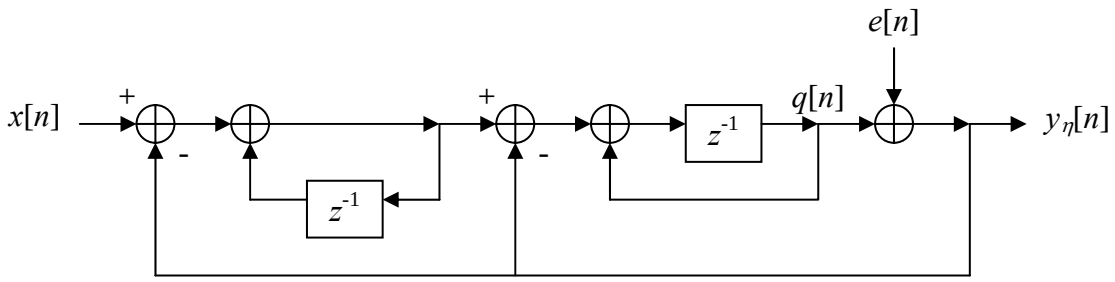


Figure 32. Linear discrete-time model of second-order delta-sigma modulation.

Equation 22 calculates the transfer function of the second-order modulator using superposition. While the first-order modulator simply computes the first difference of the noise, the second-order modulator computes the second difference $e''[n] = e[n]-2e[n-1]+e[n-2]$.

Equation 22. $$Y_\eta(z) = z^{-1}X(z) + \left(1-z^{-1}\right)^2 E(z) \quad \leftrightarrow \quad y_\eta[n] = x[n-1] + e''[n]$$

Equation 23 calculates the power spectral density of the shaped noise $S_{e''}(f)$, which is plotted in Figure 33. The figure is normalised as in Figure 30 and the unshaped noise response is given as a reference. Second-order noise shaping, as compared to first-order, flattens the noise response more at low frequencies but quadruples it at higher frequencies.

Equation 23. $$S_{e''}(f) = \left|H_e(f)\right|^2 S_e(f) = \left|1-e^{-j\omega T_\eta}\right|^4 \frac{\sigma_e^2}{2\eta f_B} = \frac{8\sigma_e^2}{\eta f_B}\sin^4\frac{\pi f}{2\eta f_B}$$

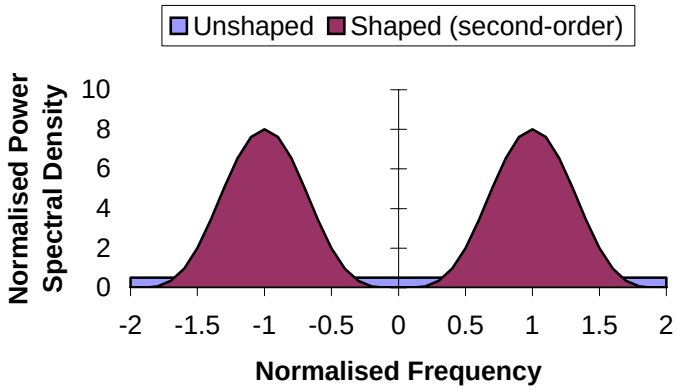


Figure 33. Noise power spectrum of second-order delta-sigma modulation.

Assuming perfect decimation, Equation 24 computes the noise power in the final signal $y[n]$. The noise power drops by 15 dB for every doubling of η, adding 2½ bits of resolution.[7] As with first-order delta-sigma modulation, the noise power depends on the dynamic range A.

Equation 24.

$$\sigma_\varepsilon^2 = \int_{-f_B}^{f_B} S_\varepsilon(f)\,df = \frac{8\sigma_e^2}{\eta f_B}\int_{-f_B}^{f_B} \sin^4\frac{\pi f}{2\eta f_B}\,df \approx \frac{8\sigma_e^2}{\eta f_B}\int_{-f_B}^{f_B}\left(\frac{\pi f}{2\eta f_B}\right)^4 df, \quad \eta >> 1$$
$$= \frac{\pi^4\sigma_e^2}{5\eta^5} = \frac{\pi^4 A^2}{15\eta^5\left(2^N-2\right)^2} \quad \left(Q = \frac{2A}{2^N-2}\right)$$

Compared to first-order delta-sigma modulation, second-order modulation requires a lower oversampling ratio to achieve the same resolution. Second-order modulation can realise 16-bit resolution at an oversampling ratio of around 128, so that compact disc quality sound (16-bit samples at 44kHz) can be digitised with a sampling frequency of around 6MHz.[7] However, the decimator must provide sharper anti-aliasing, requiring more silicon. Analogue precision is also needed since the transition bandwidth for analogue anti-aliasing is narrower.

E. Higher-order delta-sigma modulation

Because first and second-order delta-sigma modulators have been implemented successfully,[15] an obvious question is whether third and higher-order modulators can be constructed by iterations of the first-order feedback loop, as shown in Figure 34. A linear discrete-time analysis of such an architecture would suggest that every doubling of the oversampling ratio would in-

crease the resolution by L½ bits, where L is the order of the modulator (the number of loops).[7] However, these modulators are difficult to realise for $L > 2$ due to stability problems.[15]

There are two kinds of stability considerations for noise shaping modulators. The first requirement is that all poles of the filters $H_x(f)$ and $H_e(f)$ lie within the unit circle. The second requirement is not explicitly modelled by the discrete-time analysis. If the input to the quantiser is too large then the quantiser saturates and the noise power σ_e^2 is no longer equal to $Q^2/12$. The increase in noise power will depend on the degree of saturation. As a result of the circuit feedback, an increase in the noise will cause an increase in the input $q[n]$ to the quantiser, saturating it even more. This relationship is equivalent to positive feedback, whereby an increase in the noise causes an increase in the quantiser input and vice-versa. The end result is a large amplitude oscillation at a low frequency. These oscillations occur if the input signal saturates the quantiser when the loop gain is too high (≥ 2).[15]

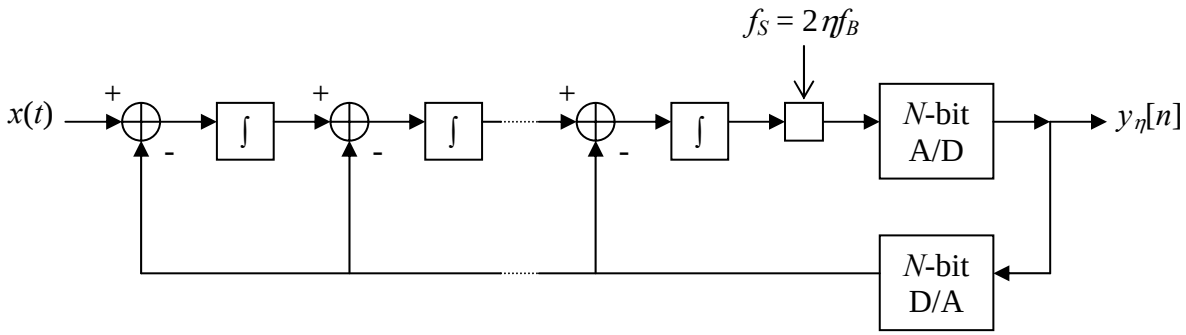


Figure 34. A higher-order delta-sigma modulator.

Higher-order modulators may be built by choosing a different architecture. Generally, higher-order noise shaping modulators implement a high-order low-pass filter $F(z)$ in place of the integrator in the standard delta-sigma modulator.[15] A low-pass filter limits the gain in the pass band, which is beneficial for stability, whereas the gain of an integrator gets larger and larger at lower and lower frequencies. There are techniques to implement most rational polynomials $F(z) = P(z)/Q(z)$, providing the degree of $P(z)$ is not more than that of $Q(z)$.

As the order increases, higher-order modulators implement more ideal low-pass filters while meeting stability criteria. However, this approach is tantamount to requiring sharp analogue anti-aliasing. The decimation requirements must also increase to compensate for the sharp rise in noise power above the input bandwidth. Although they do not quite achieve $L½$ bits for every doubling of η, higher-order noise shaping modulators do allow the same resolution to be achieved with a lower oversampling ratio, than first or second-order delta-sigma modulation,[7] but even these architectures show a diminishing return above fourth-order.[15]

F. Delta^2-sigma modulation

Since the output noise power in delta-sigma modulation is directly related to σ_e^2, which equals $Q^2/12$, a reduction in the quantiser step-size Q would increase the SNR. However, the unsaturated range of the quantiser would also be reduced, reducing the input dynamic range that can be accommodated. The unsaturated range must, in fact, be greater than the input range because the quantiser must quantise two signals simultaneously, namely the input and the circulating noise.[15] As the modulator order increases, the circulating noise power increases (for first-order delta-sigma modulation, $q[n]$ equals $x[n\text{-}1]\text{-}e[n\text{-}1]$ whereas, for second-order delta-sigma modulation, $q[n]$ equals $x[n\text{-}1]\text{-}2e[n\text{-}1]+e[n\text{-}2]$). The value of Q must therefore increase as the modulator order increases, as given in Equation 21 and Equation 24.

If the input is correlated then the dynamic range of the signal entering the delta-sigma modulator may be reduced by modulating the difference between the input and a simple prediction of the input. Figure 35 introduces an original circuit, named a *delta^2-sigma modulator*, where a delta-sigma modulator quantises the prediction error of a delta modulator.

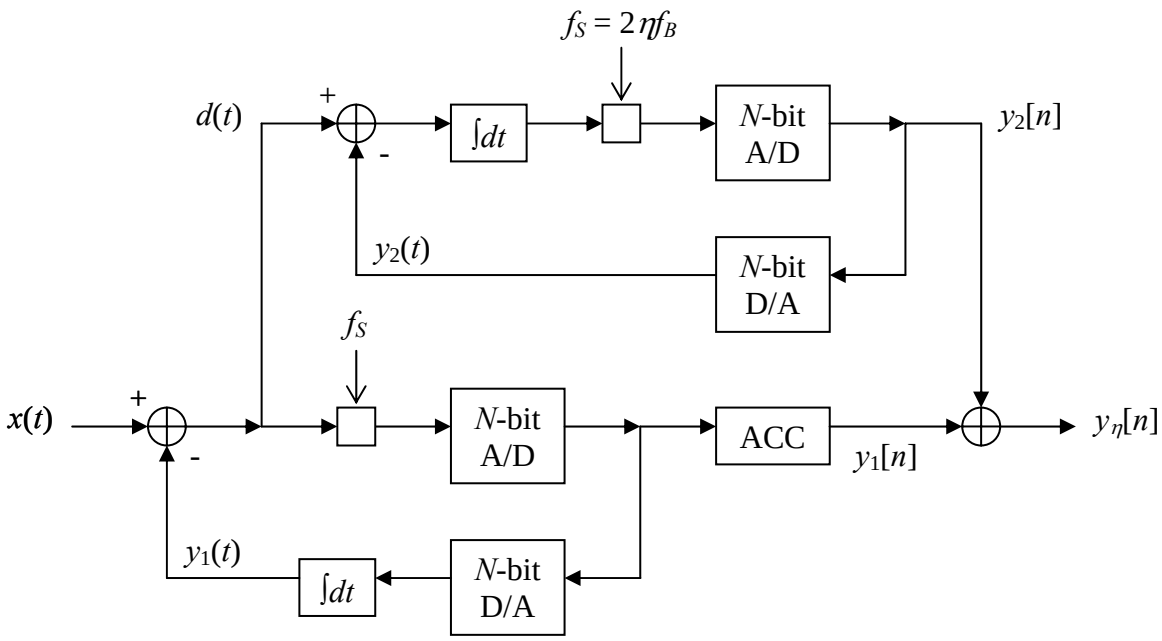


Figure 35. A delta^2-sigma, or cascaded delta delta-sigma, modulator.

Delta^2-sigma modulation, compared to delta modulation, is like linear predictive coding with an enhanced embedded excitation encoder. The output is the sum of two digitised signals $y_1[n]$ and $y_2[n]$ from a cascaded delta and delta-sigma modulator. Figure 36 gives a model of the modulation, with an extra delay in the delta modulator path for synchronisation.

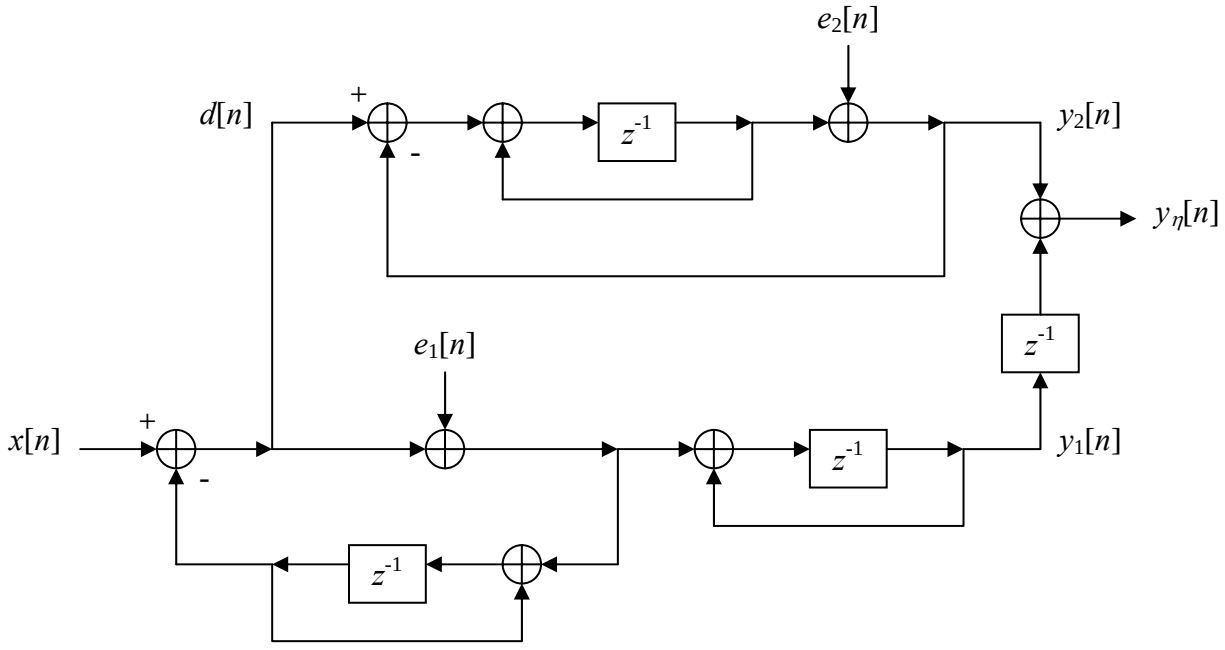


Figure 36. Linear discrete-time model of delta^2-sigma modulation.

The transfer function of the delta^2-sigma modulator is given in Equation 25. Note that the noise $e_1[n]$ of the delta modulator does not appear in the output. Effectively, the delta-sigma modulator resolves the granular noise of the delta modulator. Only the noise $e_2[n]$ of the delta-sigma modulator contaminates the output and it is shaped towards high frequencies.

Equation 25.
$$\begin{aligned} Y_\eta(z) &= z^{-1}Y_1(z) + Y_2(z) = z^{-1}Y_1(z) + z^{-1}\big(X(z) - Y_1(z)\big) + \big(1 - z^{-1}\big)E_2(z) \\ &= z^{-1}X(z) + \big(1 - z^{-1}\big)E_2(z) \quad \leftrightarrow \quad y_\eta[n] = x[n-1] + e_2'[n] \end{aligned}$$

Assuming perfect decimation of $y_\eta[n]$ to $y[n]$, the output noise power σ_ε^2, given in Equation 26, is the same as for delta-sigma modulation (with e replaced by e_2).

Equation 26. $$\sigma_\varepsilon^2 = \frac{\pi^2\sigma_{e_2}^2}{3\eta^3} = \frac{\pi^2\delta^2}{36\eta^5 f_B^2\left(2^N - 1\right)^4} \quad \left(Q_2 = \frac{Q_1}{2^N - 1}, \quad Q_1 = \frac{2T_\eta\delta}{2^N - 1}\right)$$

However, the step-size Q_2 of the embedded delta-sigma quantiser does not depend on the dynamic range A of the input signal but on the dynamic range of the prediction error $d(t)$. The latter is bounded by $\pm Q_1/2$, where Q_1 is the step-size of the quantiser in the delta modulator. Since Q_1 depends on the maximum signal slew rate δ, as in delta modulation, the overall noise performance of the delta^2-sigma modulator depends on this term and not on the input dynamic range. Note that the noise power drops as η^5, like double loop modulation.

G. Simulation results

The four modulators described in this chapter were simulated in Matlab. Test signals $x(t)$ were realisations of a stationary random process $X(t)$, abstracted in Figure 37. The process was the sum of a narrow-band white process $A(t)$ and a wide-band white process $B(t)$, having uniform probability densities $f_A(x)$ and $f_B(x)$. The narrow-band process lay in the band $|f| \le f_A$ with a power spectral density equal to A and the wide-band process lay in the band $|f| \le f_B$ with a power spectral density equal to B. Because these two processes were statistically independent,

the power spectral density of their sum equalled the sum of their power spectral densities. The process $X(t)$ was oversampled at $f_S = 2\eta f_B$, which introduces some correlation.

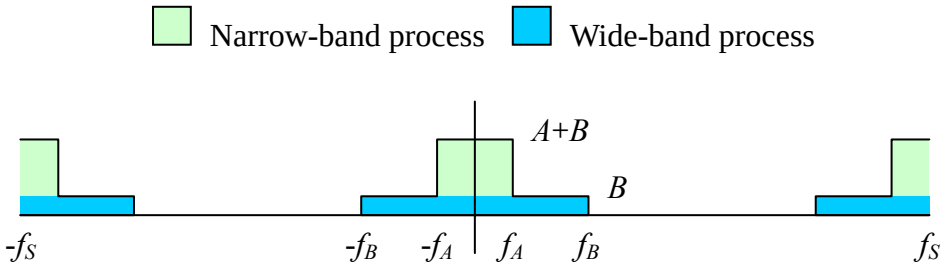


Figure 37. Power spectrum of an input process constructed for simulation.

Three simulations were undertaken, varying the oversampling ratio η, the bandwidth ratio f_B/f_A, and the power ratio Af_A/Bf_B (the power ratio relates directly to the dynamic range ratio of the narrow-band to the wide-band process). For each set of parameters, ten sample signals $x(t)$ were generated and the SNR was averaged over these realisations. Since the quantiser in the double loop modulator needs a minimum of two bits to encompass both the input signal and the circulating noise (Equation 24), the delta modulator and both delta-sigma modulators (single and double loop) were implemented with two-bit ($N = 2$) quantisers. To ensure that all modulators had the same A/D interface capacity, the delta^2-sigma modulator was implemented with two one-bit ($N = 1$) quantisers (making Q_1 equal to Q_2).

Figure 38 shows the SNR performance of the four modulators versus the oversampling ratio. For this simulation, the bandwidth ratio equalled 64 and the power ratio equalled 64 (meaning that the narrow-band process had 8 times the dynamic range within 1/64 times the bandwidth of the wide-band process). Given that such a process represents a slowly varying signal of high dynamic range superimposed on a quickly varying signal of low dynamic range, there is enough correlation for the predictive modulators to outperform the other two.

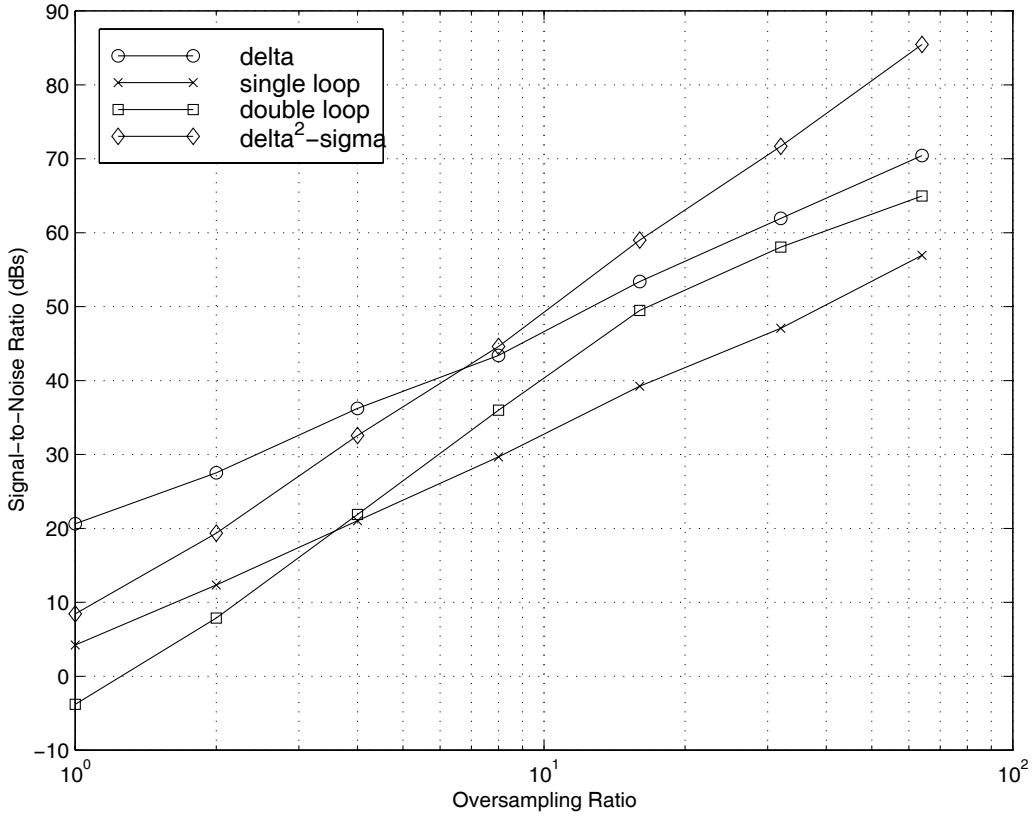


Figure 38. Modulator signal-to-noise ratio versus the oversampling ratio.

As expected, Figure 38 shows that the SNR improves with increasing η. The first-order modulators (delta and single loop) steadily improve at a rate of 9 dB per octave, consistent with the theory. At low oversampling ratios, they are better than the second-order modulators because the constants in Equation 18 and Equation 21 are smaller than those in Equation 24 and Equation 26. However, the second-order modulators (double loop and delta^2-sigma) improve at a rate of 12 to 15 dB per octave, also consistent with the theory. The double loop modulator warrants further study as its performance seems to degrade to first-order at $\eta = 16$. Finally, delta^2-sigma modulation is the best modulation scheme above $\eta = 8$.

Figure 39 plots the SNR of the modulators versus the bandwidth ratio, using an over-sampling ratio of 32 and a power ratio of 64. The bandwidth ratio corresponds to the separation, in rate of variation, between the high and low dynamic range processes $A(t)$ and $B(t)$.

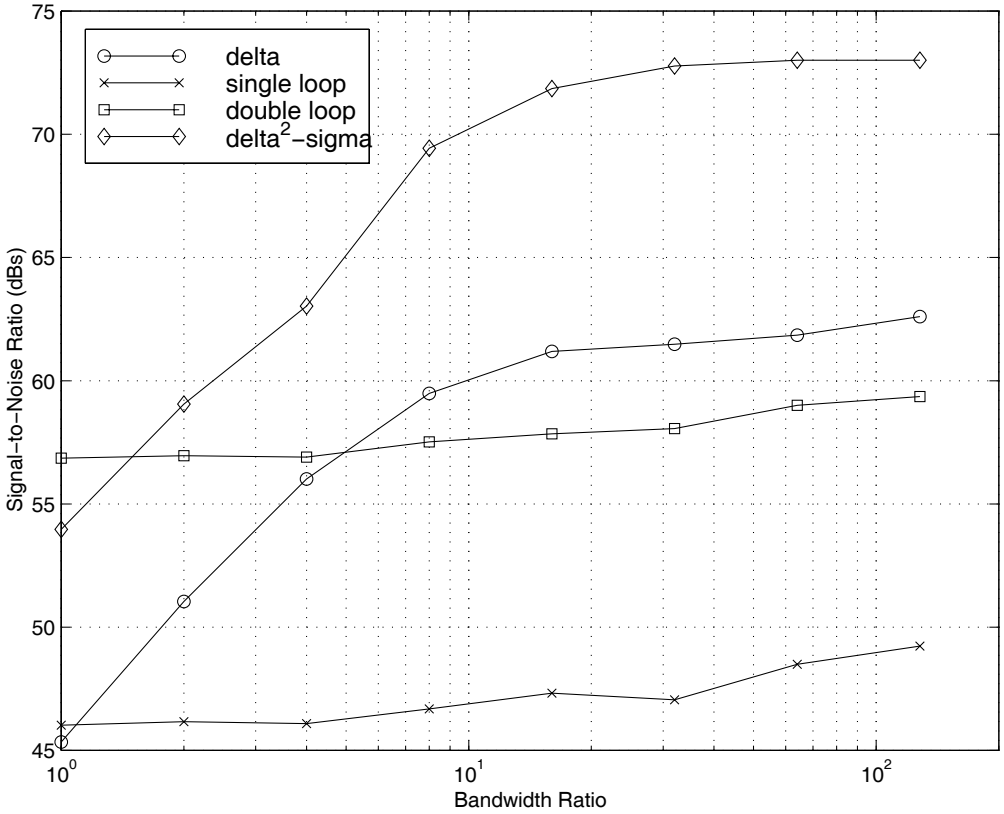


Figure 39. Modulator signal-to-noise ratio versus the bandwidth ratio.

The leftmost point in Figure 39 corresponds to a unit bandwidth ratio, meaning there is no frequency distinction between the high and low dynamic range component processes. In other words, the input process is the sum of two white processes of equal bandwidth but different average power. Such a process is relatively uncorrelated for low oversampling ratios. The predictive modulators are only slightly worse than the noise shaping modulators for this situation. However, at a bandwidth ratio of two, each predictive modulator immediately surpasses the noise shaping modulator of similar order due to a slight increase in correlation.

The performance of the noise shaping modulators is not really affected by the bandwidth ratio. Increasing the ratio does not affect the maximum amplitude of the input process but does decrease its maximum slew rate. Thus, the predictive modulators steadily improve their performance until it saturates at a bandwidth ratio of 16. At high ratios, the narrow-band

component is effectively a DC signal of random amplitude, limiting the minimum value of the process slew rate to the maximum slew rate of the wide-band component.

As shown in Figure 38, second-order modulation surpasses first-order modulation once η is sufficiently high. Each second-order modulator is better than its first-order counterpart because of an improvement in or an addition of noise shaping. This result is also evident in Figure 40, which charts the SNR performance of the modulators against the power ratio of the process. The oversampling ratio was 32 for this simulation and the bandwidth ratio was 64.

A unit power ratio, at the centre of Figure 40, corresponds to $A(t)$ and $B(t)$ having equal dynamic range. Delta^2-sigma modulation is then slightly worse than double loop modulation but delta modulation is better than single loop modulation. At the left end of the graph, the narrow-band process $A(t)$ is one tenth the dynamic range of the wide-band process $B(t)$. Delta^2-sigma is again slightly worse than double loop modulation but delta and single loop modulation are similar. However, when the narrow-band component has a large dynamic range, the predictive modulators are much better than their noise shaping counterparts.

As with the bandwidth ratio in Figure 39, the performance of the noise shaping modulators does not really change with increasing power ratio. A change in the dynamic range of $X(t)$ affects both the signal power and noise power equally, leaving the SNR results of noise shaping modulators unchanged. However, the correlation in $X(t)$ increases with an increasing power ratio since more power is allocated to the slowly varying component $A(t)$. As a result, the SNR results of predictive modulation improve with increasing power ratio.

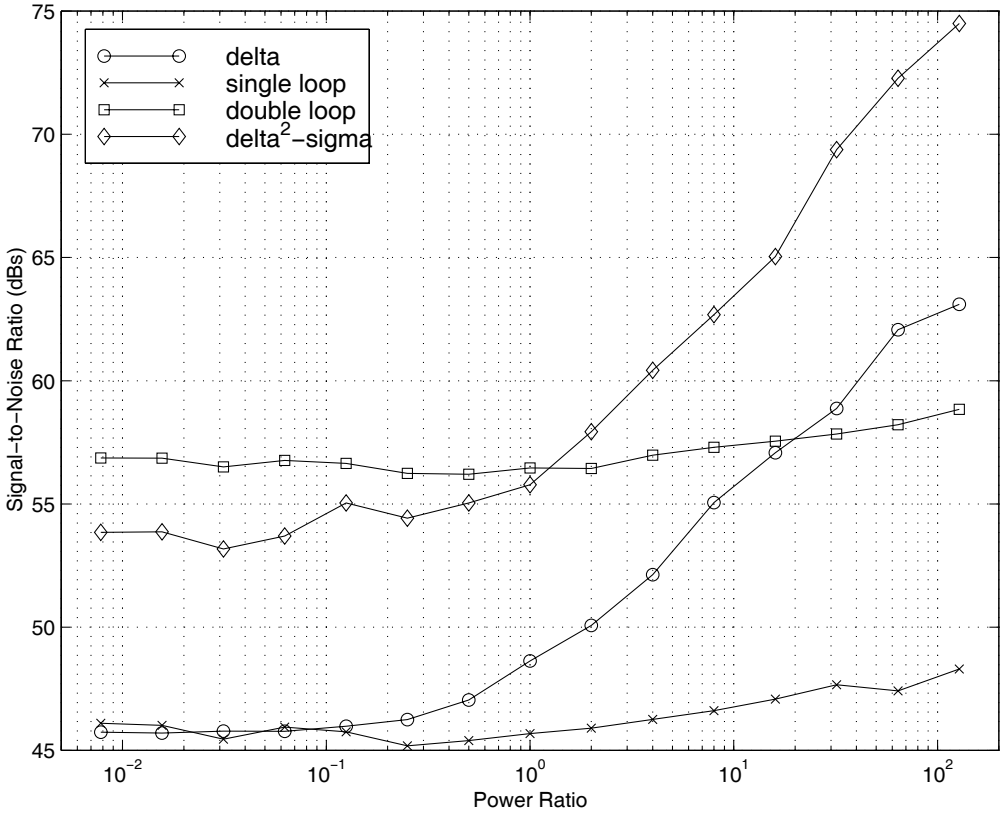


Figure 40. Modulator signal-to-noise ratio versus the power ratio.

In conclusion, this chapter has investigated source coding circuitry that improves A/D conversion of oversampled signals. Delta modulation uses prediction to reduce the dynamic range of a signal before quantisation. Delta-sigma modulation shapes the quantisation noise spectrum so that less falls in the information band. This report combined prediction and noise shaping by introducing a delta^2-sigma modulator, ideal for converting the superposition of a high dynamic range narrow-band process and a low dynamic range wide-band process.

V. Conclusion

A. Source coding principles

Previous chapters reviewed techniques for source coding of speech and of oversampled signals and suggested new algorithms and circuit architectures to improve results. From these chapters, several conclusions may be drawn about source coding.

Given a particular information source, the complexity of source coding needed to satisfy a certain distortion tolerance is inversely proportional by a high order to the channel capacity. The fact that complexity is inversely proportional to capacity is not surprising since more work must be done to squeeze the same information through a smaller bottleneck. However, at least with speech coding, the rate of increase in complexity as channel capacity decreases is surprising. The theoretical and computational effort to traverse linear PCM, logarithmic PCM, optimised non-uniform PCM, DPCM, adaptive scalar and vector quantisation, ADPCM, RPE, and CELP, outpaces the decrease in bit rate from 128 kbps to 4.8 kbps.

Furthermore, 4.8 kbps CELP does not offer toll-quality speech whereas 128 kbps linear PCM does. Below a certain channel capacity, a required level of speech quality may not be possible despite the complexity of source coding. Toll-quality standards below 8 kbps have not been established, although more research and better models may help overcome this barrier. The intrinsic unpredictability of most real sources of information implies that a minimum channel capacity exists below which the information cannot be adequately represented.

These principles are important to VLSI source coding. If a digital algorithm cannot meet a distortion tolerance for the planned capacity of a VLSI analogue-digital interface then no circuit technique would meet the tolerance either. In fact, the interface capacity must be significantly higher than the minimum capacity required by the information source because the complexity of source coding increases rapidly as the threshold is reached and VLSI circuits cannot hope to reproduce the performance of a complex digital source coding algorithm.

An important observation is that improvements in source coding and improvements in the quantitative measure used to compute the distortion introduced into a signal go together. The distortion measure has a significant influence on the source coder design, since the aim of the latter is to minimise the former. In the case of real world signals, distortion may be a subjective matter and an accurate formulation of the distortion becomes a difficult problem in itself. Nonetheless, an objective and computable distortion measure is imperative for source coding design to avoid subjective bias, to save time in experimental studies, and to apply mathematical techniques to reduce the distortion. The SNR is generally a good distortion measure. However, for non-stationary processes like speech, the SEGSNR is a better measure for it respects the time-varying nature of the process statistics.

Improving the distortion measure permits mathematical developments to improve the source coder but the reverse is also true. For the source coding designer, an imperfect distortion measure is still a useful tool to rank the source coding candidates for subjective evaluation. If a source coder manages consistently to minimise a particular distortion measure, the flaws in the measure become readily apparent in experimental studies on test signals. Exploration of the inconsistencies between the distortion measure and subjective evaluations may lead to improvements in the formulation of the measure.

Knowledge about the information user helps to improve the distortion measure and hence the source coder. For example, properties of the human auditory system, such as noise masking and non-uniform spectral sensitivity, are exploited in medium to low bit rate coding. The user, not the designer of the source coder, ultimately decides which features in the information are important and which are insignificant. Similarly, knowledge about the information source helps to improve the source coder. The source may have intrinsic constraints that restrict the ensemble of signals it can generate. For example, the reflection and absorption of glottis excitations by the human vocal tract are responsible for the high correlation of speech.

Signals typically exhibit structure through a non-uniform probability distribution of amplitudes or a non-uniform power spectral density (meaning that the signal is correlated). The former may be exploited by non-uniform quantisation and the latter by using prediction to reduce the dynamic range. Higher order statistical structures are possible but are more difficult to exploit. Source coding is useful because exploiting the statistics of information may reduce the distortion introduced when the information is passed through a limited channel.

The most important conclusion of the work described in this report is that source coding is possible with analogue and digital, albeit simulated, circuitry. Oversampled modulation is a clear example of source coding. Delta modulation exploits the correlations in oversampled signals and delta-sigma modulation exploits the non-uniform spectral sensitivity of the decimator to quantisation noise by shifting the noise to the least sensitive band, namely the high frequencies. These two source coding circuits may be combined, since they work in different ways, by cascading a predictive delta modulator with a noise shaping delta-sigma modulator.

The experiments on oversampled modulation show that particular source coders may be efficient only for particular information sources. For example, the cascaded modulator delivers the best performance, amongst the modulators considered, only when the information source approximates a high dynamic range narrow-band process added to a low dynamic range wide-band process. Source coding sacrifices some generality because assumptions are made about the information in the coder design, which may not apply to other signal sources.

B. Future work

This report has demonstrated the usefulness of source coding, has established the possibility of source coding signals for A/D conversion using simulated analogue and digital circuitry, and has presented some general principles about source coding. However, further work is now required to demonstrate the feasibility and usefulness of source coding signals for A/D conversion using real VLSI analogue and digital circuitry.

To demonstrate feasibility, a VLSI source coding A/D converter must be designed and implemented for a specific application that requires the conversion of a structured information source. To demonstrate usefulness, the A/D requirements of the application without VLSI source coding should exceed the capabilities of current low cost technology. A suitable application is the development of an affordable, high resolution, high frame rate, high dynamic range, colour digital video camera. An exact specification must still be determined but meeting all five requirements, not simply a few, appears to be challenging.

Video information does contain structure that can be exploited, particularly spatial and temporal correlations of pixel values.[17] Spatial correlations may be converted to temporal ones by raster scanning an image. The human visual system has perceptual properties that may also be exploited. For example, the eye is more sensitive to red or green light than to blue, suggesting fewer encoding resources should be given to the blue reading of a scene.[16] Additionally, the type of lighting in a scene, natural or artificial, affects the spectral distribution of the illumination, which may be exploited by a source coded A/D converter.

Furthermore, a scene $s(x,y)$, where x and y are pixel co-ordinates, is typically modelled as the product of an illumination $i(x,y)$ and a reflection $r(x,y)$, or the amount of illumination reflected by each point in the scene.[17] The logarithm of the pixel values thus represents the image as a sum of two independent processes ‘log $i(x,y)$’ and ‘log $r(x,y)$’. The illumination covers a large dynamic range but varies slowly in space whereas the reflection covers a small dynamic range but varies quickly. Such structuring should prove useful for source coding.

Figure 41 shows the first year of a two-year plan for this work. The work will begin by reviewing and developing simple digital algorithms to source code images. Once a suitable approach is identified, VLSI analogue and digital circuits for the camera and source coded A/D converter will be designed. Circuit layout overlaps with design since they are related in full-custom VLSI. The plan is to submit a chip for fabrication at the end of the first year.

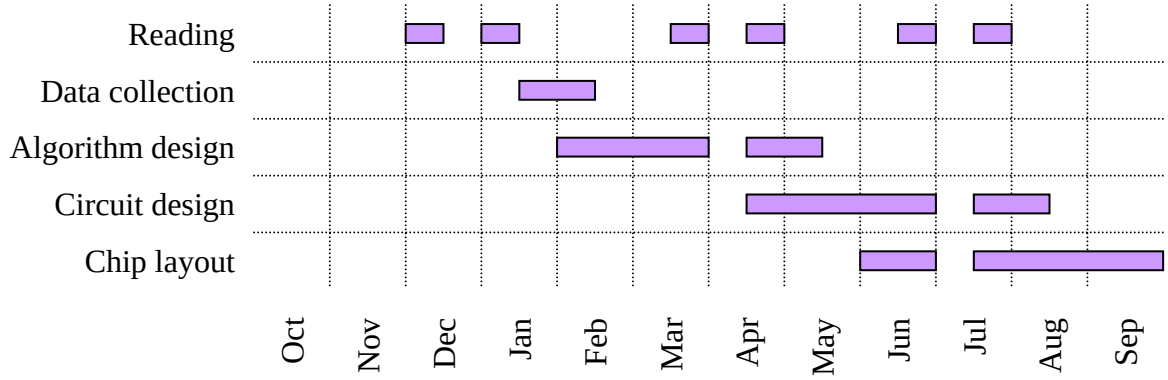


Figure 41. Plan for December 1998 to September 1999.

Figure 42 shows the second year of this plan. While the chip is being fabricated at the start of the year, the design will be documented. Equipment will also be acquired and assembled in anticipation of testing. Once the foundry returns the chip, testing will proceed for several months. Results of these tests and final conclusions will be documented at the end of the second year. Literature reviews will occur regularly over the two years, to acquire knowledge required for the project and to keep abreast of related developments. Time will also be required annually to collect test data from the Web and from high quality still cameras.

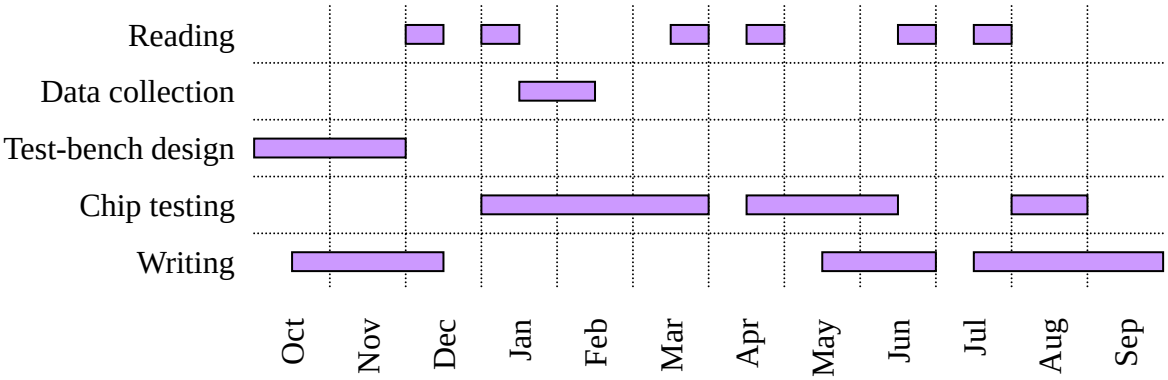


Figure 42. Plan for October 1999 to September 2000.

In conclusion, this report has hypothesised that source coding of signals, using VLSI circuitry, may be used to improve the conversion of analogue information to digital information. Theory and results from the literature and original work have been presented to argue that this idea is worth pursuing. Further work to design, implement, and test a digital video camera, incorporating a source coding A/D converter, is needed to confirm the hypothesis.

References

[1] B. K. Dolenko and H. C. Card, "Tolerance to Analog Hardware of On-Chip Learning in Backpropagation Networks", *IEEE Transactions on Neural Networks*, Vol. 6, No. 5, September 1995.

[2] M. Ismail, "Analog VLSI Neural Systems: Trends and Challenges", *Proceedings of the SPIE – The International Society for Optical Engineering*, Vol. 2492, Pt. 1, 1995.

[3] A. Murray and L. Tarassenko, *Analogue Neural VLSI – A Pulse Stream Approach*, Chapman & Hall, 1994.

[4] G. Cairns and L. Tarassenko, "Perturbation Techniques for On-chip Learning with Analogue VLSI MLPs", *Journal of Circuits, Systems, and Computers*, Vol. 6, No. 2, 1996.

[5] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[6] S. Haykin, *Communication Systems*, 3rd Ed., John Wiley & Sons, 1994.

[7] J. C. Candy and G. C. Temes, "Oversampling Methods for A/D and D/A Conversion", *Oversampling Delta-Sigma Data Converters Theory, Design and Simulation*, IEEE, 1992.

[8] S. P. Lloyd, "Least Squares Quantization in PCM", *IEEE Transactions on Information Theory*, Vol. IT-28, No. 2, March 1982.

[9] *Continuous Speech Recognition Corpus*, WSJ1, Linguistic Data Consortium, 1993.

[10] T. P. Barnwell, K. Nayebi and C. H. Richardson, *Speech Coding – A Computer Laboratory Textbook*, John Wiley & Sons, 1996.

[11] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1988.

[12] R. Steele (Ed.), *Mobile Radio Communications*, John Wiley & Sons, 1996.

[13] S. Venit and W. Bishop, *Elementary Linear Algebra*, 3rd Ed., PWS-KENT Publishing Company, 1989.

[14] A. Papoulis, *Circuits and Systems – A Modern Approach*, Holt, Rinehart and Winston, 1980.

[15] W. L. Lee and C. G. Sodini, *A Novel Higher-Order Interpolative Modulator Topology for High Resolution Oversampling A/D Converters*, M.Sc. thesis, Massachusetts Institute of Technology, June 1987.

[16] R. M. Boynton, *Human Color Vision*, Optical Society of America, 1992.

[17] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, Addison-Wesley, 1992.

Appendix

A. Lectures and seminars attended[a]

M. Zamora, “Parametric modelling and linear prediction”, Signal Processing Research Group, University of Oxford, November 1998.

IEE Analog Signal Processing Colloquium, October 1998.

Neural Computing Applications Forum, September 1998.

K. Laker and W. Sansen, “Design of analog integrated circuits for mixed-signal integrated systems”, University of Pennsylvania and K. U. Leuven, July 1998.

D. Crisan, “Stochastic filtering theory”, ICSTM London, June 1998.

G. Stein, “Geometric and photometric constraints: motion and structure from three views”, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, March 1998.

P. N. Belhumeur, “The bas-relief ambiguity”, Center for Computational Vision and Control, Yale University, February 1998.

N. Townsend, “Error prediction for neural networks (garbage in, garbage out)”, Signal Processing Research Group, University of Oxford, February 1998.

Powell, “Radial basis function methods for global optimisation”, University of Cambridge, February 1998.

T. Roska, “New results on CNN computing technology: fault tolerance, spatial adaptivity, bio-inspired devices”, Hungarian Academy of Sciences, December 1997.

A. Davidson, “Mobile robot navigation using active vision”, Robotics Research Group, University of Oxford, November 1997.

W. J. Fitzgerald, “Applications of Bayesian methods to signal processing”, Signal Processing Laboratory, University of Cambridge, October 1997.

M. Isard, “Stochastic methods for visual motion analysis”, Robotics Research Group, University of Oxford, October 1997.

Signal Processing Research Group, University of Oxford, fortnightly meetings since October 1997.

[a] These references identify the affiliations and institutions of the speakers but not the locations of their lectures or seminars. Almost all lectures and seminars were attended in Oxford.