UNIT 2

WAVEFORM CODING

Prediction filtering and DPCM - Delta Modulation - ADPCM & ADM principles-Linear Predictive Coding

2.1 LINEAR PREDICTION

Prediction use a finite set of present and past samples of a stationary process to predict a sample of the process in the *future.* The prediction is *linear* if it is a linear combination of the given samples of the process. The filter designed to perform the prediction is called a *predictor.* The difference between the actual sample of the process at the (future) time of interest and the predictor output is called the *prediction error.* According to the Wiener filter theory, a predictor is designed to minimize the mean-square value of the prediction error.

Consider the random samples $X_{n-1}, X_{n-2}, \dots, X_{n-M}$ drawn from a stationary process $X(t)$. Suppose the requirement is to make a prediction of the sample X_n. Let $\hat{X}$ denote the random variable resulting from this prediction. We can write,

$$\hat{X}_n = \sum_{k=1}^{M} h_{ok} X_{n-k} \qquad --(1)$$

where $h_{01}, h_{02}, \dots, h_{0M}$ are the *optimum predictor coefficients.* Equation (1) is illustrated in Fig. 2.1. '*M',* the number of delay elements is called as order.

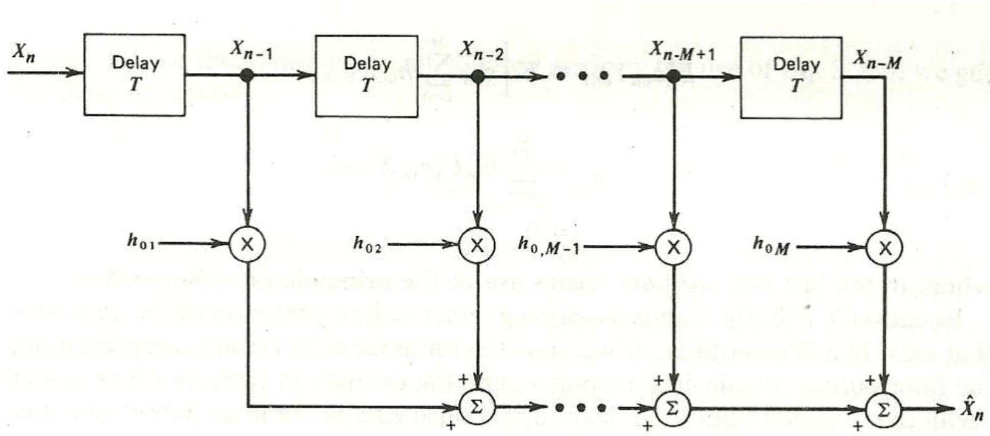


Fig. 2.1 Linear predictor

The normal equations for the predictor coefficients in Fig. 2.1 is set by minimizing the mean-square value of the prediction error. An important approach is to consider the predictor as a special case of the Wiener filter. It is given as,

1. The variance of the sample X_n, viewed as the desired response, equals

$$\sigma_X^2 = E[X_n^2]$$

$$= R_x(0)$$

   where it is assumed that X_n has zero mean.
2. The cross-correlation function of X_n, acting as the desired response, and X_{n-k} acting as the k^{th} tap input of the predictor, is given by
$$E[X_n X_{n-k}] = R_x(k), \qquad k = 1,2,\dots,M$$
3. The autocorrelation function of the predictor's tap input X_{n-k} with another tap input X_{n-m} is given by
$$E[X_{n-k} X_{n-m}] = R_x(m-k), \qquad k = 1,2,\dots,M$$

The normal equations to fit the linear prediction problem (where we look into the future by one sample) as follows:

$$\sum_{m=1}^{M} h_{om} R_x(m-k) = R_x(k), \qquad k = 1,2,\dots,M \qquad --(2)$$

in order to solve the normal equations for the predictor coefficients, the autocorrelation function of the signal for different lags are needed.

2.1.1 Prediction-error Process

The *prediction error,* denoted by ε_n, is defined by

$$\varepsilon_n = X_n = \hat{X}$$

$$= X_n - \sum_{k=1}^{M} h_{ok} X_{n-k} \qquad --(3)$$

Thus, given the present and past samples of a stationary process, namely, $X_n, X_{n-1}, \dots, X_{n-M}$, and given the predictor coefficients $h_{o1}, h_{o2}, \dots, h_{oM}$. The prediction error, ε_n is computed by using the structure shown in Fig 2.2a. This structure is called a *prediction-error filter.*

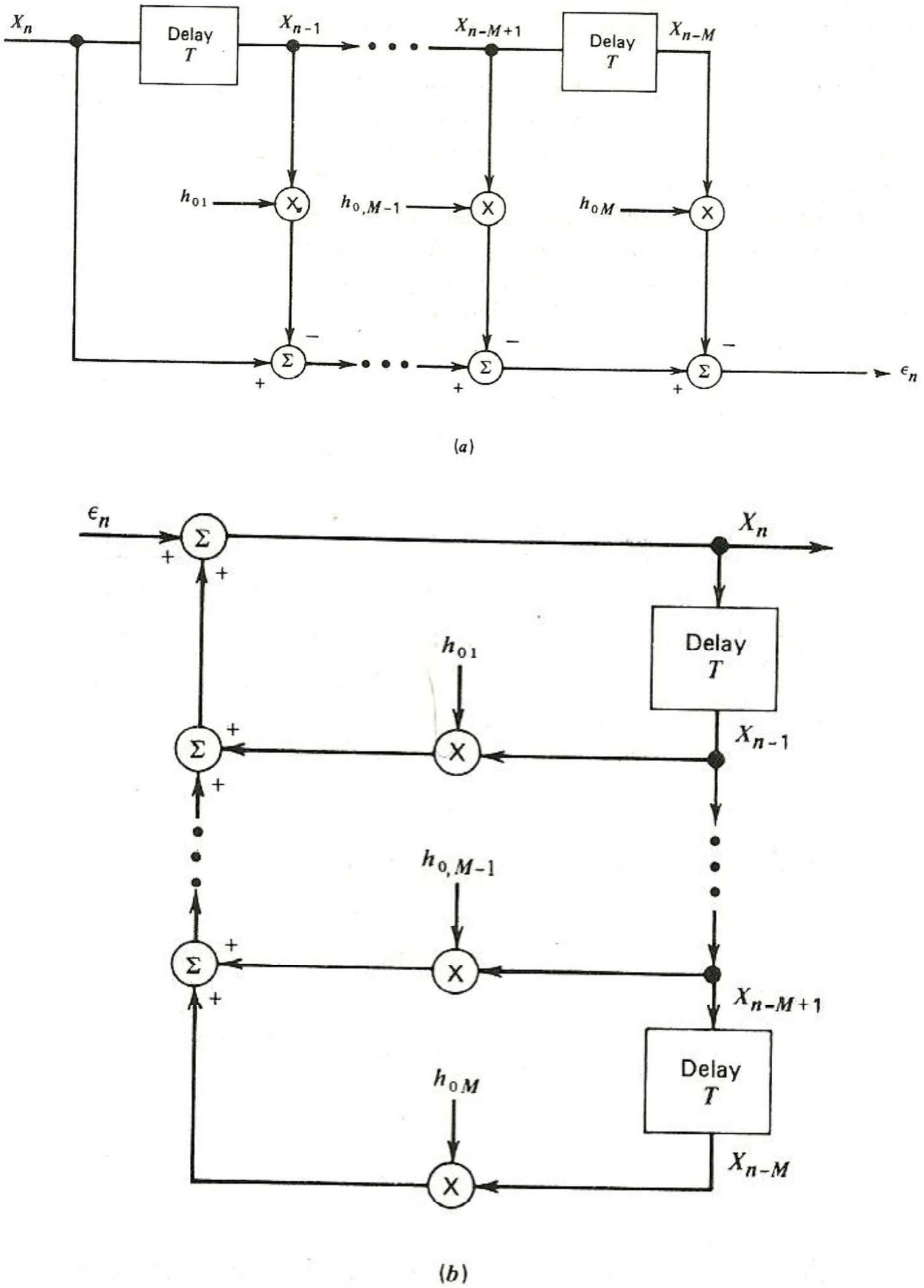


Figure 2.2 (a) Prediction-error filter. *(b)* Inverse filter.

The operation of prediction-error filtering is invertible. Specifically, we may rearrange Eq. (3) as

$$X_n = \sum_{k=1}^{M} h_{ok} X_{n-k} + \varepsilon_n$$

Hence, the "present" sample of the original process, X_n, may be computed as a linear combination of "past" samples of the process, $X_{n-1}, \dots, X_{n-M}$, plus the "present" prediction error ε_n, (where n refers to the present). Figure *2.2b* shows the structure the *inverse filter.* Note that the impulse response of the inverse filter has *infinite duration* because of feedback present in the filter, whereas the impulse response of the prediction-error filter has finite duration.

The structures of Figs. *2.2a* and 2.2*b* shows that there is a one-to-one correspondence between samples of a stationary process and those of the prediction-error process. Thus, if one is given, the other can be computed by means of a linear filtering operation. The prediction-error variance is less than σ_X^2, the variance of X_n. If X_n has zero mean, so will ε_n. Then, the prediction-error variance equals

$$\sigma_E^2 = E[\varepsilon_n^2]$$

$$= R_X(0) - \sum_{k=1}^{M} h_{ok} R_X(k) \qquad --(4)$$

The summation term on the right side of Eq. (4) is less than $R_x(0)$. Since $\sigma_X^2 = R_x(0)$, we have $\sigma_E^2 < \sigma_X^2$. This inequality is illustrated by a predictor of order one considered In Example 1

EXAMPLE 1 PREDICTOR OF ORDER ONE

Figures *2.3a* and *2.3b* show a predictor of order one and the corresponding prediction error filter, respectively. There is a single predictor coefficient, h_{01}, to be evaluated.

From Eq. (2) we get

$$h_{01} R_X(0) = R_X(1)$$

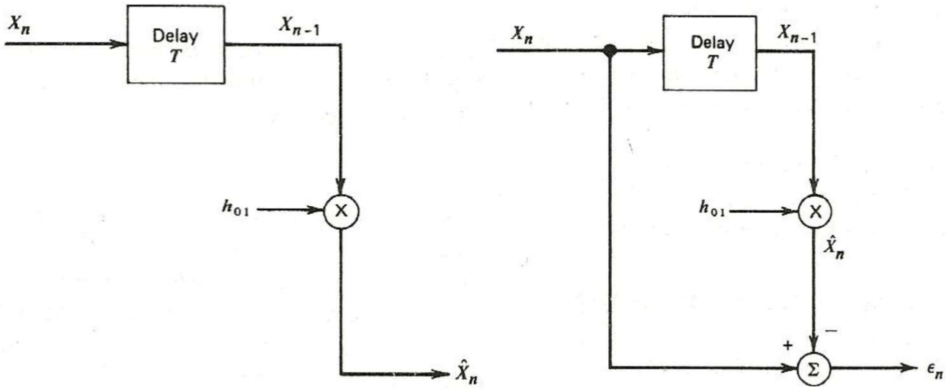
$$or \quad h_{01} = \frac{R_X(1)}{R_X(0)}$$

Using Eq. (4), the corresponding value of prediction-error variance:

$$\sigma_E^2 = R_X(0) - h_{o1} R_X(1)$$

$$= R_X(0) - \frac{R_X^2(1)}{R_X(0)} \qquad --(5)$$

Since $\sigma_X^2 = R_X(0)$, we assume that the prediction-error variance is less than the variance of the predictor input.



(a) **(b)**

Figure 2.3 ***(a)*** **Predictor of order one.** ***(b)*** **Prediction-error filter of order one.**

PROPERTY 1

The prediction-error variance decreases with increasing predictor order. In theory, this trend may go on indefinitely, or until a critical predictor order is reached, where after there is no further reduction.

PROPERTY 2

When the prediction-error variance reaches its minimum possible value, the prediction error process assumes the form of white noise. A prediction-error filter designed to whiten a stationary process is called a **whitening filter.** The resultant white noise sequence is known as the **innovations process** associated with the predictor input; the term "innovation" refers to "newness." Hence, only new information is retained in the innovations process.

Basically, prediction relies on the presence of correlation between adjacent samples of a stationary process. As we increase the predictor order, we successively reduce the correlation between adjacent samples of the process, until ultimately the prediction-error process consists of a sequence of uncorrelated samples. When this condition is reached, the prediction-error variance attains its minimum possible value, and the whitening of the original process is thereby accomplished.

2.2 DIFFERENTIAL PULSE CODE MODULATION

2.2.1 Redundant Information In PCM

The samples of a signal are highly corrected with each other. This is because any signal does not change fast. That is its value from present sample to next sample does not differ by large amount. The adjacent samples of the signal carry the same information with little difference. When these samples are encoded by standard PCM system, the resulting encoded signal contains redundant information. This is illustrated in fig. 2.4.

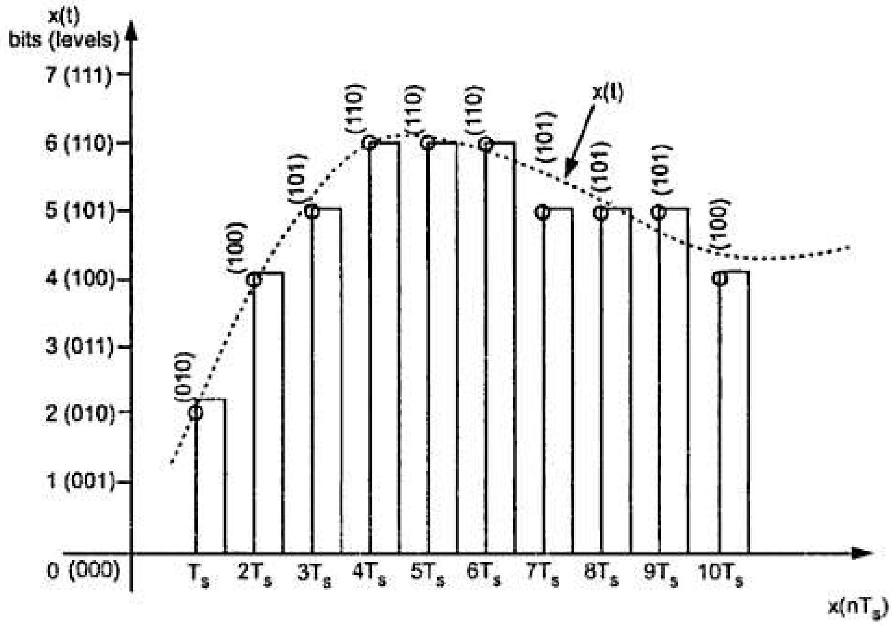


Fig.2.4 Redundant Information in PCM

Fig. 2.4 shows a continuous time signal $x(t)$ by dotted line. This signal is sampled at intervals $T_s, 2T_s, 3T_s, \dots nT_s$. The sampling frequency is selected to be higher than nyquist rate. The samples are encoded by using 3 bit (7 levels) PCM. The sample is quantized to the nearest digital level as shown by small circles in the diagram. The encoded binary value of each sample is written on the top of the samples. From Fig. 2.4 that the samples taken at $4T_s, 5T_s$ *and* $6T_s$ are encoded to same value of (110). This information can be carried only by one sample. But three samples are carrying the same information means it is redundant.

2.2.2 Principle of DPCM

If this redundancy is reduced, then **overall bit rate will decrease** and number of bits required to transmit one sample will also be reduced. This type of digital pulse modulation scheme is called Differential Pulse Code Modulation.

2.2.3 DPCM Transmitter

The differential pulse code modulation works on the principle of prediction. The value of the present sample is predicted from the past samples. The prediction may not be exact but it is very close to the actual sample value. Fig. 2.5 shows the transmitter of Differential Pulse Code Modulation (DPCM) system. The sampled signal is denoted by $x(nT_s)$ and the predicted Signal is denoted by $\hat{x}(nT_s)$. The comparator finds out the difference between the actual sample value $x(nT_s)$ and predicted sample value $\hat{x}(nT_s)$. This is called error and it is denoted by $e(nT_s)$. It can be defined as,

$$e(nT_s) = x(nT_s) - \hat{x}(nT_s) \qquad --(1)$$

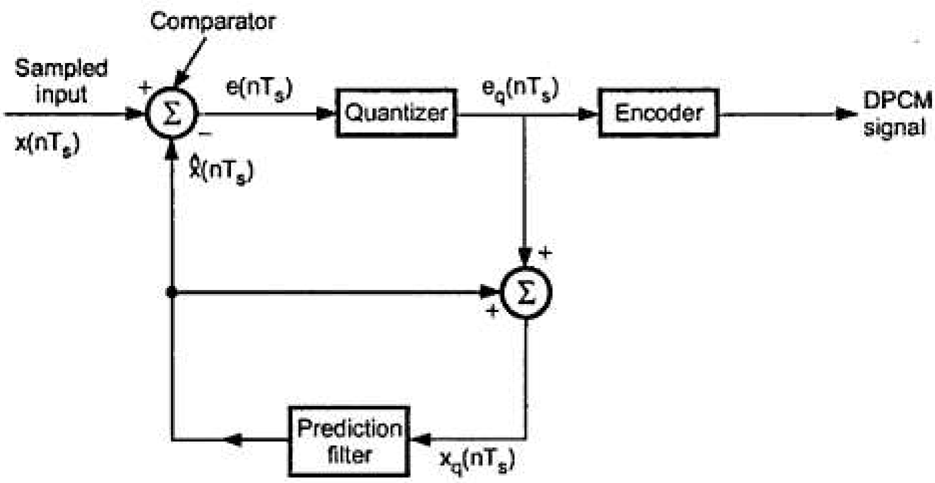


Fig. 2.5 Differential pulse code modulation transmiter

Thus error is the difference between unquantized input sample $x(nT_s)$ and prediction of it $\hat{x}(nT_s)$. The predicted value is produced by using a prediction filter. The quantizer output signal $x(nT_s)$ and previous prediction is added and given as input to the prediction filter. This signal is called $x_q(nT_s)$. This makes the prediction more and more close to the actual sampled signal. Here the quantized error signal $e_q(nT_s)$ is very small and can be encoded by using small number of bits. Thus number of bits per sample are reduced in DPCM.

The quantizer output can be written as,

$$e_q(nT_s) = e(nT_s) + q(nT_s) \qquad --(2)$$

Here $q(nT_s)$ is the quantization error. As shown in Fig. 2.5, the prediction filter input $x_q(nT_s)$ is obtained by sum $\hat{x}(nT_s)$ and quantizer output i.e.,

$$x_q(nT_s) = \hat{x}(nT_s) + e_q(nT_s)$$

Putting the value of $e_q(nT_s)$ from equation (2) in the above equation we get,

$$x_q(nT_s) = \hat{x}(nT_s) + e(nT_s) + q(nT_s) \qquad --(3)$$

Equation (1) is written as,

$$e(nT_s) = x(nT_s) - \hat{x}(nT_s)$$

$$\therefore \quad e(nT_s) + \hat{x}(nT_s) = x(nT_s)$$

∴ Putting the value of $e(nT_s) + \hat{x}(nT_s)$ from above equation into equation (3) we get,

$$x_q(nT_s) = x(nT_s) + q(nT_s) \qquad --(4)$$

Thus the quantized version of the signal $x_q(nT_s)$ is the sum of original sample value and quantization error $q(nT_s)$ The quantization error can be positive or negative. Thus equation (4) does not depend on the prediction filter characteristics.

2.2.4 Reconstruction of DPCM Signal

Fig. 2.6 shows the block diagram of DPCM receiver.

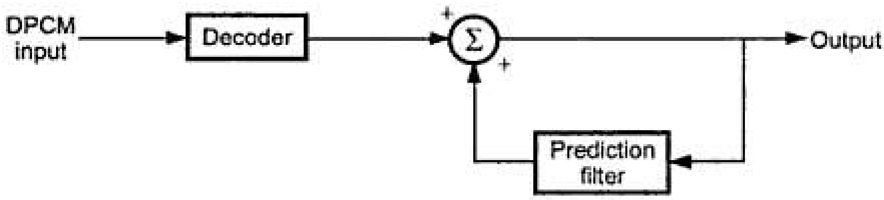


Fig. 2.6 DPCM receiver

The decoder first reconstructs the quantized error signal from incoming binary signal. The prediction filter output and quantized error signals are summed up to give the quantized version of the original signal. Thus the signal at the receiver differs from actual signal by quantization error $q(nT_s)$ which is introduced permanently in the reconstructed signal.

2.2.5 Signal to Noise ratio in DPCM

The noise power is basically mean square value $\{E[\varepsilon^2]\}$ of the noise. If the noise has zero mean (i.e. $m_\varepsilon = 0$), then

$$variance \;\; \sigma_\varepsilon^2 = E[\varepsilon^2] - m_\varepsilon^2$$

$$\therefore \quad \sigma_\varepsilon^2 = E[\varepsilon^2] \;\; since\ mean, m_\varepsilon^2 = 0$$

Thus the variance of noise is basically noise power.

Similarly variance of the signal of zero mean can be used as signal power. Variance of signal is represented as σ_x^2. Thus,

$$\sigma_x^2 = E[X^2] \;\; since\ mean, m_x^2 = 0$$

Then signal to noise ratio can be represented as,

$$\frac{S}{N} = \frac{\sigma_x^2}{\sigma_\varepsilon^2}$$

Above equation can be written as,

$$\frac{S}{N} = \underbrace{\frac{\sigma_x^2}{\sigma_e^2}}_{prediction\ gain} \cdot \underbrace{\frac{\sigma_e^2}{\sigma_\varepsilon^2}}_{PCM\ signal\ to\ noise\ ratio}$$

Here σ_x^2 is the variance of the prediction error $e\ (nT_s)$

$$\therefore \quad \left(\frac{S}{N}\right) = G_p.\left(\frac{S}{N}\right)_{PCM} \qquad --(1)$$

Here G_p is the prediction gain obtained due to differential quantization. It is given as,

$$G_p = \frac{\sigma_x^2}{\sigma_e^2}$$

$$And \;\; \left(\frac{S}{N}\right)_{PCM} = \frac{\sigma_e^2}{\sigma_\varepsilon^2}$$

Observe that the prediction error $x(nT_s)$ is input signal to quantizer in DPCM. Thus it acts like a 1-bit PCM system. Hence above equation is written and it can be calculated using signal to noise ratio equations for PCM. i.e.,

$$\left(\frac{S}{N}\right)_{PCM} = \frac{\sigma_e^2}{\sigma_\varepsilon^2} = 3\ \times 2^v$$

➔ Equation (1) can be expressed in dB as,

$$\left(\frac{S}{N}\right)_{dB} = G_p(dB) + \left(\frac{S}{N}\right)_{PCM} dB$$

$$= G_p(dB) + (4.8 + 6v)$$

- Here G_p is greater than unity. It represents the gain in signal to noise ratio due to differential quantization.
- Since σ_x^2 (in G_p is) fixed, G_p can be improved by reducing σ_e^2. This means predicted signal should be very close to actual signal.

2.3 DELTA MODULATION

In PCM, all the bits which are used to code the sample are transmitted. Hence signaling rate and transmission channel bandwidth are large in PCM. To overcome this problem Delta Modulation is used.

2.3.1 Operating Principle of DM

Delta modulation transmits only one bit per sample. That is the present sample value is compared with the previous sample value and the indication whether the amplitude is increased or decreased is sent. Input signal $x(t)$ is approximated to step signal by the delta modulator. This step size is fixed. The difference between the input signal $x(t)$ and staircase approximated signal limited to two levels, i.e, $+\,\delta$ and $-\delta$. If the difference is positive, then approximated signal is increased by one step i.e.. $'0'$. If the difference is negative, then approximated signal is reduced by '0'. When the step is reduced, '0' is transmitted and if the step is increased, ‘1’ is transmitted. Thus for each sample, only one binary bit is transmitted.

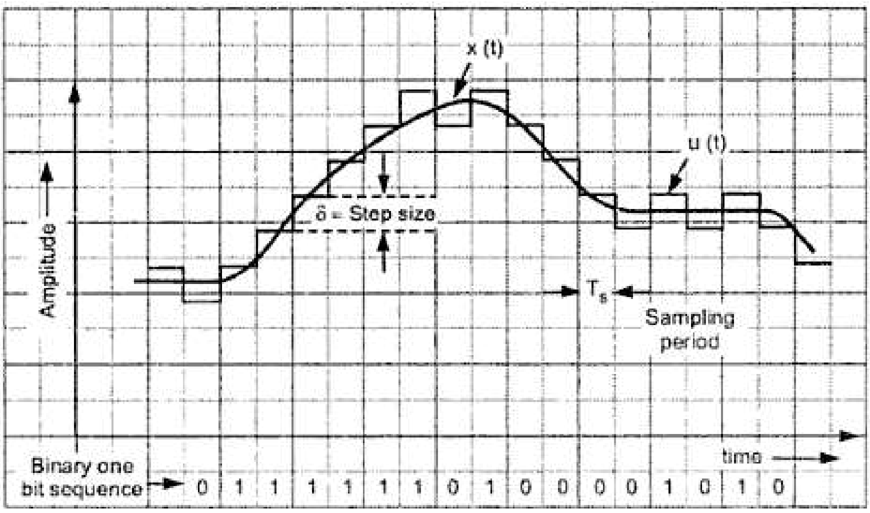


Fig. 2.7 Delta modulation waveform

Fig. 2.7 shows the analog signal $x(t)$ and its staircase approximated signal by the delta modulator.

The error between the sampled value of $x(t)$ and last approximated sample is given as,

$$e(nT_s) = x(nT_s) - \hat{x}(nT_s)$$

Here, $e(nT_s) =$ Error at present sample

$x(nT_s) =$ Sampled signal of $x(t)$

$\hat{x}(nT_s) =$ Last sample approximation of the staircase waveform

Let $u(nT_s)$ as the present sample approximation of staircase output

Then, $u[(n-1)T_s] = \hat{x}(nT_s)$

$$= Last\ sample\ approximation\ of\ staircase\ waveform$$

Let the quantity $b(nT_s)$ be defined as

$$b(nT_s) = \delta\ sgn[e(nT_s)]$$

That is depending on the sign of error $e(nT_s)$ the sign of step size δ will be decided. In other words,

$$b(nT_s) = +\delta \quad if\ \ x(nT_s) \geq \hat{x}(nT_s)$$

$$= -\delta \quad if\ \ x(nT_s) < \hat{x}(nT_s)$$

$If\ \ b(nT_s) = +\delta\ \ binary$ '1' is transmitted

$If\ \ b(nT_s) = -\delta\ \ binary$ '0' is transmitted

$$T_s = sampling\ interval$$

2.3.2 DM Transmitter

Fig. 2.8 shows the DM transmitter. The summer in the accumulator adds quantizer output $\pm\delta$ with the previous sample approximation. This gives present sample approximation. i.e.,

$$u(nT_s) = u(nT_s - T_s) + [\pm\delta]$$

$$= u[(n-1)T_s] + b(nT_s)$$

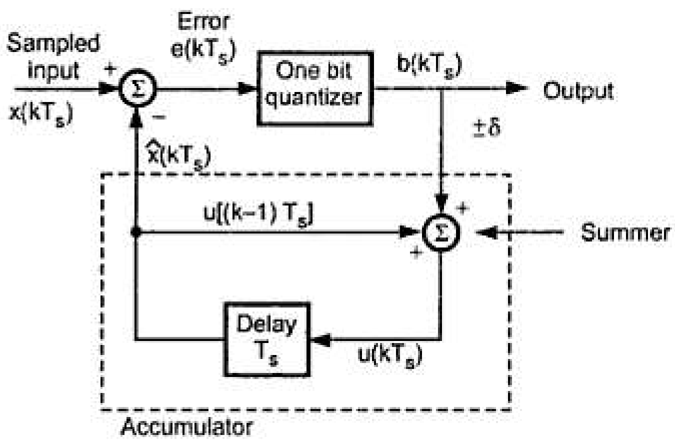


Fig. 2.8 Delta modulation transmitter

The previous sample approximation $u[(n-1)T_s]$ is restored by delaying one sample period T_s. The sampled input signal $x(nT_s)$ and staircase approximated signal $\hat{x}(nT_s)$ are subtracted to get error signal $e(nT_s)$.

Depending on the sign of $e(nT_s)$ one bit quantizer produces an output step of $+\delta$ or $-\delta$. If the step size is$+\delta$, then binary '1' is transmitted and if it is $-\delta$, then binary '0' is transmitted.

2.3.3 DM Receiver

At the receiver shown in Fig. 2.9, the accumulator and low-pass filter are used.

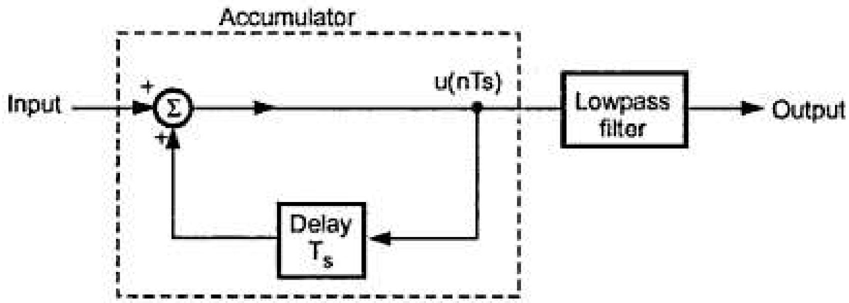


Fig. 2.9 Delta modulation receiver

The accumulator generates the staircase approximated signal output and is delayed by one sampling period T_s. It is then added to the input signal. If input is binary '1' then it adds $+\delta$ step to the previous output (which is delayed). If input is binary '0' then one step 'δ' is subtracted from the delayed signal. The low-pass filter has the cutoff frequency equal to highest frequency in $x(t)$. This filter smoothen the staircase signal to reconstruct $x(t)$.

2.3.4 Advantages of Delta Modulation

The delta modulation has following advantages over PCM,

1. Delta modulation transmits only one bit for one sample. Thus the signaling rate and transmission channel bandwidth is quite small for delta modulation.

2. The transmitter and receiver implementation is very much simple for delta modulation. There is no analog to digital converter involved in delta modulation.

2.3.5 Disadvantages of Delta Modulation

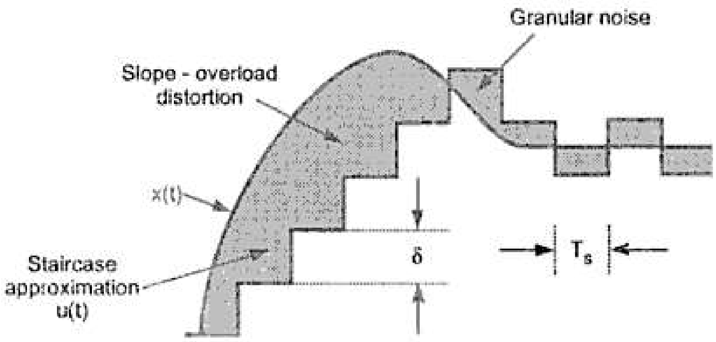


Fig. 2.10 Quantization errors in delta modulation

The delta modulation has two drawbacks

2.3.5.1 Slope Overload Distortion (Startup Error)

This distortion arises because of the large dynamic range of the input signal. As can be seen from Fig. 2.10 the rate of rise of input signal $x(t)$ is so high that the staircase signal cannot approximate it, the step size $'\delta'$ becomes too small for staircase signal $u(t)$ to follow the steep segment of $x(t)$. Thus there is a large error between the staircase approximated signal and the original input signal $x(t)$. This error is called **slope overload distortion**.

To reduce this error, the step size should be increased when slope of signal of $x(t)$ is high. Since the step size of delta modulator remains fixed, its maximum or minimum slopes occur along straight lines. Therefore this modulator is also called Linear Delta Modulator (LDM).

2.3.5.2 Granular Noise (Hunting)

Granular noise occurs when the step size is too large compared to small variations in the input signal. That is for very small variations in the input signal, the staircase signal is changed by large amount δ because of large step size. Fig. 2.10 shows that when the input signal is almost fiat, the staircase signal $x(t)$ keeps on oscillating by $\pm\delta$ around the signal. The error between the input and approximated signal is called *granular noise.*

The solution to this problem is to make step size small Thus large step size is required to accommodate wide dynamic range of the input signal (to reduce slope overload distortion) and small steps are required to reduce granular noise. Adaptive delta modulation is used to overcome these errors.

2.4 ADAPTIVE DIFFERENTIAL PULSE-CODE MODULATION (ADPCM)

Reduction in the number of bits per sample from 8 to 4 involves the combined use of **adaptive quantization** and **adaptive prediction**. The term "adaptive" means adjusting to changing level and spectrum of the input speech signal. The variation of performance with speakers and speech material, together with variations in signal level inherent in the speech communication process, make the combined use of adaptive quantization and adaptive prediction necessary to achieve best performance over a wide range of speakers and speaking situations. A digital coding scheme that uses both adaptive quantization and adaptive prediction is called **adaptive differential pulse-code modulation (ADPCM)**.

The term "adaptive quantization" refers to a quantizer that operates with a **time-varying** step size $\Delta(nT_s)$, where T_s is the sampling period. At any given time, the adaptive quantizer is assumed to have a uniform transfer characteristic. The step size $\Delta(nT_s)$ is varied in order to match the variance σ_X^2 of the input signal $x(nT_s)$. In particular, we write

$$\Delta(nT_s) = \phi\hat{\sigma}_X(nT_s) \qquad --(1)$$

where ϕ is a constant, and $\hat{\sigma}_X(nT_s)$ is an *estimate* of the standard deviation $\sigma_X(nT_s)$ (i.e., square root of the variance σ_X^2. For a nonstationary input, $\sigma_X(nT_s)$ is time-variable. Hence, the problem of adaptive quantization, according to Eq. (1), is one of estimating $\sigma_X(nT_s)$ continuously.

The estimate $\hat{\sigma}_X(nT_s)$ is done in one of two ways:

1. Unquantized samples of the input signal are used to derive forward estimates of $\sigma_X(nT_s)$, as in Fig. *2.11.* This quantization scheme is referred to as **adaptive quantization with forward estimation (AQF)**

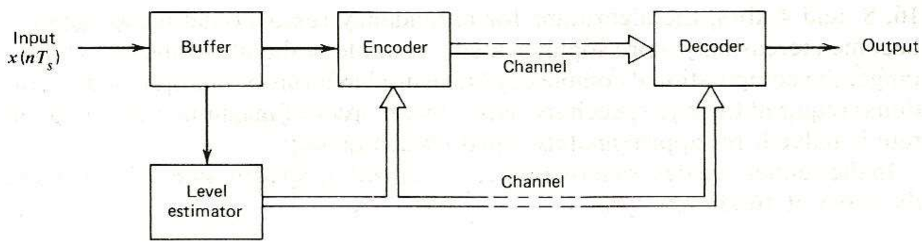


Figure 2.11 Adaptive quantization with forward estimation (AQF).

2. Samples of the quantizer output are used to derive backward estimates of $\sigma_X(nT_s)$, as in Fig. 2.12. This quantization scheme is referred to as **adaptive quantization with backward estimation (AQB).**

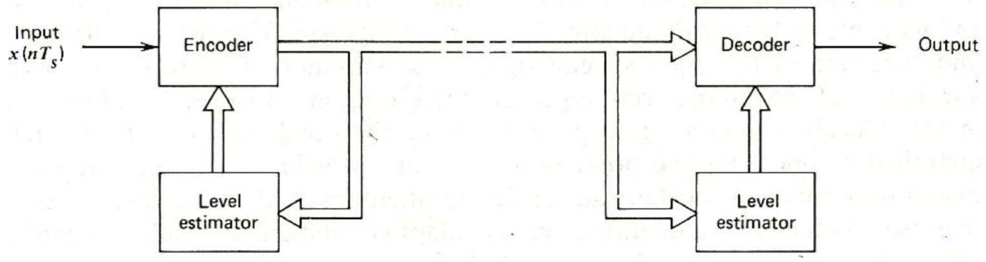


Figure 2.12 Adaptive quantization with backward estimation (AQB).

The AQF scheme of Fig. *2.11* first goes through a learning period by buffering unquantized samples of the input speech signal. The samples are released after the estimate $\hat{\sigma}_X(nT_s)$ has been obtained. This estimate is independent of quantizing noise. Therefore the step size $\Delta(nT_s)$ obtained from AQF is more reliable than that from AQB. However, the use of AQF requires the clear transmission of level information (typically, about 5 to 6 bits per step size sample) to a decoder. Thus additional **side information** has to be transmitted to the receiver. Also, a processing *delay* (on the order of 16 ms for speech) in the encoding operation results from the use of AQF, which is unacceptable in some applications.

The problems of level transmission, buffering, and delay intrinsic to AQF are all avoided in the AQB scheme of Fig. *2.12* by using the recent history of the quantizer output to extract information for the computation of the step size $\Delta(nT_s)$. Therefore, AQB is usually preferred over AQF in practice.

The adaptive quantizer with backward estimation, as in Fig. *2.12,* represents a nonlinear feedback system. Hence, the system is stable only if the quantizer input $x(nT_s)$ is *bounded* otherwise unstable.

Since the speech signals are inherently nonstationary i.e., time-varying, adaptive prediction is used. Same as adaptive quantization, there are two schemes for performing adaptive prediction:

1. **Adaptive prediction with forward estimation (APF**), in which unquantized samples of the input signal are used to derive estimates of the predictor coefficients as shown in Fig. 2.13
2. **Adaptive prediction with backward estimation (APB),** in which samples of the quantizer output and the prediction error are used to derive estimates of the predictor coefficients as shown in Fig. *2.14.*

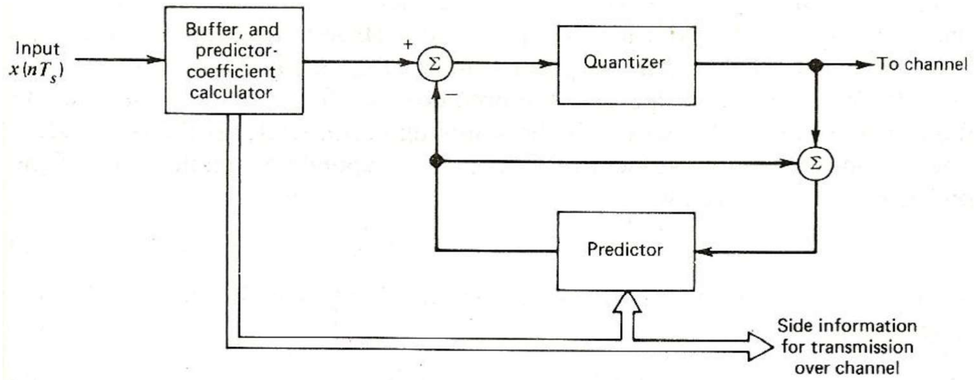


Figure 2.13 Adaptive prediction with forward estimation (APF)

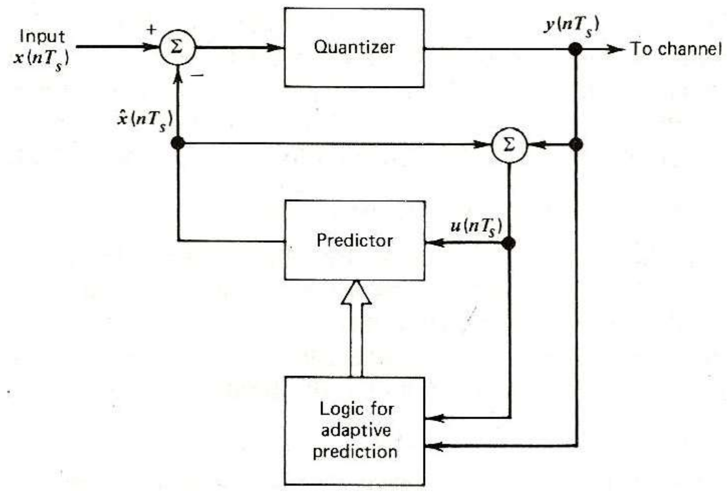


Figure 2.14 Adaptive prediction with backward estimation (APB).

In the APF scheme of Fig. 2.13, N unquantized samples of the input speech are first buffered and then released after computation of M predictor coefficients that are optimized for the buffered segment of input samples. The choice of M involves a compromise between an adequate prediction gain and an acceptable amount of side information. Likewise, the choice of learning period or buffer length N involves a compromise between the rate at which statistics of the input speech signal change and the rate at which information on predictor coefficients must be updated and transmitted to the receiver. For speech, a good choice of N corresponds to a 16ms buffer for a sampling rate of 8 kHz, and a choice of M = 10 ensures adequate use of the short-term predictability of speech.

However, APF has same disadvantages (side information, buffering, and delay) as AQF. These disadvantages are eliminated by using the APB scheme of Fig. 2.14. Since, in AQF, the optimum predictor coefficients are estimated on the basis of quantized and transmitted data, they can be updated as frequently as desired; for example, from sample to sample. Therefore, APB is the preferred method of prediction. The block "logic for adaptive prediction" is to represent the mechanism for updating the predictor coefficients. Let $y(nT_s)$ denote the quantizer output, where T_s is the sampling period and n is the time index. Then, from Fig. *2.14* the corresponding sample value of the predictor input is given by

$$u(nT_s) = \hat{x}(nT_s) + y(nT_s) \qquad \text{--(1)}$$

where $x(nT_s)$ is the prediction of the speech input sample $x(nT_s)$. We may rewrite Eq. (1) as

$$y(nT_s) = u(nT_s) - \hat{x}(nT_s)$$

where $u(nT_s)$ represents a sample value of the predictor input, and $\hat{x}(nT_s)$ represents a sample value of the predictor output and $y(nT_s)$ as the corresponding value of the prediction error

The structure of the predictor of order *M,* is shown in Fig. 2.15.

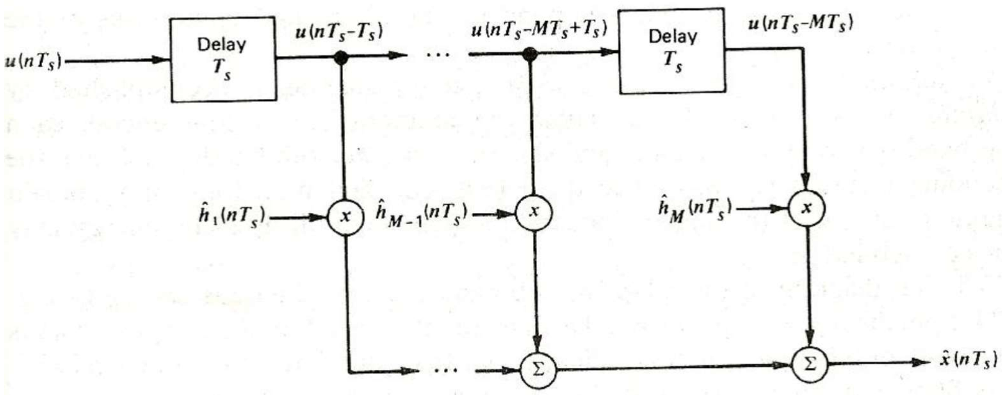


Figure 2.15 Predictor's structure

For adaptation of the predictor coefficients, we use the **least-mean-square (LMS) algorithm**. Thus,

$$\hat{h}_k(nT_s + T_s) = \hat{h}_k(nT_s) + \mu y(nT_s)u(nT_s - kT_s) \quad k = 1,2,\ldots\ldots,M \qquad --(2)$$

where μ is the adaptation constant. For the initial conditions, we set all of the predictor coefficients equal to zero at $n = 0$.The correction term in the update equation, Eq. (2), consists of the product $y(nT_s)u(nT_s - kT_s)$, scaled by the adaptation constant μ. By using a small value for μ, the correction term will decrease with the number of iterations *n.*

For **stationary** speech inputs and small quantization effects, the correction term may assume a small value in the **steady-state** condition for the average mean-squared error to closely approach the minimum possible value. For **nonstationary** inputs, care must be taken to choose the adaptation constant μ, that it has to be small enough for the fluctuations in the predictor coefficients about their optimum values to be acceptable, and large enough for the adaptation algorithm to track variations in the input statistics.

2.5 ADAPTIVE DELTA MODULATION (ADM)

2.5.1 Operating Principle

To overcome the quantization errors due to slope overload and granular noise, the step size (δ) is made adaptive to variations in the input signal $x(t)$. Particularly in the steep segment of the signal $x(t)$, the step size is increased. When the input is varying slowly, the step size is reduced. Then the method is called **Adaptive Delta Modulation (ADM)**.

The adaptive delta modulators can take continuous changes in step size or discrete changes in step size.

2.5.2 Transmitter and Receiver

Fig. 2.16(a) shows the transmitter and 2.16 (b) shows receiver of adaptive delta modulator.

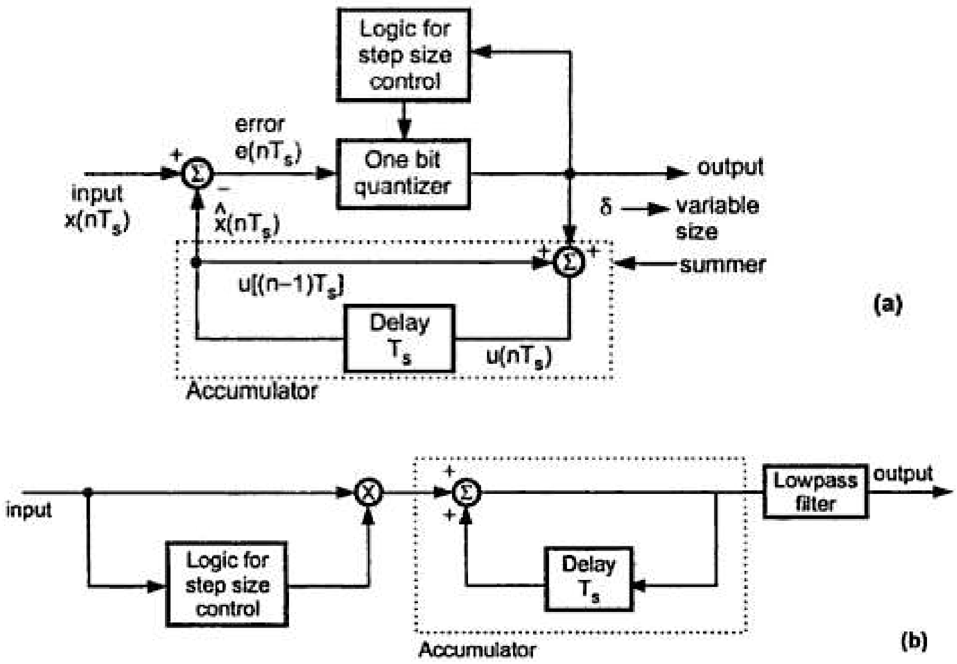


Fig. 2.16 Adaptive delta modulator (a) Transmitter (b) Receiver

The logic for step size control is added in the diagram. The step size increases or decreases according to certain rule depending on one bit quantizer output

If one bit quantizer output is high (1), then step size will be doubled for next sample. If one bit quantizer output is low, then step size will be reduced by one step. Fig. 2.17 shows the waveforms of adaptive delta modulator and sequence of bits transmitted.

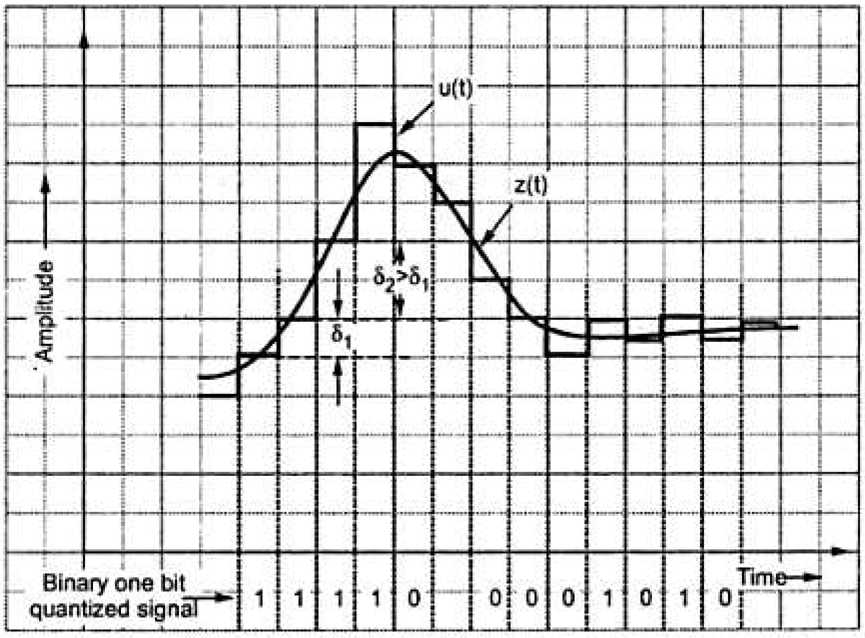


Fig. 2.17 Waveforms of adaptive delta modulation

In the receiver of adaptive delta modulator shown in Fig. 2.16 (b), the first part generates the step size from each incoming bit. Exactly the same process is followed as that in transmitter. The previous input and present input decides the step size. It is then given to an accumulator which builds up staircase waveform. The low-pass filter then smoothens out the staircase waveform to reconstruct the smooth signal.

2.5.3 Advantages of Adaptive Delta Modulation

Adaptive delta modulation has certain advantages over delta modulation. i.e.,

1. The signal to noise ratio is better than ordinary delta modulation because of the reduction in slope overload distortion and granular noise.
2. Because of the variable step size, the dynamic range of ADM is wide.
3. Utilization of bandwidth is better than delta modulation..
4. Similar to delta modulation, only one bit per sample is required and simplicity of implementation of transmitter and receiver.

2.6 LINEAR PREDICTIVE CODING

Fig. 2.18 shows the block diagram of LPC transmitter and receiver.

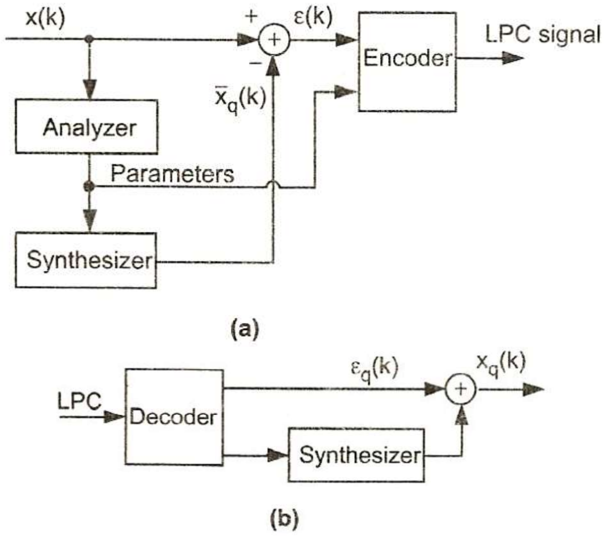


Fig. 2.18 (a) Encoder of LPC (b) Decoder of LPC

Encoder (or Transmitter): The speech signal $x\ (k)$ is the digitized signal. It is obtained by sampling the continuous time speech signal. Thus $x\ (k)$ is the sequence of speech samples. It is applied to the analyzer. The analyzer determines parameters for the synthesizer. Based on these parameters, the synthesizer reconstructs the speech signal $\hat{x}(k)$. The reconstructed signal $\hat{x}_q(k)$ and the original signal $x(k)$ are compared and the error $\varepsilon(k)$ is obtained. The analyzer parameters and the error signal is multiplexed and transmitted. This signal is called LPC signal.

Decoder (or Receiver): Now consider the decoder of Fig. 2.18 (b). The received LPC signal is applied to the demultiplexer (or decoder). It separates filter parameters and error signal $\varepsilon_q(k)$. The parameters are given to the synthesizer. The output of synthesizer is added to error signal which gives speech signal $x_q(k)$. Basically the analyzer is a digital filter. Normally this filter parameters are updated every 10 to 25 msec. For a reasonable speech quality, 10-12 parameters are used. This produces the bit rate of about 3 to 8 k bits/sec.

Advantages

(i) The encoded data rate is lowest.
(ii) The bit allocation depends upon specific characteristics of signal.

Disadvantages

(i) The parameters are evaluated for small segments.
(ii) The synthesized signal sounds monotonous.

2.7 LINEAR PREDICTIVE VOCODERS

Linear prediction provides the basis of an important source coding technique for the digitization of speech signals. The technique, known as **linear predictive vocoding,** relies on parameterization of speech signals according to a physical model for the speech production process.

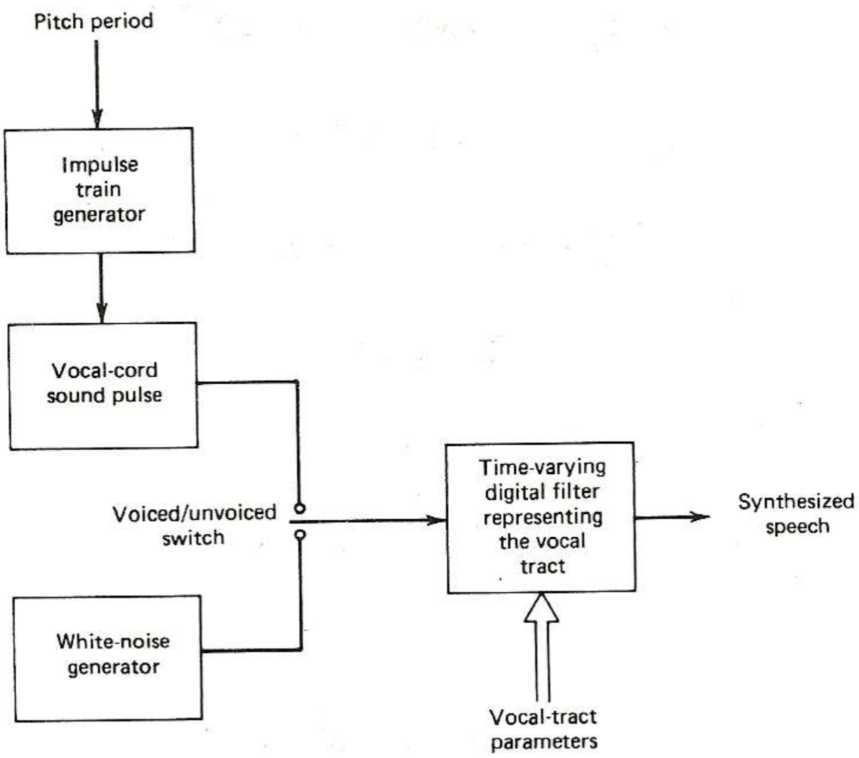


Figure 2.19 Model of speech-production process.

The term "vocoding" refers to voice *coding;* the term "vocoder" refers to the device for performing voice coding.

Figure 2.19 shows a model for the speech production process. The model assumes that the sound-generating mechanism (i.e., source of excitation*)* is linearly separable from the intelligence-modulation mechanism (i.e., *vocal tract filter).* The form of excitation depends on whether the speech sound is voiced or unvoiced. *Voiced sounds* are produced by forcing air through the glottis with the tension of the vocal cords adjusted so that they vibrate in a relaxation oscillation, thereby producing quasi-periodic pulses of air that excite the vocal tract. *Fricative* or *unvoiced sounds,* on the other hand, are generated by forming a constriction at some point in the vocal tract (usually toward the mouth end), and forcing air through the constriction at a high velocity to produce turbulence.

Examples of voiced and unvoiced sounds are represented by utterance for the" A" and "S" segments in the word "salt". The speech waveform shown in Fig. 2.20a is the result of the utterance" every salt breeze comes from the sea" by a male subject. The waveform shown in Fig. *2.20b* corresponds to the ."A" segment in the word "salt," while the magnified waveform shown in Fig. *2.20c* corresponds to the "S" segment. The generation of a voiced sound is modeled as the response of the vocal tract filter excited with a periodic sequence of impulses (very short pulses) spaced by a fundamental period equal to the *pitch* period. An unvoiced sound is modeled as the response of the vocal filter excited with a white noise sequence. The vocal tract filter is time varying, so that its coefficients can provide an adequate representation for the input segment of voiced or unvoiced sound.

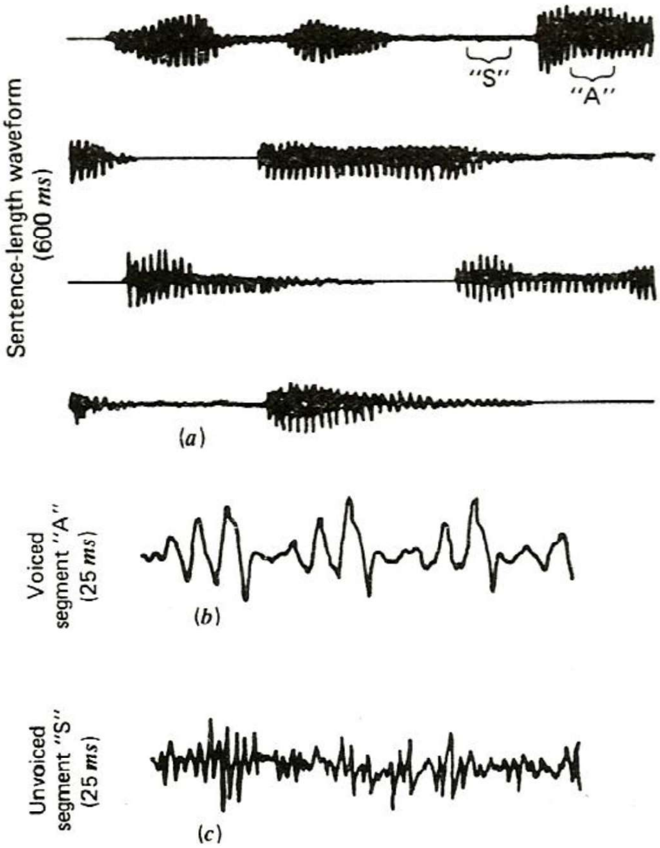


Figure 2.20 Time waveforms of speech for a long (sentence-length) segment and for short (25 ms) segments.

A linear predictive vocoder consists of a transmitter and a receiver having the block diagrams shown in Fig. 2.21.

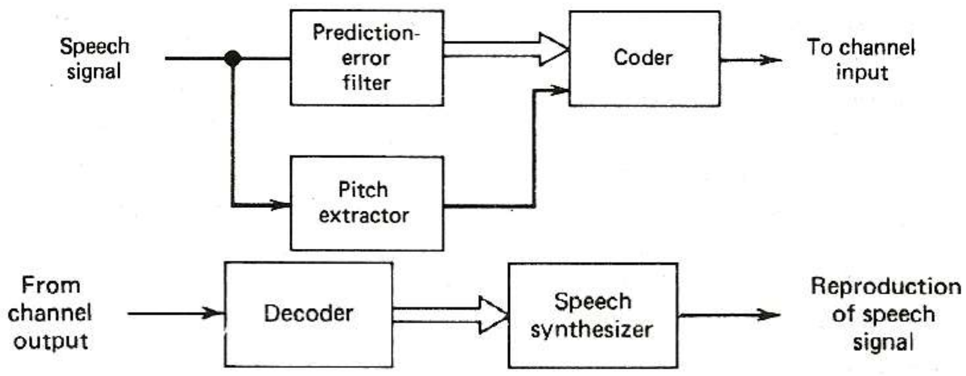


Figure 2.21 Block diagram of linear predictive vocoder. ***(a)*** **Transmitter.** ***(b)*** **Receiver.**

The transmitter, shown in Fig. *2.21a* first performs **analysis** on the input speech signal, block by block. Typically, each block is 10-30 ms long, for which the speech-production process may be treated as stationary. The parameters resulting from the analysis, namely, the prediction-error filter (analyzer) coefficients, a voiced/unvoiced parameter, and the pitch period, provides a complete description for the particular segment of the input speech signal. A digital representation of the parameters of this complete description constitutes the transmitted signal. The receiver, shown in Fig. *2.21b,* first performs decoding, followed by **synthesis** of the speech signal which utilizes the model of Fig. 2.19. The standard result of this analysis/synthesis is an artificial-sounding reproduction of the original speech signal. This poor reproduction quality of a linear predictive vocoder is tolerated for secure military communications where very low bit rates (4 *kb/s* or less) are required.

2 MARKS

1. Mention two merits of DPCM.

i) Bandwidth requirement of DPCMis less compared to PCM.
ii) Quantization error is reduced because of prediction filter.
iii) Number of bits used to represent one sample value are also reduced compared to PCM.

2. What is the main difference in DPCM and DM?

DM encodes the input sample by only one bit. It sends the information about $+\delta \ or \ -\delta$, i.e. step rise or fall. DPCM can have more than one bit for encoding the sample. It sends the information about difference between actual sample value and predicted sample value.

3. Mention the use of adaptive quantizer in adaptive digital waveform coding schemes.

Adaptive quantizer changes its step size according to variance of the input signal. Hence quantization error is significantly reduced due to adaptive quantization. ADPCM uses adaptive quantization. The bit rate of such schemes is. reduced due to adaptive quantization.