Basics of Digital Audio

Lecturer：Jun Xiao
(肖俊）
College of Software and Technology

Content

- Digitization of Sound
- MIDI： Music Instrument Digital Interface
- Quantization and Transmission of Audio

1、Digitization of Sound

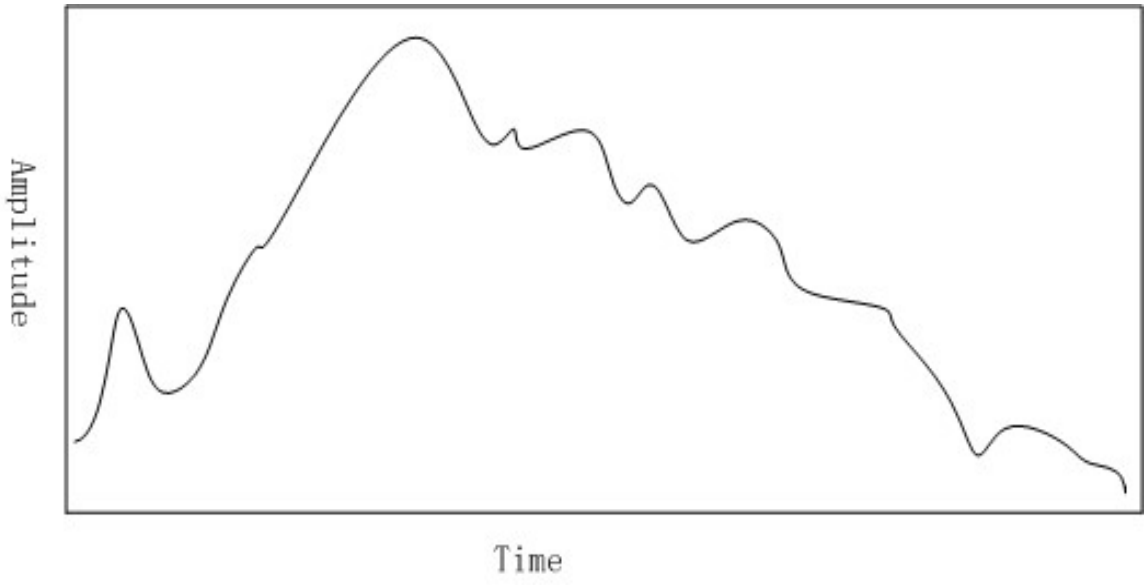
- What Is Sound ?
- Digitization
- Nyquist Theorem
- Signal-to-Noise Ratio （SNR）
- SQNR (Signal-to-Quantization-Noise Ratio)
- Linear and Nonlinear Quantization
- Audio Filtering
- Audio Quality versus Data Rate
- Synthetic Sounds

1.1 What Is Sound?

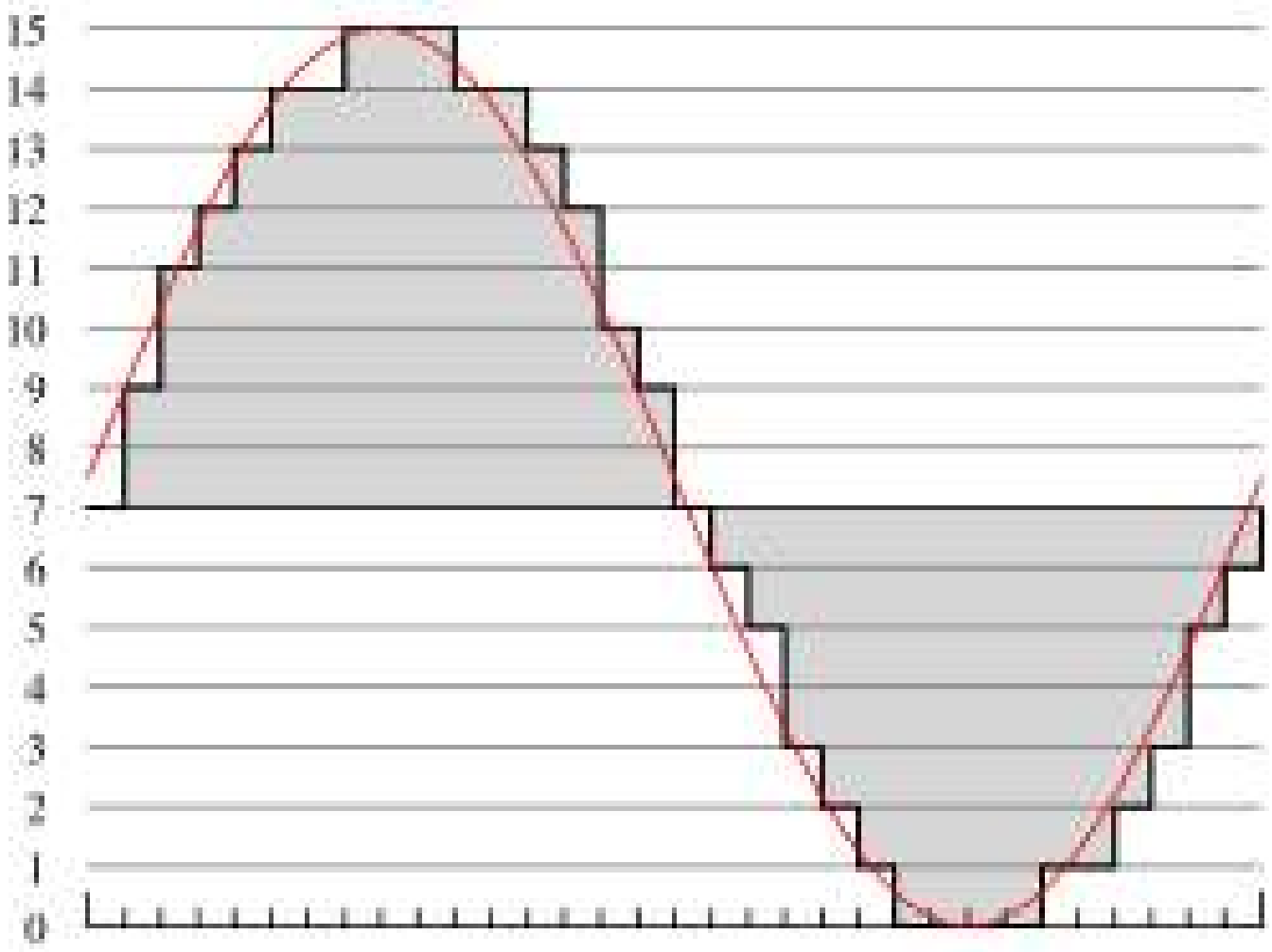
- Sound is a wave phenomenon like light
  - No air -- No sound
  - Sound is a pressure wave, taking on continuous values
- Sound has ordinary wave properties and behaviors
  - Reflection
  - Refraction
  - Diffraction
- Sound can be measured by converting pressure to voltage levels

1.2 Digitization

- Digitization means conversion to a stream of numbers, and preferably these numbers should be integers for efficiency.



An analog signal: continuous measurement of pressure wave


15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0

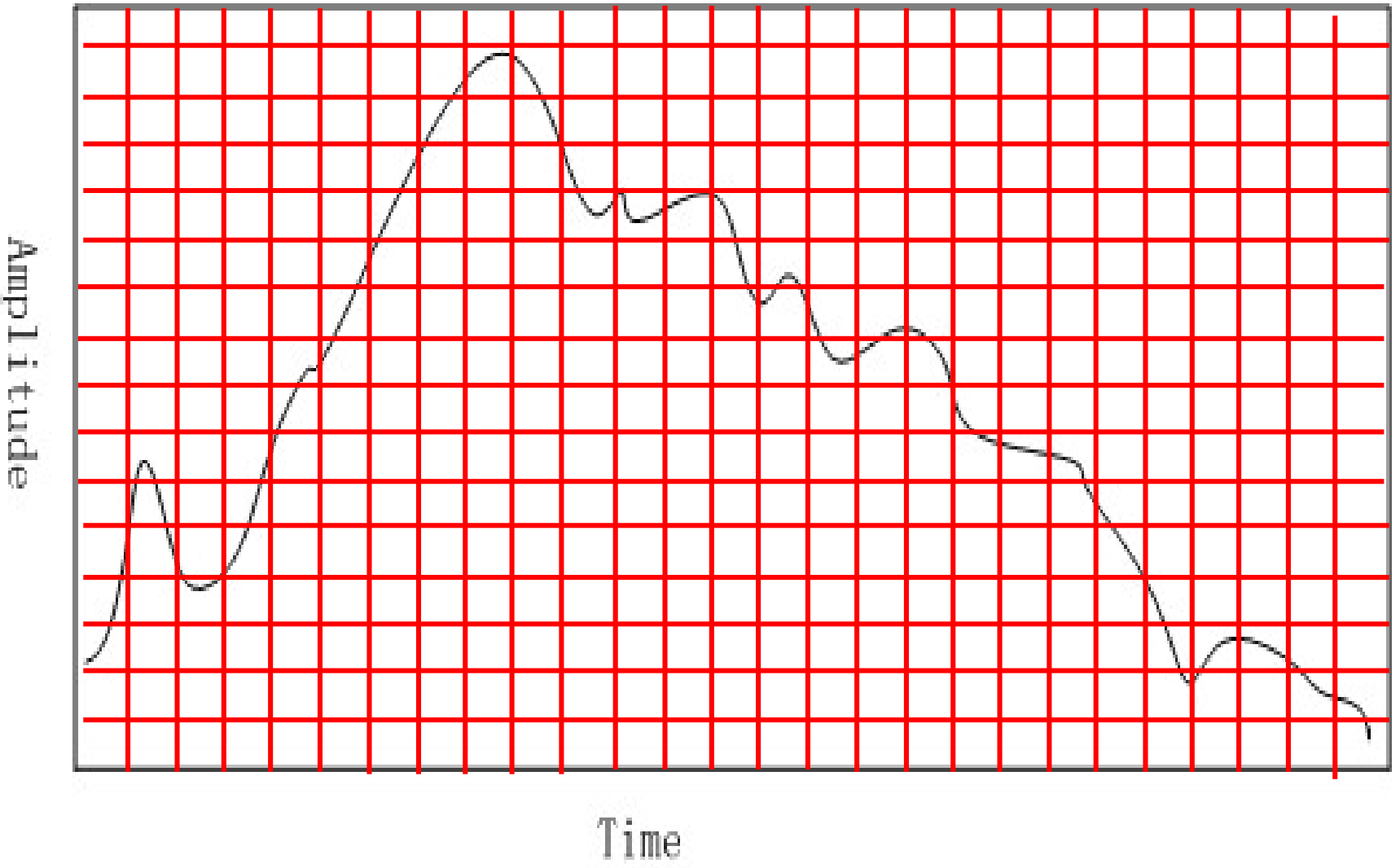
1.2 Digitization

- Amplitude -- continuous value and changes over time
  - Sampling both in time and amplitude dimensions

- Time dimension: sampling at evenly spaced intervals
  - Typical range: 8kHz to 48kHz,
  - Human can hear from 20Hz to 20kHz

- Quantization: Sampling in the amplitude dimension
  - Uniform sampling: Equally spaced sampling ;
  - Nonuniform sampling, like *u*-law rule
  - Typical uniform quantization rates:
    - 8-bit, 256 levels
    - 16-bit, 65,536 levels

1.2 Digitization



1.2 Digitization

- Thus to decide how to digitize audio data we need to answer the following questions:

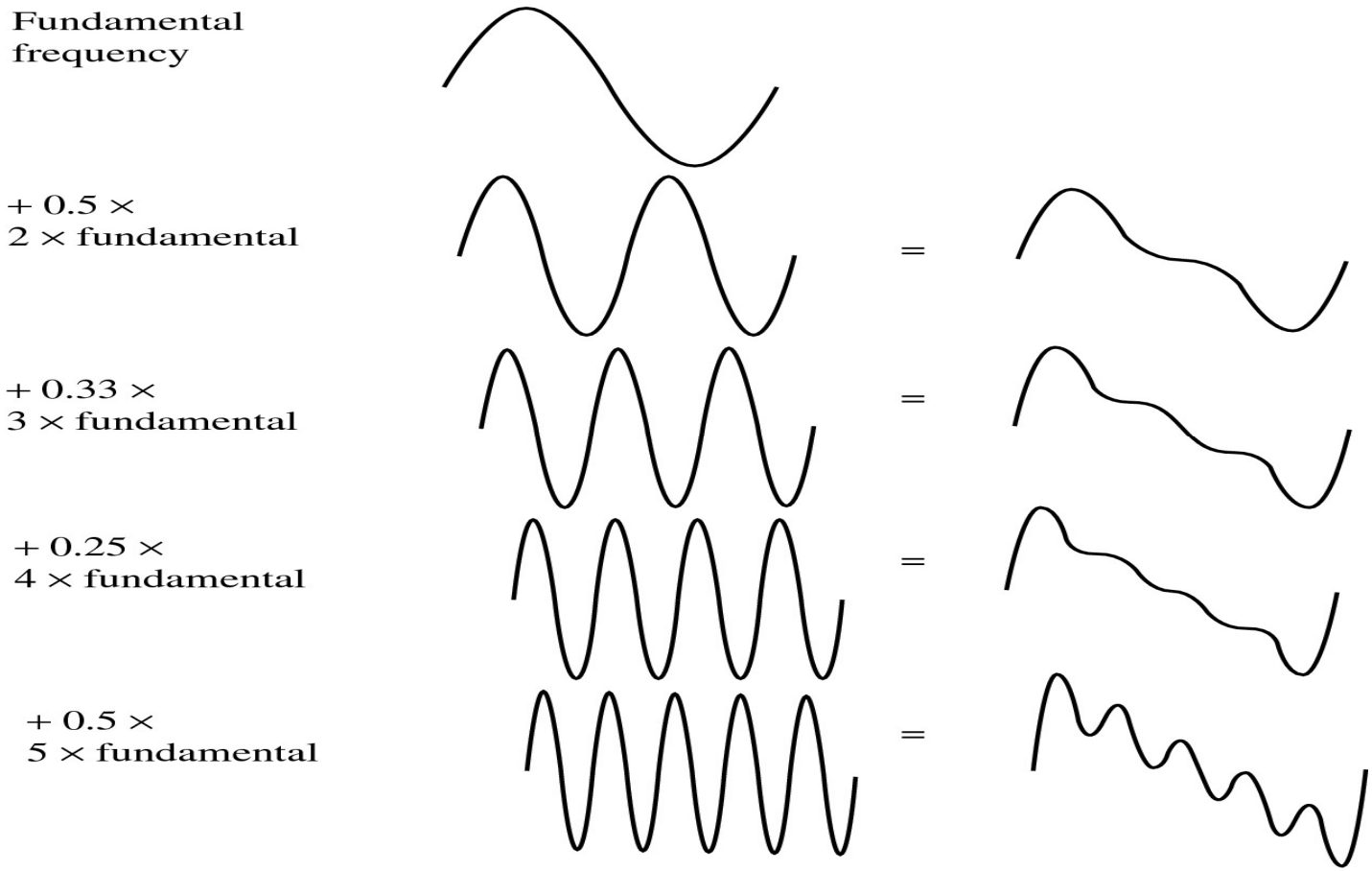
1. What is the sampling rate?

2. How finely is the data to be quantized, and is quantization uniform?

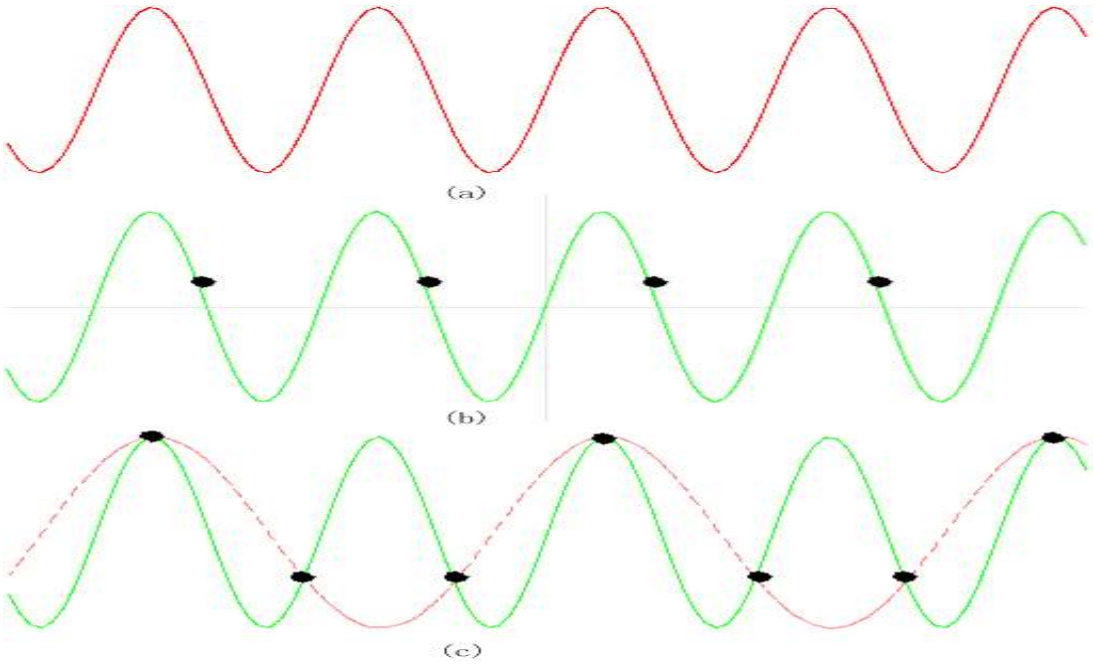
3. How is audio data formatted? (file format)

1.3 Nyquist Theorem

- Signals can be decomposed into a sum of sinusoids



1.3 Nyquist Theorem



(a) A single sinusoid
(b) Constant detected by sampling with actual frequency
(c) Alias obtained by sampling with 1.5 times the frequency

1.3 Nyquist Theorem

- Nyquist rate
  - For correct sampling, sampling rate must be at least twice the maximum frequency content in the signal
- Nyquist Theorem
  - A band-limited signal, which has a lower limit f1 and an upper limit f2 of frequency, we need a sampling rate of at least 2(f2-f1)
- Nyquist frequency: Half the Nyquist rate
  - Since it would be impossible to recover frequencies higher than Nyquist frequency in any event, most systems have an antialiasing filter that restricts the frequency content in the input to the sampler to a range at or below Nyquist frequency.

1.4 Signal-to-Noise Ratio（SNR）

- The ratio of the power of the correct signal and the noise is called the signal to noise ratio (SNR) — a measure of the quality of the signal.
- The SNR is usually measured in decibels (dB), where 1 dB is a tenth of a bel. The SNR value, in units of dB, is defined in terms of base-10 logarithms of squared voltages, as follows:

$$SNR = 10\log_{10}\frac{V_{signal}^2}{V_{noise}^2} = 20\log_{10}\frac{V_{signal}}{V_{noise}}$$

1.4 Signal-to-Noise Ratio（SNR）

- The power in a signal is proportional to the square of the voltage. For example, if the signal voltage V_{signal} is 10 times the noise, then the SNR is $20 * \log_{10}(10) = 20dB$.
- In terms of power, if the power from ten violins is ten times that from one violin playing, then the ratio of power is 10dB, or 1B.
- To know: Power — 10; Signal Voltage — 20.

1.4 Signal-to-Noise Ratio（SNR）

Threshold of hearing	0
Rustle of leaves	10
Very quiet room	20
Average room	40
Conversation	60
Busy street	70
Loud radio	80
Train through station	90
Riveter	100
Threshold of discomfort	120
Threshold of pain	140
Damage to ear drum	160

The usual levels of sound we hear around us are described in terms of decibels, as a ratio to the quietest sound we are capable of hearing.

1.5 SQNR

- Aside from any noise that may have been present in the original analog signal, there is also an additional error that results from quantization.
  - (a) If voltages are actually in 0 to 1 but we have only 8 bits in which to store values, then effectively we force all continuous values of voltage into only 256 different values.
  - (b) This introduces a roundoff error. It is not really "noise". Nevertheless it is called quantization noise (or quantization error).

1.5 SQNR

- The quality of the quantization is characterized by the Signal to Quantization Noise Ratio (SQNR).
  - (a) Quantization noise: the difference between the actual value of the analog signal, for the particular sampling time, and the nearest quantization interval value.
  - (b) At most, this error can be as much as half of the interval.
  - (c) For a quantization accuracy of N bits per sample, the SQNR can be simply expressed:

$$SQNR = 20\log_{10}\frac{V_{signal}}{V_{quan\_noise}} = 20\log_{10}\frac{2^{N-1}}{\frac{1}{2}}$$

$$= 20 \times N \times \log 2 = 6.02\,N(\text{dB})$$

1.5 SQNR

(a) We map the maximum signal to $2^{N-1} - 1$ ($\simeq 2^{N-1}$) and the most negative signal to -2^{N-1}.

(b) Eq. (6.3) is the *Peak* signal-to-noise ratio, PSQNR: peak signal and peak noise.

(c) The *dynamic range* is the ratio of maximum to minimum absolute values of the signal: V_{max}/V_{min}. The max abs. value V_{max} gets mapped to $2^{N-1} - 1$; the min abs. value V_{min} gets mapped to 1. V_{min} is the smallest positive voltage that is not masked by noise. The most negative signal, $-V_{max}$, is mapped to -2^{N-1}.

(d) The quantization interval is $\Delta V=(2V_{max})/2^N$, since there are 2^N intervals. The whole range V_{max} down to $(V_{max} - \Delta V/2)$ is mapped to $2^{N-1} - 1$.

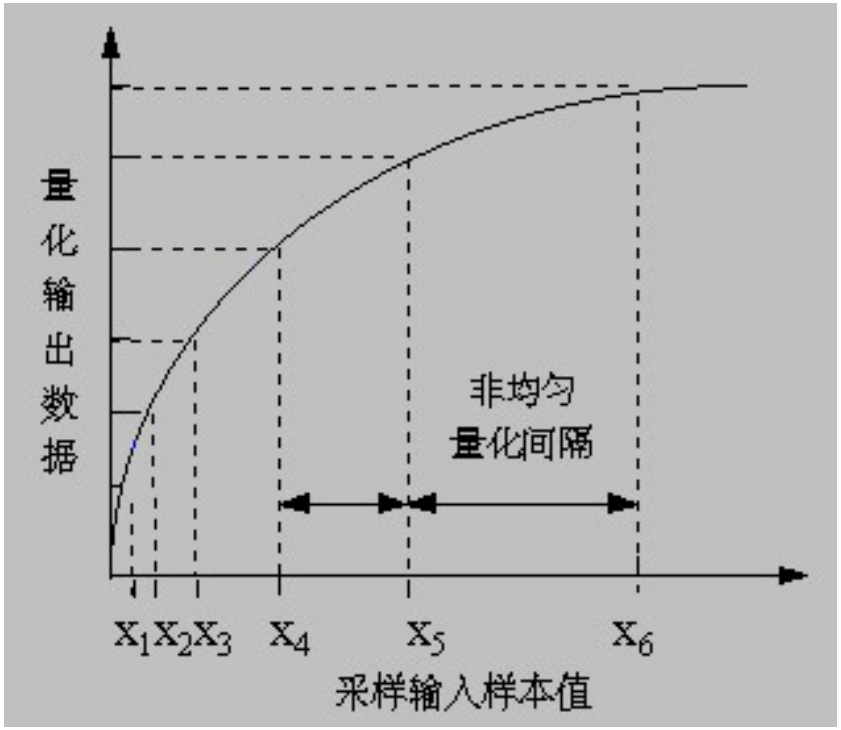
(e) The maximum noise, in terms of actual voltages, is half the quantization interval: $\Delta V/2 = V_{max}/2^N$.

1.6 Linear and Nonlinear Quantization

- Linear format : Samples are typically stored as uniformly quantized values
- Considering limited available bits and human perception properties
  - Nonuniform quantization level pay more attention to the frequency range over which humans hear best
- Nonuniform quantization schemes take advantage of human perceptual characteristics and use logarithms.

1.6 Linear and Nonlinear Quantization

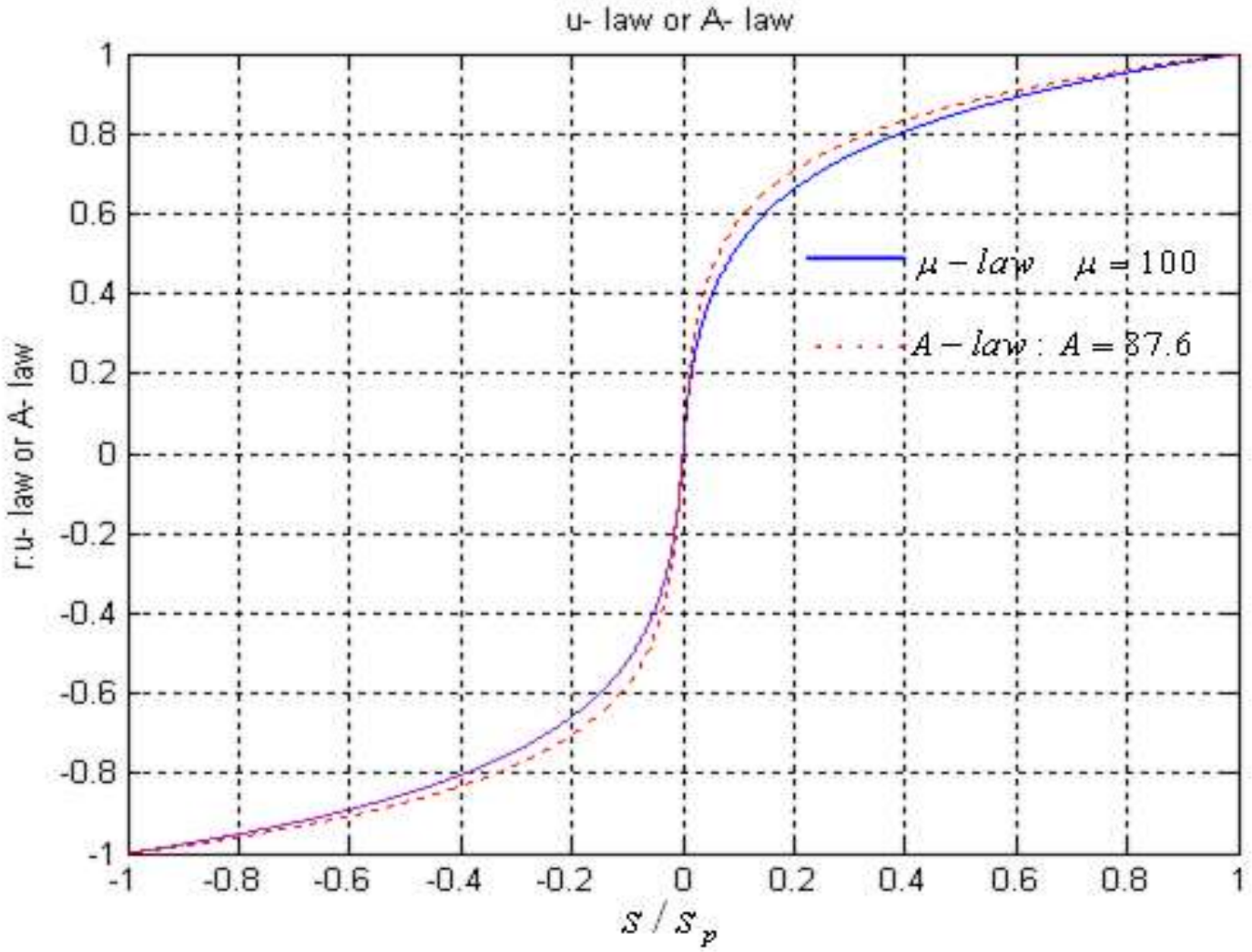
- Steps of nonlinear quantization
  - Transforming an analog signal from the raw S space into the theoretical R space;
  - Uniformly quantizing the resulting values



1.6 Linear and Nonlinear Quantization

- The equations are as follows:
- u-law $$r = \frac{\operatorname{sgn}(s)}{\ln(1+\mu)} \ln\left\{1+\mu\left|\frac{s}{s_p}\right|\right\}, \left|\frac{s}{s_P}\right| \le 1$$
- A-law $$r = \begin{cases} \frac{A}{1+\ln A}\left(\frac{s}{s_p}\right), \\ \frac{\operatorname{sgn}(s)}{1+\ln A}\left[1+\ln A\left|\frac{s}{s_p}\right|\right], \frac{1}{A} \le \left|\frac{s}{s_p}\right| \le 1 \end{cases}$$ where sgn(s) $\begin{cases} 1 & if s > 0 \\ -1 & otherwise \end{cases}$
  - μ = 100 or 255
  - A = 87.6
  - s/sp in the range -1 to 1

1.6 Linear and Nonlinear Quantization



1.7 Audio Filtering

- Prior to sampling and AD conversion, unwanted frequencies are removed by filtering the audio signal
- Kept frequencies depend on the application:
  - Speech，contain 50Hz to 10kHz；
  - Audio music signal，contain 20Hz to 20kHz
- Other frequencies are blocked by band-pass filter, also called band-limiting filter；

1.8 Audio Quality versus Data Rate

- The uncompressed data rate increases as more bits are used for quantization

- Audio quality -- data rate and bandwidth
  - Analog devices, bandwidth expressed in frequency units ,Hertz；
  - Digital devices, bits per second, bps

1.8 Audio Quality versus Data Rate

Quality	Sample Rate (Khz)	Bits per Sample	Mono / Stereo	Data Rate (uncompressed) (kB/sec)	Frequency Band (KHz)
Telephone	8	8	Mono	8	0.200-3.4
AM Radio	11.025	8	Mono	11.0	0.1-5.5
FM Radio	22.05	16	Stereo	88.2	0.02-11
CD	44.1	16	Stereo	176.4	0.005-20
DAT	48	16	Stereo	192.0	0.005-20
DVD Audio	192 (max)	24(max)	6 channels	1,200 (max)	0-96 (max)

1.9 Synthetic Sounds

- Two approaches : Digitized sound converted to analog
  - FM- frequency modulation ;
  - Wave Table (more accurate)
    - Digital samples are stored sounds from real instruments
- In FM, a carrier sinusoid is changed by adding another term involving a second, modulating frequency.

$$X(t) = A(t)\cos[\omega_c \pi t + I(t)\cos(\omega_m \pi t + \phi_m) + \phi_c]$$

  - A(t) envelope, loudness of the sound ;
  - I(t), produce harmonic feelings by changing modulation frequency ;
  - ϕ_c and ϕ_m phase constants, create time-shift

1.9 Synthetic Sounds

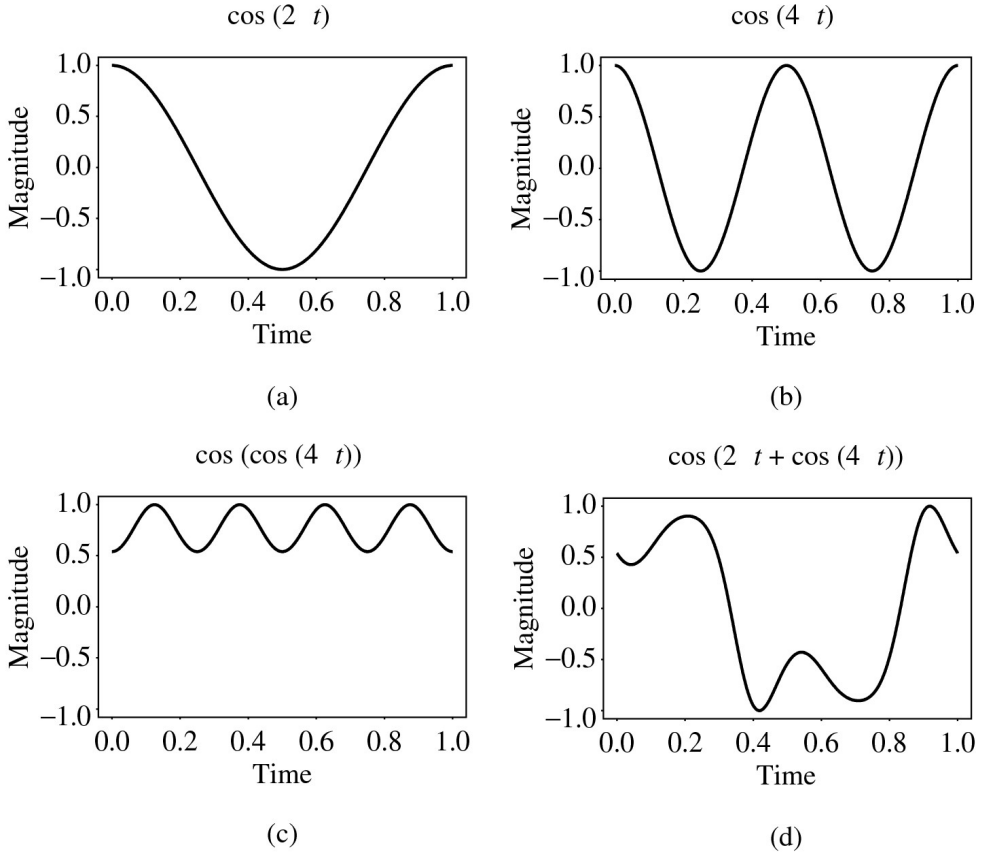
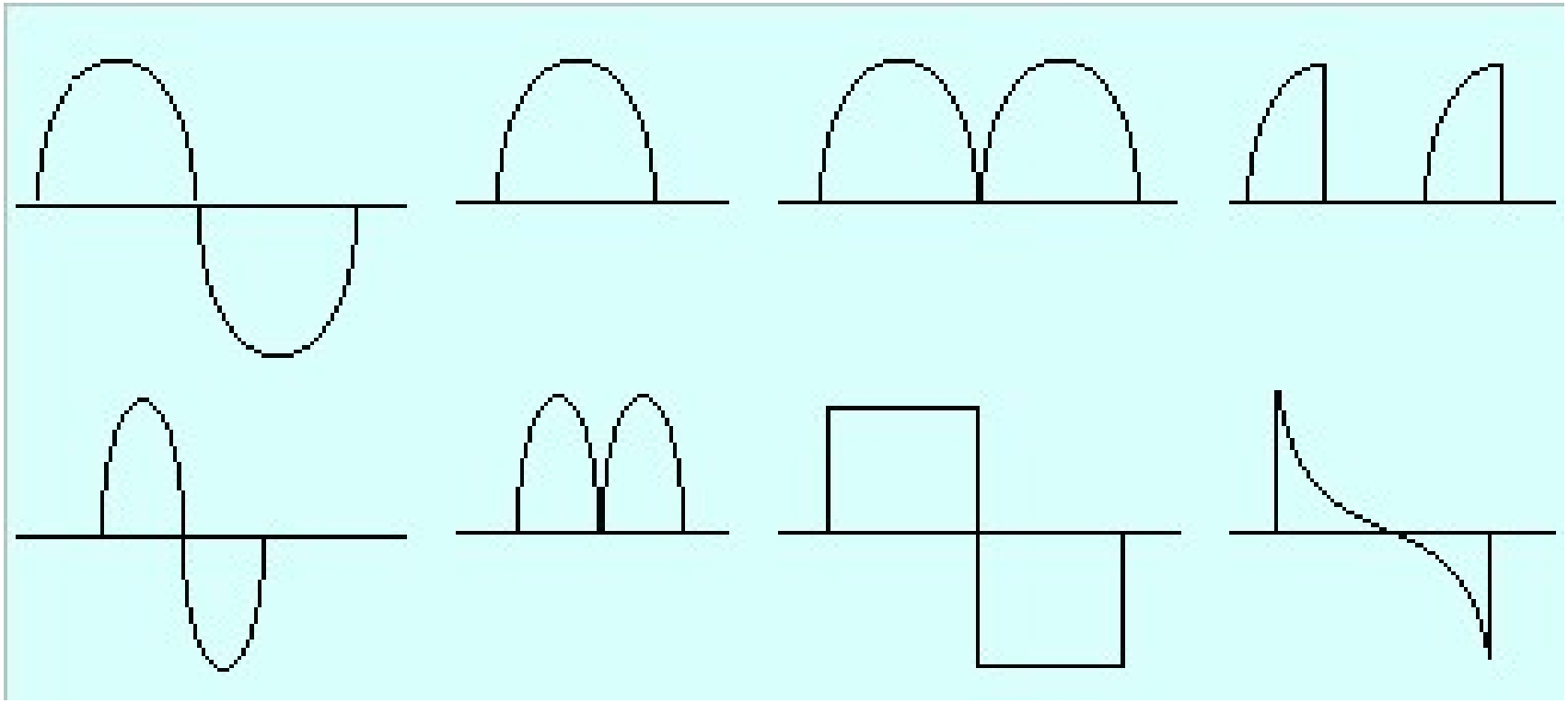


Fig. 6.7: Frequency Modulation. (a): A single frequency. (b): Twice the frequency. (c): Usually, FM is carried out using a sinusoid argument to a sinusoid. (d): A more complex form arises from a carrier frequency, $2\pi t$ and a modulating frequency $4\pi t$ cosine inside the sinusoid.



1.9 Synthetic Sounds

Wave Table synthesis: A more accurate way of generating sounds from digital signals. Also known, simply, as **sampling**.

In this technique, the actual digital samples of sounds from real instruments are stored. Since wave tables are stored in memory on the sound card, they can be manipulated by software so that sounds can be combined, edited, and enhanced.

→ Link to details.

2、Music Instrument Digital Interface

- MIDI
- Terminology
- Differences between MIDI and MP3

2.1 MIDI

- MIDI (Musical Instruments Digital Interface)
  - A protocol enables musical devices to communicate with each other
- MIDI – scripting language
  - Not audio signals, but sequences of instructions send to Midi devices to generate sound or perform some actions
- Methods of MIDI to generate music
  - FM
  - Wave table synthesis

2.2 Terminology

Synthesizer:

- Sound generator -- vary pitch, loudness, tone color
- A microprocessor, keyboard, control panels, memory and so on.

Sequencer:

- A hardware or software for editing a sequence of musical events
- One or more MIDI input (Ins) and output (OUTs).

2.2 Terminology

Channel:

- Separate MIDI messages
- 16 channels
  (associated with 16 instruments)

Timbre:

- Quality of the sound, e.g., piano, violin, etc.
- Multi-timbral – can play many different sound at the same time (e.g., piano, brass, drums, etc.)

2.3 Differences between MIDI and MP3

- MIDI file -- Some collection of instructions
  - Very small, usually about 10k size, while a MP3 file usually more than 2 Mbytes
- MP3 -- Similar voice quality like CD
  - MIDI can only generate simple music tone, it can’t regenerate singer’s voice.
- Many software can convert mp3 to midi format, like widi.

3、Quantization and Transmission of Audio

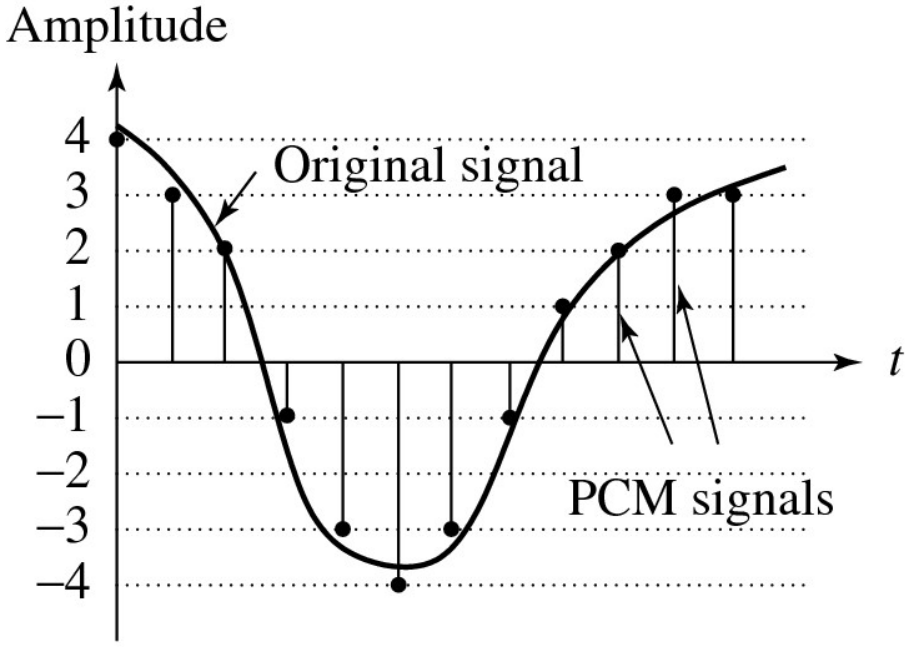
- Coding of Audio
- Pulse Code Modulation
- Differential Coding of Audio
- Lossless Predictive Coding
- DPCM
- DM
- ADPCM

3.1 Coding of Audio

- Coding -- quantization and transformation of data
- Exploiting the temporal redundancy in audio signals
  - Reduce the size of signal values
  - Differences in signals between the present and a past time can reduce the size of signal values and also concentrate the histogram of pixel values (differences, now) into a much smaller range.
  - Lossless compression methods can produce shorter bit lengths ;
- Producing quantized output for audio
  - PCM, Pulse Code Modulation
  - DPCM, Difference version of PCM
  - ADPCM, Adaptive DPCM

3.2 Pulse Code Modulation

- The basic techniques for creating digital signals from analog signals are **sampling** and **quantization**.
- Quantization consists of selecting breakpoints in magnitude, and then re-mapping any value within an interval to one of the representative output levels.



3.2 Pulse Code Modulation

a) The set of interval boundaries are called **decision boundaries**, and the representative values are called **reconstruction levels**.

b) The boundaries for quantizer input intervals that will all be mapped into the same output level form a **coder mapping**.

c) The representative values that are the output values from a quantizer are a **decoder mapping**.

d) Finally, we may wish to **compress** the data, by assigning a bit stream that uses fewer bits for the most prevalent signal values (Chap. 7).

3.2 Pulse Code Modulation

Every compression scheme has three stages:

A. The input data is ***transformed*** to a new representation that is easier or more efficient to compress.

B. We may introduce **loss** of information. ***Quantization*** is the main lossy step ⇒ we use a limited number of reconstruction levels, fewer than in the original signal.

C. ***Coding***. Assign a codeword (thus forming a binary bitstream) to each output level or symbol. This could be a fixed-length code, or a variable length code such as Huffman coding (Chap. 7).

3.2 Pulse Code Modulation

- **PCM in Speech Compression**
  - Assuming a bandwidth for speech from about 50 Hz to about 10 kHz, the Nyquist rate would dictate a sampling rate of 20 kHz.

(a) Using uniform quantization without companding, the minimum sample size we could get away with would likely be about 12 bits. Hence for mono speech transmission the bit-rate would be 240 kbps.

(b) With companding, we can reduce the sample size down to about 8 bits with the same perceived level of quality, and thus reduce the bit-rate to 160 kbps.

(c) However, the standard approach to telephony in fact assumes that the highest-frequency audio signal we want to reproduce is only about 4 kHz. Therefore the sampling rate is only 8 kHz, and the companded bit-rate thus reduces this to 64 kbps.

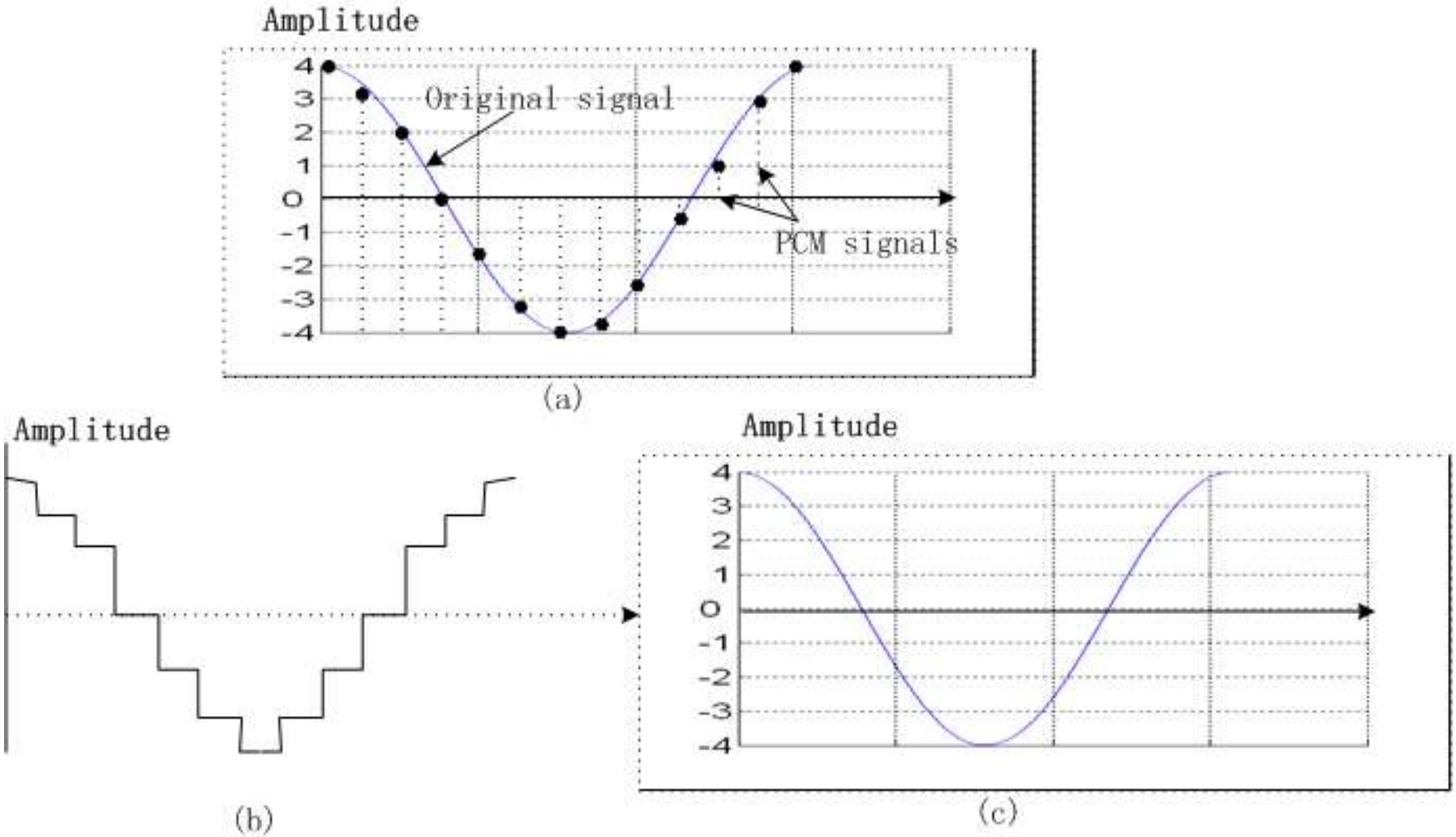
3.2 Pulse Code Modulation

However, there are two small wrinkles we must also address:

1. Since only sounds up to 4 kHz are to be considered, all other frequency content must be noise. Therefore, we should remove this high-frequency content from the analog input signal. This is done using a band-limiting filter that blocks out high, as well as very low, frequencies.

   Also, once we arrive at a pulse signal, such as that in Fig. 6.13(a) below, we must still perform DA conversion and then construct a final output analog signal. But, effectively, the signal we arrive at is the staircase shown in Fig. 6.13(b).

3.2 Pulse Code Modulation



3.2 Pulse Code Modulation

2. A discontinuous signal contains not just frequency components due to the original signal, but also a theoretically infinite set of higher-frequency components:

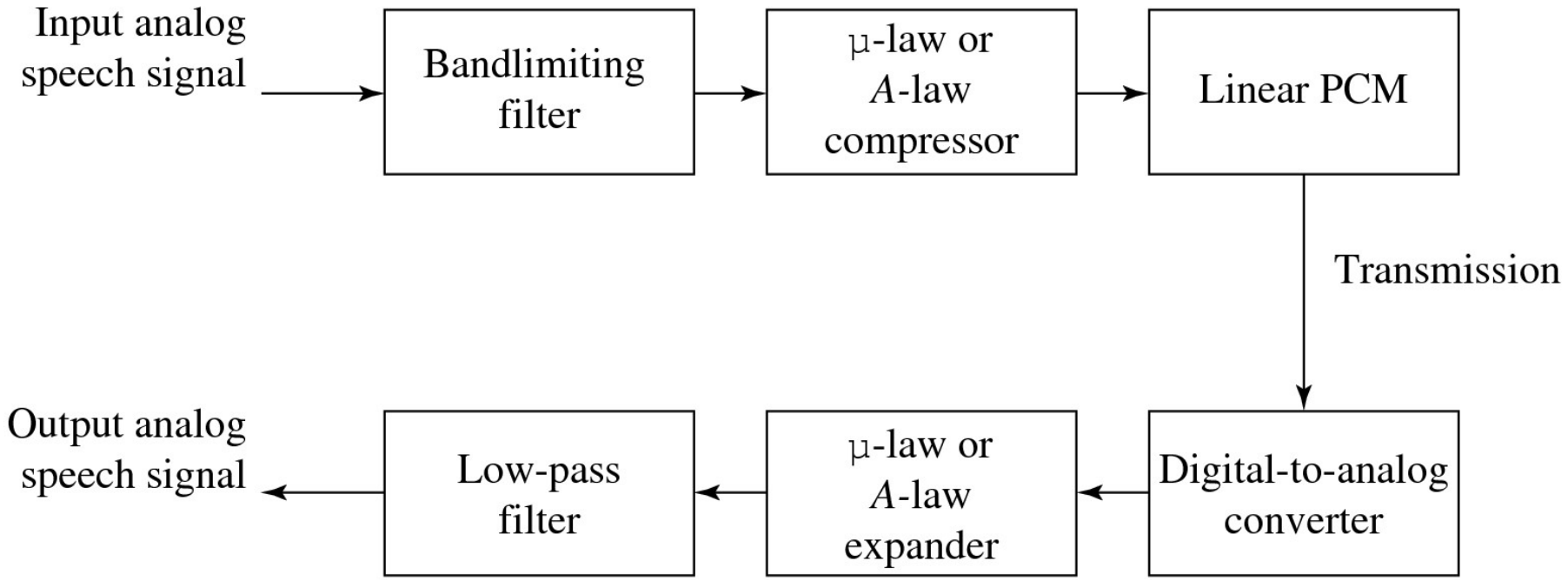
   (a) This result is from the theory of **Fourier analysis**, in signal processing.

   (b) These higher frequencies are **extraneous**.

   (c) Therefore the output of the digital-to-analog converter goes to a **low-pass filter** that allows only frequencies up to the original maximum to be retained.

3.2 Pulse Code Modulation

- The complete scheme for encoding and decoding telephony signals is shown as a schematic in Fig. 6.14. As a result of the low-pass filtering, the output becomes smoothed and Fig. 6.13(c) above showed this effect.

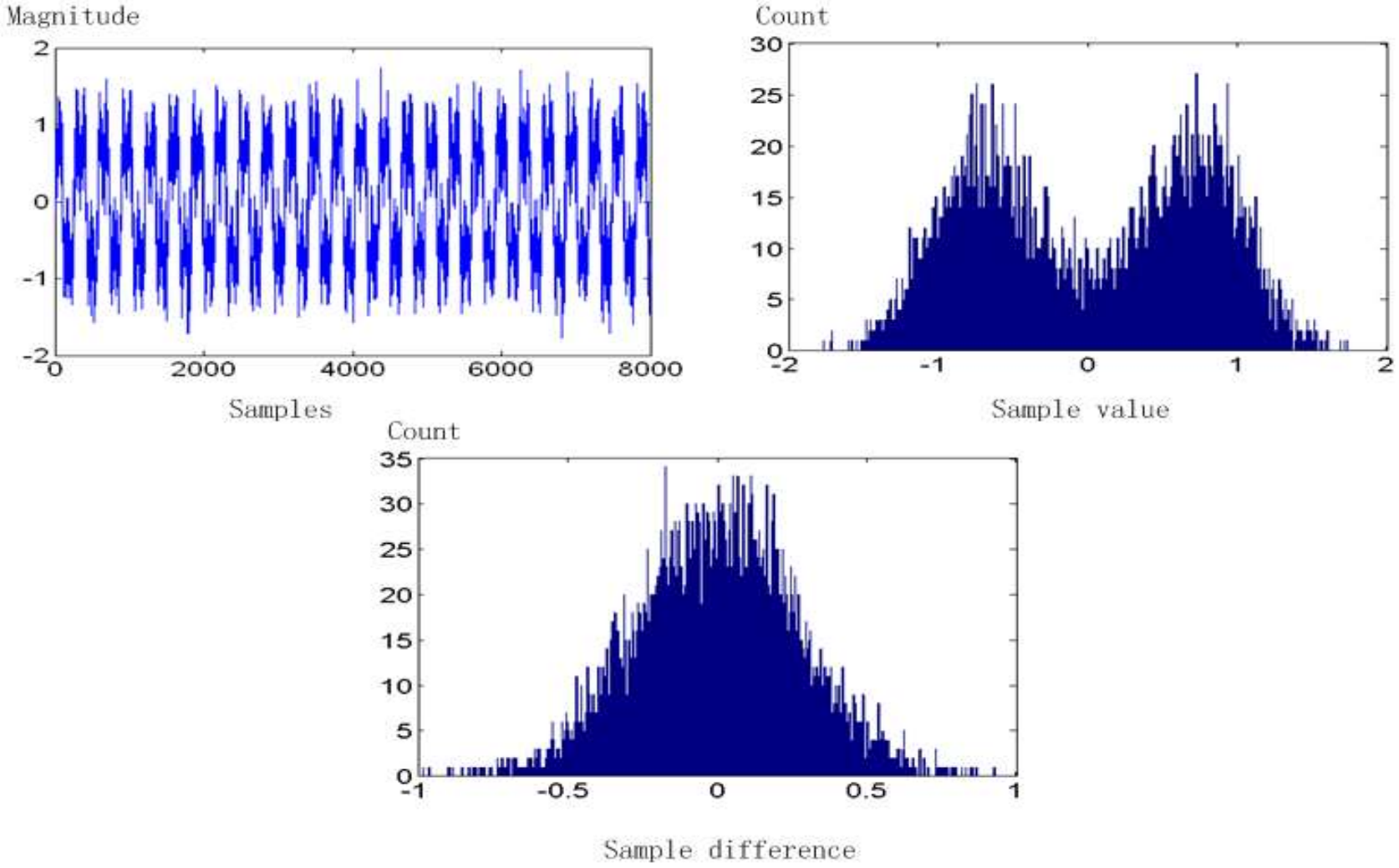


3.3 Differential coding of audio

- Audio is often stored not in simple PCM but instead in a form that exploits differences — which are generally smaller numbers, so offer the possibility of using fewer bits to store.

- If a time-dependent signal has some consistency over time ("temporal redundancy"), the difference signal, subtracting the current sample from the previous one, will have a more peaked histogram, with a maximum around zero.

3.3 Differential coding of audio



3.4 Lossless Predictive Coding

- **Predictive coding**: simply means transmitting differences — predict the next sample as being equal to the current sample; send not the sample itself but the difference between previous and next.

  (a) Predictive coding consists of finding differences, and transmitting these using a PCM system.

  (b) Note that differences of integers will be integers. Denote the integer input signal as the set of values f_n. Then we **predict** values $\hat{f}_n$ as simply the previous value, and define the error e_n as the difference between the actual and the predicted signal:

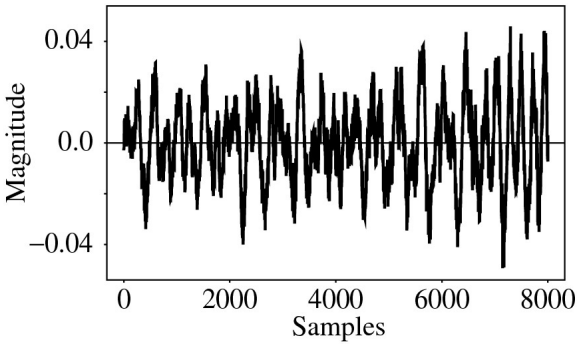
$$\hat{f}_n = f_{n-1} \qquad e_n = f_n - \hat{f}_n \tag{6.12}$$

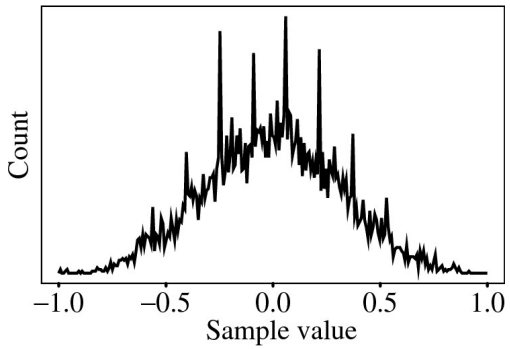
3.4 Lossless Predictive Coding

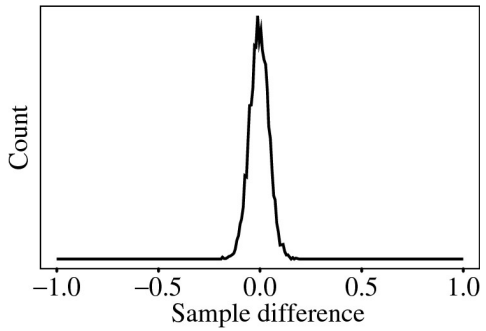
(c) But it is often the case that some function of a few of the previous values, fn−1, fn−2, fn−3, etc., provides a better prediction. Typically, a linear predictor function is used:

$$\hat{f}_n = \sum_{k=1}^{2to4} a_{n-k} f_{n-k}$$

The idea of forming differences is to make the histogram of sample values more peaked.







3.4 Lossless Predictive Coding

- One problem: suppose our integer sample values are in the range 0..255. Then differences could be as much as -255..255 —we've increased our dynamic range (ratio of maximum to minimum) by a factor of two → need more bits to transmit some differences.
  - (a) A clever solution for this: define two new codes, denoted SU and SD, standing for Shift-Up and Shift-Down. Some special code values will be reserved for these.
  - (b) Then we can use codewords for only a limited set of signal differences, say only the range −15..16. Differences which lie in the limited range can be coded as is, but with the extra two values for SU, SD, a value outside the range −15..16 can be transmitted as a series of shifts, followed by a value that is indeed inside the range −15..16.
  - (c) For example, 100 is transmitted as: SU, SU, SU, 4, where (the codes for) SU and for 4 are what are transmitted (or stored).

3.4 Lossless Predictive Coding

- Lossless predictive coding — the decoder produces the same signals as the original. As a simple example, suppose we devise a predictor for $\hat{f}_n$ as follows:

$$\hat{f}_n = \left\lfloor \frac{1}{2}(f_{n-1} + f_{n-2}) \right\rfloor$$

$$e_n = f_n - \hat{f}_n$$

3.4 Lossless Predictive Coding

- Let's consider an explicit example. Suppose we wish to code the sequence f1, f2, f3, f4, f5 = 21, 22, 27, 25, 22. For the purposes of the predictor, we'll invent an extra signal value f0, equal to f1 = 21, and first transmit this initial value, uncoded:

$$\hat{f}_2 = 21, e_2 = 22 - 21 = 1;$$

$$\hat{f}_3 = \left\lfloor \frac{1}{2}(f_2 + f_1) \right\rfloor = \left\lfloor \frac{1}{2}(22 + 21) \right\rfloor = 21,$$

$$e_3 = 27 - 21 = 6;$$

$$\hat{f}_4 = \left\lfloor \frac{1}{2}(f_3 + f_2) \right\rfloor = \left\lfloor \frac{1}{2}(27 + 22) \right\rfloor = 24,$$

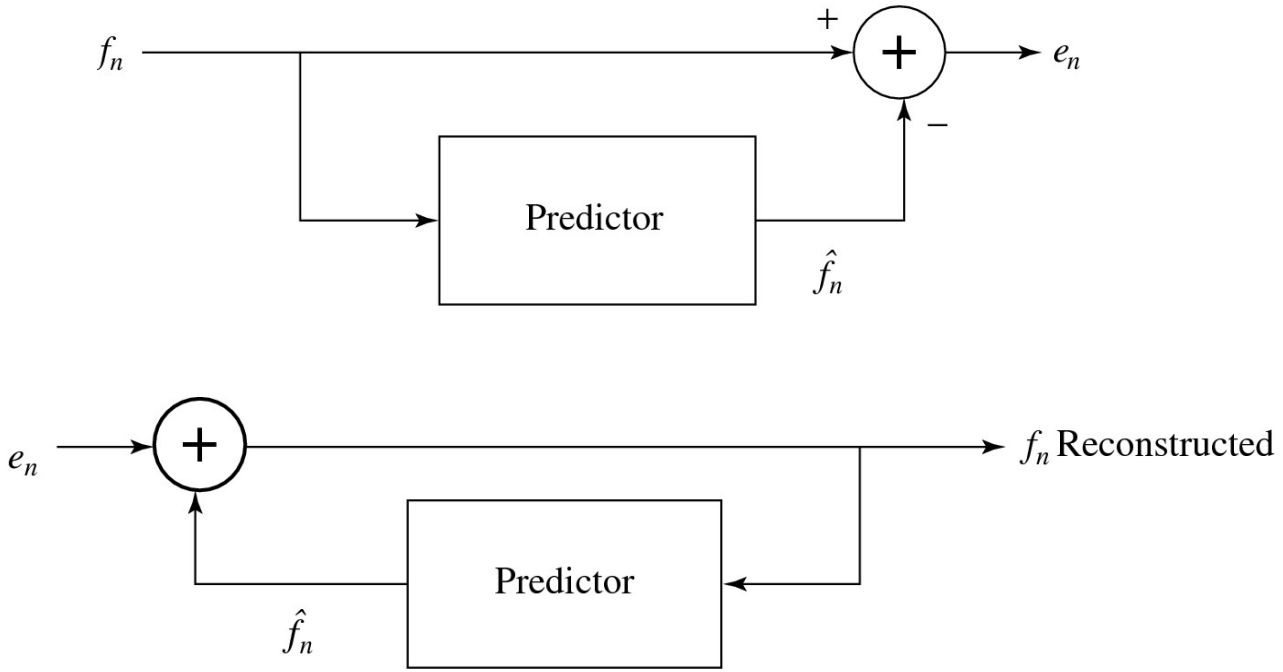
$$e_4 = 25 - 24 = 1;$$

$$\hat{f}_5 = \left\lfloor \frac{1}{2}(f_4 + f_3) \right\rfloor = \left\lfloor \frac{1}{2}(25 + 27) \right\rfloor = 26,$$

$$e_5 = 22 - 26 = -4$$

3.4 Lossless Predictive Coding

- The error does center around zero, we see, and coding (assigning bit-string codewords) will be efficient. Fig. 6.16 shows a typical schematic diagram used to encapsulate this type of system:



3.5 DPCM

- Differential PCM is exactly the same as Predictive Coding, except that it incorporates a quantizer step.

$$\hat{f}_n = function\_of(\tilde{f}_{n-1}, \tilde{f}_{n-2}, \tilde{f}_{n-3}, \ldots)$$

$$e_n = f_n - \hat{f}_n$$

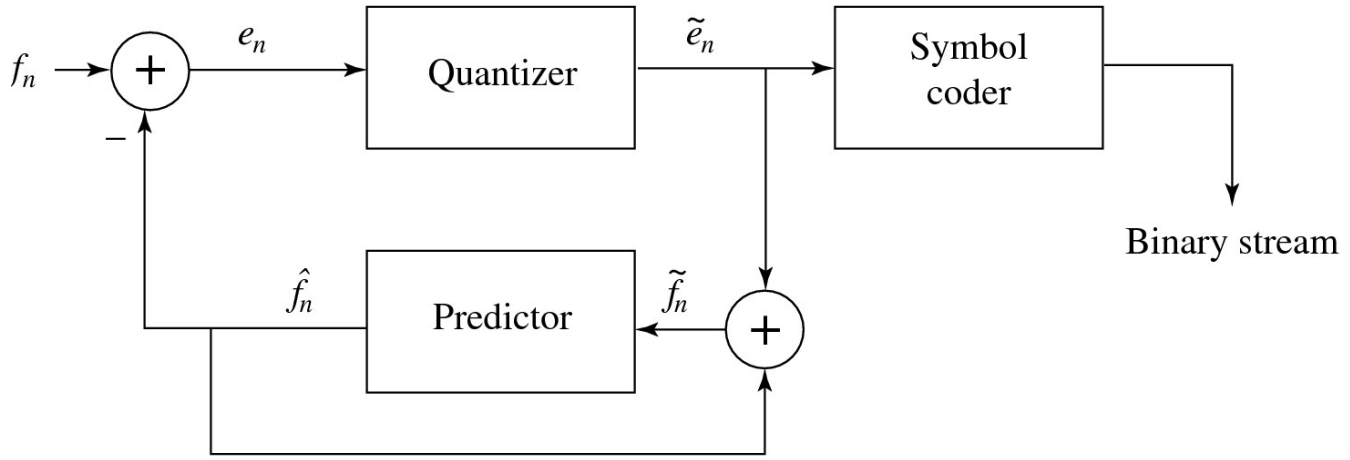
$$\tilde{e}_n = Q[e_n]$$

$$transmit\_codeword(\tilde{e}_n)$$

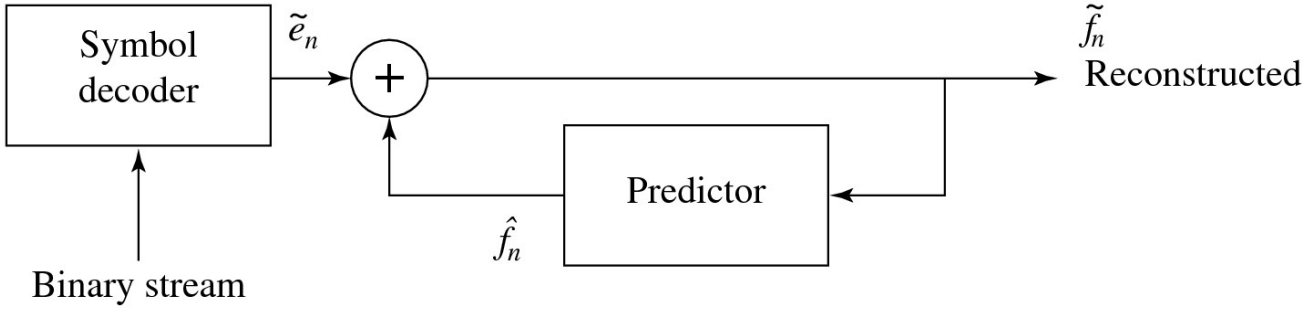
$$reconstruct: \tilde{f}_n = \hat{f}_n + \tilde{e}_n$$

Then *codewords* for quantized error values are produced using entropy coding, e.g. Huffman coding (Chapter 7).

3.5 DPCM



(a) Encoder



(b) Decoder

3.5 DPCM

- Notice that the quantization noise, $f_n - \tilde{f}_n$, is equal to the quantization effect on the error term, $e_n - \tilde{e}_n$.

- Let's look at actual numbers: Suppose we adopt the particular predictor below:

$$\hat{f}_n = trunc\left(\tilde{f}_{n-1} + \tilde{f}_{n-2}\right) \tag{6.19}$$

so that $e_n = f_n - \hat{f}_n$ is an integer.

- As well, use the quantization scheme:

$$\tilde{e}_n = Q[e_n] = 16 * \text{trunc}\left(\left(255 + e_n\right)/16\right) - 256 + 8$$

$$\tilde{f}_n = \hat{f}_n + \tilde{e}_n \tag{6.20}$$

3.5 DPCM

- First, we note that the error is in the range −255..255, i.e., there are 511 possible levels for the error term. The quantizer simply divides the error range into 32 patches of about 16 levels each. It also makes the representative reconstructed value for each patch equal to the midway point for each group of 16 levels.

e_n in range	Quantized to value
-255 .. -240	-248
-239 .. -224	-232
.	.
.	.
.	.
-31 .. -16	-24
-15 .. 0	-8
1 .. 16	8
17 .. 32	24
.	.
.	.
.	.
225 .. 240	232
241 .. 255	248

3.5 DPCM

- As an example stream of signal values, consider the set of values:

f_1	f_2	f_3	f_4	f_5
130	150	140	200	300

- Prepend extra values f = 130 to replicate the first value, f_1. Initialize with quantized error $\tilde{e}_1 \equiv 0$, so that the first reconstructed value is exact: $\tilde{f}_1$ = 130. Then the rest of the values calculated are as follows (with prepended values in a box):

$\hat{f}$	= [130],	130, 142, 144, 167
e	= [0]	20, −2, 56, 63
$\tilde{e}$	= [0]	24, −8, 56, 56
$\tilde{f}$	= [130],	154, 134, 200, 223

- On the decoder side, we again assume extra values $\tilde{f}$ equal to the correct value f_1, so that the first reconstructed value $\tilde{f}_1$ is correct. What is received is $\tilde{e}$, and the reconstructed $\tilde{f}_n$ is identical to that on the encoder side, provided we use exactly the same prediction rule.

3.6 DM

- **DM** (Delta Modulation): simplified version of DPCM. Often used as a quick AD converter.

  1. **Uniform-Delta DM**: use only a single quantized error value, either positive or negative.

     (a) ⇒ a 1-bit coder. Produces coded output that follows the original signal in a staircase fashion. The set of equations is:

$$\hat{f}_n = \tilde{f}_{n-1},$$

$$e_n = f_n - \hat{f}_n = f_n - \tilde{f}_{n-1},$$

$$\tilde{e}_n = \begin{cases} +k & \text{if } e_n > 0, \textit{where } k \textit{ is a constant} \\ -k & \textit{otherwise} \end{cases}$$

$$\tilde{f}_n = \hat{f}_n + \tilde{e}_n .$$

3.6 DM

(b) Consider actual numbers: Suppose signal values are

F_1	f_2	f_3	f_4
10	11	13	15

As well, define an exact *reconstructed* value $\tilde{f}_1 = f_1 = 10$.

(c) E.g., use step value k = 4:

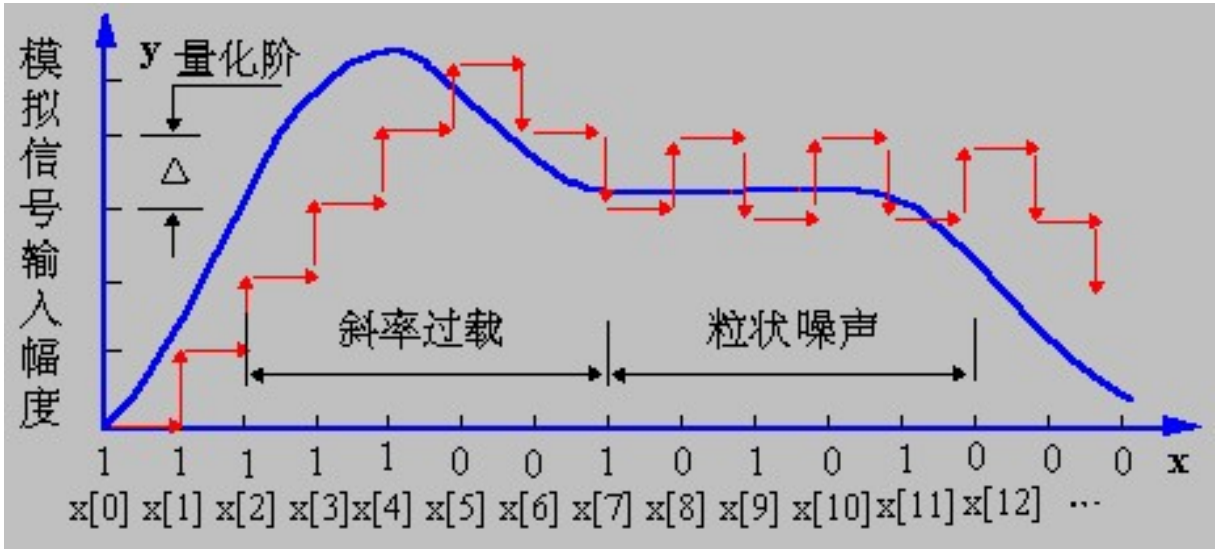
	e2 = 11 − 10 = 1,			
	e3 = 13 − 14 = −1,			
	e4 = 15 − 10 = 5,			

The reconstructed set of values 10, 14, 10, 14 is close to the correct set 10, 11, 13, 15.

(d) However, DM copes less well with rapidly changing signals. One approach to mitigating this problem is to simply increase the sampling, perhaps to many times the Nyquist rate.

3.6 DM

2. **Adaptive DM**: If the slope of the actual signal curve is high, the staircase approximation cannot keep up. For a steep curve, should change the step size k adaptively.



Adaptive DM: simply change the step size k adaptively – that is, in response to the signal's current properties.

3.6 ADPCM

- Adaptive DPCM, adapting the coder to suit the input much further.
- Adapted the quantizer step size to suit the input：
  - Using the properties of the input signal； forward adaptive quantization
  - Using the properties of quantized output； backward adaptive quantization
- Adaptive Predicative Coding： changing the prediction coefficients
  - If we use M previous values, then M coefficients a_i， i=1..M
  - Using least-squares approach， find the best values of ai：

$$\hat{f}_n = \sum_{i=1}^{M} a_i \tilde{f}_{n-i} \qquad \min \sum_{n=1}^{N} (f_n - \hat{f}_n)^2$$

The End

Thanks!

Email: junx@cs.zju.edu.cn