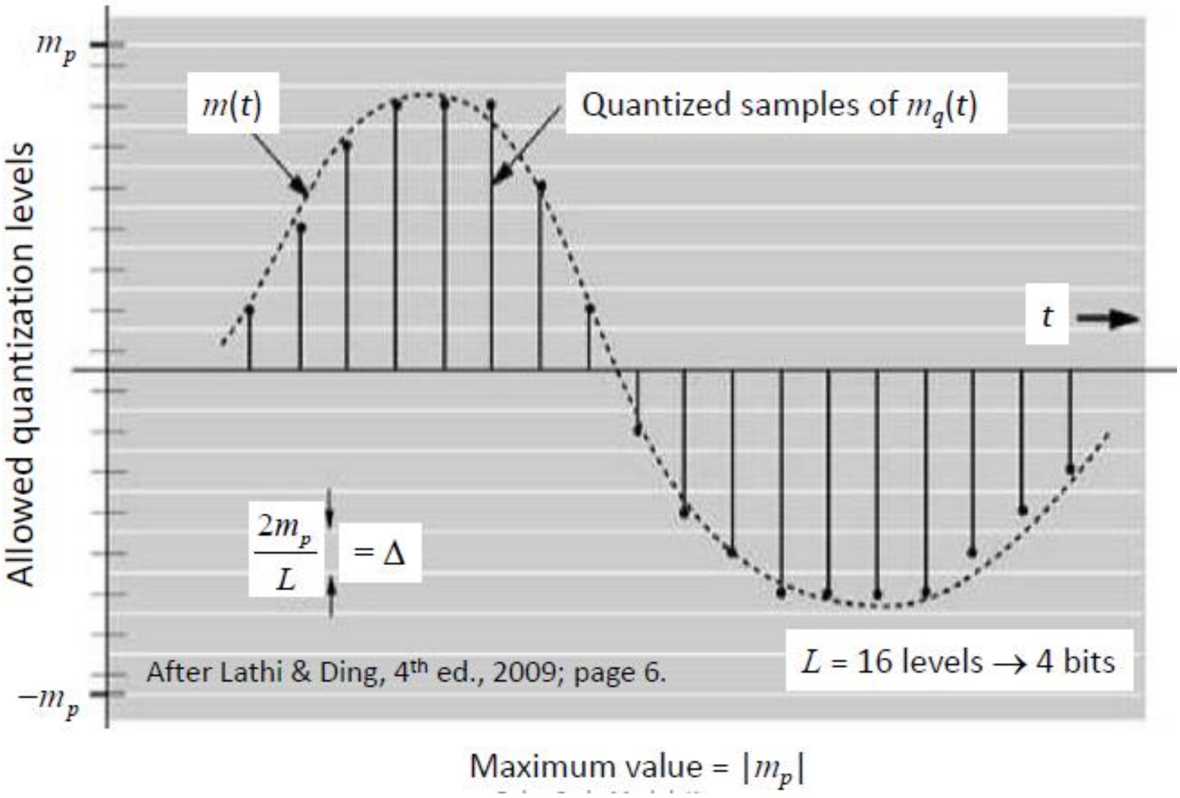


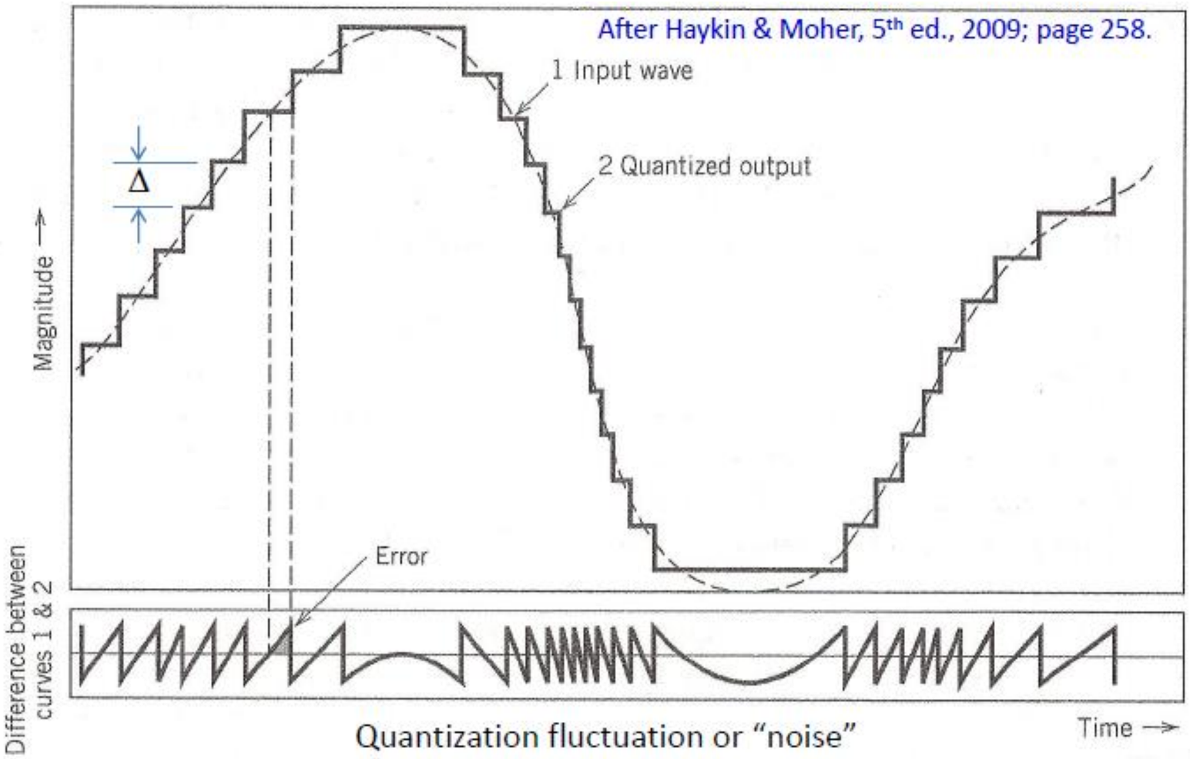
Fig. 2.5. Quantization

We start with a sampled signal {call it m (t)} and now we want to quantize
The quantized amplitude is limited to a range, say from - m_p to + m_p .
(Note: the range of m (t) may extend beyond – m_p , + m_p) in some cases.
Divide the range (-m_p , +m_p) into L uniformly spaced intervals. The number intervals is L and the separation between quantized levels is:

$$\Delta = \frac{2m_p}{L} \tag{2.1}$$

The k^{th} sample point of m (t) is designated as $m(kT_S)$ and is assigned a value equal to the midpoint between two adjacent levels. Define:
$m(kT_S)$ = k^{th} sample's value, and
$m_q(kT_S)$ = k^{th} quantized sample's value.
Then the quantization error $q(kT_S)$ is equal to $m_q(kT_S) - m(kT_S)$

Error Generated by Quantization (Quantization Noise)
Quantization noise $q(t) = m(t) - m_q(t)$



2.2 Differential Pulse Code Modulation (DPCM)

PCM is not really efficient because it generates so many bits taking up a lot of bandwidth. Can we improve on this? YES.
Suppose we have a slowly varying signal m(t), then we exploit this by using the difference between two adjacent samples. This will form the basis of differential pulse code modulation (DPCM).
Let m(k) be the k^{th} sample reading of signal m(t).
Then we can express the difference between two adjacent samples as:

$$d(k) = m(k) - m(k-1) \quad (2.2)$$

Principle: Instead of transmitting m(k), we transmit d(k)

At the receiver knowing d(k) and the previous value of m(k – 1) allows us to construct the value of m(k)
How do we benefit from doing this?
The difference of successive samples almost always is much smaller than the full range of the sample values of m(t) (full range covers - m_p to + m_p).
We use this fact to improve upon the efficiency of PCM by requiring fewer bits.
Furthermore, we can make use of the estimate of m(k), denoted by m_{est} (k).
We use previous sample values of m(t) to make this.
Suppose
m_{est} (k) is the estimate of the k^{th} sample, then the difference d(k) is defined by:

$$d(k) = m(k) - m_{est}(k) \tag{2.3}$$

and it is the difference d(k) that is transmitted.

Receiver Concept:
At the receiver we determine the estimate m_{est}(k) from previous sample values, and then generate m(k) by adding the received d(k) values to the estimate m_{est} (k).
Thus, the reconstruction of the samples is done iteratively.

Example: Simple example of DPCM
Code the following value sequence:
1.4 1.75 2.05 2.5 2.4, quantization step= 0.2.
Solution:

-Error = 1.75 – 1.4 = 0.35 ≈ 0.4
-Prediction value = 1.4 + 0.4 = 1.8

-Error = 2.05 – 1.8 ≈ 0.2
-Prediction value = 1.8 + 0.2 = 2.0

-Error = 2.5 – 2 = 0.5 ≈ 0.4
-Prediction value = 2 + 0.4 = 2.4

-Error = 2.4 – 2.4 = 0
-Prediction value 2.4 + 0 = 2.4
DPCM sends 1.4 1.8 2.0 2.4 2.4