Chapter 3: Pulse Modulation

Dr. Samir Alghadhban

Content

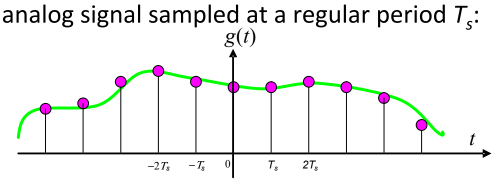
Overview of Pulse Modulation:

- We saw in EE370 that sampling of analog signals can be used to represent signals by discrete samples without loss of information. **What are the conditions ?**
- Provided that the signal is band-limited, and the sampling rate is faster than the minimum Nyquist rate.
- For communication applications, the transmission of these discrete signal pulses (instead of the full analog signal) offers many advantages, such as
  - Power savings (since the samples will occupy short duty cycles)
  - Efficient processing (more sophisticated signal processing will be possible)
  - Robust transmission (against the effect of noise, etc)
- In this lecture, we study the different types of pulse modulation schemes, and focus more closely on the Pulse Amplitude Modulation (PAM) combined with digital coding as found in Pulse Code Modulation (PCM)
- Applications that combine PCM with time division multiplexing (TDM) are widely used in practice, and will also be discussed in the context of digital telephony networks

3

Types of Pulse Modulation:

- Consider an analog signal sampled at a regular period T_s:

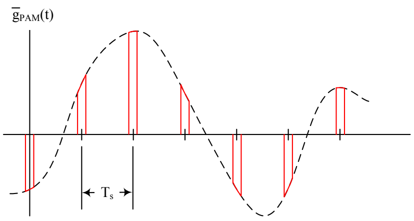


- There are different ways of representing the sample values by using a pulse train (sequence of regular pulses spaced at the sampling period). This pulse train is like an unmodulated carrier that will be altered according to the sample values being sent
- We can have several methods of Pulse Modulation. For example, we may use:
  - Pulse Amplitude Modulation - PAM (sample value mapped to pulse amplitude)
  - Pulse Width Modulation - PWM (sample value mapped to pulse width)
  - Pulse Position Modulation - PPM (sample value mapped to pulse position)

4

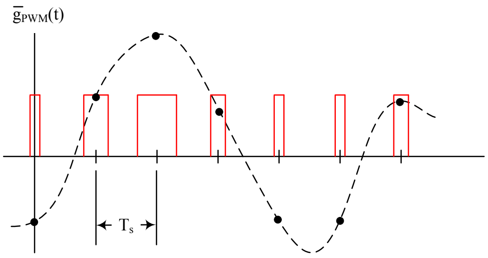
Pulse Amplitude Modulation (PAM):

- This is the simplest form of pulse modulation, where the amplitudes of the successive pulses are changed in accordance with the value of the analog samples as illustrated in the following figure
- The figure illustrates PAM with a "sample & follow" format, but we can also have "sample & hold" schemes where the amplitudes are held constant at their initial values



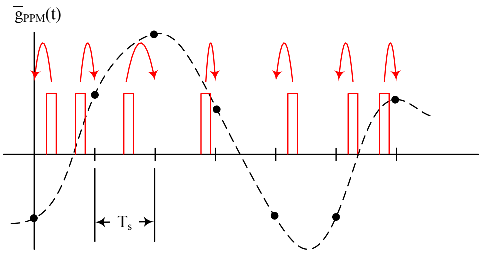
Pulse Width Modulation (PWM):

- In this scheme, the default constant duration (or width) of the basic pulses is altered according to the corresponding values of the analog samples, as illustrated in the following figure



Pulse Position Modulation (PPM):

- In this scheme, the default fixed starting position of the basic pulses is altered according to the values of the analog samples, as illustrated in the figure



7

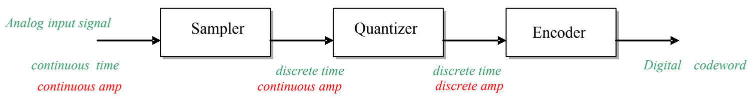
Observations

- PAM, PWM and PPM schemes are all used to transmit samples of an analog signal in a more efficient way than transmitting the full analog signal completely
- PAM offers a clear power savings (by using short-duration pulses), but PWM and PPM in particular can offer additional advantages in terms of robustness to noise and distortion. This is because the "information" (i.e., message values) is contained in pulse duration or position, and not in its amplitude (which is more sensitive to noise). So, if the pulse limits are sharp (i.e., quick rise/fall times), then PWM and PPM will be detected more robustly
- However, on the downside, PAM, PWM and PPM will use more transmission bandwidth than the original analog signal, so there is a tradeoff between power and bandwidth efficiency by using these schemes

8

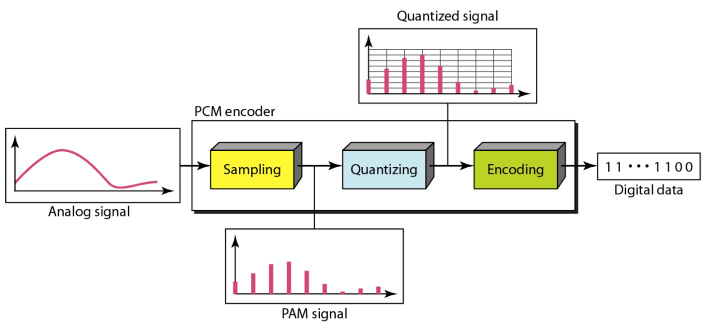
Digital Pulse Modulation of Analog Signals:

- We saw in previous lectures that analog signals can be fully converted to digital bit streams by performing three operations:
  - Sampling: to select discrete-time samples from the analog signal
  - Quantization: to confine the sampled values to a finite number of amplitude levels
  - Encoding: to represent the finite number of amplitudes by digital codewords
- These stages are illustrated in the following diagram:



Pulse Code Modulation (PCM):

- These operations can be viewed as a combination of PAM modulation with quantization and digital encoding.
- The resulting scheme is called Pulse Code Modulation (PCM), and is widely used in practical transmission systems.

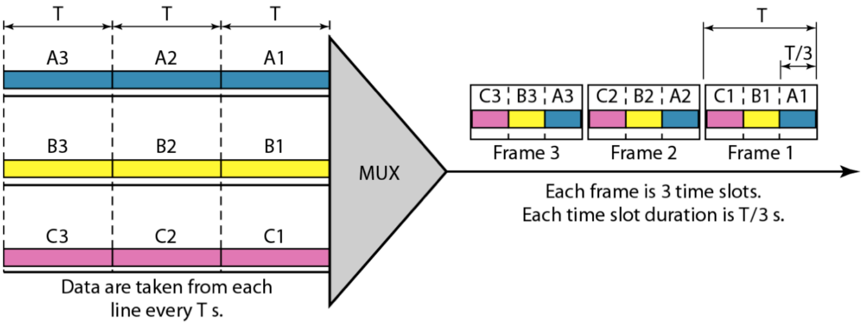


Features of PCM:

- PCM is the most commonly used method of pulse modulation because it offers good performance in noisy environments, and is easily implemented in digital hardware
- PCM is usually combined with time division multiplexing to aggregate many low-rate signals onto a high-rate transmission line (discussed next). Timing synchronization is an important requirement in these situations
- A disadvantage of PCM is that the bandwidth consumed is much larger (compared to direct analog signal transmission). However, there are techniques that can reduce the PCM data rates, and consequently bandwidth consumption (to be discussed in the next lecture)
- A very typical example of PCM transmission is found in telephony networks, where the voice signals are limited to a band less than 4KHz before being sampled at a rate of 8000 samples/sec, and then quantized and encoded with 8 bits per sample. The PCM bit rate (per voice line) is therefore 64 Kbps.

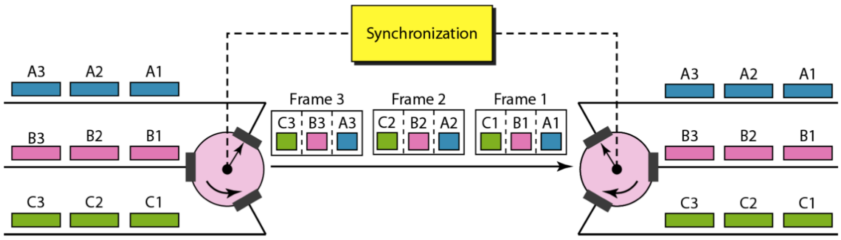
11

Illustration of TDM (with 3 signals):



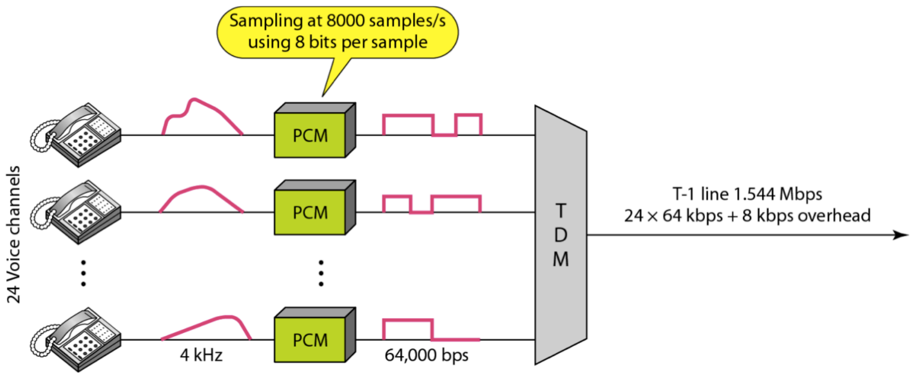
12

Demultiplexing of TDM Signals:



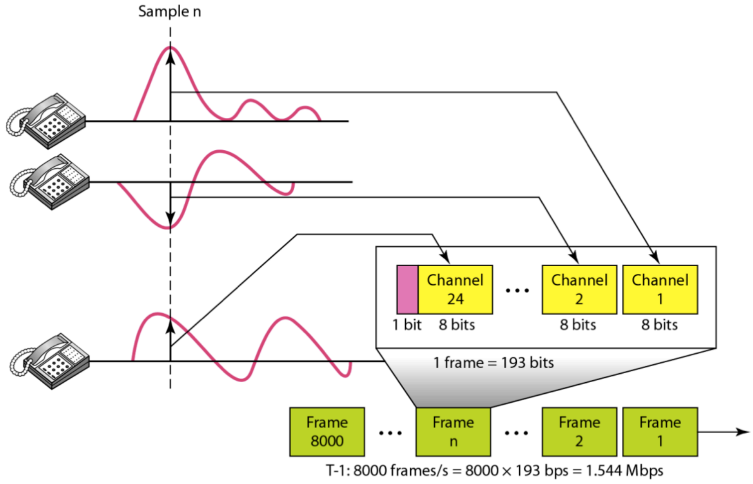
13

Example: T1 Line in Digital Telephony Transmission



14

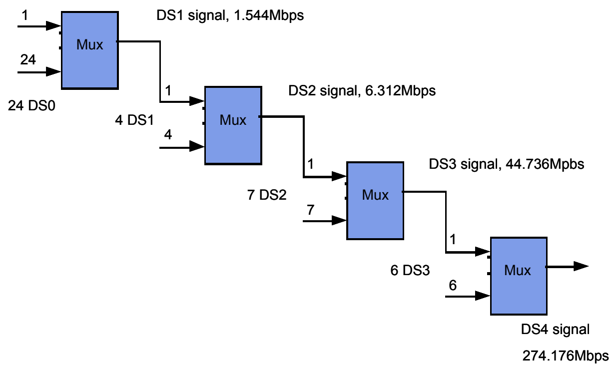
T1 Example cont'd:



15

Digital Multiplexing Hierarchy in Telephony Networks:

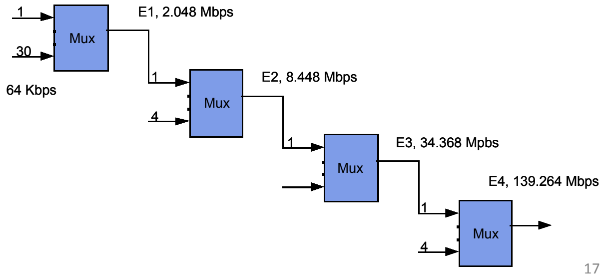
- Standard Digital Signal (DS) multiplexing rates are used in telephony networks. For example, the PCM basic rate is denoted by DS0, 24 DS0 lines form a DS1 (T1 line), 4 DS1 lines form a DS2 line, etc.



16

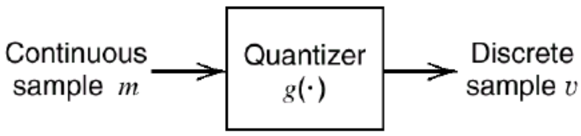
Digital Multiplexing Hierarchy in Telephony Networks (cont'd):

- The DS rates are mainly used is North America and Japan. In most other countries, another hierarchy is used, which is also based on the basic 64Kbps PCM rate. 30 PCM voice channels are aggregated to form a 2.048Mbps E1 line (two other 64Kbps channels are reserved for control), 4 E1 lines form a 8.44bMbps E2 line, etc.



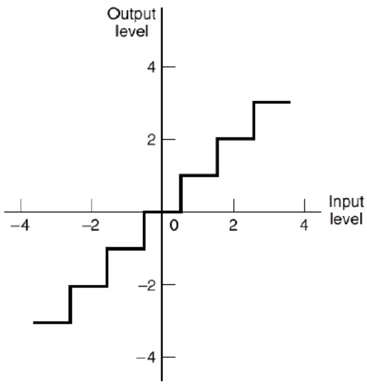
3.6 Quantization Process

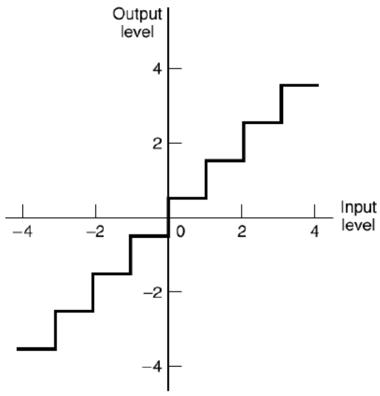
Amplitude *quantization* is defined as *the process of transforming the sample amplitude $m(nT_s)$ of a message signal $m(t)$ at time $t = nT_s$ into a discrete amplitude $v(nT_s)$ taken from a finite set of possible amplitudes.*



3.6 Quantization Process

Midtread **Midrise**

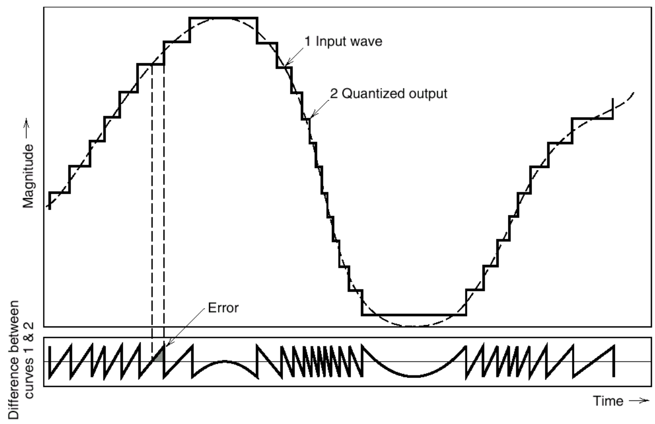




19

3.6 Quantization Noise

The use of quantization introduces an error defined as the difference between the input signal *m* and the output signal *v.* The error is called *quantization noise.*



20

3.6 Quantization Noise

- Let the quantizer input *m* be the sample value of a zero-mean random variable *M*. *(If the input has a nonzero mean, we can always remove it by subtracting the mean from the input and then adding it back after quantization.)*
- A quantizer g(.) maps the input random variable *M* of continuous amplitude into a discrete random variable V.
  - their respective sample values *m* and *v* are related by v=g(m)
- Let the quantization error be denoted by the random variable *Q* of sample value *q*.
- We may thus write *q=m-v* or *Q=M-V*

21

3.6 Quantization Noise

- With the input M having zero mean, and the quantizer assumed to be symmetric, it follows that the quantizer output V and therefore the quantization error *Q*, will also have zero mean.
- Thus for a partial statistical characterization of the quantizer in terms of output **signal-to-(quantization) noise ratio**, we need only find the mean-square value of the quantization error *Q*.

22

3.6 Quantization Noise

- Consider then an input *m* of continuous amplitude in the range $(-m_{max}, m_{max})$.
- Assuming a uniform quantizer of the midrise type with ***L*** levels, we find that the step-size of the quantizer is given by:

$$\Delta = \frac{2m_{\max}}{L}$$

- For a uniform quantizer, the quantization error *Q* will have its sample values bounded by:

$$-\frac{\Delta}{2} \leq q \leq \frac{\Delta}{2}$$

3.6 Quantization Noise

- If the step-size is sufficiently small *(i.e., the number of representation levels L is sufficiently large)*, it is reasonable to assume that the quantization error *Q* is a *uniformly distributed* random variable.

$$f_Q(q) = \begin{cases} \frac{1}{\Delta}, & -\frac{\Delta}{2} < q \leq \frac{\Delta}{2} \\ 0, & \text{otherwise} \end{cases}$$

- With the mean of the quantization error being zero, its variance σ^2_Q is the same as the mean-square value:

$$\sigma_Q^2 = E[Q^2]$$
$$= \int_{-\Delta/2}^{\Delta/2} q^2 f_Q(q)\, dq$$

3.6 Quantization Noise

- The quantization noise variance will be:

$$\sigma_Q^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} q^2\, dq = \frac{\Delta^2}{12}$$

- Typically, the L-ary number *k,* denoting the k^{th} representation level of the quantizer, is transmitted to the receiver in binary form.
- Let R denote the number of *bits per sample* used in the construction of the binary code. We may then write $\mathbf{L=2^R}$ or equivalently, $\mathbf{R=\log_2 L}$

25

3.6 Quantization Noise

- The step size is:

$$\Delta = \frac{2m_{\max}}{L} = \frac{2m_{\max}}{2^R}$$

- Thus, noise variance (power) will be:

$$\sigma_Q^2 = \frac{1}{3} m_{\max}^2 2^{-2R}$$

- Let *P* denote the average power of the message signal *m(t).* We may then express the *output signal-to-noise ratio of* a uniform quantizer as:

$$(\mathrm{SNR})_O = \frac{P}{\sigma_Q^2} = \left(\frac{3P}{m_{\max}^2}\right) 2^{2R}$$

26

3.6 Quantization Noise

- the output signal-to-noise ratio of the quantizer increases ***exponentially*** with increasing number of bits per sample, *R.* Recognizing that an increase in *R* requires a proportionate increase in the channel (transmission) bandwidth B_T.
- There is a fundamental tradeoff between quantization noise and channel bandwidth.

27

3.6 Quantization Noise

Example: Sinusoidal Modulating Signal

- Consider the special case of a full-load sinusoidal modulating signal of amplitude A_m. Using a quantizer with L levels and R bits for each level.
- Find the average signal power, the noise power and the signal to noise ratio.
- The average signal power is $P = \frac{A_m^2}{2}$
- The quantization noise power is $\sigma_Q^2 = \frac{1}{3}A_m^2 2^{-2R}$
- The output signal-to-noise ratio is $(\mathrm{SNR})_O = \frac{A_m^2/2}{A_m^2 2^{-2R}/3} = \frac{3}{2}(2^{2R})$

28

3.6 Quantization Noise

Example: Sinusoidal Modulating Signal

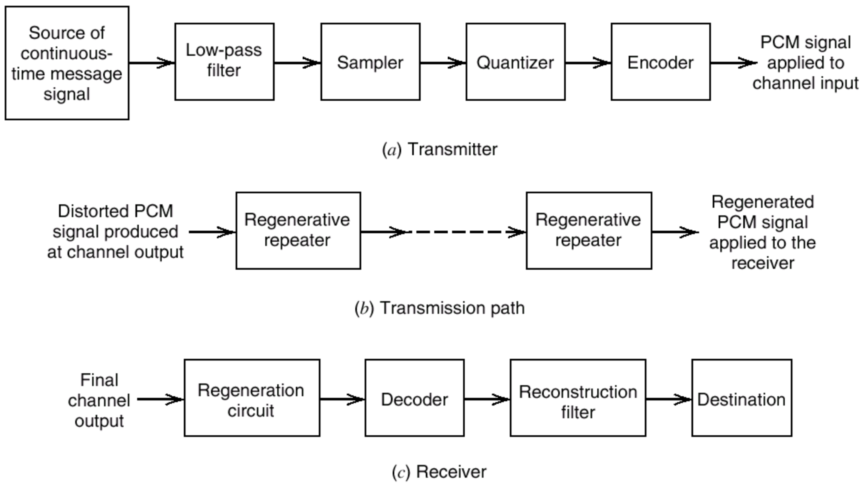
- Expressing the signal-to-noise ratio in decibels, we get

$$SNR_{dB}=10\log_{10}(SNR)_0=1.8+6R$$

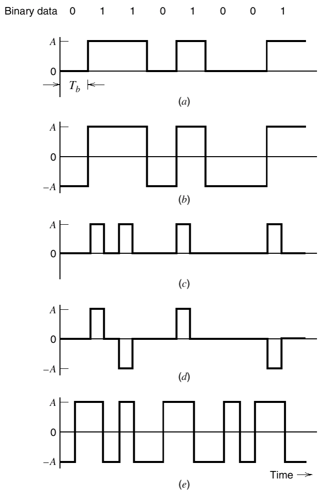
TABLE 3.1 *Signal-to-(quantization) noise ratio for varying number of representation levels for sinusoidal modulation*

Number of Representation Levels, L	*Number of Bits per Sample, R*	*Signal-to-Noise Ratio (dB)*
32	5	31.8
64	6	37.8
128	7	43.8
256	8	49.8

3.7 Pulse-Code Modulation



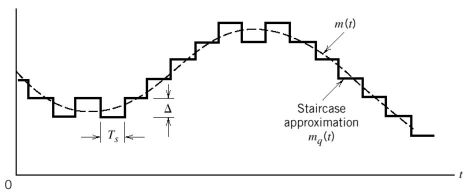
3.7 Pulse-Code Modulation



Line codes for the electrical representations of binary data.
(*a*) Unipolar NRZ signaling.
(*b*) Polar NRZ signaling.
(*c*) Unipolar RZ signaling.
(*d*) Bipolar RZ signaling.
(*e*) Split-phase or Manchester code.

3.12 Delta Modulation

- In *delta modulation** (DM), an incoming message signal is oversampled (i.e., at a rate much higher than the Nyquist rate) to purposely increase the correlation between adjacent samples of the signal. This is done to permit the use of a simple quantizing strategy for constructing the encoded signal

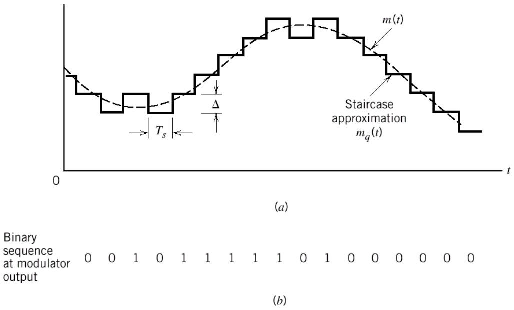


(*a*)

Binary sequence at modulator output 0 0 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0

(*b*)

3.12 Delta Modulation



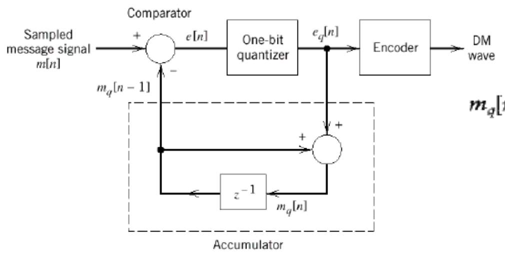
$$e[n] = m[n] - m_q[n-1]$$
$$e_q = \Delta \operatorname{sgn}(e[n])$$
$$m_q[n] = m_q[n-1] + e_q[n]$$

where *e[n]* is an *error signal* representing the difference between the present sample *m[n]* of the input signal and the latest approximation $m_q[n — 1]$ to it,
$e_q[n]$ is the quantized version of *e[n],*
and sgn(.) is the signum function.
Finally, the quantizer output $m_q[n]$ is coded to produce the DM signal.

33

3.12 Delta Modulation

- At the sampling instant nT_s , the accumulator increments the approximation by a step A in a positive or negative direction, depending on the algebraic sign of the error sample *e[n].*
- If the input sample *m[n]* is greater than the most recent approximation $m_q[n]$, a positive increment +A is applied to the approximation.
- If, on the other hand, the input sample is smaller, a negative increment —A is applied to the approximation.
- In this way, the accumulator does the best it can to track the input samples by one step (of amplitude +A or —A) at a time.

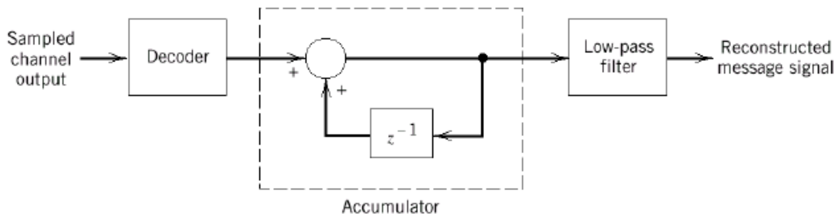


$$m_q[n] = \Delta \sum_{i=1}^{n} \operatorname{sgn}(e[i])$$
$$= \sum_{i=1}^{n} e_q[i]$$

34

3.12 Delta Modulation

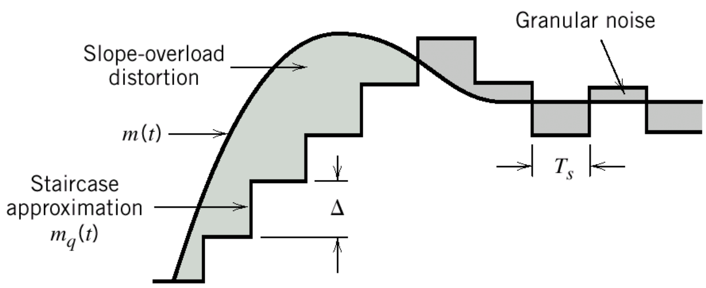
- The staircase approximation $m_q(t)$ is reconstructed by passing the sequence of positive and negative pulses, produced at the decoder output, through an accumulator in a manner similar to that used in the transmitter.
- The out-of- band quantization noise in the high-frequency staircase waveform $m_q(t)$ is rejected by passing it through a low-pass filter with a bandwidth equal to the original message bandwidth.



35

3.12 Delta Modulation

- Delta modulation is subject to two types of quantization error: slope overload distortion and granular noise



36

3.12 Delta Modulation

- **Slope overload distortion**: step-size A is too small for the staircase approximation $m_q(t)$ to follow a steep segment of the input waveform $m(t)$.

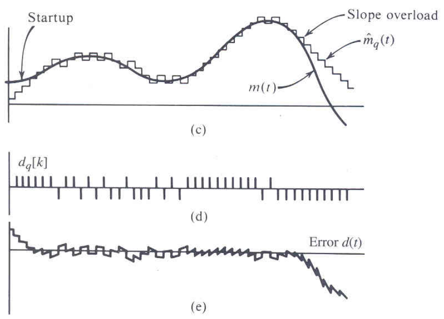


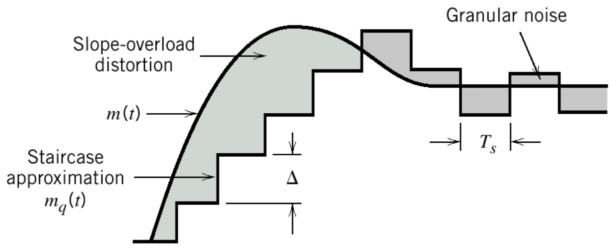
Figure 6.20 Delta modulation.

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

37

3.12 Delta Modulation

- **Granular noise:** occurs when the step size A is too large relative to the local slope characteristics of the input waveform $m(t)$.



38

3.12 Delta Modulation

We thus see that there is a need to have a large step-size to accommodate a wide dynamic range, whereas a small step size is required for the accurate representation of relatively low-level signals.

To satisfy such a requirement, we need to make the delta modulator "adaptive," in the sense that the step size is made to vary in accordance with the input signal.

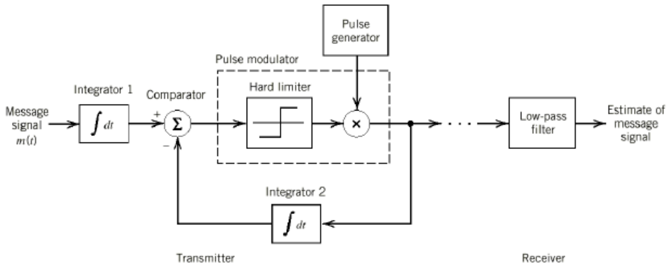
39

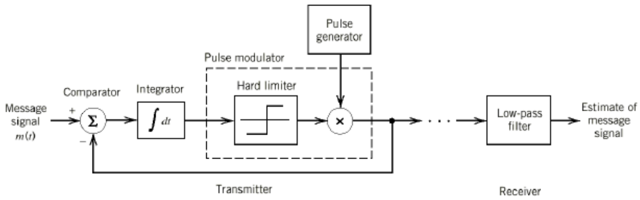
3.12 Delta-Sigma Modulation

- The quantizer input in the conventional form of delta modulation may be viewed as an approximation to the ***derivative*** of the incoming message signal.
- This behavior leads to a drawback of delta modulation in that transmission disturbances such as noise result in an accumulative error in the demodulated signal.
- This drawback can be overcome by ***integrating*** the message signal prior to delta modulation.
  - The low-frequency content of the input signal is pre-emphasized.
  - Correlation between adjacent samples of the delta modulator input is increased which tends to improve overall system performance by reducing the variance of the error signal at the quantizer input.
  - Design of the receiver is simplified.

40

3.12 Delta-Sigma Modulation





41

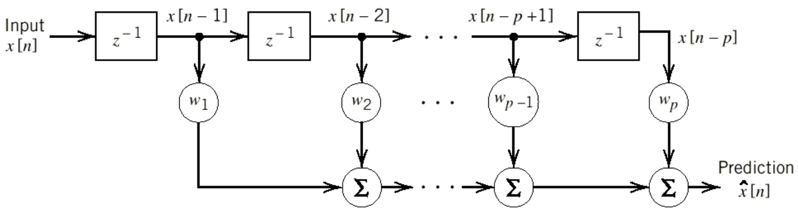
3.13 Linear Prediction

- In delta modulation, simplicity of implementations of both the transmitter and receiver is attained by using a sampling rate far in excess of that needed for pulse-code modulation.
- The price paid for this benefit is a corresponding increase in the transmission and therefore channel bandwidth.
- We may wish to trade increased system complexity for a reduced channel bandwidth.
- A signal-processing operation basic to the attainment of this latter design objective is prediction.

42

3.13 Linear Prediction

- Consider a *finite-duration impulse response (FIR) discrete-time filter* which involves the use of three functional blocks:
  - Set of p unit-delay elements, each of which is represented by z^{-1}.
  - Set of multipliers involving the filter coefficients $w_1, w_2, \ldots, w_p$.
  - Set of "adders" used to sum the scaled versions of the delayed inputs $x[n-1]$, $x[n-2]$, ..., $x[n-p]$ to produce the output $\hat{x}[n]$
- The filter output $\hat{x}[n]$ or more precisely, the *linear prediction* of the input, is thus defined by the *convolution sum* $\hat{x}[n] = \sum_{k=1}^{p} w_k x[n-k]$



43

3.13 Linear Prediction

- The ***prediction error***, denoted by $e[n]$, is defined as the difference between $x[n]$ and the prediction $\hat{x}[n]$

$$e[n] = x[n] - \hat{x}[n]$$

- The design objective is to choose the filter coefficients w_1, $w_2, \ldots, w_p$ so as to minimize an *index of performance, **J***, defined as the mean-square error:

$$J = E\left[e^2[n]\right]$$

44

3.13 Linear Prediction

- Expanding the mean-squared error:

$$J = E[x^2[n]] - 2\sum_{k=1}^{p} w_k E[x[n]x[n-k]] + \sum_{j=1}^{p}\sum_{k=1}^{p} w_j w_k E[x[n-j]x[n-k]]$$

- We assume that the input signal *x(t)* is the sample function of a stationary process *X(t)* of zero mean; that is, E[x[n]] is zero for all *n.* Define

$$\begin{aligned}\sigma_X^2 &= \text{variance of a sample of the process } X(t) \text{ at time } nT_s \\ &= E[x^2[n]] - (E[x[n]])^2 \\ &= E[x^2[n]]\end{aligned}$$

$$\begin{aligned}R_X(kT_s) &= \text{autocorrelation of the process } X(t) \text{ for a lag of } kT_s \\ &= R_X[k] \\ &= E[x[n]x[n-k]]\end{aligned}$$

45

3.13 Linear Prediction

- Accordingly, we may rewrite **J** in the simplified form:

$$J = \sigma_X^2 - 2\sum_{k=1}^{p} w_k R_X[k] + \sum_{j=1}^{p}\sum_{k=1}^{p} w_j w_k R_X[k-j]$$

- Differentiating the index of performance **J** with respect to the filter coefficient w_k and setting the result equal to zero, and then rearranging terms, we obtain:

$$\sum_{j=1}^{p} w_j R_X[k-j] = R_X[k] = R_X[-k], \qquad k = 1, 2, \ldots, p$$

- The optimality equations are called the *Wiener-Hopf equations* for linear prediction.

46

3.13 Linear Prediction

Wiener-Hopf Equations in Matrix Form:

$$\mathbf{w}_o = p\text{-by-1 optimum coefficient vector} = [w_1, w_2, \ldots, w_p]^T$$

$$\mathbf{r}_X = p\text{-by-1 autocorrelation vector} = [R_X[1], R_X[2], \ldots, R_X[p]]^T$$

$$\mathbf{R}_X = p\text{-by-}p\text{ autocorrelation matrix} = \begin{bmatrix} R_X[0] & R_X[1] & \cdots & R_X[p-1] \\ R_X[1] & R_X[0] & \cdots & R_X[p-2] \\ \vdots & \vdots & & \vdots \\ R_X[p-1] & R_X[p-2] & \cdots & R_X[0] \end{bmatrix}$$

We may thus simplify the set of equations as $\mathbf{R}_X \mathbf{w}_o = \mathbf{r}_X$

47

3.13 Linear Prediction

- We assume that the autocorrelation matrix $\mathbf{R}_x$ is *nonsingular,* so that its inverse exists.
- We may then solve for the coefficient vector w_0 by multiplying both sides of this equation by the *inverse matrix* $\mathbf{R}_x^{-1}$ obtaining the optimum solution:

$$\mathbf{w}_o = \mathbf{R}_X^{-1}\mathbf{r}_X$$

- The minimum mean-square value of the prediction error is:

$$J_{\min} = \sigma_X^2 - \mathbf{r}_X^T\mathbf{R}_X^{-1}\mathbf{r}_X$$

- The quadratic term $\mathbf{r}_x^T\mathbf{R}_x^{-1}\mathbf{r}_x$ is always positive

48

3.13 Linear Adaptive Prediction

- Calculating the weight vector of a linear predictor requires knowledge of the autocorrelation function $R_x[k]$ of the input sequence $[x[n]]$ for lags $k = 0,1,. . . ,p,$ where p is the prediction order.
- What if knowledge of $R_x[k]$ for varying k is not available? In these situations, which occur frequently in practice, we may resort to the use of an *adaptive predictor.*
- The predictor is adaptive in the following sense:
  - Computation of the tap weights w_k, $k = 1,2,..., p,$ proceeds in a "recursive" manner, starting from some arbitrary initial values of the tap weights.
  - The algorithm used to adjust the tap weights (from one iteration to the next) is "self-designed," operating solely on the basis of available data.

49

3.13 Linear Adaptive Prediction

- The aim of the algorithm is to find the minimum point of the bowl-shaped *error surface* that describes the dependence of the cost function **J** on the tap weights. It is therefore intuitively reasonable that successive adjustments to the tap-weights of the predictor be made in the direction of the steepest descent of the error surface, that is, in a direction opposite to the ***gradient vector*** whose elements are defined by:

$$g_k = \frac{\partial J}{\partial w_k}, \qquad k = 1, 2, \ldots, p$$

- This is the idea behind the ***method of steepest descent.***

50

3.13 Linear Adaptive Prediction

- Let $w_k[n]$ denote the value of the k^{th} tap-weight at iteration *n.* Then the updated value of this weight at iteration *n* + 1 is defined by:

$$w_k[n+1] = w_k[n] - \frac{1}{2}\mu g_k, \qquad k = 1, 2, \dots, p$$

- Where μ is a *step-size parameter* that controls the speed of adaptation, and the factor *1/2* is included for convenience of presentation.
- Differentiating the cost function **J** with respect to w_k, we readily find that:

$$\begin{aligned} g_k &= -2R_X[k] + 2\sum_{j=1}^{p} w_j R_X[k-j] \\ &= -2E[x[n]x[n-k]] + 2\sum_{j=1}^{p} w_j E[x[n-j]x[n-k]], \qquad k = 1, 2, \dots, p \end{aligned}$$

51

3.13 Linear Adaptive Prediction

- Thus, the estimate of g_k at iteration *n* is:

$$\hat{g}_k[n] = -2x[n]x[n-k] + 2\sum_{j=1}^{p} w_j[n]x[n-j]x[n-k], \qquad k = 1, 2, \dots, p$$

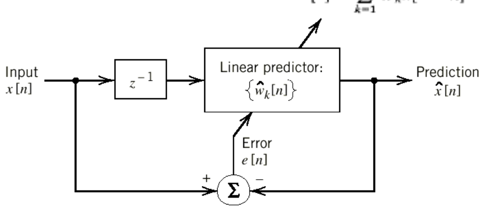
- The adaptive tap weights will be updated as:

$$\begin{aligned} \hat{w}_k[n+1] &= \hat{w}_k[n] + \mu x[n-k]\left(x[n] - \sum_{j=1}^{p} \hat{w}_j[n]x[n-j]\right) \\ &= \hat{w}_k[n] + \mu x[n-k]e[n], \qquad k = 1, 2, \dots, p \end{aligned}$$

- where *e[n]* is the *prediction error* defined as:

$$e[n] = x[n] - \sum_{j=1}^{p} \hat{w}_j[n]x[n-j]$$

$$\hat{x}[n] = \sum_{k=1}^{p} w_k x[n-k]$$



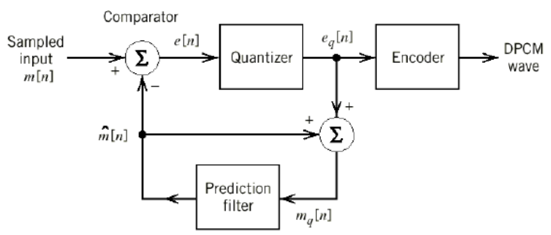
3.14 Differential Pulse-Code Modulation

- When a voice or video signal is sampled at a rate slightly higher than the Nyquist rate as usually done in pulse-code modulation, the resulting sampled signal is found to exhibit a high degree of correlation between adjacent samples.
- The meaning of this high correlation is that, in an average sense, the signal does not change rapidly from one sample to the next, and as a result, the difference between adjacent samples has a variance that is smaller than the variance of the signal itself.
- When these highly correlated samples are encoded, as in the standard PCM system, the resulting encoded signal contains *redundant information.* This means that symbols that are not absolutely essential to the transmission of information are generated as a result of the encoding process.
- By removing this redundancy before encoding, we obtain a more efficient coded signal, which is the basic idea behind differential pulse-code modulation.

53

3.14 Differential Pulse-Code Modulation

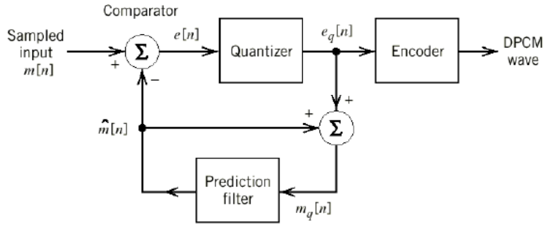
- Now if we know the past behavior of a signal up to a certain point in time, we may use prediction to make an estimate of a future value of the signal.
- Suppose then a baseband signal *m(t)* is sampled at the rate $f_s = 1/T_S$ to produce the sequence *m[n]* whose samples are T_s seconds apart. The fact that it is possible to predict future values of the signal *m(t)* provides motivation for the *differential quantization* scheme.
- In this scheme, the input signal to the quantizer is defined by $e[n] = m[n] - \hat{m}[n]$



54

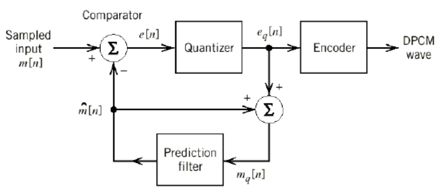
3.14 Differential Pulse-Code Modulation (DPCM)

- The quantizer output may be expressed as $e_q[n] = e[n] + q[n]$
- The prediction-filter input is $m_q[n] = \hat{m}[n] + e_q[n]$
- Then $m_q[n] = \hat{m}[n] + e[n] + q[n]$
- Since $\hat{m}[n] + e[n] = m[n]$, we get $m_q[n] = m[n] + q[n]$



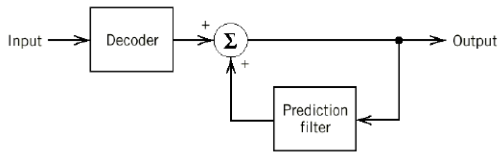
3.14 Differential Pulse-Code Modulation (DPCM)

- $m_q[n] = m[n] + q[n]$ represents a quantized version of the input sample *m[n].*
- That is, irrespective of the properties of the prediction filter, the quantized sample $m_q[n]$ at the prediction filter input differs from the original input sample *m[n]* by the quantization error *q[n].*
- Accordingly, if the prediction is good, the variance of the prediction error *e[n]* will be smaller than the variance of *m[n],* so that a quantizer with a given number of levels can be adjusted to produce a quantization error with a smaller variance than would be possible if the input sample *m[n]* were quantized directly as in a standard PCM system.



3.14 DPCM Receiver

- The receiver for reconstructing the quantized version of the input consists of a decoder to reconstruct the quantized error signal.
- The quantized version of the original input is reconstructed from the decoder output using the same prediction filter used in the transmitter.
- In the absence of channel noise, we find that the encoded signal at the receiver input is identical to the encoded signal at the transmitter output.
- Accordingly, the corresponding receiver output is equal to $m_q[n]$ which differs from the original input *m[n]* only by the quantization error *q[n]* incurred as a result of quantizing the prediction error *e[n].*



57

3.14 DPCM vs. Delta Modulation

- What is the difference between DPCM and Delta Modulation ?
- Differential pulse-code modulation includes delta modulation as a special case.
- In particular, comparing the DPCM system the DM system, we see that they are basically similar, except for two important differences:
  - the use of a one-bit (two-level) quantizer in the delta modulator
  - the replacement of the prediction filter by a single delay element (i.e., zero prediction order).
- Thus, DM is the 1-bit version of DPCM.

58

3.14 DPCM

The output signal-to-noise ratio of the DPCM system is:

$$SNR_O = \frac{\sigma_M^2}{\sigma_Q^2}$$

Where σ^2_M is the variance of the original input sample *m[n],* assumed to be of zero mean, and σ^2_Q is the variance of the quantization error *q[n].*

We may rewrite the SNR to be:

$$SNR_O = \left(\frac{\sigma_M^2}{\sigma_E^2}\right)\left(\frac{\sigma_E^2}{\sigma_Q^2}\right) = G_P\left(SNR_Q\right)$$

where σ^2_E is the variance of the prediction error AND G_p is the ***processing gain*** produced by the differential quantization scheme