DESIGN IMPLEMETATION AND STUDY OF A CODE (DPC) MODULATOR AND DE-MODULATOR USING MATLAB AND SIMULINK

NITIABALAN KARUPPAYAH

Universiti Sains Malaysia

2005

ABSTRACT

This thesis presents a detailed study of linear Pulse Code Modulation and also Differential Pulse Code Modulation using SIMULINK. Theoretical coding concepts are discussed extensively, each providing examples to accommodate better understanding of both PCM and DPCM concept. Non-linear PCM and DPCM cases are not discussed in this thesis. Basic introduction for the proposed systems are explained together with introduction to SIMULINK program. A step-by-step explanation of each stage is provided, along with accompanying visual parameter fields in the APPENDIX section. Two types of test input signals are given to both constructed models; namely, a sine wave of 100 radians/second and a simple ramp function. Each stage of the blocks operations are studied in detail and the outputs are presented in graphical waveforms. Signal-to-Quantization Noise ratio is calculated for each system and compared. Arising problems, such as the need of amplitude equalization, during the implementation of these circuits are also mentioned, along with steps to overcome the mentioned irregularities and redundant results. Two types of tests can be done to reach the final conclusion; the first, an objective test, where SNR are calculated and graphical results are produced. The second test is a subjective test, where Mean Opinion Score (MOS) is obtained to verify whether the speech coding encoded is discernable to a human ear and determining the quality of the output. Differences between PCM and DPCM are stated based on the results obtained from the simulation of both systems; namely, in terms of bandwidth requirements, quantization noise, coding, simplicity of the implemented circuit, and also the quality of the output obtained.

ABSTRAK

Tesis ini mempersembahkan satu laporan lengkap berkenaan Pemodulatan Kod Denyut (PCM) lelurus dan juga Pemodulatan Kod Denyut Kebezaan (DPCM) dengan menggunakan program SIMULINK. Konsep-konsep teori berkenaan kaedah-kaedah tersebut dikaji dan diterangkan secara menyeluruh, dengan setiap satu kaedah diterangkan dengan contoh-contoh yang berkaitan untuk memudahkan pemahaman untuk pihak-pihak yang ingin mempelajari kaedah-kaedah yang disebutkan di atas. Kaedah Pemodulatan Kod Denyut (PCM) tak lelurus dan juga Pemodulatan Kod Denyut Kebezaan (DPCM) tak lelurus tidak dibincangkan dalam tesis ini. Hanya kes-kes lelurus diberikan perhatian. Penjelasan langkah demi langkah diberikan pada setiap peringkat bersama parameternya dalam bentuk visual di bahagian Lampiran. Dua jenis isyarat masukan ujian dikenakan terhadap kdua-dua model yang dibina; iaitu satu isyarat sinus 100 radians/saat dan satu isyarat fungsi tanjakan. Operasi setiap peringkat blok yang dibina dibina dan diterangkan secara mendalam dan menyeluruh. Keluaran setiap blok dipersembahkan dalam bentuk graf. Nilai nisbah isyarat terhadap hangar, yang juga dikenali sebagai nilai SNR dikira bagi setiap system dalam kedua-dua kes dan seterusnya dibandingkan. Masalah berbangkit, seperti perlunya penyamaan amplitud, semasa implementasi litar-litar tersebut juga dibincangkan bersama dengan langkah-langkah untuk mengatasi sebarang ketaksekataan dan masalah lelebihan keputusan. Dua kaedah boleh dijalankan untuk mendapatkan kesimpulan terakhir dalam tesis ini. Yang pertama ialah, satu ujian objektif, dimana nilai nisbah SNR dikira dan keputusan dalam bentuk graf diperoleh. Kaedah kedua melibatkan satu ujian subjektif, dimana Min Skor Pendapat - *Mean Opinion Score* (MOS) diperoleh untuk mengesahkan pengkodan suara. Perbezaan di antara Pemodulatan Kod Denyut (PCM) lelurus dan Pemodulatan Kod Denyut Kebezaan (DPCM) yang dikemukakan dalam tesis ini adalah berdasarkan keputusan-keputusan yang diperoleh daripada simulasi model yang dibina untuk kedua-dua system. Keputusan-keputusan ini melibatkan keperluan lebar jalur, hangar pengkuantuman, pengkodan, kemudahan litar yang dibina dan diimplementasi, serta kualiti isyarat keluaran yang diperoleh.

ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere gratitude to all those who have contributed in completing this project. First of all, my sincere thanks goes to my final year project supervisor, Professor Farid Ghani for his professional supervision and constant attention throughout my final year project. It has been a privilege to work under a knowledgeable and supportive supervisor. I appreciate the efforts and numerous suggestions of Prof. Farid Ghani, to improve the performance and outcome of the project. I would also like to extend my sincere appreciation to the Second Examiner, Associate Professor Lim Chee Peng, who reviewed my final draft of this thesis. My special thanks goes to the Administration of School of Electrical & Electronic Engineering, for providing this wonderful opportunity to carry out this final year project.

The project has indefinitely exposed me to the current technology and also to increase my knowledge and skills in software design implementation. I was also exposed to do a proper research in order to get a clear idea on the selected topic. Finally, my sincere thank to my parents and my friends for their support and motivation throughout the completion of this project.

Nitiabalan Karuppayah
Science University of Malaysia
nitiabalan@yahoo.com

LIST OF FIGURES

Figure	Caption	Page Number
1.1	Analog to digital conversion	2
1.2	Block diagram of digital communication system	4
2.1	Pulse Code Modulation (PCM) Coder	7
2.2	Basic communication system	8
2.3	A simple PCM communication system	9
2.4 (a)	Nonuniform Quantizer ($L=7$)	11
2.4 (b)	Uniform Quantizer ($L=8$)	11
2.4 (c)	Nonuniform Quantizer ($L=7$)	12
2.4 (d)	Uniform Quantizer ($L=8$)	12
3.1 (a)	Block diagram of DPCM encoder	25
3.1 (b)	Block diagram of DPCM decoder	25
3.2	Linear predictor of order N	30
3.3 (a)	Definitions of sequences in PCM	35
3.3 (b)	Definitions of sequences in DPCM	35
3.4	Error propagation in DPCM encoding	36
3.5	Transmission error patterns	37
3.6	Delta modulation encoder and decoder	38
3.7	Slope overload and granular noise of delta modulation system	39
3.8	Performance of several DPCM coding schemes	40
4.1	Basic Simulation Diagram of Pulse Code Modulation	47
4.2	Basic Simulation Diagram of Differential Pulse Code Modulation	53
4.3	Pulse Code Modulation (PCM) Simulation Diagram for Sine Wave Input	57
4.4	Graphical waveform obtained from Scope 1 (Figure 4.3)	60

Figure	Caption	Page Number
4.5	Graphical waveform obtained from Scope 2 (Figure 4.3)	61
4.6	Graphical waveform obtained from Scope 5 (Figure 4.3)	62
4.7	Graphical waveform obtained from Scope 3 (Figure 4.3)	63
4.8	Graphical waveform obtained from Scope 6 (Figure 4.3)	64
4.9	Graphical waveform obtained from Scope (Figure 4.3)	65
4.10	Pulse Code Modulation (PCM) Simulation Diagram for Ramp Function Input	67
4.11	Graphical waveform obtained from Scope 1 (Figure 4.10)	68
4.12	Graphical waveform obtained from Scope 2 (Figure 4.10)	69
4.13	Graphical waveform obtained from Scope 3 (Figure 4.10)	70
4.14	Graphical waveform obtained from Scope 4 (Figure 4.10)	71
4.15	Graphical waveform obtained from Scope 6 (Figure 4.10)	72
4.16	Graphical waveform obtained from Scope (Figure 4.10)	72
4.17	Differential Pulse Code Modulation (DPCM) Simulation Diagram for Sine	73
4.18	Graphical waveform obtained from Scope 6 (Figure 4.17)	74
4.19	Graphical waveform obtained from Scope 1 (Figure 4.17)	75
4.20	Graphical waveform obtained from Scope 2 (Figure 4.17)	76
4.21	Graphical waveform obtained from Scope 3 (Figure 4.17)	77
4.22	Graphical waveform obtained from Scope 4 (Figure 4.17)	78
4.23	Graphical waveform obtained from Scope 5 (Figure 4.17)	79
4.24	Differential Pulse Code Modulation (DPCM) Simulation Diagram for Ramp Function Input	81
4.25	Graphical waveform obtained from Scope 5 (Figure 4.24)	82
4.26	Graphical waveform obtained from Scope 1 (Figure 4.24)	83
4.27	Graphical waveform obtained from Scope 2 (Figure 4.24)	83
4.28	Graphical waveform obtained from Scope 3 (Figure 4.24)	84
4.29	Graphical waveform obtained from Scope 4 (Figure 4.24)	85

LIST OF TABLES

Table	Caption	Page Number
2.1	Ordinal numbers and its representations in binary	16
4.1	Quantization partitions and its representations	58
4.2	Number of Bits and its corresponding Signal-to-Quantization Noise ratio.	66

CHAPTER 1
INTRODUCTION

1.1 Background

One of the major applications of digital technology is signal processing. Basically, signal processing refers to an operation on a signal to enhance or transform its characteristics. Signal processing can be applied to either analog or digital waveforms. Amplification, equalization, modulation and filtering are common examples of signal processing functions.

Digital signal processing (DSP) refers to the use of digital logic and arithmetic circuits to implement signal processing functions on digitized signal waveforms. Usually, analog signals are converted to digital representations for the purpose of processing them digitally and storing the result. Then, the digital representations of the processed signals are converted back to analog form. Digital signal processing is much desired for analog signal mainly due to the following reasons;

- Reproducibility : The immunity of digital circuits to small imperfections and parasitic elements implies that circuits can be produced with consistent operational characteristics without fine adjustments or aging tolerances.

- Programmability : A single basic structure can be used for a variety of signal types and applications by merely changing an algorithmic or parametric specification in a digital memory.

- Time sharing : A single digital signal processing circuit cab be used for multiple signals by storing temporary results of each process in random-access memory and processing each signal in a cyclic (time-divided) fashion.

- Automatic test : Since the inputs and outputs of a digital signal processing circuit are digital data, test can be performed routinely by comparing test responses to data patterns stored in memory.

- Versatility : Because of the decision-making capabilities of digital logic, digital signal processing can perform many functions that are impossible or impractical with analog implementations.

An analog signal can always be converted into digital form by combining three basic operation; *sampling*, *quantizing*, and *encoding* as shown in Figure 1.1 below.

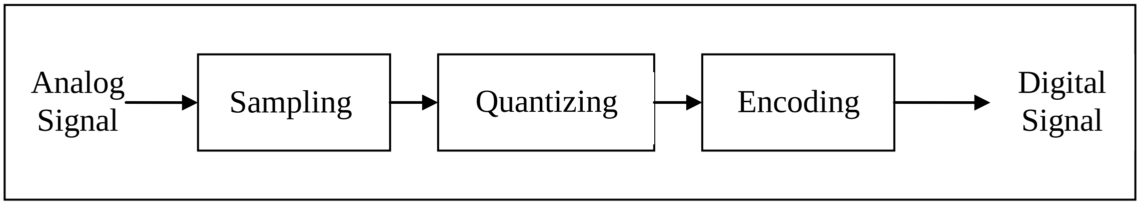

Figure 1.1 Analog to digital conversion

In the sampling operation, only sample values of the analog signal at uniformly spaced discrete instants of time are retained. In the quantizing operation, each sample value is approximated by the nearest level in a finite set of discrete levels. In the encoding operation, the selected level is represented by a *code word* that consists of a prescribed number of code elements.

As a result of sampling and quantizing operations, *errors* are introduced into the digital signal. These errors are nonreversible in that it is not possible to produce an exact replica of the original analog signal from its digital representation. However, the errors are under the circuit designer’s control. By a proper selection of sampling rate and code-word length (i.e. number of quantizing levels), the errors due to sampling and quantizing can be made so small that the difference between the analog signal and its digital reconstruction is not discernible by a human observer.

In source coding, the *encoder* maps the digital signal generated at the source output into another signal in digital form. The mapping is one-to-one, and the objective is to eliminate or reduce redundancy so as to provide an efficient representation of the source output. Since the source encoder mapping is one-to-one, the *source decoder* simply performs the inverse mapping and thus delivers to the user destination a reproduction of the original digital source output. The primary benefit thus gained from the application of source coding is a reduced bandwidth requirement.

In channel coding, the objective is for the *encoder* to map the incoming digital signal into a channel input and for the decoder to map the channel output into an output digital signal in such a way that the effect of channel noise is minimized. That is, the combined role of the channel encoder and decoder is to provide for reliable communication over noisy channel. This provision is satisfied by introducing *redundancy* in a prescribed fashion in the channel encoder and exploiting it in decoder to reconstruct the original encoder input as accurately as possible. Thus, in source coding, redundancy is removed, while in channel coding, redundancy is controlled.

Clearly, source coding may be performed alone, channel coding alone or the two together. In the latter case, the source coding is performed first, followed by channel encoding in the transmitter, as illustrated in Figure 1.2. In the receiver, the reverse order is applied; channel decoding is performed first, followed by source decoding. Whichever combination is used, the resulting improvement in system performance is achieved at the cost of increased bandwidth.

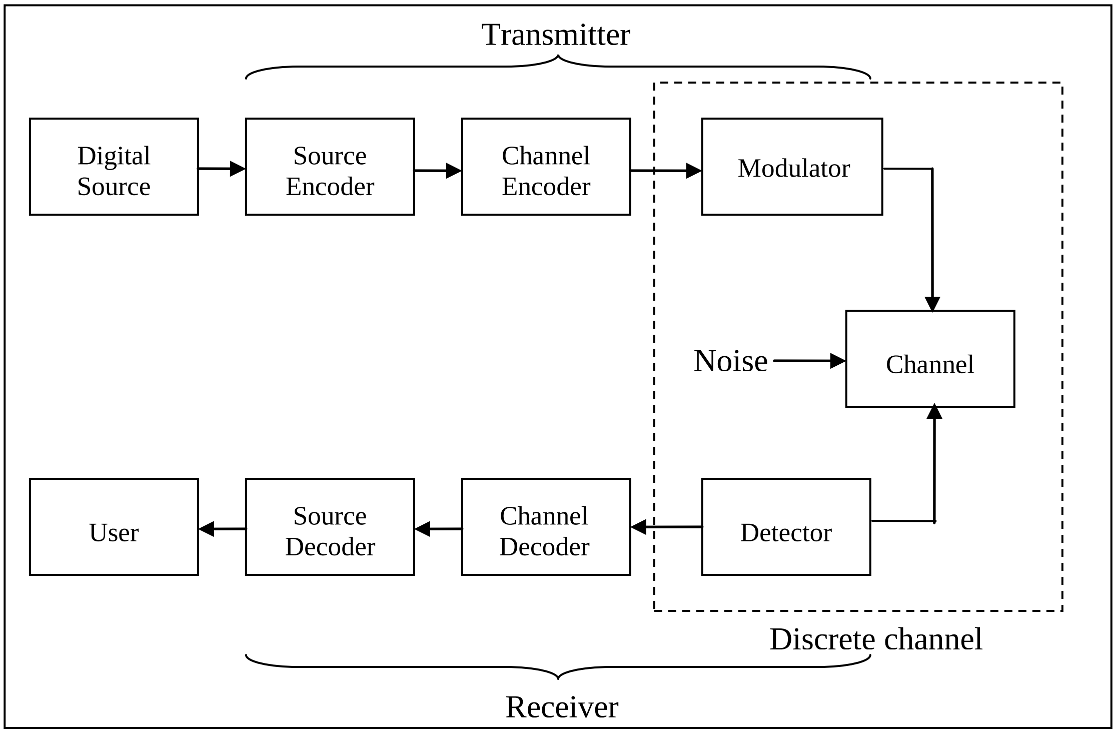

Figure 1.2 Block diagram of digital communication system.

As for modulation, it is performed with the purpose of providing for the efficient transmission of the signal over the channel. In particular, the *modulator* (last stage of the transmitter in Figure 1.2), operates by keying shifts in the *amplitude*, *frequency*, or *phase* of a sinusoidal carrier wave to the channel encoder output. The digital modulation technique for doing so is referred to as *amplitude-shift keying*, *frequency-shift keying*, or *phase-shift keying*, respectively. The detector, performs demodulation (inverse of modulation), thereby producing a signal that follows the time variations in the channel encoder output.

The details of modulation and coding used in a digital communication system depend on the characteristics of the channel and the applications of interest. The two channel characteristics, bandwidth and power, are the primary communication resources available to the designer. Other channel characteristics of particular concern are the degree to which the amplitude and phase responses of the channel are determined, whether the channel is *linear* or *non-linear*, and how free the channel is from external interference.

1.2 Scope of the Project

This project is mainly concerned on coding signals using Pulse Code Modulation and Differential Pulse Code Modulation, using the help of simulated diagram and environment of SIMULINK. The results of both methods are obtained and compared to determine any losses or improvements in the reconstructed signal.

Due to the availability of digital signal processing components, some of which contain internal A/D converters, digital processing is generally the most effective means of implementing a DPCM algorithm. In fact, most DPCM applications involve processing speech signals that have been digitized into standard 64-kbps PCM formats.

The Differential Pulse Code Modulation implemented circuit usually requires no analog processing. The decoders in these implementations are exactly like the feedback implementations in the corresponding encoder. This reinforces the fact that the feedback loop generates an approximation of the input signal (delayed by one sample). If no channel errors occur, the decoder output (before filtering) is identical to the feedback signal. Thus the closer the feedback signal matches the input, the closer the decoder output matches the encoder input.

DPCM uses an algorithm to predict an information signal’s future value. Thus, it reduces the redundancy in a signal and allows the information contained in it to be transmitted using fewer symbols, shorter time and/or lower signal power. DPCM coding technique achieves its bit rate reduction by removing redundancy from the input signal. This technique also has a high probability of being implemented on modern telecommunication field as it proves to have a wide range of variable implementation characteristics. These are the characteristics that shall be investigated in this project.

To do so, a PCM model is built using SIMULINK according to the process stages involved. Then a DPCM model is built to compare the results of both models. Sine waves at appropriate stages are captured, compared and studied in detail.

1.3 Report Organization

Chapter 1 is mainly about the reason why signal processing is important in our daily life today. This chapter also describes the background of basic signal coding processes, briefly. Source coding and channel coding are discussed to ensure a clear picture of the project is drawn prior to explaining the involved processes. Key characteristics of modulation techniques are also laid out.

Chapter 2 meanwhile discusses the theory behind Pulse Code Modulation and its processes. Each process is explained in length using appropriate diagrams and graphs. Possible errors in this technique are mentioned here, to prepare the reader to expect redundancies in the processed output. Examples of applications where Pulse Code Modulation is used is also given for better understanding.

Chapter 3 describes Differential Pulse Code Modulation in detail. DPCM is introduced in comparison to PCM. Additional steps involved in DPCM, such as the linear prediction is described with calculation examples of Quantization-to-Noise ratio values. Then, DPCM is compared with PCM using theoretical aspects of both techniques.

Chapter 4 is the core part of this project, where both PCM and DPCM techniques are implemented using SIMULINK. MATLAB and SIMULINK are introduced accordingly, to familiarize the reader with its functions. Blocksets and Toolboxes necessary for this project are mentioned and explained. During implementation, two types of test input signal are given; a sine wave and a ramp function. The outputs are compared with the original input signal to be analyzed. Results of the implementations are discussed extensively.

Finally, *Chapter 5* shall provide the overall conclusions which are derived from the results and discussion from the previous chapter.

CHAPTER 2

Pulse Code Modulation

2.1 Introduction

PCM is the best established and most implemented and also the most applied of all digital coding systems. One reason for this is simply the fact that PCM is the earliest developed and also the best understood coding system. It is also the conceptually simplest system; further, most versions of PCM coders are instantaneous, implying a coding delay of no more than one sample period. Another reason for the importance of PCM is that, it is not signal-specific, but actually it is versatile. For example, PCM is not mismatched to voice band data waveforms in a way speech-specific DPCM coders are. Thus, PCM is widely accepted as a standard against which to calibrate other approaches to waveform digitization. All waveform coders, with the exception of Deltamodulation, involves stages of PCM coding and decoding.

A PCM coder is simply a waveform sampler followed by an amplitude quantizer as shown in figure below.

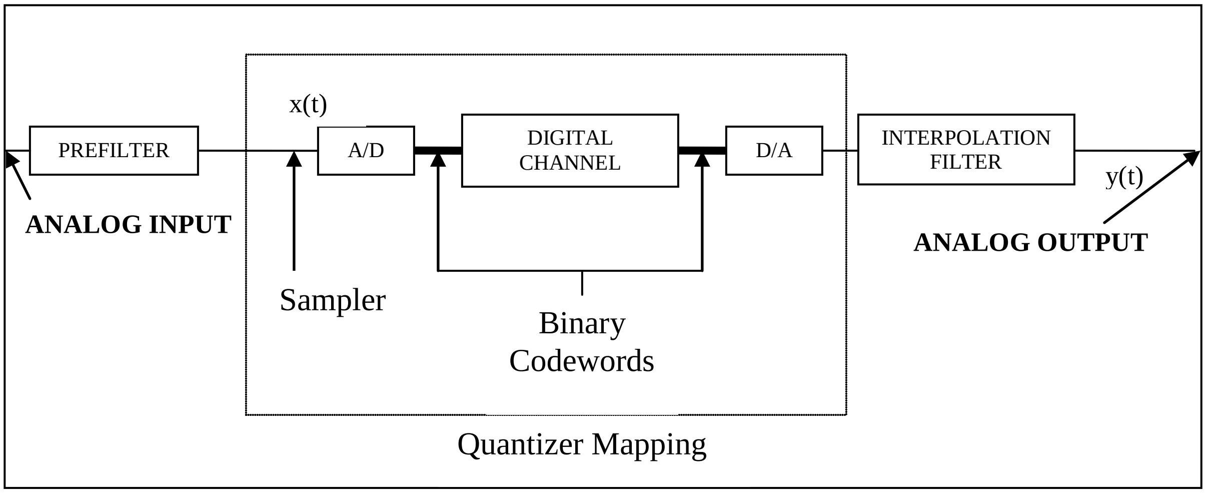

Figure 2.1 Pulse Code Modulation (PCM) Coder

2.2 Sampling

Refering to the waveform communication system in Figure 2.2(a) below, all digital waveform systems components are the devices labeled sampler and interpolation filter.

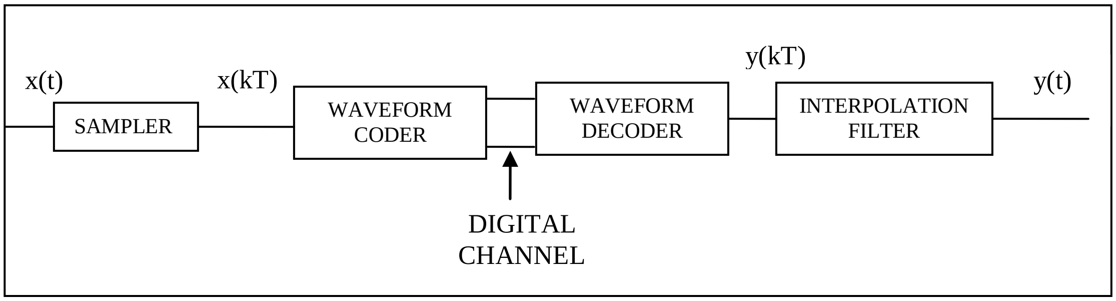

Figure 2.2 Basic communication system

The quantities x (kT), are discrete-time samples, derived from a continuous time signal x (t) at times kT, where T is the sampling period. T is the reciprocal of the sampling frequency or sampling rate f_s.

$$T = 1/f_s \tag{2.1}$$

The samples x (kT) are equally spaced in time; almost assumed as uniform sampling for simplicity. The aim of the interpolation filter at the receiver is to derive a continuous waveform y (t) from the decoder samples y (kT) using an appropriate interpolation procedure. Under certain conditions, sampling and interpolation operations are error-free for bandlimited x (t), and the source of reconstruction noise is in fact caused by the waveform coder and waveform decoder boxes in Figure 2.2.

A simple example of a PCM communication system is shown in Figure 2.3. Sampling and coding operations can be combined into a single box which is widely known as an analog-to-digital or A/D converter; and the PCM decoder can be simply represented as a digital-to-analog or D/A converter. Prefiltering or bandlimiting is a prerequisite for error-free sampling and interpolation, unless input $x(t)$ is already bandlimited. The prefilter is also called an anti-aliasing filter.

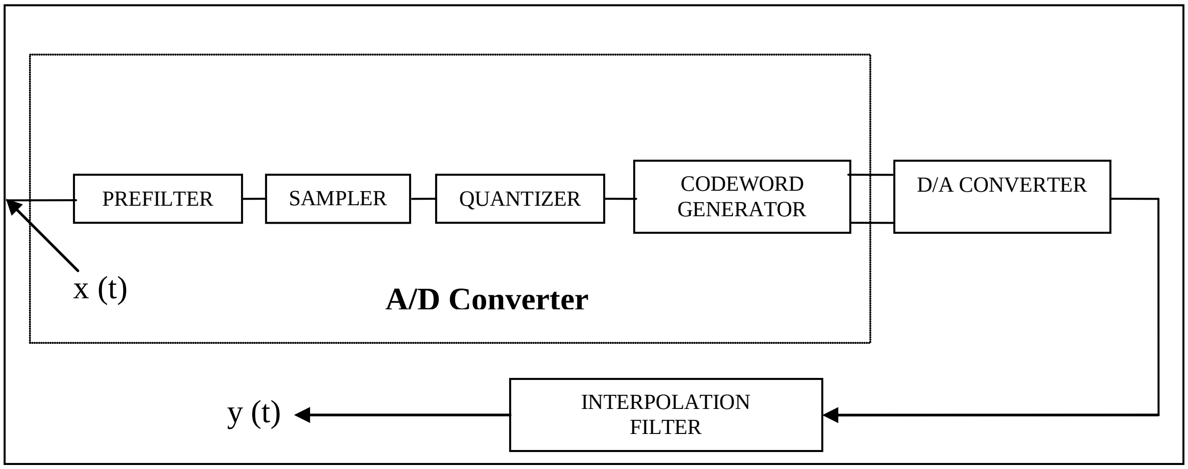

Figure 2.3 A simple PCM communication system

Prefiltering is a critical step in any digital coding system. The purpose of prefiltering is to prevent or minimize aliasing effects which can disrupt speech and image coding. Prefiltering operation introduces its own distortion into aliasing if the nominal bandwidth is appropriately chosen. SNR characterizations of coding systems are based on the comparison of prefiltered waveforms with coded prefiltered waveforms. These SNR characterizations will not include the contribution of prefiltering errors; rather, they will reflect only the distortions introduced into the prefiltered waveform in the coding process [Jayant and Noll, 1984].

$$SNR(dB)\big|_{PCM} = \begin{cases} 6.02R - 10, speech \\ 6.02R, image \end{cases} \text{, where } R \text{ is the number of bits/sample.} \quad (2.2)$$

The transmission rate, I of a PCM system, as for any other coding system is given by the following equation:

$$I = f_s R \text{ bits/second} \quad (2.3)$$

Coder inputs can either be the amplitude-continuous samples $x(kT)$ or finely amplitude-quantized versions, as obtained at the output of an A/D converter preceding the coder box.

Once a certain desired value of nominal bandwidth is established, the need to prevent aliasing will allow very little flexibility in the design of prefilter or in the design of minimum sampling rate f_s; and the only way to reduce the transmission rate, I of the digital coding system will be by controlling the parameter R in the Equation 2.3.

2.3 Quantization

Information-bearing waveforms are typically continuous-amplitude and continuous-time in nature. Hence, analog-to-digital (A/D) conversion is needed to produce a discrete representation of the waveform. The process of amplitude quantization is needed to limit the number of possible amplitudes to a finite value. The operations of sampling and quantization could be performed in either order; but in practice, sampling precedes quantization. Amplitude quantization is a crucial step in digital communication, because it determines to a great extent the overall distortion as well as the bit rate necessary to communicate the waveform to the receiver.

Amplitude quantization is the procedure of transforming a given signal amplitude $x(n)$ at the time n into an amplitude $y(n)$ taken from a finite set of possible amplitudes. Memoryless and instantaneous quantization is a situation where the transformation at time n is not affected by earlier or later input samples. Memoryless quantization is inefficient if the samples are statistically dependent. These dependencies are therefore eliminated prior to quantization in the coder.

The time index is dropped when dealing with memoryless quantizers, and symbols such as x is used instead of $x(n)$. Quantizer input need not always be the amplitude of the primary waveform to be coded. It may as well be a function of this waveform, for example, a difference between neighbouring samples or some type of spectral information about input samples; and the quantizing process may only be one component of the overall coding system.

The quantizer characteristic, a staircase function by definition, is given by $y = Q(x)$. Several types of $Q(x)$ are shown in Figure 2.4 [Jayant and Noll, 1984].

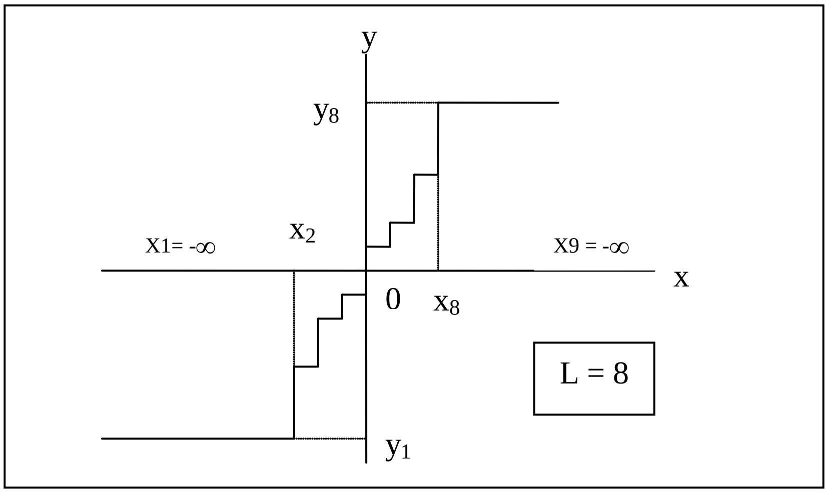

Figure 2.4 (a)

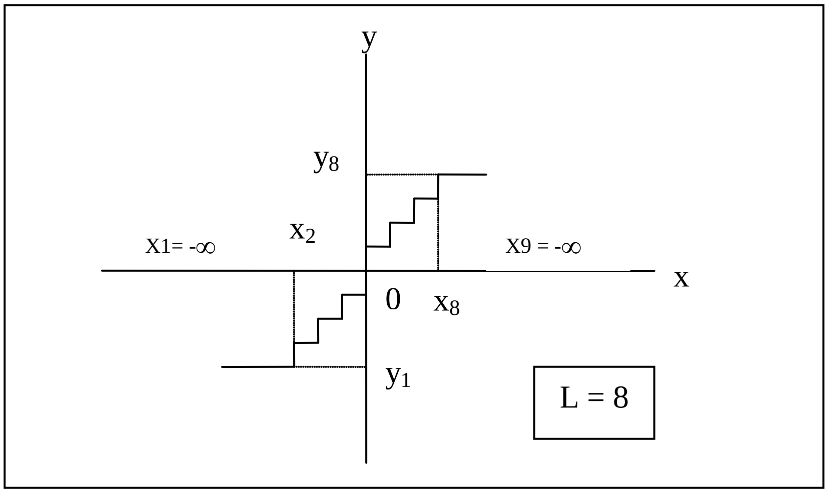

Figure 2.4 (b)

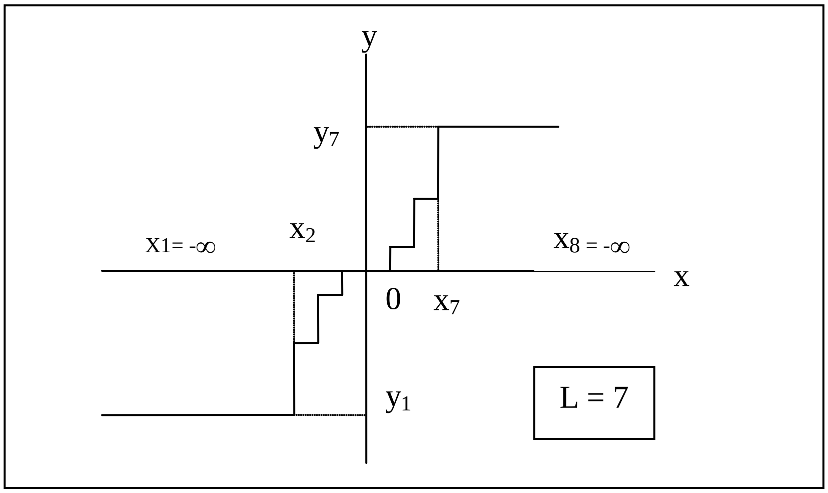

Figure 2.4 (c)

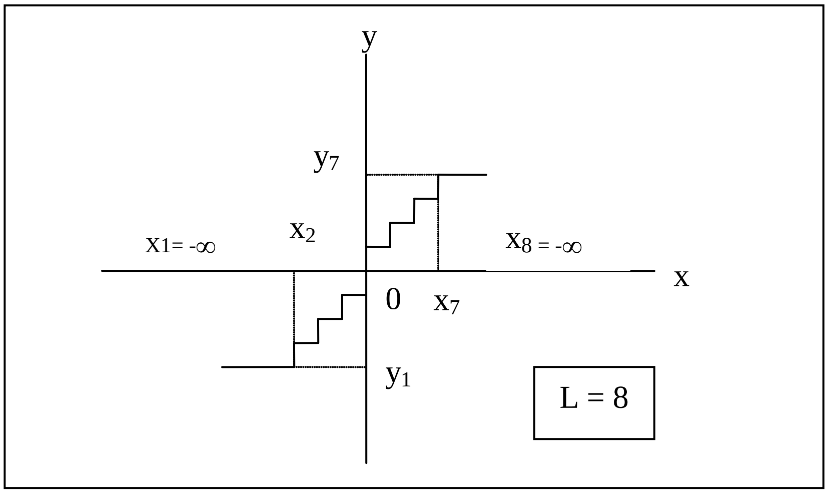

Figure 2.4 (d)

All the above characteristics are symmetric about zero. The quantizers in parts (a) and (c) of Figure 2.4 are nonuniform, with decision level and reconstruction level intervals being different in length and being functions of *k*. The quantizers in parts (b) and (d) of Figure 2.4 are uniform quantizers.

Uniform Quantizers : $y_{k+1} - y_k = \Delta$; $k = 1,2,\ldots..,L-1$ (2.4)

$x_{k+1} - x_k = \Delta$; for finite x_k, x_{k+1} (2.5)

The characteristic $Q(x)$ can also be of *midtread* or *midrise* type, depending on whether zero is one of the output levels or not. For symmetry, L is chosen to be odd for midtread quantizers, as in parts (c) and (d) of Figure 2.4. Usually, *midrise* quantizers are assumed to have even L.

Inherent in the quantization process is an error between input x and output y. This error is called quantization error, defined as the difference between x and y :

$$q = x - y = x - Q(x), \qquad \text{where } y \in \{y_1, y_2, ...y_L\} \tag{2.6}$$

The quantization error is totally deterministic for a given *x* because an input $x \in f_k$, always leads to a single output $\mathrm{y_k}$. The term ‘quantization noise’ is usually used to describe the errors introduced by the quantizers.

Here, we let the random variable *Q* denote the quantization error, and *q* denotes its sample value. Assuming that the random variable *Q* is uniformly distributed over the range of $-\Delta/2$ to $\Delta/2$,

$$f_Q(q) = \begin{cases} \dfrac{1}{\Delta} & -\dfrac{\Delta}{2} \le q \le \dfrac{\Delta}{2} \\ 0 & otherwise \end{cases} \tag{2.7}$$

where *f_Q(q)* is the probability density function of the quantization error. To ensure that the incoming signal does not overload the quantizer, the mean of the quantization error is set to be zero, and its variance σ_Q^2 is the same as the mean-square value, as shown below

$$\sigma_Q^2 = E\left[Q^2\right] \tag{2.8}$$

$$= \int_{-\aleph}^{\infty} q^2 f_Q(q) dq \tag{2.9}$$

Substituting, we get

$$\sigma_Q^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} q^2 dq \tag{2.10}$$

$$= \frac{\Delta^2}{12} \tag{2.11}$$

Thus, the variance of the quantization noise, produced by a uniform quantizer, grows as the square of the step size. Letting the variance of the baseband signal at the quantizer input be denoted by σ_Q^2, original signal plus quantization noise is obtained when the signal is reconstructed at the receiver output. Therefore, an *output-to-quantization signal noise ratio (SNR)* may be represented as;

$$(SNR)_o = \frac{\sigma_X^2}{\sigma_Q^2} = \frac{\sigma_X^2}{\Delta^2/12} \tag{2.12}$$

From the equation, the smaller the step size Δ, the larger will the SNR be.
SNR is also expressed in decibels as,

$$10\log_{10}(SNR)_o \approx 6n - 7.2 \tag{2.13}$$

Channel bandwidth *B*, for binary PCM system is typically *nW*. Thus, the equation can be rewritten as,

$$10\log_{10}(SNR)_o \approx 6\left(\frac{B}{W}\right) - 7.2 \tag{2.14}$$

.

2.4 Coding

In combining the processes of sampling and quantizing, the specification of a continuous baseband signal becomes limited to a discrete set of values, but not in the form best suited for transmission over a line, radio path, or optical fiber. To exploit the advantages of sampling and quantizing, the use of *encoding process* is required to translate the discrete set of sample values to a more appropriate form of signal. Any plan for representing each member of this *code*. One of the discrete events in a code is called a code element or symbol. For example, the presence or the absence of a pulse is considered as a symbol. A particular arrangement of symbols used in a code to represent a single value of the discrete set is called a code-word or character.

In a *binary code*, each symbol may be either of two distinct values or kinds, such as the presence or absence of a pulse. The two symbols of a binary code are usually denoted as 0 and 1. In a *ternary code*, each symbol may be one of three distinct values or kinds, and so on for other codes. However, the maximum advantage over the effects of noise in a transmission medium is obtained using a binary code, because a binary symbol withstands a relatively high level of noise and is easy to regenerate.

Assuming that in a binary code, each code-word consists of *n* bits. Then, using such a code, a total of 2^n distinct numbers can be represented. For example, a sample quantized into one of $2^4 = 16$ levels may be represented by a 4-bit code-word. There are several ways of establishing a one-to-one correspondence between representation levels and code-words. A more convenient way is to express the ordinal number of the representation level as a binary number. In the binary number system, each bit has a place-value that is a power of 2, as illustrated in the table below for the case of $n = 4$.

Table 2.1 Ordinal numbers and its representations in binary.

Ordinal Number of Representation Level	Level Number Expressed as Sum of Powers of 2	Binary Number
0		0000
1	2^0	0001
2	2^1	0010
3	$2^1 + 2^0$	0011
4	2^2	0100
5	$2^2 + 2^0$	0101
6	$2^2 + 2^1$	0110
7	$2^2 + 2^1 + 2^0$	0111
8	2^3	1000
9	$2^3 + 2^0$	1001
10	$2^3 + 2^1$	1010
11	$2^3 + 2^1 + 2^0$	1011
12	$2^3 + 2^2$	1100
13	$2^3 + 2^2 + 2^0$	1101
14	$2^3 + 2^2 + 2^1$	1110
15	$2^3 + 2^2 + 2^1 + 2^0$	1111

2.5 Regeneration

The most important feature of PCM system lies in the ability to control the effects of distortion and noise produced by transmitting PCM wave through a channel. This capability is accomplished by reconstructing the PCM wave by means of chain of *regenerative repeaters* located at sufficiently close spacing along the transmission route. Three basic functions are performed by a regenerative repeater, namely, *equalization*, *timing*, and *decision making*. The equalizer shapes the received pulses so as to compensate for the effects of amplitude and phase distortions produced by imperfections in the transmission characteristics of the channel. The timing circuit provides periodic pulse train, derived from the received pulses, for sampling the equalized pulses at the instants of time where the signal-to-noise ratio is at maximum. The decision device is enabled when, at the sampling time determined by the timing circuit, the amplitude of the equalized pulse plus noise exceeds a predetermined voltage level. Thus, for example, in a PCM system with on-off signaling, the repeater makes a decision in each bit interval as to whether or not a pulse is present. If the decision is "yes", a clean new pulse is transmitted to the next repeater. If, on the other hand, the decision is "no", a clean base line is transmitted. In this way, the accumulation of distortion and noise in a repeater span is completely removed, provided that the disturbance is not too large to cause an error in the decision-making process. Ideally, except for delay, the regenerated signal is exactly the same as the signal originally transmitted. In practice however, the regenerated signal differs slightly from the original signal for two main reasons:

- The presence of channel noise and interference causes repeater to make wrong decisions occasionally, thereby introducing *bit errors* into the regenerated signal.
- If the spacing between received pulses deviates from its original value, a jitter is introduced into the regenerated pulse position, thereby causing distortion.

2.6 Decoding

The first operation in the receiver is to regenerate (i.e., reshape and clean-up) the received pulses. These clean pulses are then regrouped into code-words and decoded (i.e. mapped back0 into a quantized PAM signal. The *decoding process* involves generating a pulse with the amplitude of which is the linear sum of all the pulses in the code-word, with each pulse weighted by its place-value (2^0, 2^1, 2^2, 2^3, …) in the code.

2.7 Reconstruction

The final operation in the receiver is to recover the analog signal. This is done by passing the decoder output through a low-pass reconstruction filter whose cutoff frequency is equal to the message bandwidth *W*. Assuming that the transmission path is error-free, the recovered signal include no noise with the exception of the initial distortion introduced by the quantization process. [Simon Haykin, 1988]

It should be noted that for high quality representation, the minimum value of R is 8 bits/sample for speech and video. Speech can only be encoded with high quality using 8-bit PCM quantizer, so that the coder can cope with expected variations in input level. In the PCM coding of video, although some amount of companding may be advantageous, associated gains are much less than in speech, and 8-bit linear (i.e. uniformly quantized) PCM is sufficient to provide very high output quality. An 8-bit linear format implies a maximum-intensity to minimum-intensity ratio of $2^8 = 256$.

The adaptive quantizers have been applied widely in low bit rate speech systems using techniques such as DPCM, SBC, and TC. At these low bit rates (say, $R < 4$ bits/sample), adaptive quantization provides a significant performance gain over time-invariant non-uniform quantization. Commercial PCM systems, on the other hand, are based on the use of 8 bits/sample; at these higher bit rates, a time-invariant quantizer can provide *toll quality* speech, provided an appropriate form of log-companding is employed. Log-companding increases the range of input level variation over which a specific level of

SNR can be maintained; and in particular, ensures an adequate SNR frequently transmits low-level segments in speech communication.

With a speech sampling rate of f_s = 8 kHz, and coding with R = 8 bits/sample, the transmission rate is I = 64 kb/s. This is a very important number in digital transmission practice. The bit rate 64 kb/s is the smallest unit for the well-established digital hierarchies and is used as an important standard rate for emerging digital networks. This includes Integrated Services Digital Networks (ISDN) where a channel rate of 64 kb/s, or multiples or sub-multiples thereof, may be allocated not only for digital subscribers, but for digital transmission of other signals such as data.

Taking an example of a speech input sampled at f_s = 8 kHz, and quantized with R = 8, 4, 2 and 1 bits/sample, the 8-bit log-quantizer is equivalent in dynamic range (ratio of maximum output magnitude to minimum output magnitude) to a 13-bit uniform quantizer. A PCM system with either of the above quantizers provides toll quality digitized speech, where 8-bit log-quantization will be used for communication of high-quality telephone speech at a transmission rate of I = 64 kb/s. The lower bit rate outputs will show the effect of *amplitude overload* or *clipping*; resulting reductions in speech intelligibility are surprisingly small, even in the worst case of *infinite clipping* (R = 1). However, output speech quality falls rapidly with decreasing R. In particular, the quality of R = 4 log-quantizer is not very useful. It has an MOS score of about 3, as compared with the score of 4.5 typical of 8-bit PCM.

In piecewise linear systems, the quantizer step size as well as the number of quantizer intervals are constant quantities within each linear segment. Step-sizes in consecutive segments are related by powers of 2. Digitally linearizable PCM with a bit rate of R = 8 bits/sample is a widely accepted standard for speech digitization. Lower bit rate systems are often designed to be compatible with the above mentioned standard even if they are not equivalent to it in performance. These systems include lower rate PCM systems such as *Nearly Instantaneous Companding* as well as lower rate systems based on differential PCM or related procedures.

Log-PCM at 64 kb/s is a time-honored basis for speech coding because it is a low-complexity technique with zero delay. It also provides *toll quality* in a strict communication network sense, a property not always possessed by the *high quality* typical speech coders. The motivation for these coders is that they offer better speech quality at bit rates considerably lower than 64 kb/s. [Jayant and Noll,1984]

2.8 Errors in Pulse Code Modulation

Assuming that random errors are introduced in the bits that convey information about quantizer output level, a binary symmetric channel, which implies that information about quantizer output level will be transmitted as a sequence of $R = \log_2 L$ bits; and the random bit error probability, or *bit error rate*, is denoted by p_e.

Channel error variance, σ_c^2 is expressed as a product of bit error rate p_e and a channel coefficient ξ that was a function of quantizer and binary code. The total reconstruction error variance, σ_r^2 is expressed as the sum of quantization error variance, σ_q^2 and channel error variance, σ_c^2 ;

$$\Sigma_r^2 = \sigma_q^2 + \sigma_c^2 = \sigma_q^2 + \xi\, p_e\, \sigma_x^2 \qquad (2.15)$$

In the special case of a uniform probability density function (pdf) input (a good model for images), and uniform quantization with NBC coding, the channel coefficient can be analytically expressed by :

$$\xi = 4(\, 1 - 2^{-2R}\,) \qquad (2.16)$$

Using the equation SNR (dB) = 6.02R – 10 $\log_{10} \left(f_1^{\,2}/3\right)$; R>>1, and the above equations, a closed-form SNR result for $R >> 1$ is obtained;

$$SNR(dB) \approx 6.02R - 10\log_{10}\left(1 + 2^{2(R+1)}\, p_e\right) \text{ for images;} \qquad R >> 1 \qquad (2.17)$$

From the last equation, if R = 8 and $p_e = 10^{-4}$, the SNR loss due to channel error is 14.3 dB. This shows that quantizing and coding systems can be made more robust in the sense of designing them for an optimum ξ.

For example, if the least significant 2 of the 8 bits in PCM system are stolen for algebraic error-protection, the maximum possible SNR is reduced by 12 dB, and the error-protected 6-bit system performs better than an unprotected 8-bit system only in the channel-noise-dominated situation described by $p_e > p_{eqc}$, an appropriately defined threshold probability. Estimates of peqc based on perceptual evaluations will in general be smaller in PCM coding because channel errors are much more noticeable than quantization errors of identical variance. Explicit error protection is extremely critical in the PCM transmission of *digital audio* with high fidelity. If the expected error rate p_e is in the order of 10^{-5} or more, and if the total bit rate is fixed, it is useful to steal 1, or even 2 least significant bits, in order to provide parity checks on the more significant PCM bits; this is done although stealing of 2 bits may result in perceptible increase of quantization noise.

2.9 Applications

Applications of digitized audio are in digital recording and in the distribution of broadcast music material. Digitizers for music require extremely high values of R for high-quality representation. This is due to the very large dynamic range of audio signals. However, in many applications of music digitization, the PCM coder input is pre-recorded music with dynamic range greater than that of the recording tape. This range is in the order of 75dB. As a result, 14-bit uniform resolution (output magnitude range of $20 \log_{10} 2^{13}$ = 78 dB for a zero-mean input) is then an adequate design goal. However, in intra-studio applications, 16-bit resolution may be called for, to allow a sufficient quality margin in the presence of subsequent signal-processing operations such as mixing. A 16 bit/sample format is also standard for *compact disc* recordings that offer a signal-to-noise ratio of more than 90 dB. The sampling rate used in this recording system is 44.1 kHz.

The minimum adequate value of number of bits *R* can be decreased as in the case of video, by preemphasis and dithering techniques, but the greatest reductions in *R* are those due to companding, and/or adaptive quantization. Transmission systems for digital audio are designed to operate at bit rates that are multiples of 64 kb/s. Examples of such systems are a 12 bit/samples, 32 kHz sampling rate system with I = 384 kb/s, and a 12 bit/sample, 16 kHz sampling rate system with I = 192 kb/s. The respective sampling rates are appropriate for bandwidths in the order of 15 and 7 kHz.

The bit-economical coding systems of later chapters have been developed mainly for speech and video; they can be adapted to music coding as well, resulting in digitizing systems with *R* values much less than 12 bits/sample.

CHAPTER 3

DIFFERENTIAL PULSE CODE MODULATION

3.1 Introduction

Differential PCM (DPCM) coders, which are based on the notion of quantizing a prediction error signal, are important examples of predictive coding systems. DPCM systems with one-bit quantizers constitute an important subclass, *delta modulation*.

The complexity of a DPCM system is directly related to that of the predictor algorithm. Predictors based on recent waveform history and time-invariant predictor coefficients lead to a class of coders which constitutes one example of the *low-to-medium-complexity* designs. These coders utilize *time-invariant* or *fixed* speech predictors and *intraframe* image predictors for high-quality digitization at bit rates in the order of R = 3 or 4 bits/sample, in each case. DPCM systems of medium-to-high-complexity are characterized by the use of *adaptive predictors* matched to the short-time input spectrum, and/or the use of *distant-sample-memory* for utilizing waveform periodicities. Examples of the latter are *pitch predictors* for speech and interframe predictors for video. These complex approaches are necessary for high-quality coding with R = 2 or 1 bits/sample. The quantizer refinements are also applicable to the *prediction error quantizers* of DPCM systems. For example, non-uniform and adaptive quantizers are taken for granted in DPCM systems for video and speech, respectively; and in both case, entropy-coded quantizers lead to significant gains, especially at lower bit rates.

By representing a correlated waveform in terms of appropriate difference samples, or prediction error samples, one can achieve an increased *SNR* at a given bit rate; or equivalently, a reduction of bit rate for a given requirement of *SNR*. This can be appreciated at least qualitatively for the simple but important case where the correlation ρ_1 between adjacent samples approaches unity. Assuming that the encoder represents the waveform as a succession of adjacent sample differences $\{x(n) - x(n-1)\}$; then these difference samples are quantized for transmission; and that a decoder recovers an approximation to the input essentially by integrating quantized adjacent sample differences. With $\rho_1 \to 1$, the variance of the quantizer input is much smaller than that of the coder input $x(n)$; and since quantization error variance is proportional to quantizer input variance for a given number of bits per sample, *R*, the above reduction of quantizer input variance leads to a reduction of reconstruction error variance σ_r^2 for a given value of *R*. In general, the quantizer input in a DPCM coder is a prediction error or difference signal

$$d(n) = x(n) - \hat{x}(n) \tag{3.1}$$

where $\hat{x}(n)$ is a prediction of $x(n)$, and the ratio

$$G_p = \sigma_x^2 / \sigma_d^2 \tag{3.2}$$

is a corresponding *prediction gain*. With some explanation, it will be noted that the prediction gain will be seen to represent the *SNR* improvement in going from PCM to DPCM. For the important sub-class of linear predictors, the prediction gain is upper-bounded by the inverse of spectral flatness measure γ_x^2. The performance of DPCM systems will be bounded by the following expression for minimum mean square value of reconstruction error $r(n) = x(n) - y(n)$;

$$\min\{\sigma_r^2\} = \varepsilon_q^2 \gamma_x^2 \sigma_x^2 \; ; \qquad \gamma_x^2 \leq 1 \tag{3.3}$$

Noting in contrast that the distortion in a PCM system is $\varepsilon_q^2 \sigma_x^2$. Thus, by exploiting input signal redundancies, DPCM coding will realize a decrease of error variance by a factor that could be as much as γ_x^2. Typical values of γ_x^2 for long-term speech and image spectra are in the order of 1/8 and 1/16 respectively. Much smaller values are realized in short-term voiced speech spectra (say, $\gamma_x^2 \leq 1/256$) and in the spectra of low-activity images.