

## Chapter 2

# Waveform Encoding

### 2.1 Introduction

‘Any natural signal is in analog form’. To respect the said statement and to meet the basic requirement of any type of digital signal processing and digital communication, the essential and prior step is converting the electrical form (through transducer) of the natural analog signal into digital form, as digital modulator or any type of digital signal processor does not accept analog signal as its input. Therefore, to consider the digital transmission of analog signal, it is very important to encode the waveform 1st. This process of *waveform encoding* is done through sampling, quantization and encoding; finally the analog information is converted to digital data.

The digitally coded analog signal produces a rugged signal with high immunity to distortion, interference and noise. This source coding also allows the uses of *regenerative repeater* for long distance communication. In the process of quantization, the approximation results in quantization noise and with a target of removing the noise, the bandwidth becomes comparable to the analog signal. Hence, a trade-off between the noise and bandwidth is to be established.

According to the specification in terms of accuracy-bandwidth trade off different schemes of source encoding are employed. The basic procedure of digital transmission of analog signal is coded pulse modulation or PCM. Quantization can be made non-uniformly [4] to catch up intimate details. To reduce the bandwidth in an efficient way, the process of differential pulse code modulation (DPCM) and delta-modulation (DM) [7] are used. In the prediction process of DM, reduction of hardware can be ensured by a nice engineering called as delta-sigma modulation (DSM). At the end of the chapter, the linear prediction is discussed with the help of Genetic Algorithm (GA).

### 2.2 Pulse Code Modulation (PCM)

Pulse code modulation (PCM) is a digital transmission system of analog signal having an analog to digital converter (ADC) at the input and a digital to analog converter (DAC) at the output.

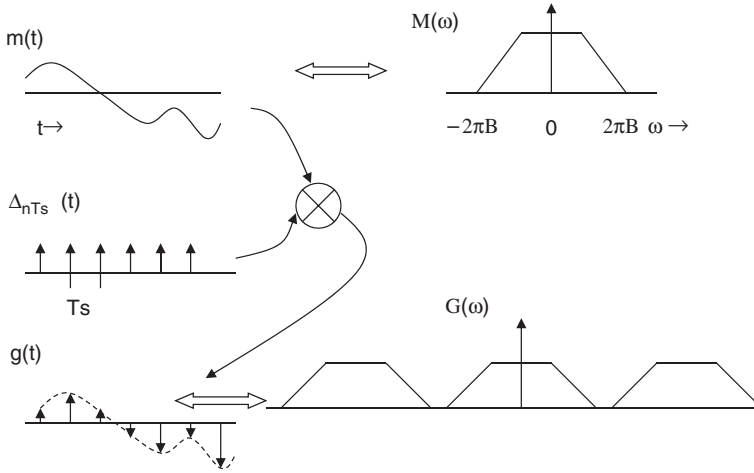
From the previous discussion it is granted that, the analog signals found in nature are essentially continuous time continuous valued signal. To make the signal compatible to feed into digital communication system, the signal needs to be converted to another form which is essentially represented using digits only. This analog-to-digital conversion follows principally three chronological steps namely, (a) Sampling, (b) Quantization and (c) Encoding. By the process of sampling, the time continuity is broken taking care of sufficiently large (almost 1) probability of regeneration. Quantization reshapes each sample height in voltage or in current mode. Here we get defined levels of voltage/current with respect to defined instants of time. That's why this signal obtained after quantization is called as discrete time discrete valued signal. Now, the remaining task is the representation of the defined set of finite number of probable time and voltage/current values. That can be done by mapping each of the elements of the set to a digital code. Automatically an encoder can map the entire signal into digit space.

When the digits are received un-erroneously (for the simplicity of understanding, it's assumed that no errors occur into the communication channel; if error comes, the problem can be taken care of by error detection and correction in channel coding to be discussed in the Chap. 9) at the destination end, formation of the word is essential. Each word is the tag of each quantization level. Next, a synchronized clock oscillating in sampling frequency can segregate the levels, define the time instants respective of the each level and then a discrete time discrete valued signal is obtained here again merely identical to the input signal of the encoder of the transmitter. The signal is obviously a multi-frequency signal where high frequency component arises due to the process of sampling. Now a low pass filter can reconstruct the original signal with some quantization error, which can be minimized by just increasing the number of quantization levels, i.e., by increasing the number of bits to represent each level.

Here a question of bandwidth noise trade-off arises. The objective of communication must play an important role here. There may a priority on accuracy, may be a priority of speed, or there may exist a compromise between these two. Depending upon the user or system requirement, a lot of variations of PCM are designed.

### ***2.2.1 Process of Sampling***

The word 'sample' means a small portion of physical element/phenomena which singularly or collectively has all the characteristics of the original physical element/phenomena. As discussed in Chap.1, this is the 1st step towards analog to digital signal conversion. Let's consider the signal  $m(t)$  of bandwidth  $B$  Hz. When the signal is multiplied by a train of impulses, the resultant signal is obviously the sampled version of  $m(t)$ . The impulses are equi-spaced by an amount of time  $T_s$ . After sampling, the signal  $g(t)$  becomes discrete in time domain (Fig. 2.1).



**Fig. 2.1** The process of sampling to get discrete time signal

To establish the relationship between the analog signal and discrete (sampled) signal, let's take the example of sampling of a sinusoid  $s(t)$ , sampled at a rate of  $F_s$ , i.e., sampling interval  $T_s$ .

The analog signal is given by,

$$S(t) = A \sin(\Omega t + \phi) \quad (2.1)$$

where,

$A$  is the amplitude,

$\Omega$  is the analog angular frequency  $= 2\pi F$ ,

$F$  is the analog frequency,

$\phi$  is the initial phase

After sampling, we are taking the signal at  $t = 0, T_s, 2T_s, \dots, NT_s$ .

Or,  $t = nT_s$ , ( $n=1, 2, 3, \dots, N$ ),  $N$ =number of samples.

Therefore, from Eq. (2.3),

$$\begin{aligned} S(nT_s) &= A \sin(2\pi F n T_s + \phi) \\ \Rightarrow S(n) &= A \sin\left(2\pi \times \frac{F}{F_s} + \phi\right) \text{ as } T_s = 1/F_s \\ \Rightarrow S(n) &= A \sin(2\pi \times f + \phi) \\ \Rightarrow S(n) &= A \sin(\omega n + \phi) \end{aligned} \quad (2.2)$$

taking

$$f = \frac{F}{F_S}$$

From Eq. (4.1) and (4.2), the correspondence of sampling is shown. The sequence formed as  $S(n)$  and the input signal  $S(t)$  to the sampler, are correspondent.

Looking closely regarding the units of the different variations of frequencies, generated as a bi-product of the process of sampling, we have the following derived units.

Unit of  $F$  = cycles/s

Unit of  $F_S$  = samples/s

Therefore,

Unit of digital frequency,  $f$  = cycles/sample

Unit of analog angular frequency,  $\Omega$ , i.e.,  $2\pi F$  = rad.cycles/s

Unit of digital angular frequency  $\omega$  = rad/samples.

Here, if one decrease the spacing between the samples ( $T_S$ ), i.e., increase the rate of sampling  $F_S$ , the reconstruction would be easier at the receiver side. But, the bandwidth of transmission would be increased thereafter, it will affect the processing time of the sampled signal. So, there is a trade-off between noise and transmission bandwidth. However, it's really important to know the lower limit of the choice of the sampling frequency for successful reconstruction.

### 2.2.1.1 Sampling Theorem

An analog signal can be reconstructed from its sampled values un-erroneously, if the sampling frequency is at least twice the bandwidth of the analog signal.

Say an analog signal  $m(t)$  has three different frequency components  $f_1$ ,  $f_2$ , and  $f_3$ , or combination of all of them, where  $f_1 < f_2 < f_3$  then the bandwidth of that  $m(t)$  signal must be  $f_3$ . For ease of calculation, we are taking bandwidth of that signal  $B$  Hz, in general.

Therefore we can say the spectrum  $M(\omega)$  is band limited by  $2\pi B$  in  $\omega$  scale, and band limited by  $B$  Hz in  $f$  scale. Now, sampled signal is nothing but the signal obtained by multiplying  $m(t)$  by unit impulse train  $\Delta_{nT_S}(t)$ . From the figure the sampled signal is  $g(t) = m(t) \times \Delta_{nT_S}(t)$ . Time period of the impulse train is  $T_S$ . therefore frequency is  $f_s = 1/T_S$ . Let's now expand the  $\Delta_{nT_S}(t)$  signal in Fourier series so that we can study the spectrum of  $g(t)$  signal. As  $\Delta_{nT_S}(t)$  is even function of time, By Fourier series,

$$\Delta_{nT_S}(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_s t + b_n \sin n\omega_s t) \quad (2.3)$$

Here,

$$\begin{aligned} a_0 &= \frac{1}{T_S} \int_{T_S} \Delta_{nT_S}(t) dt = \frac{1}{T_S} \\ a_n &= \frac{2}{T_S} \int_{T_S} \Delta_{nT_S}(t) \cos n\omega_0 t dt = \frac{2}{T_S} \\ b_n &= 0, \text{ for even function} \end{aligned}$$

Here,  $\int_{T_S}$  signifies integration within upper limit to lower limit difference  $T_S$ .  
Therefore,

$$\begin{aligned} \Delta_{nT_S}(t) &= \frac{1}{T_S} (1 + 2 \cos \omega_s t + 2 \cos 2\omega_s t + 2 \cos 3\omega_s t + \dots) \\ \Rightarrow g(t) &= m(t) \times \Delta_{nT_S}(t) \\ &= \frac{1}{T_S} (m(t) + 2m(t) \cos \omega_s t + 2m(t) \cos 2\omega_s t + 2m(t) \cos 3\omega_s t + \dots) \\ &= \frac{1}{T_S} \left( m(t) + 2m(t) \times \frac{e^{j\omega_s t} + e^{-j\omega_s t}}{2} + 2m(t) \times \frac{e^{2j\omega_s t} + e^{-2j\omega_s t}}{2} \right. \\ &\quad \left. + 2m(t) \times \frac{e^{3j\omega_s t} + e^{-3j\omega_s t}}{2} + \dots \right) \\ &= \frac{1}{T_S} \left( m(t) + m(t) (e^{j\omega_s t} + e^{-j\omega_s t}) + m(t) (e^{2j\omega_s t} + e^{-2j\omega_s t}) \right. \\ &\quad \left. + m(t) (e^{3j\omega_s t} + e^{-3j\omega_s t}) + \dots \right) \end{aligned} \tag{2.4}$$

From the 1st term of the equation above, it is very clear that spectrum of  $g(t)$  will be similar to  $M(\omega)$  except the amplitude. Amplitude will be  $1/T_S$  times of that of  $G(\omega)$ . 2<sup>nd</sup> term indicates  $M(\omega)$  shifted in both sides by the amount of  $\omega_s$  (using negative frequency concept). Similarly 3<sup>rd</sup> term indicates  $M(\omega)$  shifted in both sides by the amount of  $2\omega_s$  and so on.

Now if we can extract the central spectrum by low pass filter from the assembly of spectrum, then we can easily reconstruct the  $m(t)$  signal.

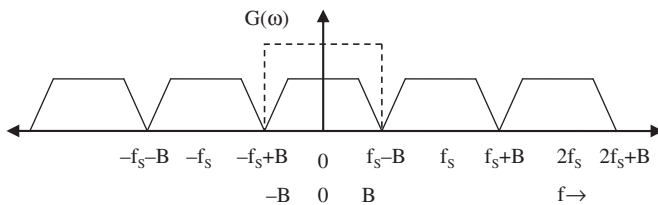
## Case I

From the Fig. 2.2 above specific fs i.e.  $\omega_s$  is taken into account. It is very clear that central spectrum can be extracted using LPF. Therefore it is *success* of reconstruction.

Here  $\omega_s - 2\pi B = 2\pi B$

Or,  $\omega_s = 4\pi B$

Or,  $f_s = 2B$



**Fig. 2.2** Sampled spectra for  $f_s = B$

## Case II

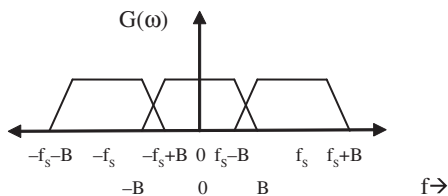
From the Fig. 2.3 above other  $f_s$  i.e.  $\omega_s$  is taken into account. It is very clear that central spectrum cannot be extracted using LPF; because, there is an overlap region of two spectra. The error due to this overlap region is called as *aliasing error*. Therefore it is *failure* of reconstruction.

$$\text{Here } \omega_s - 2\pi B < 2\pi B$$

$$\text{Or, } \omega_s < 4\pi B$$

$$\text{Or, } f_s < 2B$$

**Fig. 2.3** Sampled spectra for  $f_s < B$



## Case III

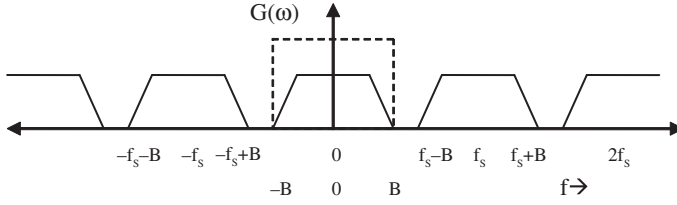
From the Fig. 2.4 above specific  $f_s$  i.e.  $\omega_s$  is taken into account. It is very clear that central spectrum can be extracted using LPF more easily than case I. Therefore it is *success* of reconstruction.

$$\text{Here } \omega_s - 2\pi B > 2\pi B$$

$$\text{Or, } \omega_s > 4\pi B$$

$$\text{Or, } f_s > 2B$$

Therefore collectively we can say  $f_s > 2B$  i.e., sampling theorem is proved. The minimum frequency of sampling for successful reconstruction is Nyquist sampling rate ( $f_s = 2B$ )



**Fig. 2.4** Sampled spectra for  $f_s > B$

Collectively, the condition for successful reconstruction of the message signal can be derived therefore as,

$$f_s \geq 2B \quad (2.5)$$

### 2.2.1.2 Aliasing

Say, two sequences are there  $s_1(n)$  and  $s_2(n)$ . If the digital angular frequency difference between these two sequences is integral multiple of  $360^\circ$ , the sequences would be identical.

Say,

$$\begin{aligned} s_1(n) &= A \sin(\omega_1 n + \phi) \\ s_2(n) &= A \sin(\omega_2 n + \phi) \end{aligned} \quad (2.6)$$

and

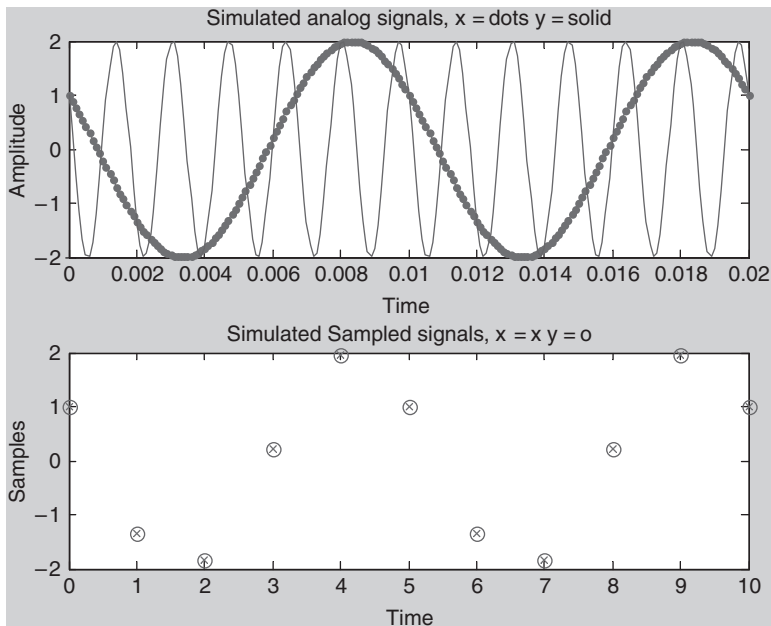
$$\omega_1 = \omega_2 + 2\pi$$

Then,

$$s_1(n) = A \sin(\omega_2 n + \phi + 2\pi n) = A \sin(\omega_2 n + \phi) \quad (2.7)$$

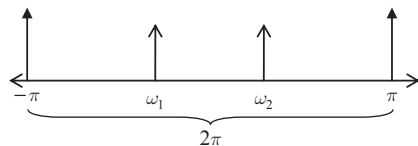
It is a severe error. The above equation signifies, after sampling the uniqueness of the signal destroys. Two different sinusoids map to a single sequence after sampling (Fig. 2.5). This error is called as aliasing error and these two sequences are called as alias of other.

To overcome the problem of aliasing, the choice of  $\omega$  should be such that, any  $\omega$  must reside within  $\pm\pi$ . Then only the difference between any two  $\omega$ 's be restricted within the range of  $2\pi$  as shown in the Fig. 2.6.



**Fig. 2.5** Illustration of the sampling theorem

**Fig. 2.6** Solution to the error of aliasing



Therefore, for 0 aliasing, the digital angular frequency should be as follows

$$\begin{aligned}
 |\omega| &\leq \pi \\
 \Rightarrow 2\pi \frac{F}{F_s} &\leq \pi \\
 \Rightarrow F &\leq \frac{F_s}{2} \\
 \Rightarrow F_s &\leq 2F
 \end{aligned} \tag{2.8}$$

Equation (2.8) is indeed the mathematical model for statement of the sampling theorem, and the process of sampled sinusoid signal correspondence is another way to prove the sampling theorem.

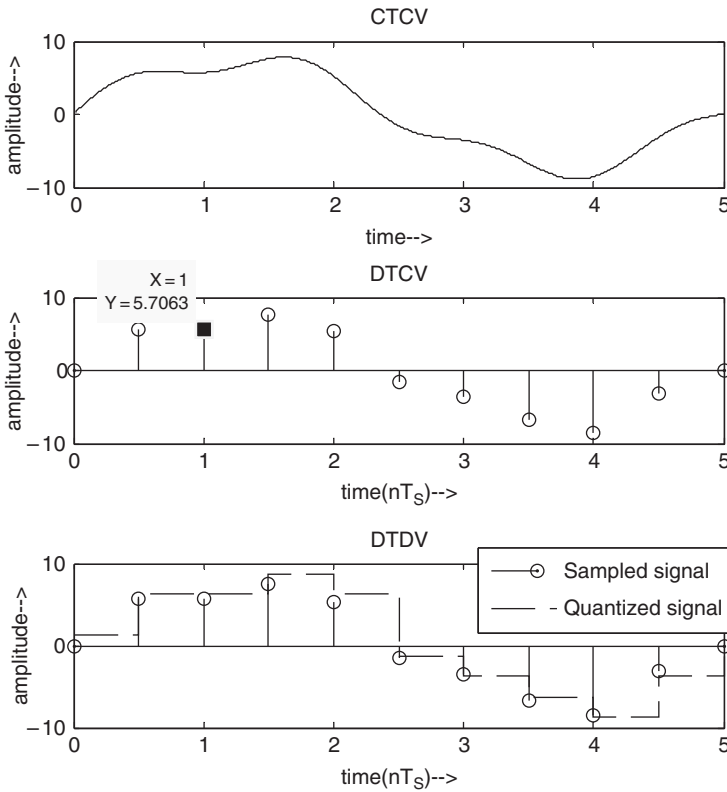
### 2.2.2 Process of Quantization

After sampling what we get is a sequence of impulses with value continuity. The impulses are separated by an amount of the sampling time  $T_s$ , but one cannot infer



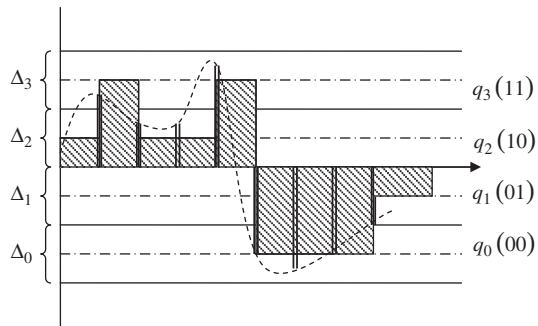
about the amplitude of the signal even after sampling in terms of discontinuity. It means, a set can be defined with finite number of time elements but a set with finite number of probable amplitude elements cannot be defined yet. The amplitude may be anything between the  $A_{\max}$  and  $A_{\min}$ . As for an example, if the signal is within voltage range  $+5\text{ V}$  and  $-5\text{ V}$ , at an instant of sampling the amplitude may be  $3.000000029\text{ V}$ . And we do not have the control over the accuracy permission, i.e., how many decimal digits we should allow, or truncate or round off and so on. This signal is therefore a discrete time continuous valued (DTCV) signal after sampling. Now, we need to make it properly a discrete time discrete valued (DTDV) signal. The process by which a set of finite number of amplitude values are also defined is called as *quantization*. Say, a set is defined with voltage values  $1, 2, 3, 4, 5$  and their negative counterpart volts. Then any original value obtained by sampling will be approximated to the closest defined value. For some cases, the sampled values may be quantized high; sometimes they may be quantized low with respect to the defined levels (Fig. 2.7).

To understand closely, let's consider the signal of Fig. 2.8. The maximum and minimum values are designated as  $A_{\max}$  and  $A_{\min}$ . Now, the entire range  $A_{\max}$



**Fig. 2.7** Time valued definition of signal

**Fig. 2.8** Illustration of uniform quantization and encoding



to  $A_{\min}$  is equally divided into 4 regions of width  $S$ . The 4 regions are named as  $\Delta_0$  to  $\Delta_3$ . The middle of each of the  $\Delta$ -regions is our defined probable amplitude level. The levels are called as q-levels or quantization levels. The algorithm of quantization states, 'If the sampled signal amplitude lies within  $\Delta_i$  region, the amplitude will be quantized (approximated) to the amplitude  $q_i$ '. Figure 2.8 clearly depicts the process.

The shaded signal in the above figure is the staircase representation of the quantized signal. Next, each q-level can easily be encoded to through 2 bit encoder or MUX to get proper digital representation of the input analog signal. After quantization, the signal  $x_q(kT_S)$  is represented as  $q_2, q_3, q_2, q_2, q_3, q_0, q_0, q_0, q_0, q_1$ . And after encoding, each level is inferred by a pair of digits in parallel. Next, by employing a parallel to serial converter, the serial bit stream is obtained which is the corresponding digital representation of the input analog signal.

There are two types of uniform quantizer namely,

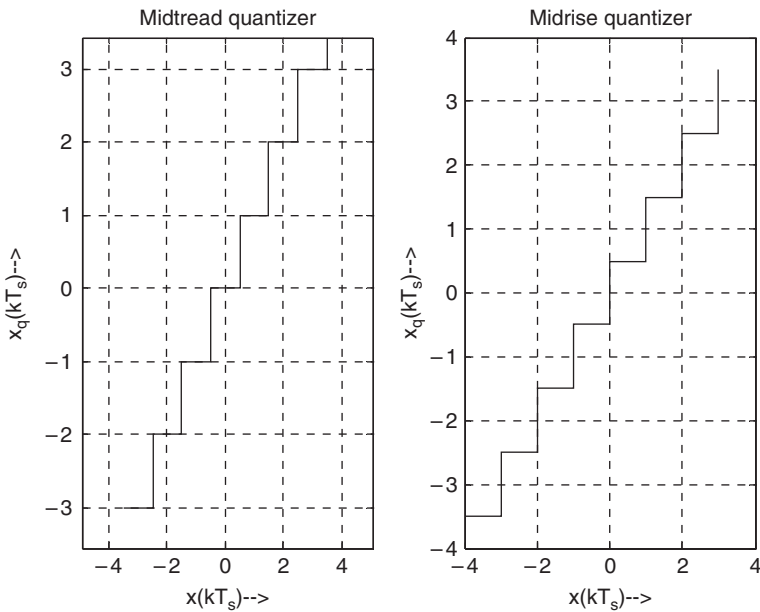
- (a) Midtread quantizer
- (b) Midrise quantizer

Figure 2.9 shows the input-output characteristics of the two types of the uniform quantizer. In midtread type quantizer, the origin lies in the middle of the tread of the staircase like graph. In midrise type, the origin the graph lies in the middle of a rising part of the staircase like graph. It can be noticed that, both midtread and midrise graphs are symmetric about the origin.

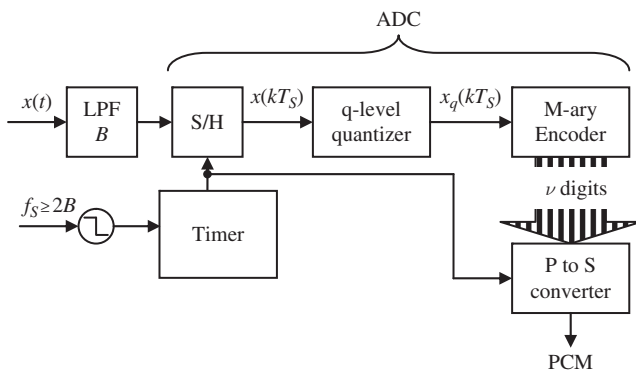
## 2.2.3 PCM Transmitter and Receiver

### 2.2.3.1 PCM Transmitter

The entire principle of analog to digital conversion is now unveiled to us. Therefore, we can just club up all the procedural blocks to construct the PCM transmitter as shown in Fig. 2.10. We have already discussed the functioning of the blocks. First the input analog signal  $x(t)$  is being passed through a low pass filter of bandwidth



**Fig. 2.9** Midtread and midrise quantizer characteristics



**Fig. 2.10** PCM transmitter

$B$  Hz to restrict the bandwidth of the signal for prevention of aliasing error. Next, the three basic steps of analog to digital conversion (ADC) is being followed i.e., sampling-quantization-encoding, as discussed before. But here the encoded bit stream is essentially parallel in form, although the objective is to get serial bit stream in response to serial analog signal input [1].

The encoder works with  $M$ -ary digits (if  $M=2$ , it becomes binary) and produces a codeword consisting of  $\nu$  digits in parallel, corresponding to each sample. Since there are  $M^\nu$  (in binary,  $2^\nu$ ) possible  $M$ -ary codewords, with  $\nu$  digits per word,

unique coding of the  $q$  different quantization levels requires that,  $M^\nu \geq q$ . The parameters should be chosen removing the inequality, such that,

$$q = M^\nu \text{ or, } \nu = \log_M q \quad (2.9)$$

For binary PCM, number of bits per q-level can be calculated as  $\nu = \log_2 q$ .

Finally, the successive code words are read out serially. The parallel to serial conversion also needs the clocking by sampling frequency. Otherwise, either one word would superimpose or overwrite the other, or one code word would be converted twice. In both the cases, we get improper conversion. Therefore, clocking in the parallel to serial converter is essential.

As, each encoded sample is represented using  $\nu$  digits, so the signaling rate at the output becomes  $r = \nu f_s$ . Therefore, for transmission of PCM signal, the bandwidth needed is

$$B.W._{tr} \geq \frac{1}{2}r = \frac{1}{2}\nu f_s \geq \nu B \quad (2.10)$$

### 2.2.3.2 PCM Receiver

The transmitted PCM signal is now passed through transmission channel towards the receiver. Contamination of noise is almost obvious during the traversal of the signal through transmission path. Though the received signal is noise accumulated, the process of regeneration yields a nearly errorless signal as the SNR becomes sufficiently large. In the receiver section ultimately we need to get the reconstructed analog signal. Therefore, one digital to analog conversion (DAC) is to be designed. As shown in the Fig. 2.11, the DAC operation is just the opposite, i.e., serial to parallel conversion (for getting proper form of input to the next stage, i.e. decoder),  $M$ -ary decoding and sample and hold. The output of the sample and hold circuit is the staircase type waveform,  $x_q(t)$ .

The waveform shown in Fig. 2.12 may be regarded as a staircase approximation  $x(t)$ . A low pass filter is then employed for smoothening. The low pass filtering produces the reconstructed analog signal  $\tilde{x}(t)$ , which differs from  $x(t)$  to the extent that the quantized samples  $(x_q(kT_s))$  differ from the exact sampled values  $(x(kT_s))$ .

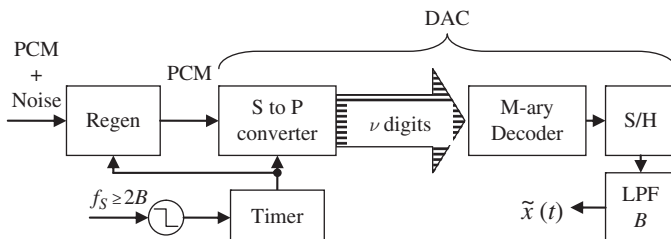
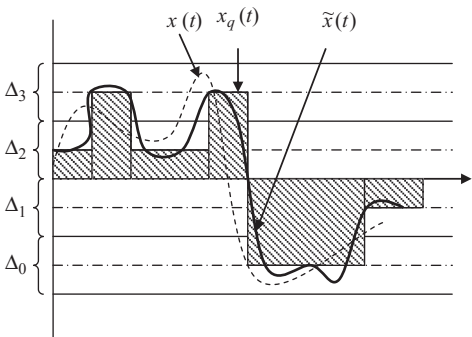


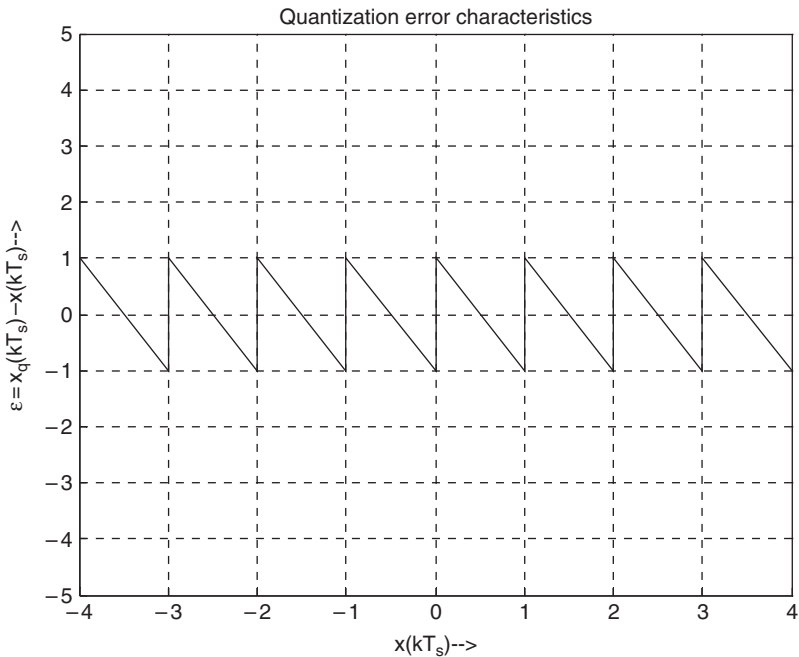
Fig. 2.11 PCM receiver

**Fig. 2.12** Waveform reconstruction through PCM receiver as shown in Fig. 2.11



**2.2.4 Quantization Error**

It is now understood that, the process of quantization is nothing but approximation to nearer level of voltage/current. Due to this approximation, there a random amount of difference occurs between actual and quantized value, named as quantization error. As the error is like a random number and adds noise to modulation process, the error can also be viewed as a noise called as *quantization noise* Fig. 2.13.

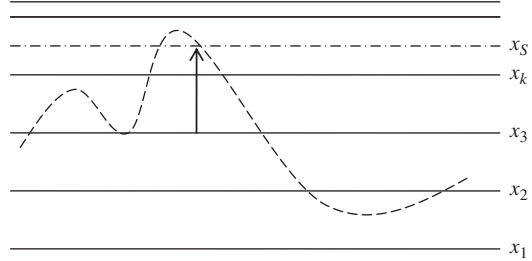


**Fig. 2.13** Quantization error characteristics

The noise is now to be calculated as mean-square quantization error,  $e^2$  where  $e$  is the difference between the actual and approximated value of voltage/current.

The peak to peak amplitude of the sampled signal  $x_s(t)$  is divided into  $M$  equal levels each of width  $S$ . At the centre of each level of width the quantization levels are located as  $x_1, x_2, \dots, x_M$  as shown the Fig. 2.14. Since, in the figure  $x_s(t)$  happens to be the closest to the level  $x_k$ , the quantizer output will be  $x_k$  and obviously, the quantization error is  $e = x_s(t) - x_k$ .

**Fig. 2.14** Interpretation of quantization error



Let,  $p(x) dx$  be the probability that  $x_s(t)$  lies in the voltage/current range  $x + dx/2$  to  $x - dx/2$ . Then the mean square quantization error [3] is

$$\overline{e^2} = \int_{x_1-S/2}^{x_1+S/2} p(x) (x - x_1)^2 dx + \int_{x_2-S/2}^{x_2+S/2} p(x) (x - x_2)^2 dx + \dots \quad (2.11)$$

Now, the PDF (probability density function)  $p(x)$  of the message signal will not certainly be constant throughout each division. Assuming large  $M$ , i.e., with sufficiently small  $S$ ,  $p(x)$  can be taken as constant throughout each division. Then, the 1st term of the right hand side of the Eq. (2.11),  $p(x) = p^{(1)} = \text{constant}$ . The 2nd term is  $p(x) = p^{(2)}$ , and so on. Hence, the constant terms may be taken out of the integration sign. If we now substitute  $y \equiv x - x_k$ , the expression in Eq. (2.11) becomes

$$\begin{aligned} \overline{e^2} &= \left( p^{(1)} + p^{(2)} + \dots \right) \int_{-S/2}^{S/2} y^2 dy \\ &= \left( p^{(1)} + p^{(2)} + \dots \right) \frac{S^3}{12} \\ &= \left( p^{(1)}S + p^{(2)}S + \dots \right) \frac{S^2}{12} \end{aligned} \quad (2.12)$$

Now, according to definition,  $p^{(1)}S$  be the probability that the signal lies within the 1st quantization range,  $p^{(2)}S$  be the probability that the signal lies within the 2nd quantization range, and so on. Hence, the sum of the terms in the bracket in the

Eq. (2.12) must have unity value, as signal must lie within the peak-to-peak value, i.e., summation of all quantization range. Therefore, the mean square quantization error is

$$\overline{e^2} = \frac{S^2}{12} \quad (2.13)$$

### 2.2.5 Signal to Noise Ratio (SNR) for Quantized Pulses

A signal with maximum and minimum amplitudes as  $+A$  and  $-A$  is assumed. The signal is quantized in  $M$  levels with each of width  $S$ . Then the peak to peak range of amplitude (voltage or current) is  $A_{pp} = A - (-A) = 2A$ . As understood from the graph shown in Fig. 2.13, the maximum value of quantization error is  $S/2$  or  $-S/2$  in either direction. Therefore, the degradation of the signal is limited to the amount of  $S/2$  in additive or subtractive manner. It is obvious that, the peak to peak amplitude can also be expressed in terms of number and width or quantization levels as  $A_{pp} = M \times S$ .

The peak power of the analog signal (normalized to 1  $\Omega$ ) can be expressed as

$$A_p^2 = \left( \frac{A_{pp}}{2} \right)^2 = \left( \frac{MS}{2} \right)^2 = \frac{M^2 S^2}{4} \quad (2.14)$$

Now, we do have already the expression for quantization noise. Therefore, from Eq. 2.13 and Eq. 2.14 we can derive the signal to noise ratio for uniformly quantized pulses as [6]

$$\frac{S_t}{N_q} = \frac{\frac{M^2 S^2}{4}}{\frac{S^2}{12}} = 3M^2 \quad (2.15)$$

The above equation supports our common understanding that, SNR will be improved as the number of levels to be squared is increased. As a limit if  $M \rightarrow \infty$ , the PCM signals will be converted to PAM signal, i.e., no quantization will be done.

*Example 2.1* Consider an audio signal as given

$$s(t) = 3 \cos 500\pi t \text{ V}$$

- (i) Find the SNR when  $s(t)$  is quantized using 10 bit PCM.
- (ii) How many bits of quantization is needed for achieving SNR of 40 dB.

$$(i) \text{ SNR} = \frac{S_t}{N} = \frac{12P_S}{S^2},$$

where  $P_S$  is the RMS signal power defined as,

$$\begin{aligned}
 P_S &= \frac{3^2}{2} = 4.5 \text{ W} \\
 S &= \frac{V_H - V_L}{M} = \frac{3 - (-3)}{2 \times 10^{10}} = 5.86 \text{ mV} \\
 \therefore SNR &= \frac{S_t}{N} = \frac{12 \times 4.5}{(5.46 \times 10^{-3})^2} = 1.56 \times 10^6 \\
 \Rightarrow SNR_{\text{dB}} &= \left. \frac{S_t}{N} \right|_{\text{dB}} = 10 \log (1.57 \times 10^6) \approx 62 \text{ dB}
 \end{aligned}$$

(ii) We know,  $40 \text{ dB} = 4 \times 10 \log 10 = 10 \log 10^4$

Therefore, to find out the number of bits of quantization,

$$\begin{aligned}
 \frac{12 \times 4.5}{(6/2^v)^2} &= 10^4 \\
 \Rightarrow \frac{2^{2v}}{(6)^2} &= \frac{10^4}{12 \times 4.5} \\
 \Rightarrow 2^{2v} &= \frac{36 \times 10^4}{12 \times 4.5}
 \end{aligned}$$

Taking  $\log_2$  of both sides, we get

$$\begin{aligned}
 \Rightarrow 2v &= \log_2 \left( \frac{36 \times 10^4}{12 \times 4.5} \right) = 12.7027 \\
 \therefore v &= \left\lceil \frac{12.7}{2} \right\rceil = 7 \text{ bits}
 \end{aligned}$$

### 2.2.6 Non-uniform Quantization: Companding

For most of the voice communication channels, very low speech volumes predominate for more than half of the times, the voltage characterizing detected speech energy is 25% of the RMS value. Large amplitude values are relatively rare; only 10–15% of the time the voltage exceeds the RMS value. This is a peculiar statistics of speech signals. Therefore, the spacing at the high amplitude level becomes a wastage and for incorporation of that unused q-levels, we do have to go for high bandwidth transmission of low accuracy transmission, if we employ uniform quantization.

In case of the design of uniform quantization, the quantization error varies directly proportional to the square of the uniform width of quantization levels, i.e., the quantization error is inversely proportional to the square of number of quantization levels. Now, to reduce the error, if we plan to increase the number

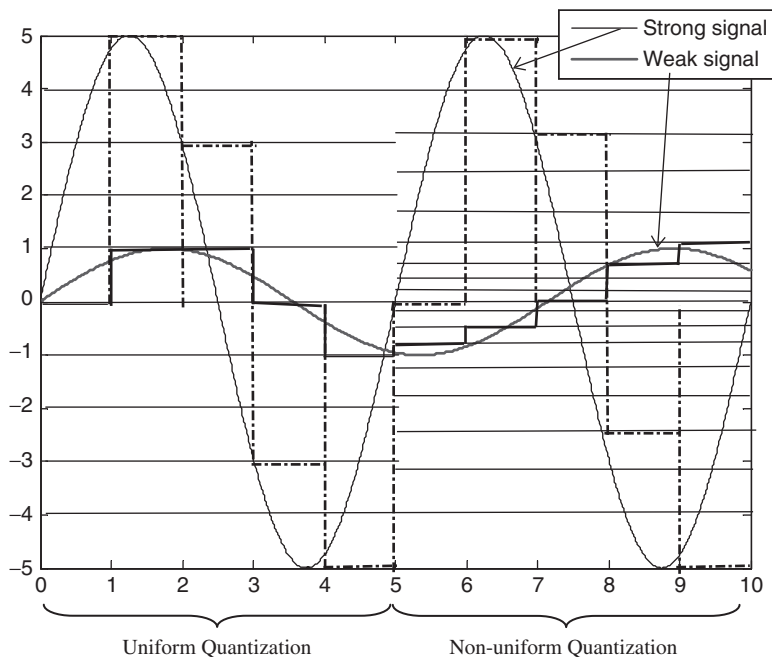


of  $q$ -levels, it requires more number of bits to represent each level. Hence, the transmission bandwidth will be increased significantly. To bypass the trade-off between these two (transmission bandwidth and quantization error), an overall reduction of quantization error is achieved by judiciously choosing the widths (non-uniformly) of the levels keeping the total number of quantization levels unchanged [4, 5].

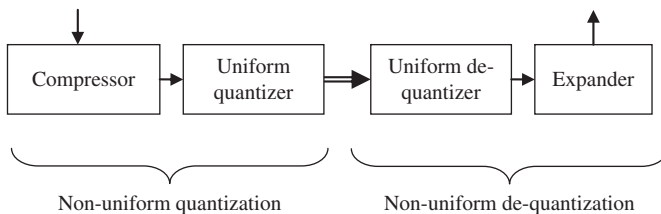
The ratio of voltage levels covered by voice signals from peaks of loud talk to the weak passages of weak talk is of the order of 1000:1. Such variation is taken care of by *non-uniform quantization* of the signal. The non-uniform quantization helps the practical variability in such a way that, the step size automatically increases as the separation from the origin of the input-output amplitude characteristics is increased. In weak passages, the step size becomes smaller and the details are not missed at all. On the other hand, by employing uniform quantization, either number of steps should be made higher or the weaker signal details have to be compromised.

The procedure for getting non-uniformly quantized samples is compression followed by uniform quantization. At the receiver side also, the reverse operation, i.e., expanding followed by uniform quantization is done, as shown in Fig. 2.15. The inclusion of COMPRESSION and EXPANDING renames the process of non-uniform quantization as a contraction COMPANDING (Fig. 2.16).

It can be easily understood that, the compression characteristic and parameter of the characteristic is really an important factor to know for reliable and successful communication. A typical compressor characteristic is shown in



**Fig. 2.15** Uniform quantization vs. non-uniform quantization



**Fig. 2.16** The process of COMPANDING

**Fig. 2.17** Compressor characteristics

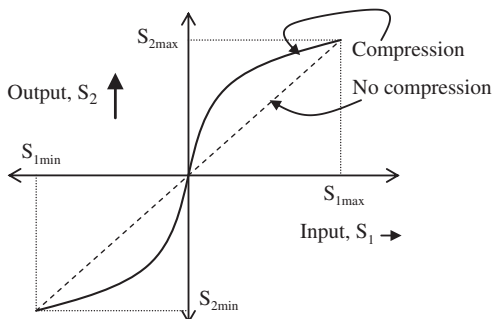


Fig. 2.17. A lot of compressor laws with different parameter orientation are there. Among them A-law and  $\mu$  law are accepted by CCITT<sup>1</sup> and are discussed here.

$\mu$  law are mainly used in North America and Japan. A-law is used in Europe and rest of the country.

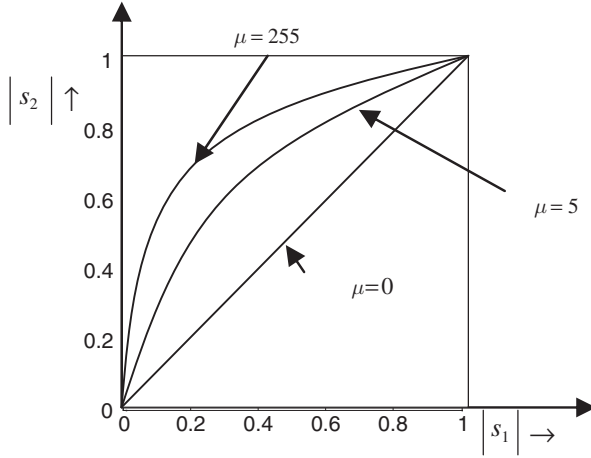
### 2.2.6.1 $\mu$ -Law

The compression algorithm following  $\mu$ -law is given by

$$|s_2| = \frac{\log(1 + \mu|s_1|)}{\log(1 + \mu)} \quad (2.16)$$

where,  $s_1$  and  $s_2$  are normalized input and output voltage/current respectively.  $\mu$  is a non-negative parameter which determine the degree of compression. For  $\mu = 0$ , we obtain uniform quantization. The plot of the normalized output with normalized

<sup>1</sup>CCITT is the contraction of Consultative Committee for International Telegraphy and Telephony.



**Fig. 2.18** Compression by  $\mu$ -law

input with different values of  $\mu$  is shown in Fig. 2.18. The quantization step is defined as the reciprocal of the slope of the compression curve given by

$$\frac{d|s_1|}{d|s_2|} = \frac{\log(1 + \mu)}{\mu} (1 + \mu |s_1|) \quad (2.17)$$

For lower values of input,  $\mu |s_1| \ll 1$ , therefore,

$$|s_2| \approx \frac{\mu |s_1|}{\log(1 + \mu)} \quad (2.18)$$

and

$$\frac{d|s_1|}{d|s_2|} \approx \frac{\log(1 + \mu)}{\mu} \quad (2.19)$$

For higher values of input,  $\mu |s_1| \gg 1$ , therefore,

$$|s_2| \approx \frac{\log(\mu |s_1|)}{\log(1 + \mu)} \quad (2.20)$$

and

$$\frac{d|s_1|}{d|s_2|} \approx |s_1| \log(1 + \mu) \quad (2.21)$$

Therefore, the  $\mu$ -law is linear at low input levels and approximately logarithmic at high input levels.

### 2.2.6.2 A-Law

Another well known compression algorithm following A-law is given by

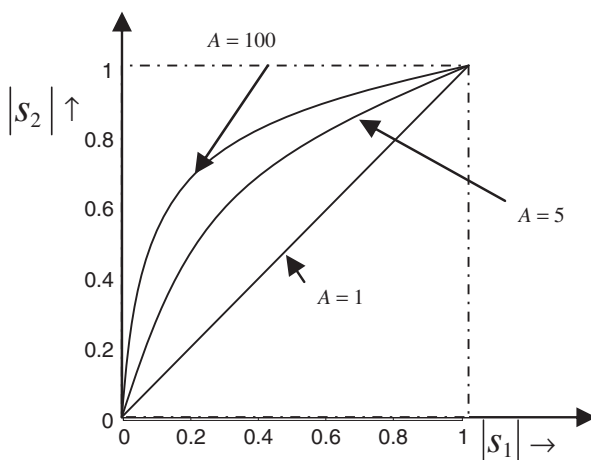
$$|s_2| = \frac{A |s_1|}{1 + \log A}; \quad 0 \leq |s_1| \leq \frac{1}{A} \quad (2.22)$$

$$|s_2| = \frac{1 + \log(A |s_1|)}{1 + \log A}; \quad \frac{1}{A} \leq |s_1| \leq 1 \quad (2.23)$$

Here also,  $A$  is a non-negative parameter of compression.  $A=1$  signifies uniform-quantization. The plot of the normalized output with normalized input with different values of  $A$  is shown in Fig. 2.19. The quantization step is defined as the reciprocal of the slope of the compression curve given by

$$\frac{d|s_1|}{d|s_2|} = \begin{cases} \frac{1 + \log A}{A}; & 0 \leq |s_1| \leq \frac{1}{A} \\ (1 + \log A) |s_1|; & \frac{1}{A} \leq |s_1| \leq 1 \end{cases} \quad (2.24)$$

Thus, the quantization steps over the central linear segment are diminished by factor  $A/(1+\log A)$ . These steps over the linear segment have a dominant effect on small signals and are diminished practically by about 25 dB as compared to uniform quantization.



**Fig. 2.19** Compression by A-law

*Example 2.2* When a 256 level quantization is employed for a sinusoid with peak voltage 40 Volts, what is the voltage interval without compression?

For  $\mu = 255$ , what is the maximum and minimum separation between the levels?

- (i) For uniform quantization, the step size,  $S$  can be determined easily as

$$S = \frac{2 \times V}{256} = \frac{2 \times 40}{256} = 312.5 \text{ mV}$$

- (ii) To apply  $\mu$ -law for non-uniform quantization, for  $\mu = 255$   
The smallest height  $h$  would be increased in GP as

$$\begin{aligned} h + 2h + 4h + 8h + 16h + 32h + 64h + 128h &= 2V \\ \Rightarrow 255h &= 2V \\ \Rightarrow h &= 0.0078 \text{ V} = 7.8 \text{ mV} \\ \Rightarrow 128h &= 1.0039 \text{ V} \end{aligned}$$

Therefore the maximum and minimum step size employing  $\mu$ -law for non-uniform quantization, for  $\mu = 255$  are 1.0039 V and 7.8 mV respectively.

## 2.3 Differential Pulse Code Modulation (DPCM)

In analog messages we can make a good guess about a sampled value from the knowledge of the past sampled values. In other words, the sampled values are not independent, and generally there is a great deal of redundancy in the Nyquist samples. Proper exploitation of this redundancy leads to encoding a signal with lesser number of bits. Consider a sampling scheme where instead of transmitting the sampled values, we transmit the difference between the successive samples. By employing the technique of transmitting the quantized difference values of the successive samples we can efficiently use the bandwidth provided by the transmitting channel.

### 2.3.1 Cumulative Error in PCM

In general PCM system, a quantization error  $e_{q_i}$  is added to the quantized output while quantizing the  $i$ th sample. Now just take a look on how the quantization error affects the DPCM output.

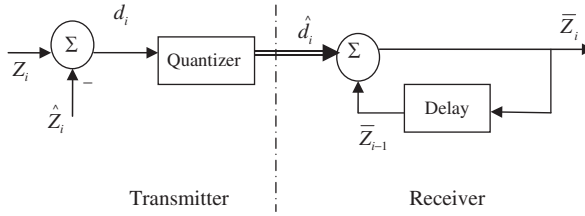
Here  $Z_i$  is the sampled value at  $i$ th instant and  $\hat{Z}_i$  is the predicted sample (considered as delayed sample) at  $i$ th instant.

Now, from the Fig. 2.20,

$$\begin{aligned} d_i &= Z_i - \hat{Z}_i \\ &= Z_i - Z_{i-1} \end{aligned} \tag{2.25}$$

Or recovered sample at the receiver is  $\bar{Z}_i = \hat{d}_i + \bar{Z}_{i-1}$

For the 1st sample,  $i=1$ . So,  $\bar{Z}_1 = \hat{d}_1 + \bar{Z}_0$



**Fig. 2.20** Basic DPCM communication system

$$= \hat{d}_1 + Z_0 \text{ (As at 0th instant } \bar{Z}_0 = Z_0)$$

And the quantized difference

$$\hat{d}_i = d_i + e_{qi} \quad (2.26)$$

So, from the diagram presented in Fig. 2.20,

$$\begin{aligned} \bar{Z}_1 &= d_1 + e_{q1} + Z_0 \\ \text{Or } \bar{Z}_1 &= Z_1 + e_{q1} \end{aligned} \quad (2.27)$$

Now, for  $i=2$ , we get from Eq. (2.25),

$$\begin{aligned} \bar{Z}_2 &= \hat{d}_2 + \bar{Z}_1 \\ \bar{Z}_2 &= d_2 + e_{q2} + Z_1 + e_{q1} \\ \bar{Z}_2 &= Z_2 + e_{q1} + e_{q2} \end{aligned} \quad (2.28)$$

Proceeding in this way we can see

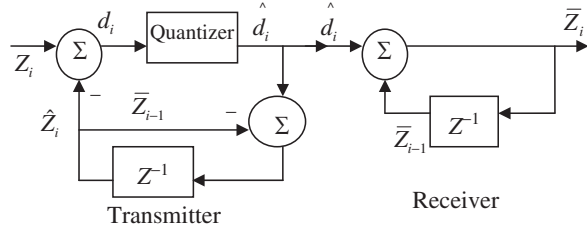
$$\bar{Z}_n = Z_n + \sum_{i=1}^n e_{qi} \quad (2.29)$$

So, here we see that in case of basic configuration of DPCM, the output at  $n$ th instant is affected not only by the quantization error of  $n$ th instant, but also by the errors of previous instants. The error becomes severe as the term of samples increases. To overcome the problem, a new configuration is designed.

### 2.3.2 Prevention of Cumulative Error by Applying Feedback

In the receiver section, a delay feedback path is used for proper recovery of the message sample. To solve the problem of cumulative quantization error in the transmitted samples, we have just copied the feedback loop of the receiver section and pasted to the transmitter section as shown in Fig. 2.21.

**Fig. 2.21** DPCM communication system with error suppressor



feedback circuit has been added to the transmitter side. The feedback circuit is basically same as the receiver circuit. Now let us analyze the transmitter. Here the block  $Z^{-1}$  signifies 1 unit delay (delay by an amount of  $T_S$ ).

At  $i$ th instant

$$d_i = Z_i - \bar{Z}_{i-1} \text{ and } \hat{d}_i + \bar{Z}_{i-1} = \bar{Z}_i \quad (2.30)$$

Also,

$$\hat{d}_i = d_i + e_{qi} \quad (2.31)$$

For the 1st sample, i.e. for  $i=1$ ,

$$\begin{aligned} \hat{d}_1 &= d_1 + e_{q1} \\ \text{So, } \bar{Z}_1 &= \hat{d}_1 + \bar{Z}_0 \\ &= d_1 + \bar{Z}_0 + e_{q1} \\ &= Z_1 + e_{q1} \end{aligned} \quad (2.32)$$

For  $i=2$ , from Eq. (2.31) we get

$$\hat{d}_2 = d_2 + e_{q2} \quad (2.33)$$

Now,

$$\hat{d}_2 + \bar{Z}_1 = \bar{Z}_2$$

So,

$$\begin{aligned} \bar{Z}_2 &= d_2 + e_{q2} + \bar{Z}_1 \\ &= Z_2 + e_{q2} \end{aligned} \quad (2.34)$$

Proceeding in this way, we can say reconstructed output at any instant

$$\bar{Z}_n = Z_n + e_{q_n} \quad (2.35)$$

So, comparing Eqs. (2.29) and (2.35), we can say that in the present case, the reconstructed output depends only on the quantization error of that particular instant. So it's free from cumulative error.

Thus if  $m[k]$  is the  $k$ th sample instead of transmitting  $m[k]$  we transmit the difference  $d[k] = m[k] - m[k - 1]$ . At the receiver knowing  $d[k]$  and the previous sample  $m[k - 1]$ , we reconstruct  $m[k]$  iteratively at the receiver. Now, the differences between successive samples are generally much smaller than the sample values. Thus the peak amplitude  $m_p$  of the transmitted value is reduced considerably because the quantization interval  $A_v = m_p/l$ , for a given  $l$  (no of bits per sample) we can reduce the quantization noise given by  $A_v^2/12$ . This also means that for a given transmission bandwidth (which is related to number of bits per sample as  $BW = \frac{l \times \text{samples/s}}{2}$ ) and for a given SNR we can reduce BW.

We can improve upon this scheme by estimating (predicting) the value of  $k$ th sample  $\hat{m}[k]$  from a knowledge of the previous sample. If this estimate is  $\hat{m}[k]$ , then we transmit the difference (prediction error)  $d[k] = m[k] - \hat{m}[k]$ . At the receiver also, we determine the estimated  $\hat{m}[k]$  from the previous sample, and then generate  $m[k]$  by adding the received  $\hat{d}[k]$  to the estimated  $\hat{m}[k]$ .

Now if our prediction is quite worth, the difference between  $\hat{m}[k]$  and  $m[k]$  will be much smaller than the difference between two sample values  $m[k]$  and  $m[k - 1]$ . Thus the number of bits required will be even lower than the previous case and hence the bandwidth will also be lower.

In the basic assumption, the prediction for a particular sample is taken granted as the delayed sample. In the next section, we'll try to discuss the reason and logic behind this assumption.

### 2.3.3 How We Can Predict the Future?

The approach to signal prediction (estimation) is discussed here using the spirit of Taylor [2].

Let us consider, a signal  $m(t)$ , which have derivatives of all order at  $t$ . Using Taylor series we can express  $m(t + T_S)$  as

$$m(t + T_S) = m(t) + \frac{T_S}{1!} m'(t) + \frac{T_S^2}{2!} m''(t) + \dots \quad (2.36a)$$

$$\approx m(t) + T_S m'(t) \text{ For small } T_S \quad (2.36b)$$

Equation (2.36a) shows that from knowledge of the signal and its derivative at instant  $t$ , we can predict a future signal at  $t = T_S$ . In fact, even if we know just the first derivative, we can still predict this value approximately, as shown in



Eq. (2.36b). Let us denote the  $k$ th sample of  $m(t)$  as  $m[k]$ , that is  $m(kT_s) = m[k]$ , and  $m(kT_s \pm T_s) = m[k \pm 1]$ .

Now, the derivative

$$m'(kT_s) \approx \frac{m[kT_s] - m[kT_s - T_s]}{T_s}.$$

So from Eq. (2.36b), we obtain

$$\begin{aligned} m[k+1] &\approx m[k] + T_s \left[ \frac{m[k] - m[k-1]}{T_s} \right] \\ &= 2m[k] - m[k-1]. \end{aligned} \quad (2.37)$$

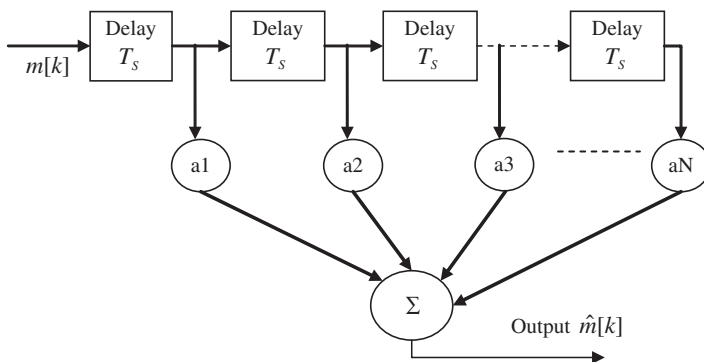
It shows that, we can find a crude prediction of the  $k+1$ th sample from the two previous samples. The approximation improves if we take more terms in Eq. (2.36a). To determine the higher order derivatives in the services, we require more samples from the past. Larger the number of past samples, we can get a better approximation.

$$m[k] \approx a_1 m[k-1] + a_2 m[k-2] + \dots + a_N m[k-N] \quad (2.38)$$

The right hand side is  $\hat{m}[k]$  the predicted value of  $m[k]$ . Thus  $\hat{m}[k] = a_1 m[k-1] + a_2 m[k-2] + \dots + a_N m[k-N]$ .

This is the equation of an  $N$ th order prediction. Thus we can design one prediction filter whose output is  $\hat{m}[k]$  and input is  $m[k-1], m[k-2], \dots, m[k-N]$ . But we can get these by simply delaying  $m[k]$  by  $T_s$ .

The parameters  $a_1, a_2, \dots, a_N$  are called prediction coefficients. This is also called linear prediction. It is actually a transversal filter (a tapped delay line), where the tap gains are set equal to the prediction coefficients as shown in Fig. 2.22.



**Fig. 2.22** Basic idea of linear prediction

### 2.3.4 Analysis of DPCM

DPCM transmit  $d[k]$ , the difference between  $m[k]$  and  $\hat{m}[k]$  and at receiver we generate  $\hat{m}[k]$  from the past sample values to which the received  $d[k]$  is added to generate  $m[k]$ .

There is, however, one difficulty in this scheme. At the receiver, instead of past sample  $m[k-1]$ ,  $m[k-2]$ ,  $\dots$   $m[k-N]$  as well as  $d[k]$ , we have there quantized versions,  $m_q[k-2]$ ,  $\dots$   $m_q[k-N]$ . Hence, we cannot determine  $\hat{m}[k]$ . We can only determine  $\hat{m}_q[k]$  the predicted value of quantized sample from previous quantized samples. This will increase the error in reconstruction (because the error is here additive, error in each quantized signal is added together).

In such case, a better strategy is to determine  $\hat{m}_q[k]$ , the estimate of  $m_q[k]$  instead of  $m[k]$  at the transmitter also from the quantized sample. The difference  $d[k] = m[k] - \hat{m}_q[k]$  is now transmitted using PCM (Fig. 2.23). At the receiver, we can generate  $\hat{m}_q[k]$ , and from the received  $d[k]$ , we can reconstruct  $m_q[k]$  (Fig. 2.24).

Here  $d_q[k] = d[k] + q[k]$ , where  $q[k]$  is the quantization error. The predictor output  $\hat{m}_q[k]$  is fed back to its input so that the predictor input  $m_q[k]$  is  $m_q[k] = \hat{m}_q[k] + d_q[k]$

$$\begin{aligned} &= m[k] - d[k] + d_q[k] \\ &= m[k] + q[k] \end{aligned} \quad (2.39)$$

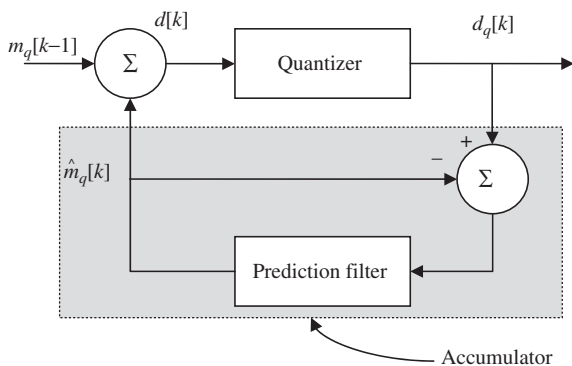


Fig. 2.23 DPCM transmitter

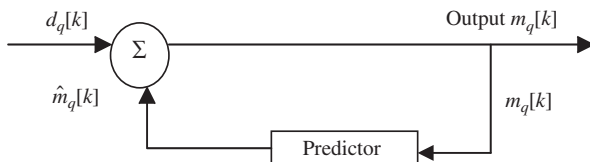


Fig. 2.24 DPCM receiver

This shows that  $m_q[k]$  is a quantized form of  $m[k]$ . The predictor input is indeed  $m_q[k]$  as assumed.

The receiver shown in Fig. 2.24 is identical to the shaded portion of the transmitter. The inputs in both cases are also the same  $d_q[k]$ . The predictor output must be  $m_q[k]$ .

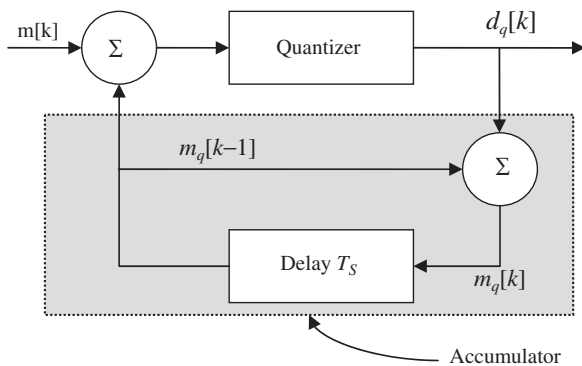
## 2.4 Delta Modulation

Sample correlation used in DPCM is further explained in Delta Modulation (DM) by over sampling (typically 4 times the Nyquist rate) the baseband signal. This increases the correlation between adjacent samples, which results in a small prediction error that can be encoded using only one bit (no. of levels = 2). Thus DM is basically a one bit DPCM, that is, a DPCM that uses only two levels ( $L=2$ ) for quantization of the  $m[k] - \hat{m}_q[k]$ . For more simple scheme the difference between  $m[k]$  and  $m[k - 1]$  is encoded by only one bit. That is if  $m[k]$  is higher than  $m[k]$  then 1 is transmitted, otherwise 0.

In comparison to PCM (and also DPCM), it is very simple method of A/D conversion. In DM the word framing is unnecessary in transmitter and receiver. This strategy allows us to use fewer bits per sample for encoding a base band signal.

In delta modulation, we use a first order predictor, which, as seen earlier, is just a time delay of  $T_S$ . Thus the DPCM transmitter and receiver can be reconfigured to delta modulation as shown in the following Fig. 2.25.

**Fig. 2.25** Delta modulation as a special case of DPCM



Here

$$m_q[k] = m_q[k - 1] + d_q[k]$$

Hence,

$$m_q[k - 1] = m_q[k - 2] + d_q[k - 1]$$

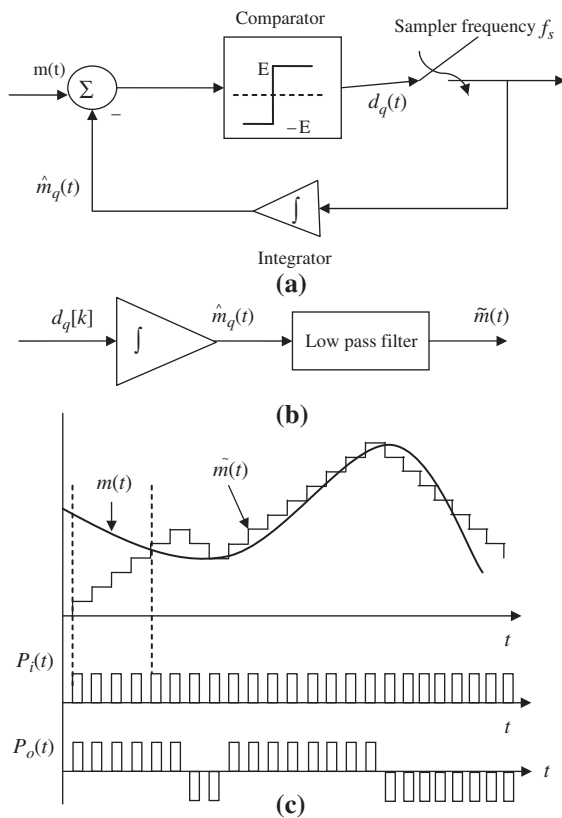
So,

$$m_q[k] = m_q[k-2] + d_q[k-1] + d_q[k] \quad (2.40)$$

Proceeding iteratively in this manner and considering 0 initial condition, i.e.

$$m_q[0] = 0, \text{ we have } m_q[k] = \sum_{n=0}^k d_q[n] \quad (2.41)$$

This shows that the receiver is just an accumulator. If the output  $d_q[k]$  is represented by impulses, then the accumulator (receiver) may be realized by an integrator because its output is the sum of the strengths of input impulses (sum of the areas under the impulses). We may replace the feedback portion of the modulator (which is identical to the demodulator) by an integrator. The demodulator output is  $m_q[k]$ , which when passed through an LPF gives us the retrieved signal. The Fig. 2.26 bellows shows a practical implementation of the delta modulator and demodulator.



**Fig. 2.26** (a) Delta modulator, (b) Delta demodulator, (c) DM waveforms

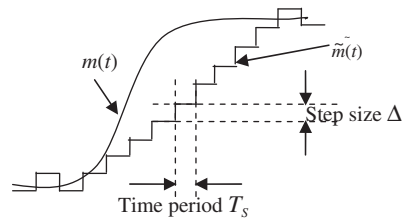
### 2.4.1 Drawbacks of Delta Modulation

The waveform  $\tilde{m}(t)$  needs to closely follow the waveform of  $m(t)$  in order to the recovered waveform  $\tilde{m}(t)$  resembles  $m(t)$ . Taking a careful look at the waveform shows situations where  $\tilde{m}(t)$  is unable to follow  $m(t)$  as the slope of  $m(t)$  is higher than that of  $\tilde{m}(t)$ . Also when  $m(t)$  is varying slowly, the change of  $\tilde{m}(t)$  is higher enough to cause an error.

#### 2.4.1.1 Slope Overloading

When, at any instant or any duration of time the slope of  $m(t)$  is higher then the slope of  $\tilde{m}(t)$ , i.e. step size/time period,  $\tilde{m}(t)$  is unable to follow the input message signal  $m(t)$ . So the recovered waveform will be distorted. Figure 2.27 shows the case.

**Fig. 2.27** Slope overloading error in delta modulation (DM)



The condition for slope over loading is then,

$$\left. \frac{dm(t)}{dt} \right|_{\max} \geq \frac{\Delta}{T_s} \quad (2.42)$$

*Example 2.3* For the signal  $s(t) = A \sin 2\pi ft$  calculate the sampling frequency to avoid slope overload error, considering 256 quantization levels and 1 kHz signal frequency.

$$\begin{aligned} \left. \frac{ds(t)}{dt} \right|_{\max} &= A \cdot 2\pi f \\ \therefore f_s = \frac{1}{T_s} &\leq \frac{2A\pi f}{\Delta} \\ \Rightarrow f_s &\leq M\pi f \quad \left[ \because M = \frac{2A}{\Delta} \right], \end{aligned}$$

M = Number of levels,  
2A = peak to peak amplitude.

Now, putting M=256, f=1,

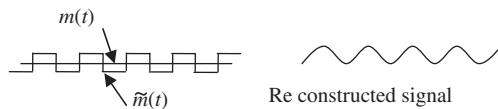
$$f_s \geq 256 \times 3.14 \times 1 \text{ kHz} = 803.84 \text{ kHz}$$

Therefore, to avoid the error of slope overload, the sampling frequency must be greater than 803.84 kHz.

### 2.4.1.2 Granular Noise

In other hand, when the input message signal  $m(t)$  is varying slowly, the change in  $\tilde{m}(t)$  is comparatively higher that causes error in reconstruction. This type of error or distortion is called Granular noise. Figure 2.28 shows the case.

**Fig. 2.28** Granular noise error in delta modulation



## 2.5 Adaptive Delta Modulation

The drawbacks of delta modulation are primarily due to the slope. These drawbacks can be overcome by suitably changing the slope. Slope can be changed either by changing the time period or the step size. Changing the time period will cause in changing the frequency and bandwidth. So it is not possible to vary the time period for a single message. Rather we can vary the step size instead to overcome the limitation. One may question that, by increasing the step size we are increasing the quantization error, because the quantization error is directly proportional to the square of the step size. But here the basic assumption of squared error calculation is missing and does not hold for the present case.

A Delta Modulation system with variable step size is known as the Adaptive Delta Modulation (ADM).

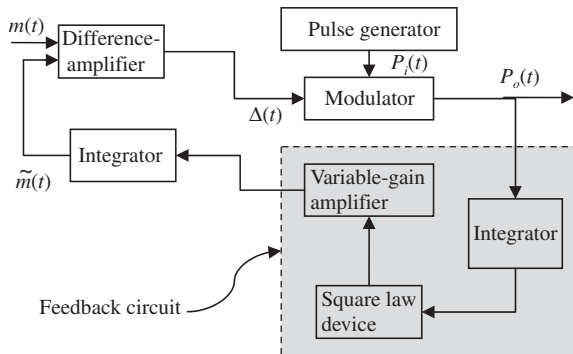
The block diagram of an ADM system is shown in the following Fig. 2.29.

In the concept discussed above, the squaring circuit is responsible for the size of step ( $\Delta$ ) adaptation (tuning). The shaded region shown in the figure above is the 'adaptation algorithm' in generic. Two other well known adaptation algorithms are discussed below.

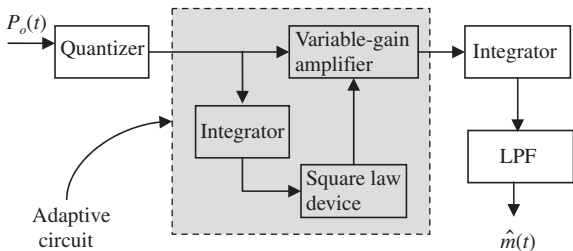
### 2.5.1 Song Algorithm

In the year of 1971, C. L. Song et al. [7] proposed a nice algorithm for step size adaptation by which the step size of the predicted waveform ( $\tilde{m}(t)$ ) can nicely be adapted. According to the algorithm, positive slope of the prediction with respect to time results in the next prediction equal to the previous step size added with the

**Fig. 2.29** Adaptive delta modulation (ADM) communication system.



**(a) ADM transmitter**



**(b) ADM receiver**

step size at 0th prediction ( $\Delta_0$ ), similarly, for negative slope of the prediction with respect to time results in the next prediction equal to the previous step size minus the step size at 0th prediction ( $\Delta_0$ ).

If the  $n$ th prediction sample is  $\Delta(n)$ , the algorithm prescribes the adaptation as shown in the following two equations.

$$\Delta(n) = \Delta(n-1) + \Delta_0 \quad \text{if} \quad m(t) > \tilde{m}(t) \quad (2.43)$$

$$\Delta(n) = \Delta(n-1) - \Delta_0 \quad \text{if} \quad m(t) < \tilde{m}(t) \quad (2.44)$$

This addition and subtraction is called as *accumulation* in other terms. To combine the Eqs. (2.43) and (2.44) let us assume another parameter called as discrepancy parameter ( $d_n$ ) as

$$d_n = \begin{cases} +1 & \text{if} \quad m(t) > \tilde{m}(t) \\ -1 & \text{if} \quad m(t) < \tilde{m}(t) \end{cases} \quad (2.45)$$

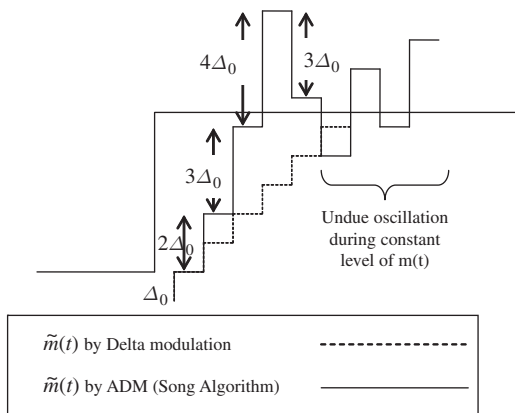
Now, combining Eqs. (2.43) and (2.44), the  $n$ th sample can be calculated using song algorithm of adaptation as

$$\Delta(n) = |\Delta(n-1)d_n + \Delta_0 d_{n-1}| \quad (2.46)$$

The song algorithm can now be used for digitization of any type of analog signal shape. Let us take a typical step size analog signal for prediction through accumulation. The ADM waveform through song algorithm is shown below.

The level of decrement is controlled by the new algorithm. In the previous section (adaptation by squaring), the increment and decrement amount is only the step size of 0th instant ( $\Delta_0$ ). But, as shown in the Fig. 2.30, the ADM predicted waveform is showing undue oscillation for typically step like analog signal. For low slope and high slope signals of other types can be tracked more efficiently by this method of adaptation. The slope overload problem is solved totally by the song algorithm. The undue oscillation can be viewed as a special type of granular noise.

**Fig. 2.30** Adaptive delta modulation by song algorithm



### 2.5.2 Space-Shuttle Algorithm

Space-shuttle algorithm is a modification over song algorithm only. We have seen that, the negative slope or maintenance of constant levels of the analog signal cannot be properly tracked by the prediction through adaptive delta modulation by song algorithm. The step size of the accumulated waveform is incremented and decremented by same amount ( $\Delta_0$ ). But, allowance of this linear decrement gives rise to an undue oscillation during constant level of  $m(t)$ . The new algorithm only changes the rule of decrement. Instead of linear decrement space shuttle algorithm suggests constant decrement of amount  $\Delta_0$ .

$$\Delta(n) = \Delta(n-1) + \Delta_0 \quad \text{if} \quad m(t) > \tilde{m}(t) \quad (2.47)$$

$$\Delta(n) = \Delta_0 \quad \text{if} \quad m(t) < \tilde{m}(t) \quad (2.48)$$



Otherwise, the algorithm is very much similar to the song algorithm. Here also, the step size is incremented if the direction of the step at the current clock edge is the same as the previous clock edge. If the direction is opposite, the step size is decremented.

2.6 Sigma-Delta Modulation (SDM)

Delta modulation requires two integrators for modulation and demodulation processes as shown in Fig. 2.31 below. Since integration is a linear operation, the second integrator can be moved before the modulator without altering the overall input/output characteristics. Furthermore, the two integrators in Fig. 2.31a can be combined into a single integrator by the linear operation property (Fig. 2.31b).

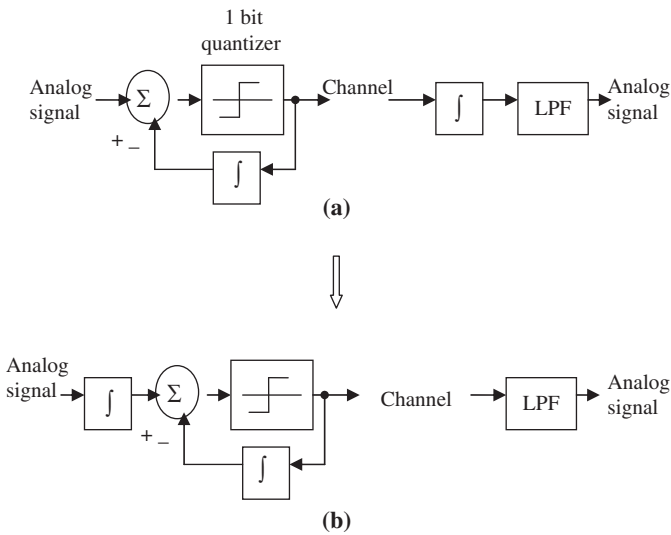


Fig. 2.31 (a) Delta modulator and demodulator, (b) Modification of (a) by integrator matching

The arrangement shown in Fig. 2.32 is the design diagram of a Sigma-Delta (S-D) Modulation system. This structure, besides being simpler, can be considered as being a ‘smoothed version’ of a 1-bit pulse code modulation (delta modulator). Here, the *smoothness* signifies low pass filtering (i.e., integrated) of the comparator

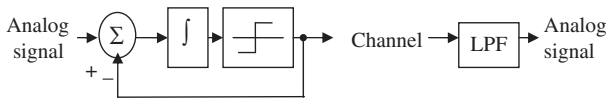


Fig. 2.32 Sigma-Delta modulator with a single integrator

output as shown in Fig. 2.32 and 1 bit PCM signifies a hard limiter of 2 output levels of saturation at the output (Fig. 2.26).

The name Sigma-Delta modulator comes from putting the integrator (sigma) in front of the delta modulator. The use of the integrator in the manner described here has also the following benefits:

1. The low frequency content of the input analog signal is pre-emphasized.
2. Correlation between adjacent samples of the delta modulator input is increased, which improves the overall system performance by reducing the variance of the error signal at the quantizer input.
3. Design of the receiver is simplified.
4. Noise performance is improved as discussed in the following section.

### 2.6.1 Noise Performance

Sometimes, the S-D modulator is referred to as an interpolative coder. The quantization noise characteristic (noise performance) of such a coder is frequency dependent in contrast to delta modulation. As will be discussed further, this noise-shaping property is well suited to signal processing applications such as digital audio and communication. Like delta modulators, the S-D modulators use a simple coarse quantizer (comparator). However, unlike delta modulators, these systems encode the integral of the signal itself and thus their performance is insensitive to the rate of change of the signal (Fig. 2.33).

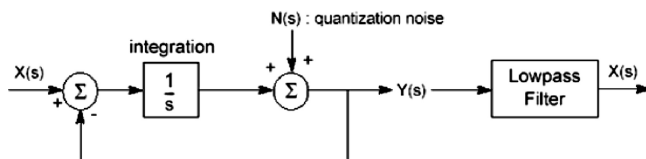


Fig. 2.33 Delta-Sigma modulation in frequency domain representation

- Signal transfer function (when  $N(S)=0$ ):

$$Y(s) = [X(s) - Y(s)] \times \frac{1}{s} \quad (2.49)$$

$$\frac{Y(s)}{X(s)} = \frac{\frac{1}{s}}{1 + \frac{1}{s}} = \frac{1}{1 + s} \quad (2.50)$$

Clearly, it signifies transfer function of LPF.

- Noise transfer function (when  $X(S)=0$ ):

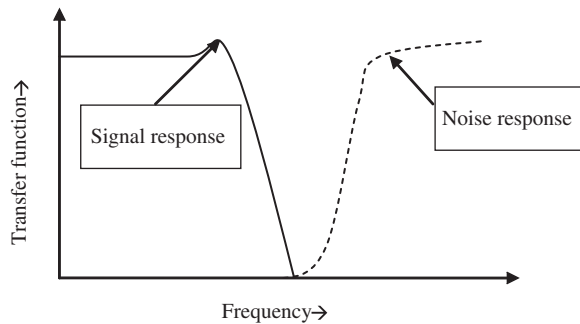
$$Y(s) = -Y(s) \times \frac{1}{s} + N(s) \quad (2.51)$$

$$\frac{Y(s)}{N(s)} = \frac{1}{1 + \frac{1}{s}} = \frac{s}{s + 1} \quad (2.52)$$

Clearly, it signifies transfer function of HPF.

Transfer function shown in Fig. 2.34 illustrates the modulator's main action. As the loop integrates the error between the sampled signal and the input signal, it low pass filters the signal and high pass filters the noise. In other words, the signal is left unchanged as long as its frequency content doesn't exceed the filter's cutoff frequency, but the S-D loop pushes the noise into a higher frequency band. Grossly over-sampling the input causes the quantization noise to spread over a wide bandwidth and the noise density in the bandwidth of interest (baseband) to significantly decrease.

**Fig. 2.34** Noise performance of signal delta modulation



## 2.7 Linear Predictive Coder (LPC)

### 2.7.1 Concept

From the Sect. 2.3.3 (DPCM prediction), it is understood that, for prediction of the next sample, only the delayed sample may not reduce the error of prediction to our threshold level of satisfaction. For that, we have incorporated a series of delayed samples with suitable scaling factors. The word 'suitable' is important in respect of the design (Fig. 2.35)

The accuracy of the prediction is dependent on the proper choice of the scaling factors. Here the predicted sample is expressible as linear combination of the series of samples as under:

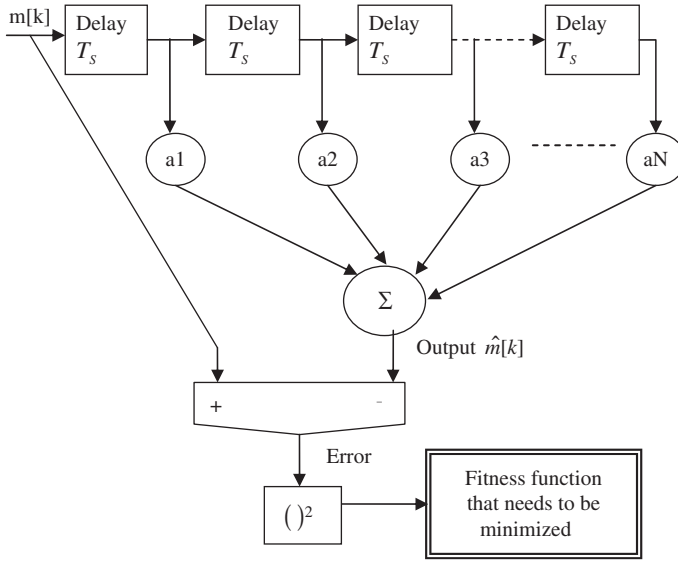


Fig. 2.35 Linear predictive coding with optimization of fitness of prediction

$$\begin{aligned}
 \hat{m}[k] &= a_1 m[k-1] + a_2 m[k-2] + \dots + a_N m[k-N] \\
 &= \sum_{i=1}^N a_i m[k-i]
 \end{aligned} \tag{2.53}$$

Next, the error is calculated with respect to real  $m[k]$  and the squared value of error (or we can use one squaring circuit to make it sign free) is minimized by any optimization algorithm to find out ‘suitable’ scaling factors  $a_i$ ’s for proper prediction. This is LSE (least square error) estimation algorithm. Once the scaling factors are fixed for training samples, now our LPC (linear predictive coder) is ready to use for the other query samples also. The optimization process needs to solve a huge number of differential equations to find out proper  $a_i$ ’s for good predictor design. To overcome the hazard of solving the differential equations, we take help of soft computing approaches like Genetic Algorithm (GA), to be discussed in the next section.

### 2.7.2 Genetic Algorithm Based Approach

Genetic algorithm is a biologically inspired algorithm based on Darwinian evolution, and is a nice algorithm for object optimization. Here, we can parallelly discuss about basic GA and application of GA to find out suitable  $a_i$ ’s for minimum squared error.

Genetic algorithms are implemented in a computer simulation in which a population of abstract representations (called chromosomes or the genotype of the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. Here, we will use real-coded GA. The evolution usually starts from a population of randomly generated individuals and allowed for reproduction. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness to the particular problem), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

A typical genetic algorithm requires:

1. A genetic representation of the solution domain,
2. A fitness function to evaluate the solution domain.

Here, the gene is represented as an array of solutions in real code. The fitness function is defined over the genetic representation and measures the *quality* of the represented solution. The fitness function is always problem dependent. In the present case, the fitness function  $f$  is defined as the squared error as given below:

$$f = \varepsilon^2 = (m[k] - \hat{m}[k])^2 = \left( m[k] - \sum_{i=1}^N a_i m[k-i] \right)^2 \quad (2.54)$$

After having the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, and then improve it through repetitive application of mutation, crossover, inversion and selection operators.

#### Step1: *Population*

A genetic pool is created with 20 chromosomes each with N number of genes with random values, covering the entire range of possible solutions. Here the scaling factors must reside within 0 and 1. Therefore, the initialized random values must be fractions only. Say, the values of genes are the trial solutions of  $a_i$ 's ( $\forall i$ ) for minimum squared error.

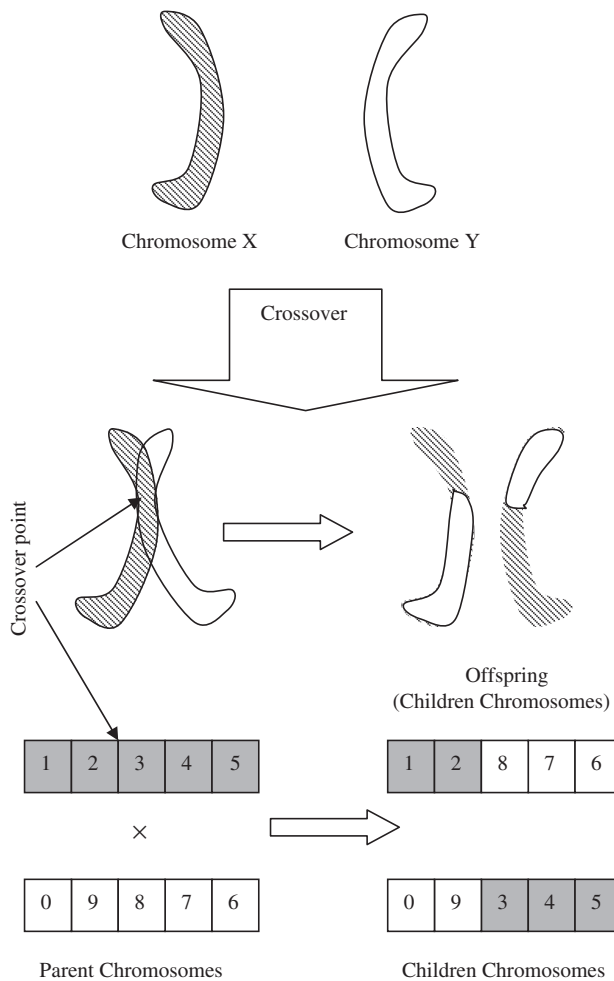
#### Step2: *Selection*

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a *fitness-based* process. It means, from that 20 strings of arrays (20 chromosomes), the most fit genes will be selected. The values of  $a_i$ 's ( $\forall i$ ) (i.e., the genes) will

be used to find the fitness value according to the Eq. (2) and from those 20 chromosomes (20 trial solutions) 10 better/ fitter will be selected.

### Step3: *Reproduction*

The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation. The Fig. 2.36 clearly shows the process of crossover. The crossover point should be chosen such that number of genes at the left of the cross point of the X chromosome must be exactly equal to the number of genes at the left of the cross point of the Y chromosome.



**Fig. 2.36** Process of crossover for a 5 variable (N=5) solution

For each new solution to be produced, a pair of ‘parent’ solutions is selected for breeding from the pool selected previously. By producing a ‘child’ solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its ‘parents’. New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated.

Here we have assumed that, only 5 delays are responsible for the prediction, i.e.,  $N=5$ . So, two fit chromosomes are chosen as shown in Fig. 2.36. In X chromosome, trial solution for  $a_1 = \frac{1}{1+2+3+4+5} = \frac{1}{15}$ ,  $a_2 = \frac{2}{1+2+3+4+5} = \frac{2}{15}$ ,  $a_3 = \frac{3}{1+2+3+4+5} = \frac{3}{15}$  and so on. Similarly, in Y chromosome, trial solution for  $a_1 = \frac{0}{0+9+8+7+6} = 0$ ,  $a_2 = \frac{9}{0+9+8+7+6} = \frac{9}{30}$  and so on. After crossover, two new solutions for the scaling factors are created. In the 1st child chromosome,  $a_1 = \frac{1}{1+2+8+7+6} = \frac{1}{24}$ ,  $a_2 = \frac{2}{1+2+8+7+6} = \frac{2}{24}$ ,  $a_3 = \frac{8}{1+2+8+7+6} = \frac{8}{24}$  and so on. Similarly, in 2nd child chromosome, solution for  $a_1 = \frac{0}{0+9+3+4+5} = \frac{0}{21} = 0$ ,  $a_2 = \frac{9}{0+9+3+4+5} = \frac{9}{21}$  and so on.

The above example shows, the processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding.

#### Step4: Termination

This reproduction and offspring (or child) generation is continuously repeated until a termination condition has been reached. Common terminating conditions are:

- A solution is found that satisfies minimum criteria (threshold of fitness).
- Fixed number of generations reached.
- Allocated budget (computation time/money) reached
- The highest ranking solution’s fitness is reaching or has reached
- a plateau such that successive iterations no longer produce better results.
- Manual inspection.
- Combinations of the above.

## 2.8 MATLAB Programs

### 2.8.1 Aliasing

```
% In CD: ch2_1.m
% Aliasing
% Output: Fig. 2.5
% Programed by Apurba Das (Jan, '10)
```

```

clear all;close all;
freq=100;phase=pi/3;
mag=2;fs=500;Ts=1/fs;
k=1; % number of repetitions
num_points=200; % How many points to use, 200 makes it smooth
num_samples=11; % How many samples to simulate reading
           % 11 puts "sampled" under "analog"
step=2/(freq*num_points); % get a nice step size
t=0:step:2*(1/freq);
n=0:num_samples-1;
% x and y are simulated analog functions
x= mag*cos(2*pi*freq*t+phase);
y= mag*cos(2*pi*(freq+k*fs)*t+phase);
% x2 and y2 are simulated sampled version of x and y
x2(n+1)=mag*cos(2*pi*freq*n*Ts+phase);
y2(n+1)=mag*cos(2*pi*(freq+k*fs)*n*Ts+phase);
% Plot the analog signal
subplot(2,1,1);
plot(t,x,'r.-');hold on;plot(t,y,'b-');
title('Simulated analog signals, x=dots y=solid');
xlabel('Time');ylabel('Amplitude');
% Plot "sampled" signals
subplot(2,1,2);
plot(n,x2,'rx');hold on;plot(n,y2,'bo');
title('Simulated Sampled signals, x=x y=o');
xlabel('Time');ylabel('Samples');
figure;
nn=0:0.002:0.02;
plot(t,x,'r.-');hold on;plot(t,y,'b-');stem(nn,x2,'kv');

```

## References

1. Carlson, A.B., Crilly, P.B., Rutledge, J.C., "Communication Systems – An Introduction to Signals and Noise in Electrical Communication (4th Ed.)", McGraw-Hill International, 2002
2. Lathi, B.P., "Modern Digital and Analog Communication Systems (3rd Ed.)", Oxford University Press, 2005
3. Taub, H., Schilling, D.L., "Principles of Communication Systems", McGraw-Hill, 1986 (Original from the University of Michigan)
4. Smith, B., "Instantaneous Companding of Quantized Signals," *Bell System Technical Journal*, Vol. 36, May 1957, pp. 653–709
5. Kaneko, H., "A Unified Formulation of Segment Companding Laws and Synthesis of Codecs and Digital Compandors," *Bell System Technical Journal*, Vol. 49, September 1970, pp. 1555–1558
6. Skler, B., "Digital Communications: Fundamentals and Applications (2nd Ed.)", Pearson Education, 2001
7. Song, C.L., Schilling, D.L., Garodnick, J. "A Variable Step-Size Robust Delta Modulator", *IEEE Transactions on Communications Technology*, Vol. COM-19, December 1971, pp. 1033–1044



<http://www.springer.com/978-3-642-12742-7>

Digital Communication

Principles and System Modelling

Das, A.

2010, XVIII, 246 p. 182 illus. With online files/update.,

Hardcover

ISBN: 978-3-642-12742-7