

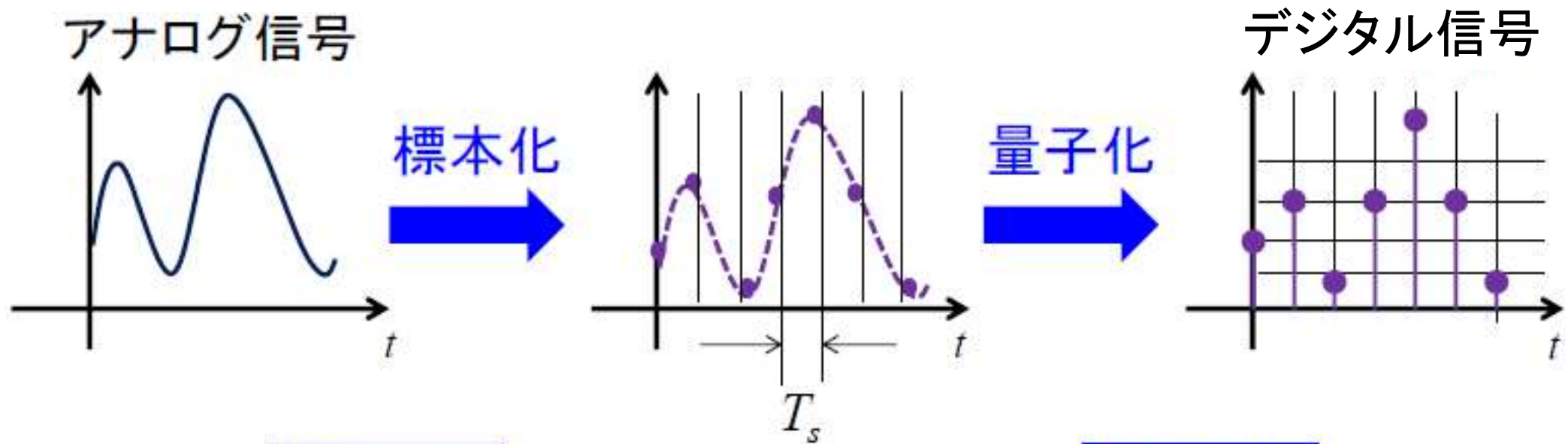
第2回
サウンドメディア論
および演習 講義編

前回の復習

- 「音」って何だっけ？
 - 空気の圧力変化が疎密波として伝わったもの
 - 「波紋」が広がるイメージ
- デジタルデータ
 - 情報量の単位は1ビットや1バイト
 - データサイズ(キロ、メガ、ギガ)の表現
 - 2進数か10進数か

今回の主な内容

アナログ信号を計算機に取り込むための処理



標本化

- 連続な時間を離散化
(とびとびの時間へ)
- 一定の間隔で分割
- 標本化定理

量子化

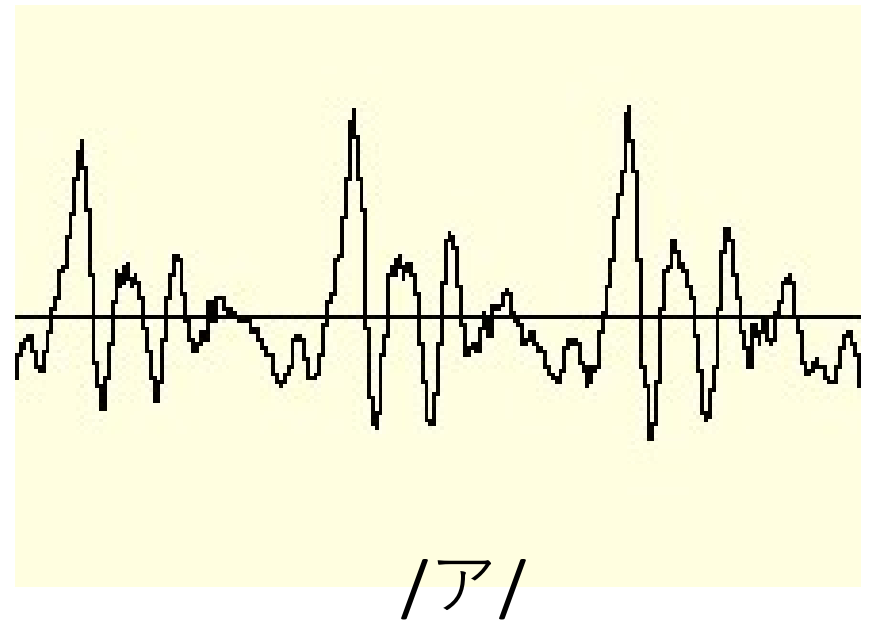
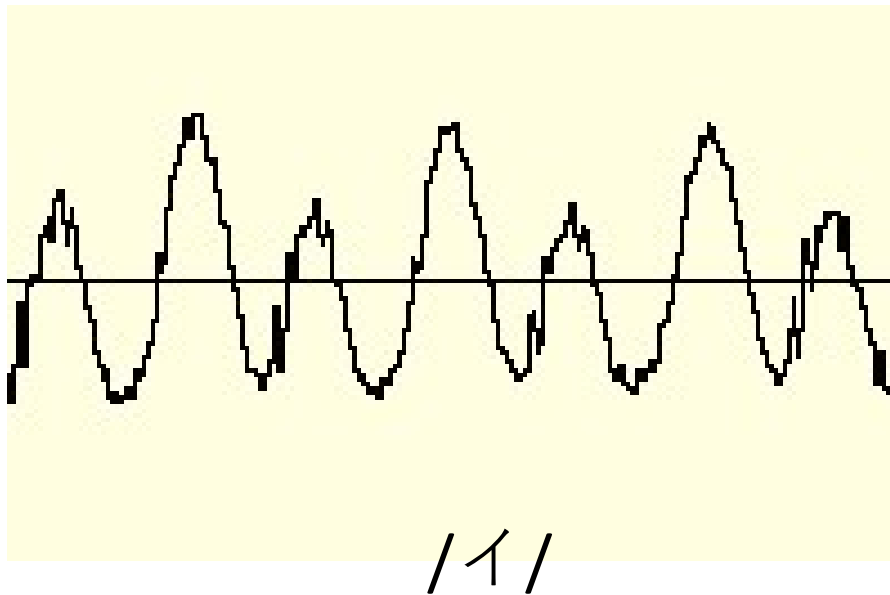
- 連続な振幅値を離散化
(とびとびの振幅値へ)
- 一定の間隔値で分割
- 量子化誤差

信号と情報

- 「信号」とは「情報」を担うもの
⇒ 何らかの物理量の変化を通して情報を担う
電圧

例：音声信号

音声の/ア/と/イ/をマイクで電気信号に変換した波形



信号と情報

- 信号と情報の例

- 電話機の電気信号と音声情報

- テレビ受像機内の信号と画像＋音声情報

- 心電図の波形と心筋の活動状況

などなど。。。

信号と情報

- 自然界の信号はまず**電気信号**に変換され、
電子機器で処理されることがほとんど
⇒なぜか？

**電気信号が他の物理量に比べて、
伝達、記録、加工するのに便利だから**

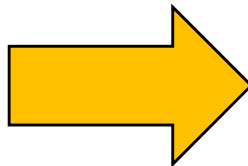
アナログ信号

- 自然界の信号はアナログ信号とも呼ばれる
 - ⇒ アナログは本来「相似」という意味
 - ⇒ 実際の物理量と相似な信号を扱う

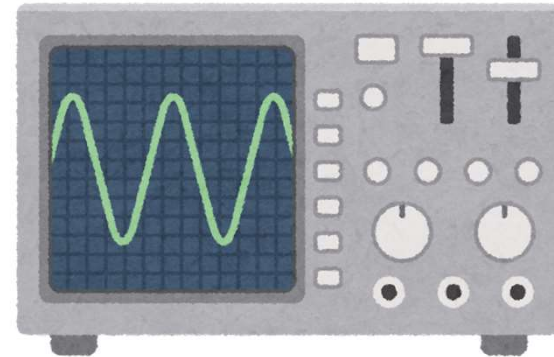
音声の場合 物理量：空気の圧力、信号：電圧



圧力変化を
電圧変化に変換



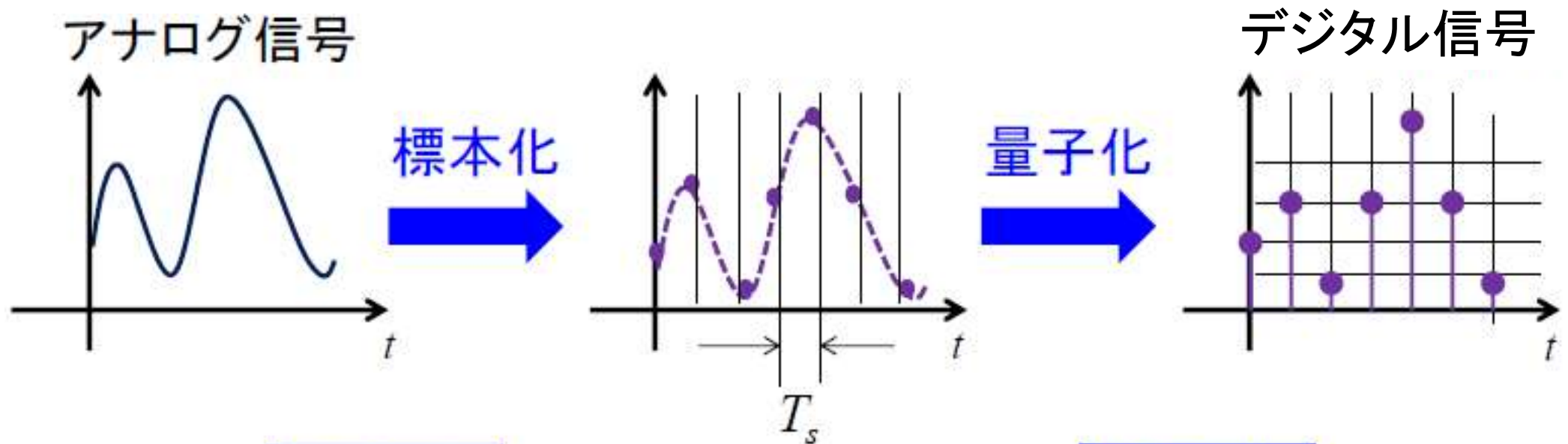
オシロスコープで表示



アナログ信号

- 値の大きさは実数値をとり、無限種類の可能性
⇒連続的な値
- 時間的にも連続（途切れない）
⇒「アナログ」を「連続的」の意味と捉える
- そのままでは計算機に取り込めない

アナログ信号を計算機に取り込む



標本化

- 連続な時間を離散化
(とびとびの時間へ)
- 一定の間隔で分割
- 標本化定理

量子化

- 連続な振幅値を離散化
(とびとびの振幅値へ)
- 一定の間隔値で分割
- 量子化誤差

デジタル信号

- デジタルとは？

数え上げる(カウントする)ことができる、という意味
⇒要するに「バラバラ」の状態

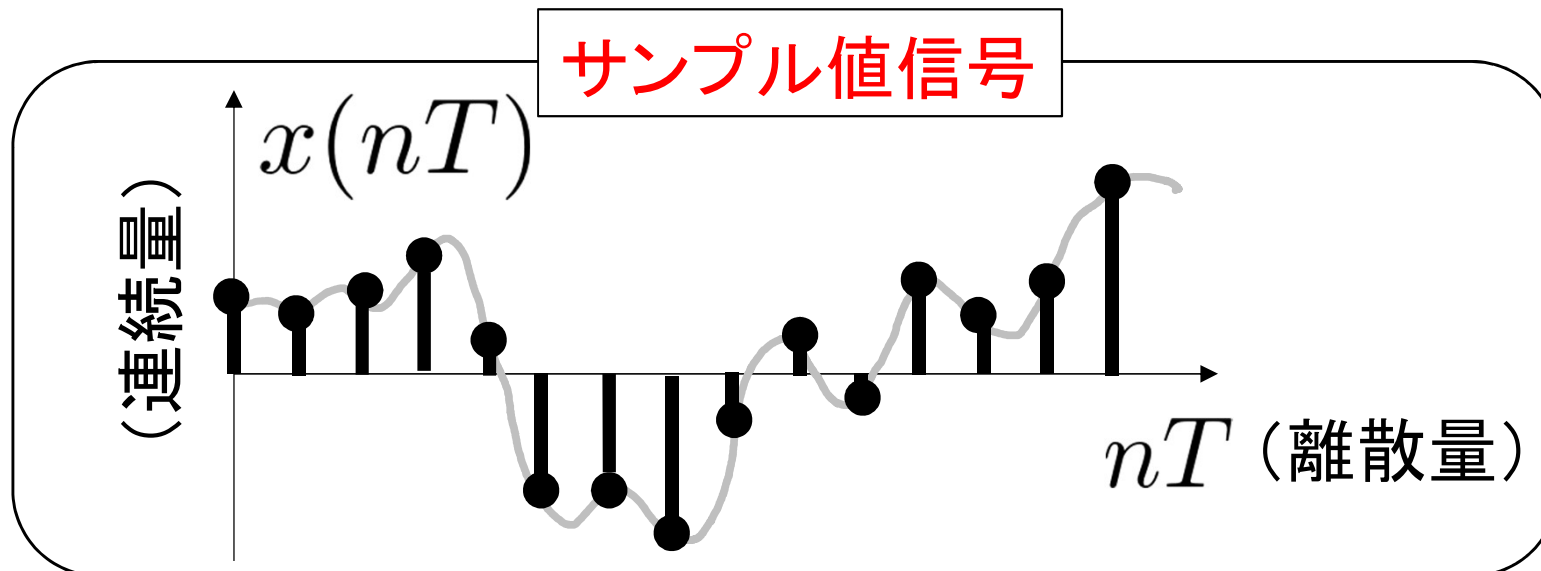
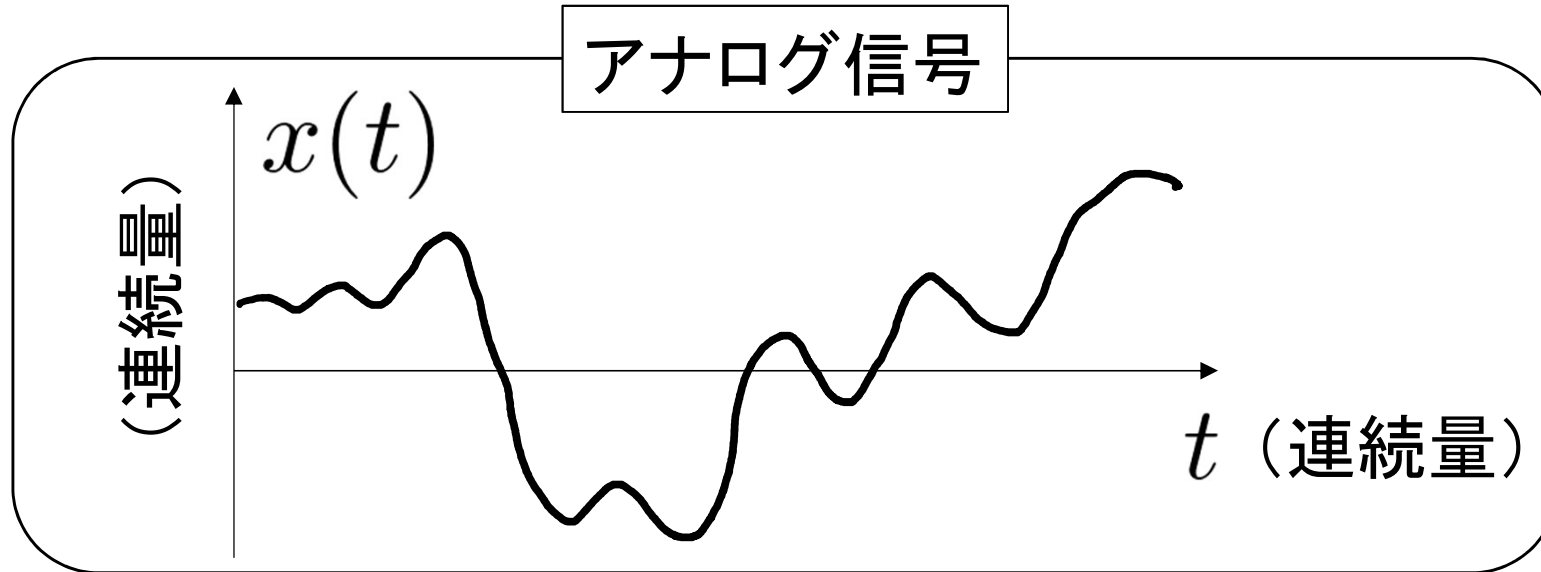
- デジタル信号ってどういうもの？

- 時間的にバラバラ: 離散的な時刻のみで値を持つ
- 振幅値もバラバラ: 離散的な(有限個の)値を持つ

⇒計算機で扱える！

標本化と量子化

- 標本化(サンプリング): 時間に関する離散化



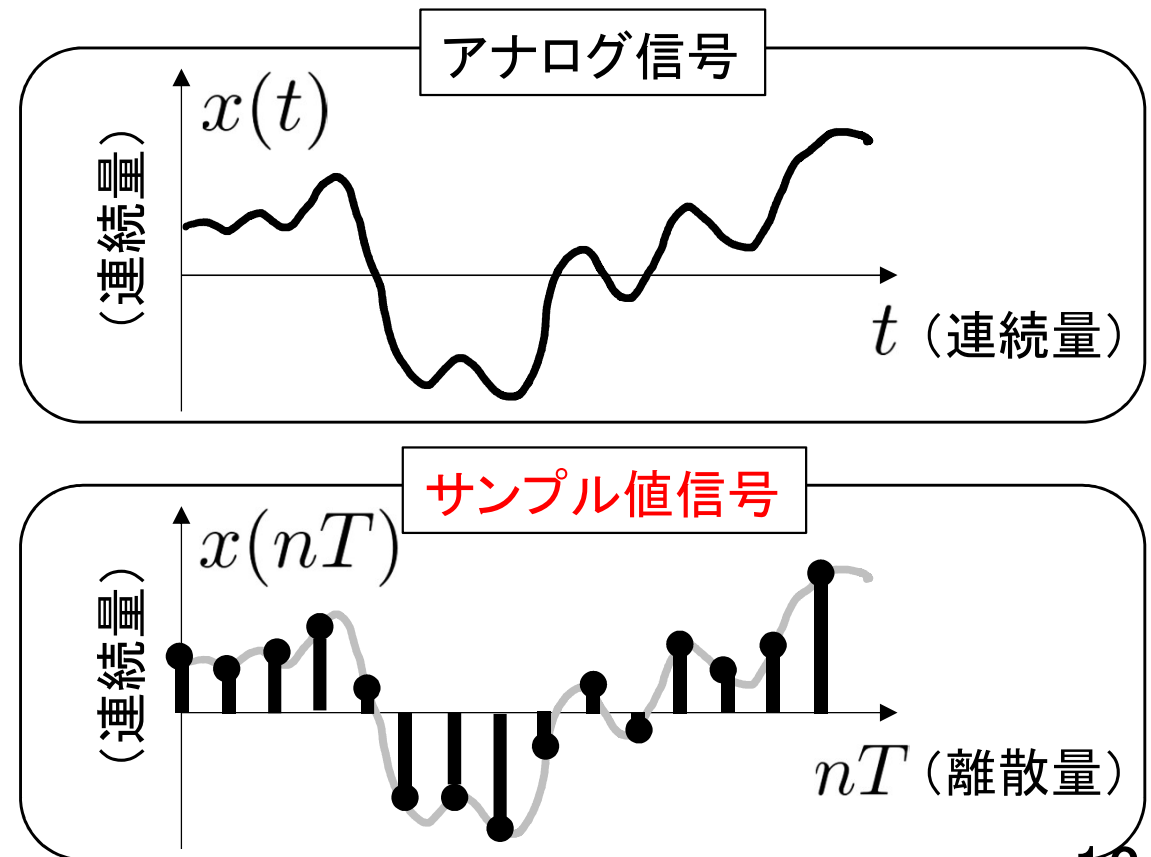
標本化と量子化

- 標本化(サンプリング): 時間に関する離散化
アナログ信号 $x(t)$ から時間間隔 T [秒]ごとの
信号値 $x(nT)$ を取り出すこと(n は整数)

サンプル値信号

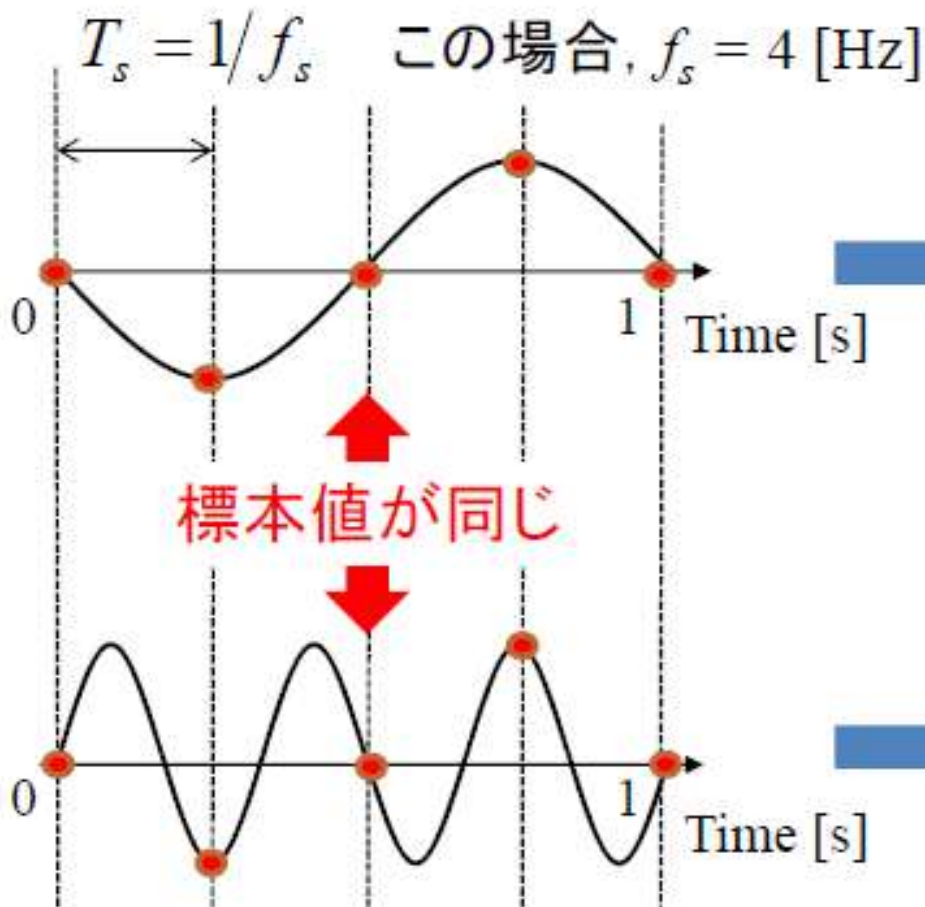
T : 標本化周期

- 標本化周期の逆数は
標本化周波数



標本化定理

アナログ信号に含まれる最大周波数成分の2倍以上で
標本化周波数を取れば元の信号を復元できる



周波数 1 [Hz] の正弦波であるため、
標本化周波数を $f_s > 2$ [Hz] にする
必要あり. $f_s = 4$ [Hz] より、元の信号を
復元可能

周波数 3 [Hz] の正弦波であるため、
標本化周波数を $f_s > 6$ [Hz] にする
必要あり. $f_s = 4$ [Hz] より、元の信号を
復元不可能

問

以下の場合について、標本化周波数はいくつ？

1. 電話の場合

⇒ 8kHz ($1/8000 \text{ sec} = 0.000125 \text{ sec}$)

2. CDの場合

⇒ 44.1kHz ($1/44100 \text{ sec} = 0.0000226 \text{ sec}$)

3. FMラジオの場合

⇒ 32kHz ($1/32000 \text{ sec} = 0.000315 \text{ sec}$)

3. デジタル放送の場合

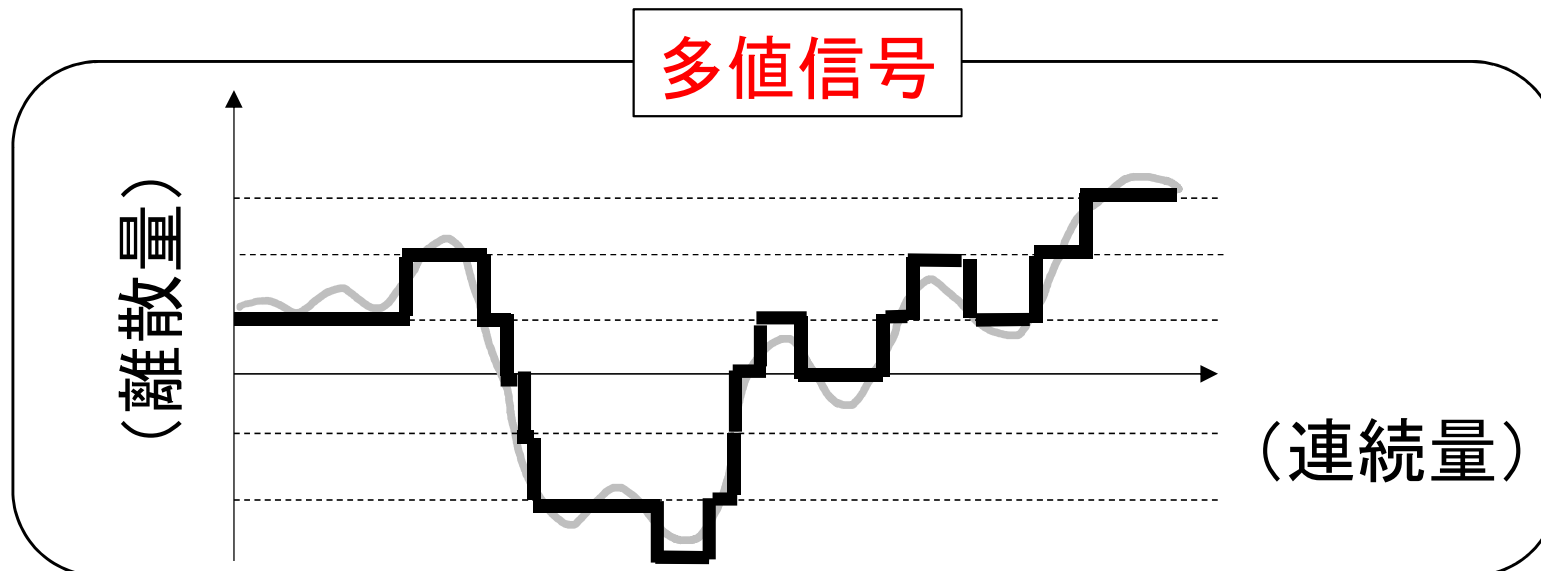
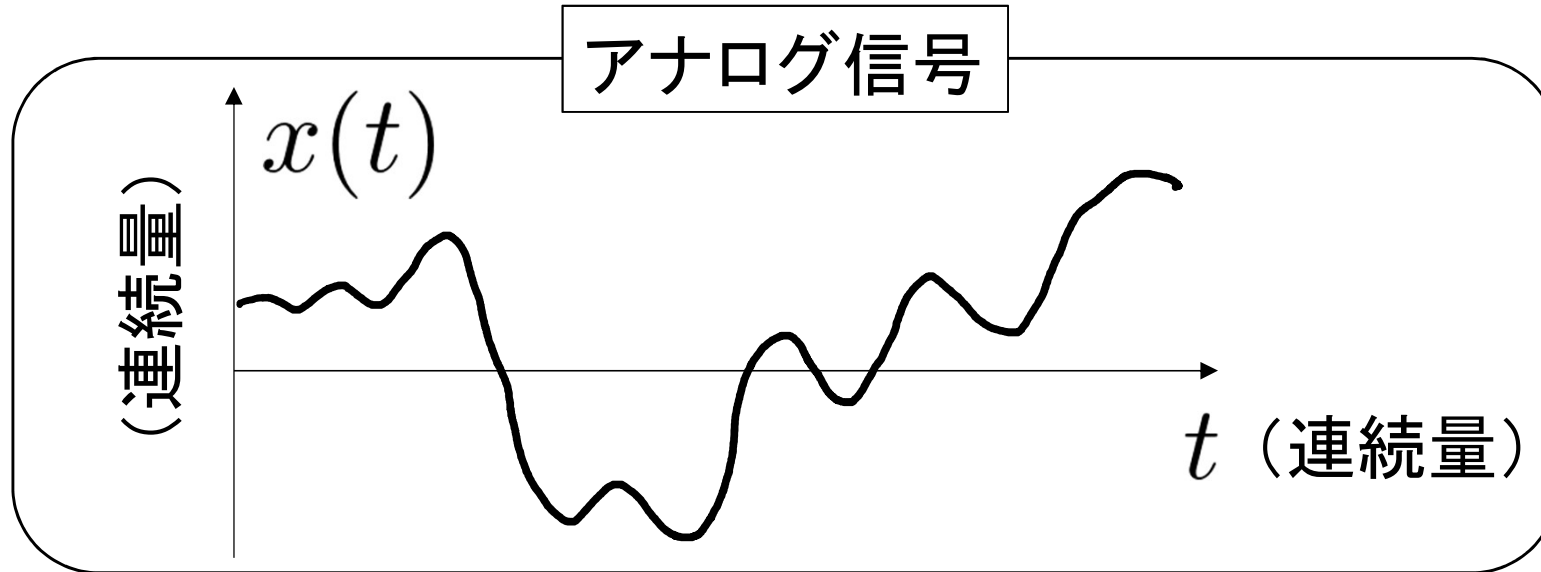
⇒ 48kHz ($1/48000 \text{ sec} = 0.0000208 \text{ sec}$)

CDの標本化周波数について

- 人間の可聴範囲は20Hz~20,000Hz
 - ⇒ 最大周波数の2倍より大きい44100Hz
 - ⇒ 標本化定理の条件は満たしている

標本化と量子化

- 量子化: 振幅値に関する離散化

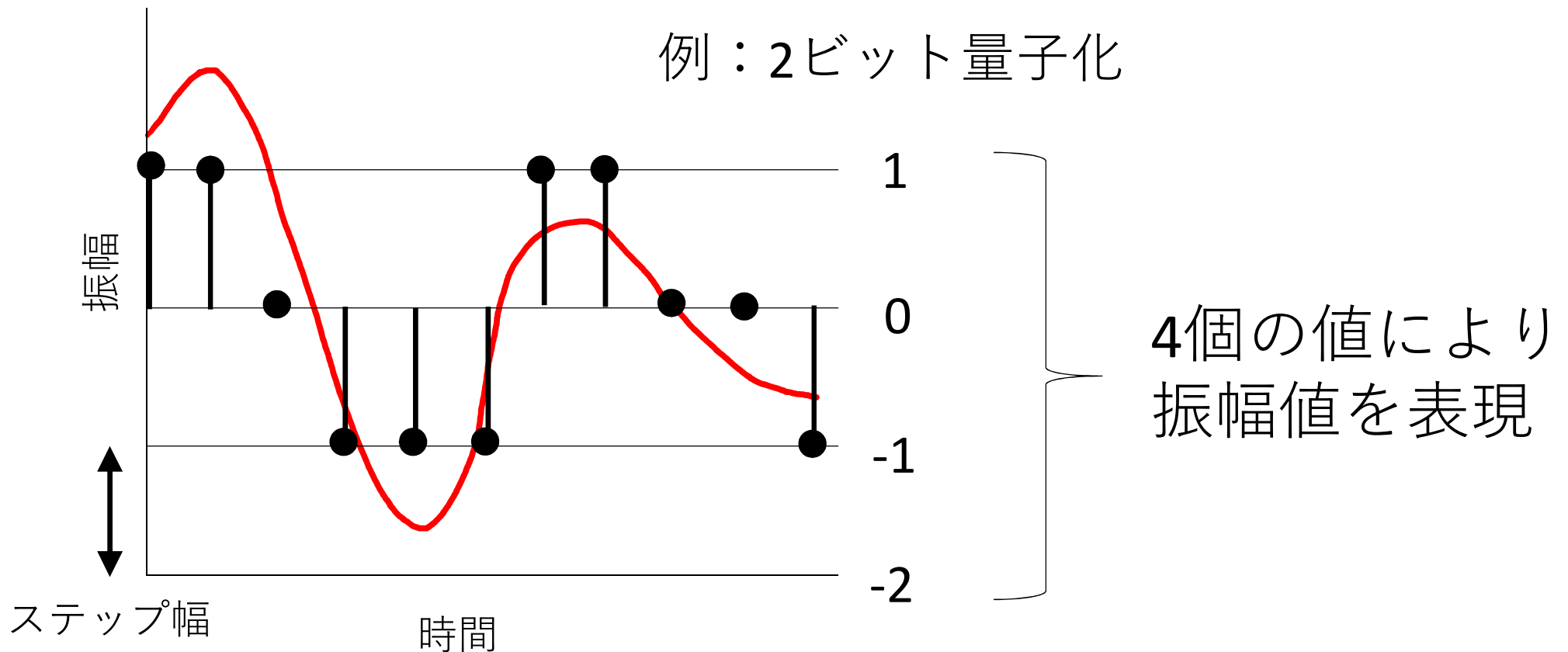


量子化ビット数(量子化精度)

- 振幅値を区切るステップ数(=間隔の細かさ)

$$N = 2^Q$$

N : ステップ数 Q : 量子化ビット数

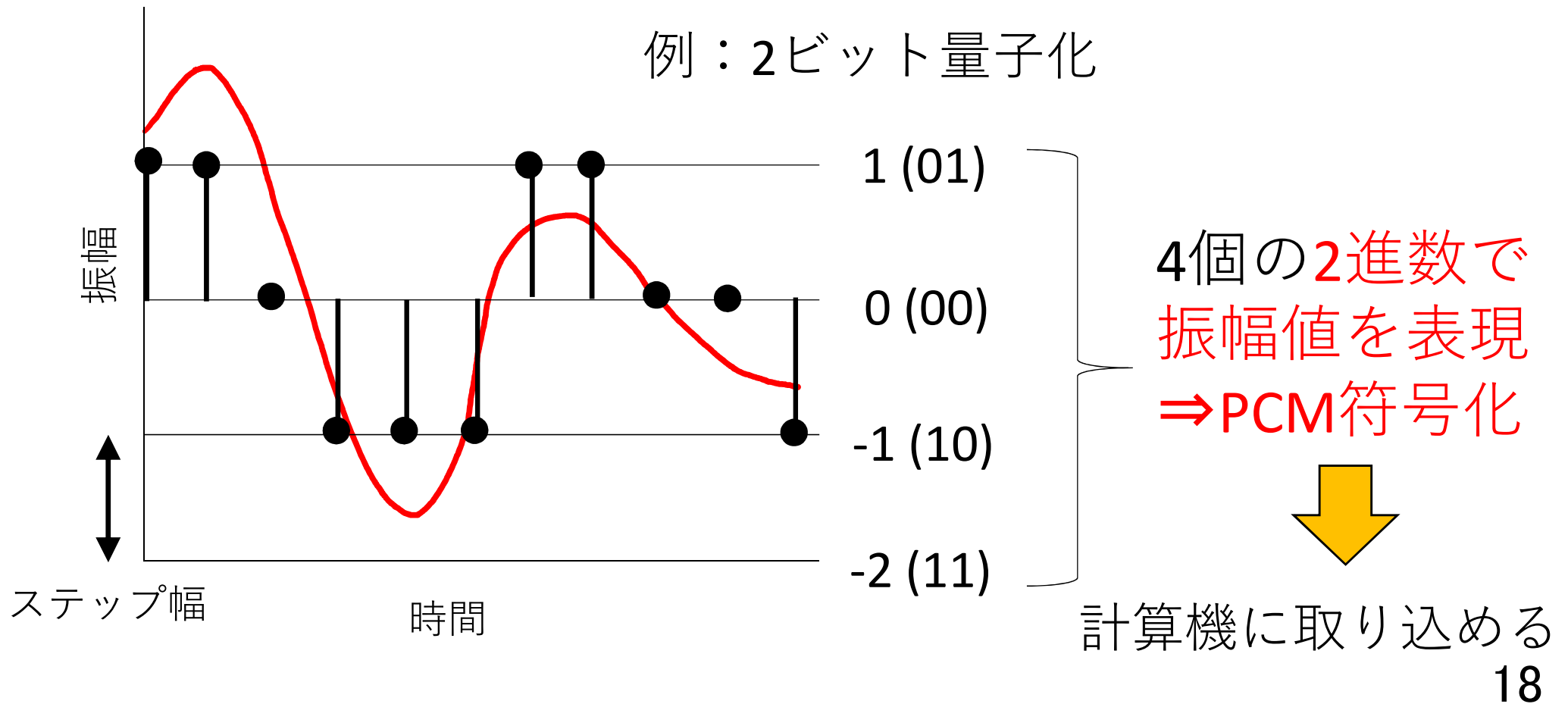


量子化ビット数(量子化精度)

- 振幅値を区切るステップ数(=間隔の細かさ)

$$N = 2^Q$$

N : ステップ数 Q : 量子化ビット数

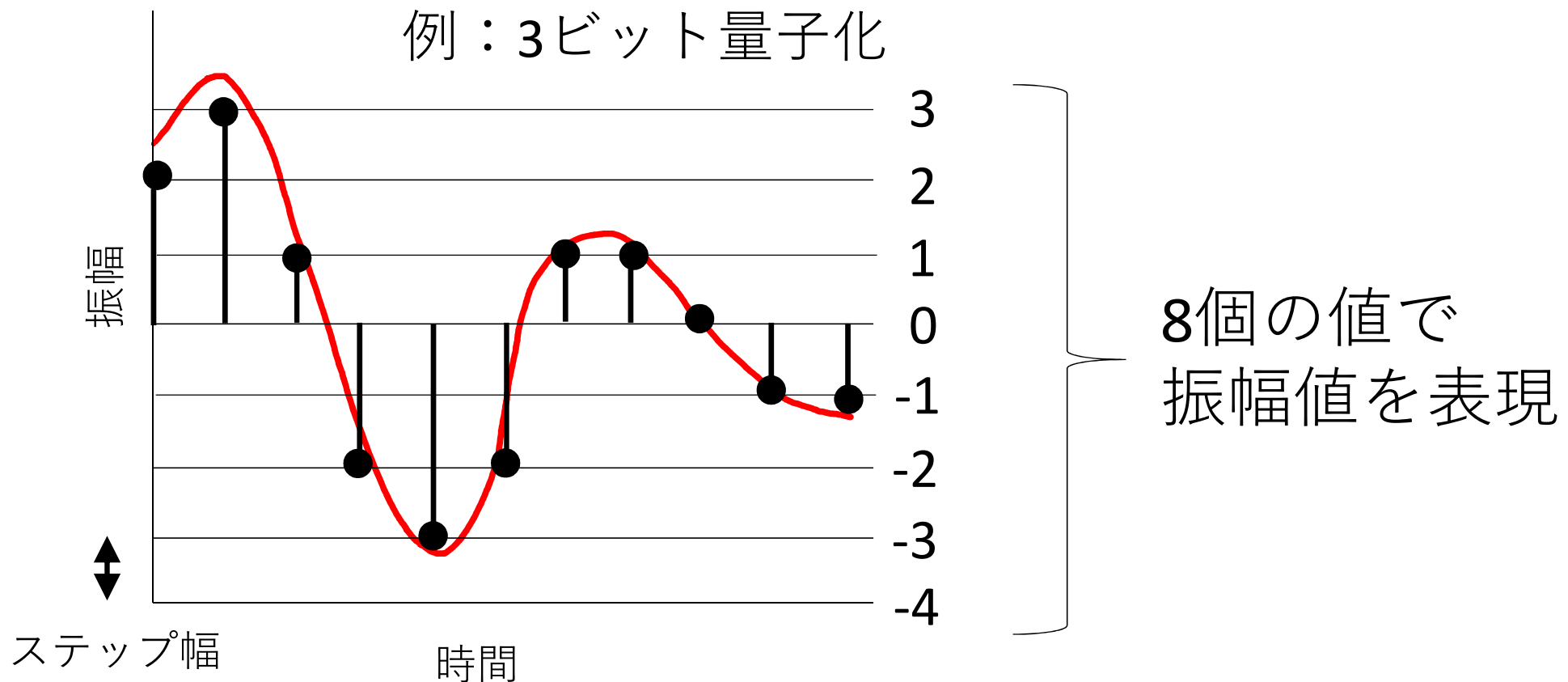


量子化ビット数(量子化精度)

- 振幅値を区切るステップ数(=間隔の細かさ)

$$N = 2^Q$$

N : ステップ数 Q : 量子化ビット数



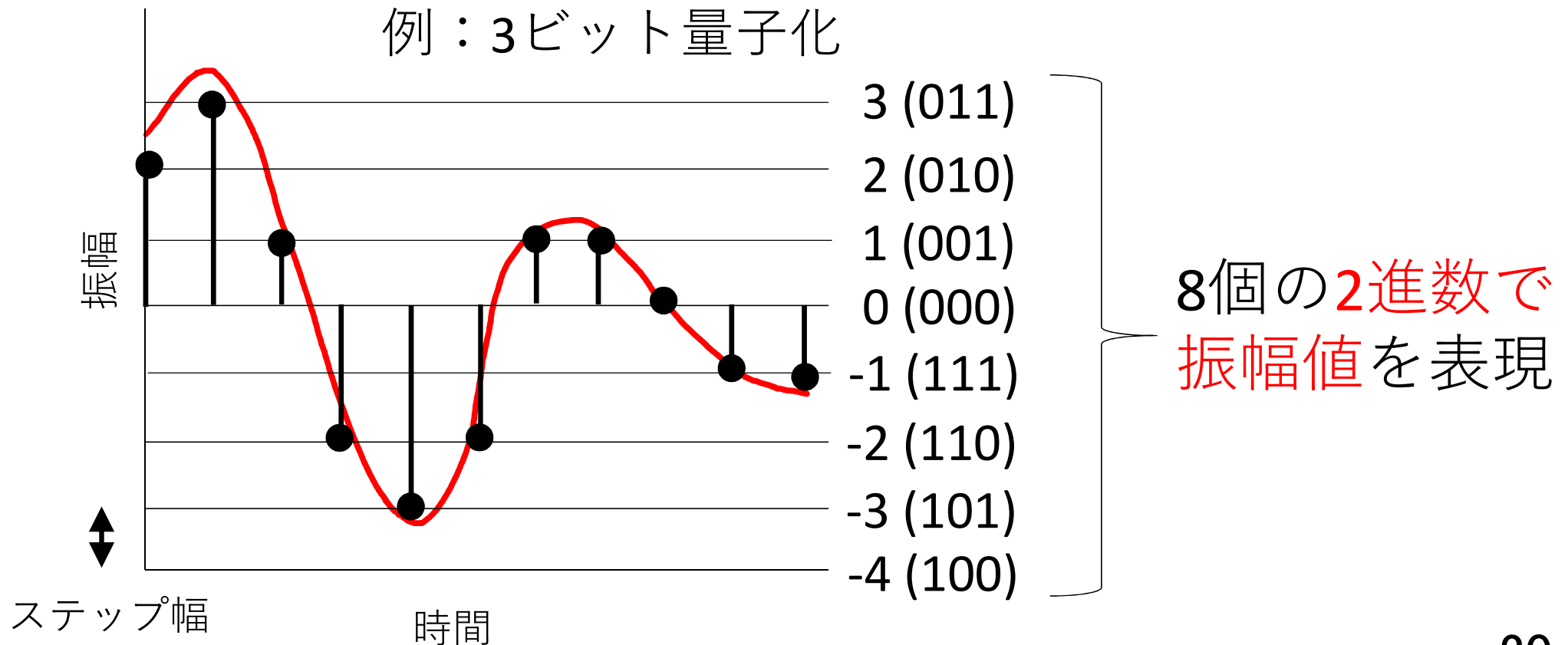
量子化ビット数(量子化精度)

- 振幅値を区切るステップ数(=間隔の細かさ)

$$N = 2^Q$$

N : ステップ数 Q : 量子化ビット数

例: 3ビット量子化



量子化ビット数(量子化精度)

- 量子化ビット数が大→ステップ幅は小
 - ⇒波形の詳細な変化を落とさない
 - ⇒元の信号との誤差(量子化誤差)が小さくて済む
 - 表現可能な値の範囲が決まる
 - 2ビット量子化⇒ -2 ~ +1 (4段階)
 - 3ビット量子化⇒ -4 ~ +3 (8段階)
 - 8ビット量子化⇒ -128 ~ +127 (256段階)
 - 16ビット量子化⇒ -32768 ~ +32767 (65536段階)
- ⇒値の範囲を決定する指針: ダイナミックレンジ

ダイナミックレンジについて

- ダイナミックレンジとは？

最大振幅値と最小振幅値の比の対数(単位はdB)

$$20 \log_{10} \left(\frac{N}{1} \right) \quad [\text{dB}] \quad N: \text{ステップ数}$$

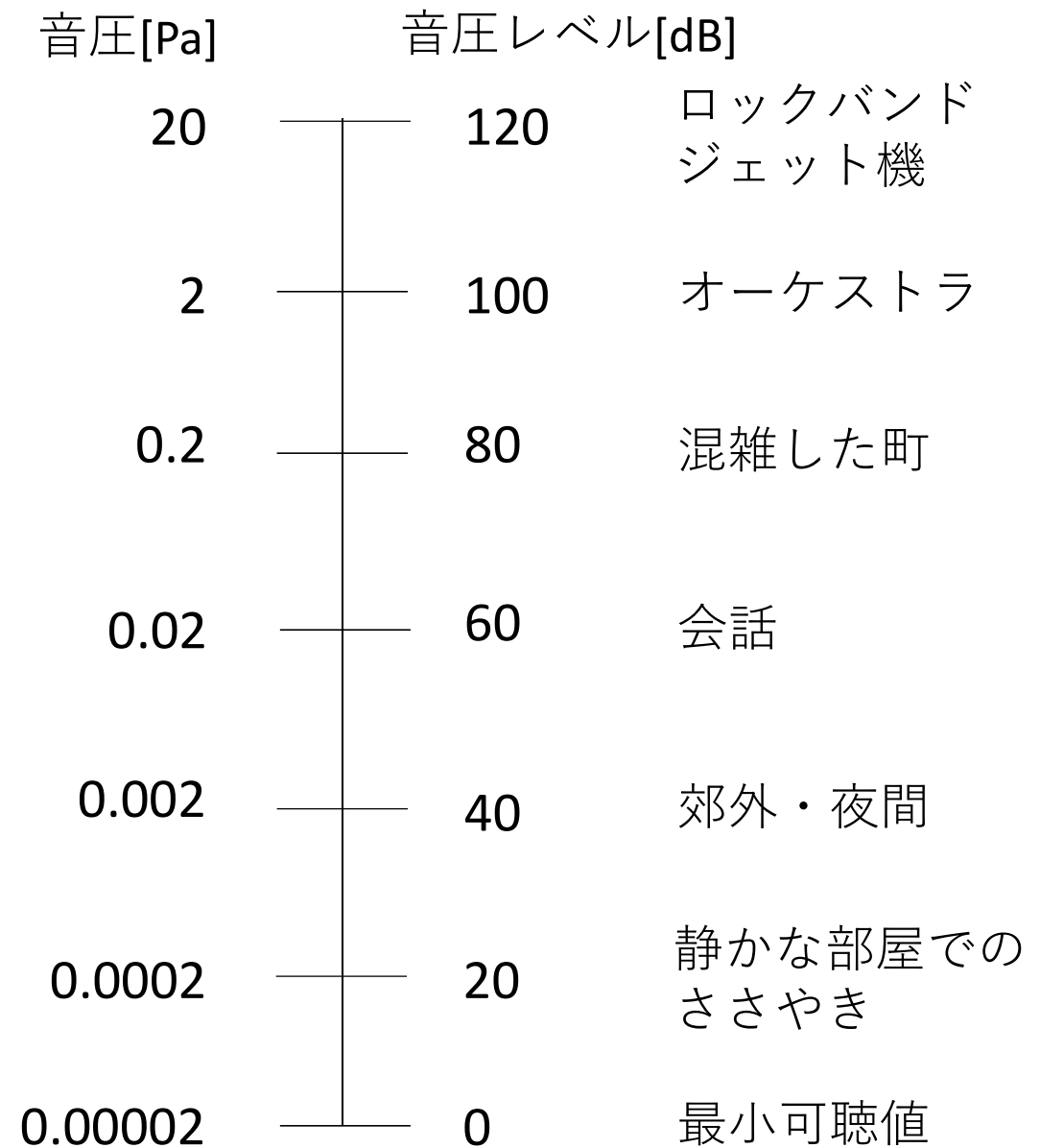
- 人が聞く音の場合

- 最小振幅値は、聞き取ることができる最も小さい音の大きさ

$$20 \log_{10} \frac{\text{音圧}}{\text{音の最小可聴値}} = 20 \log_{10} \frac{p}{2 \times 10^{-5}}$$

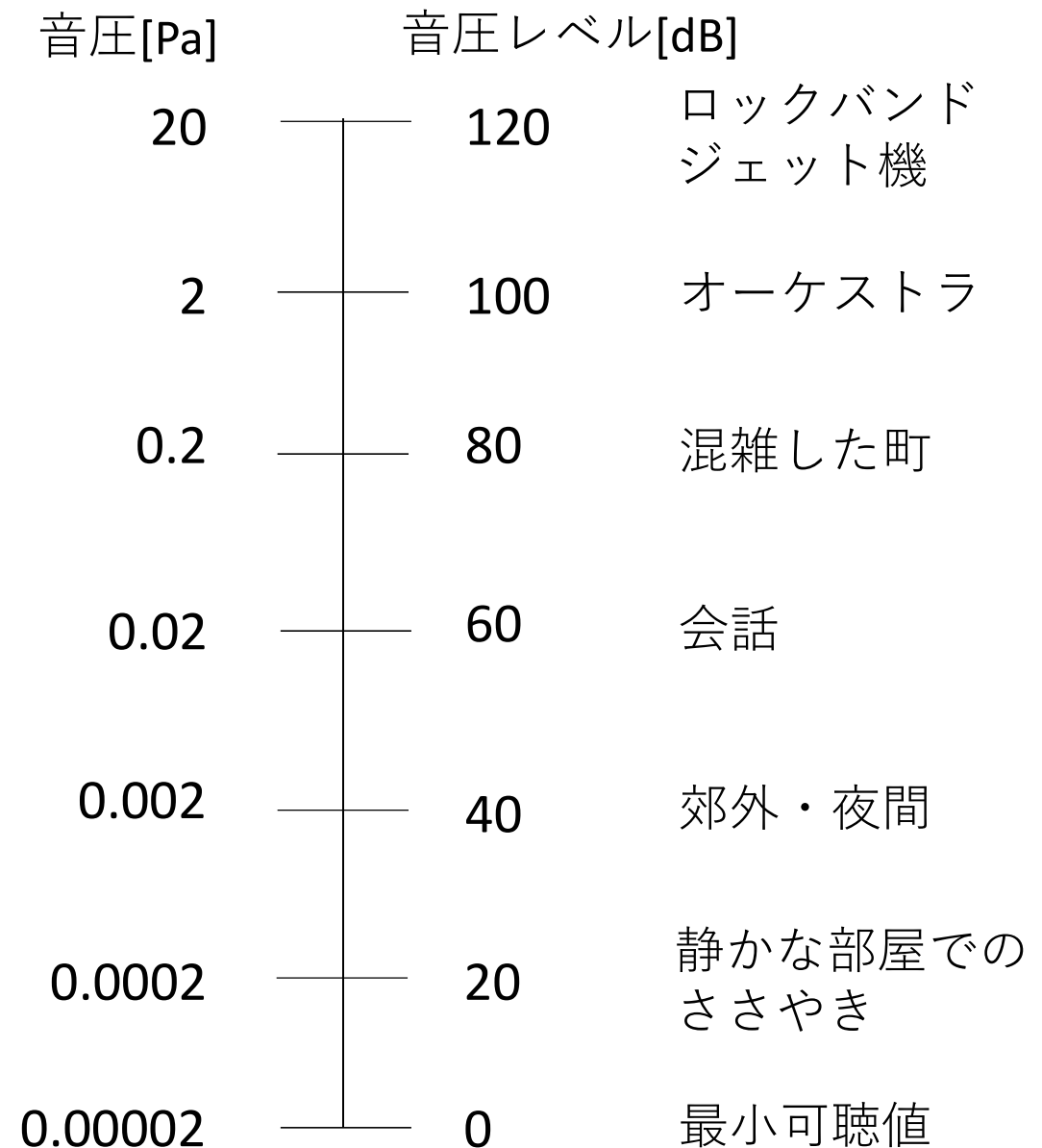
単位はPa

人間の耳のダイナミックレンジは120dB



人間の耳のダイナミックレンジは120dB

音圧の最小値と最大値
の大きさの比は**100万**！
⇒人間の聴覚はスゴイ



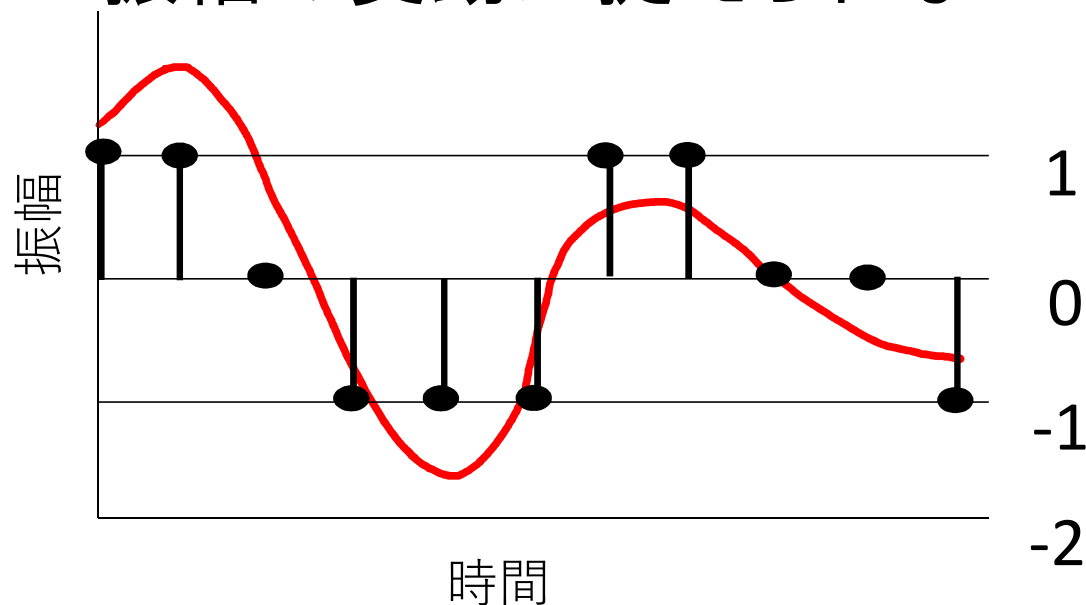
CDのダイナミックレンジは？

- CDの量子化ビット数は
 - ステップ数は65536 ($=2^{16}$)
 - $20 \log (65536/1) \rightarrow$
 \Rightarrow 人間の耳の性能(120dB)を十分に満たしていない！
- ハイレゾ音源(DVDとか)は20ビットや24ビット
 - 20ビットだと120dB、24ビットだと144dB
 - \Rightarrow 人間の耳のダイナミックレンジを十分にカバー

量子化ビット数を変えて聞き比べ

量子化 ビット数	16ビット	12ビット	8ビット	4ビット	3ビット
音					

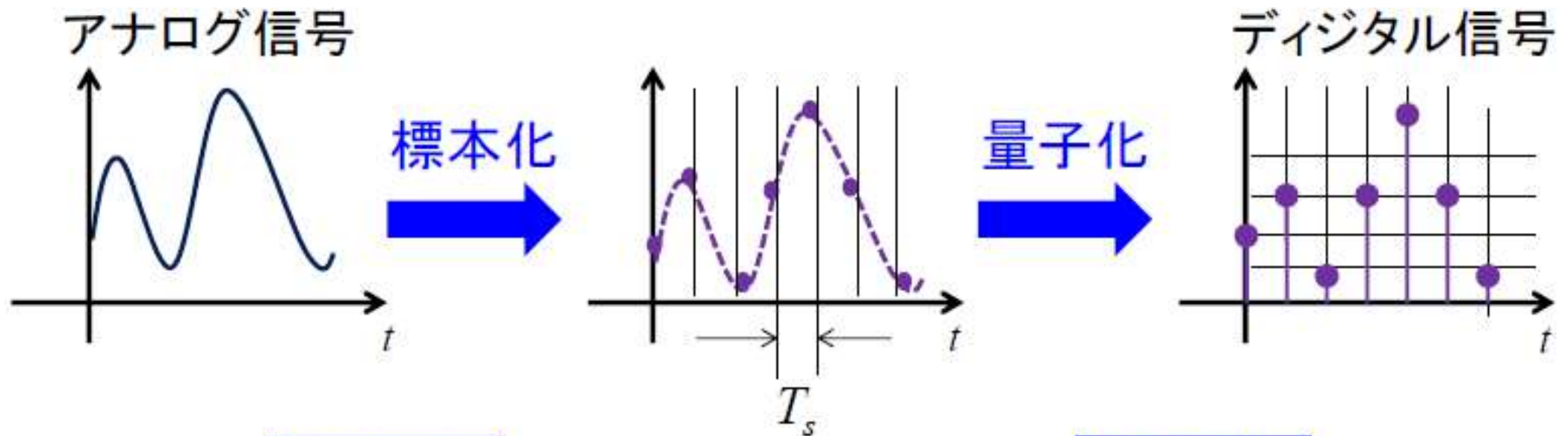
量子化ビット数が小さいほど雑音が大きい
⇒ 振幅の変動が捉えられない



例：2ビット量子化

値の範囲をはみ出したら
値が強制的に丸められる

これまでの流れ



標本化

- 連続な時間を離散化
(とびとびの時間へ)
- 一定の間隔で分割
- 標本化定理

量子化

- 連続な振幅値を離散化
(とびとびの振幅値へ)
- 一定の間隔値で分割
- 量子化誤差

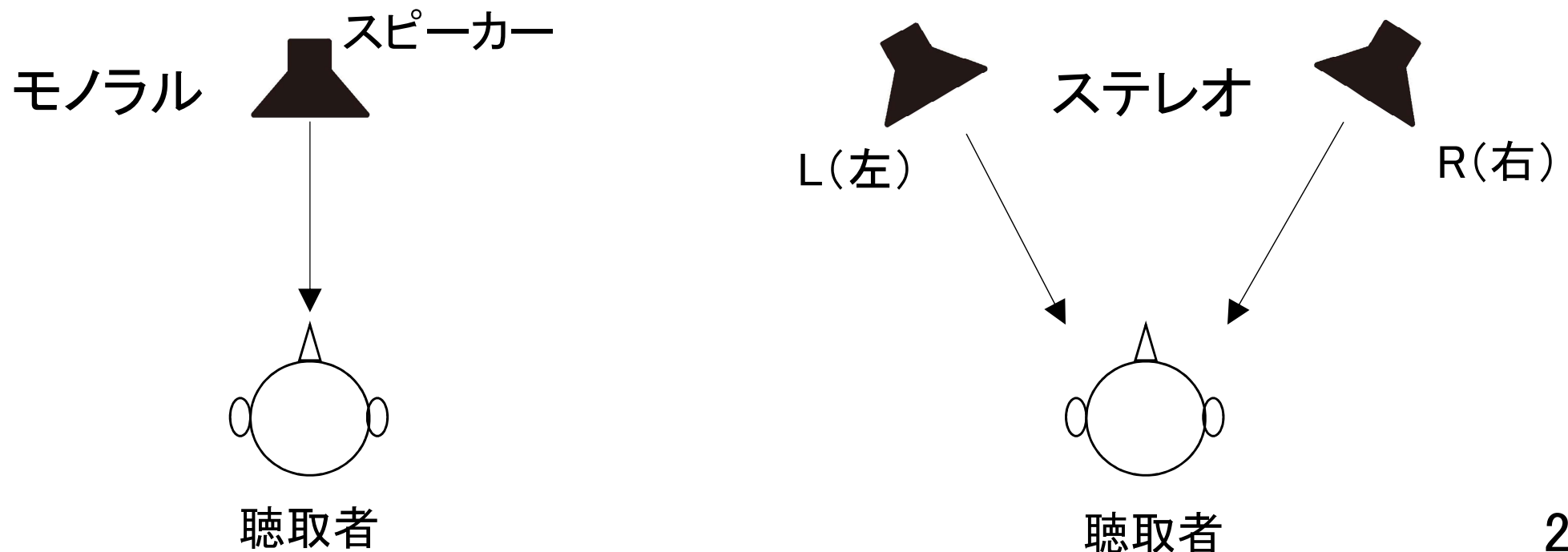
離散化のまとめ

- 時間に対する離散化: 標本化(サンプリング)
 - 振幅に対する離散化: 量子化
- ⇒両方が離散化された信号: デジタル信号

時間 \ 振幅	連続	離散
	連続	離散
連続	アナログ信号	多値信号
	(連続時間信号)	
離散	サンプル値信号	デジタル信号
	(離散時間信号)	

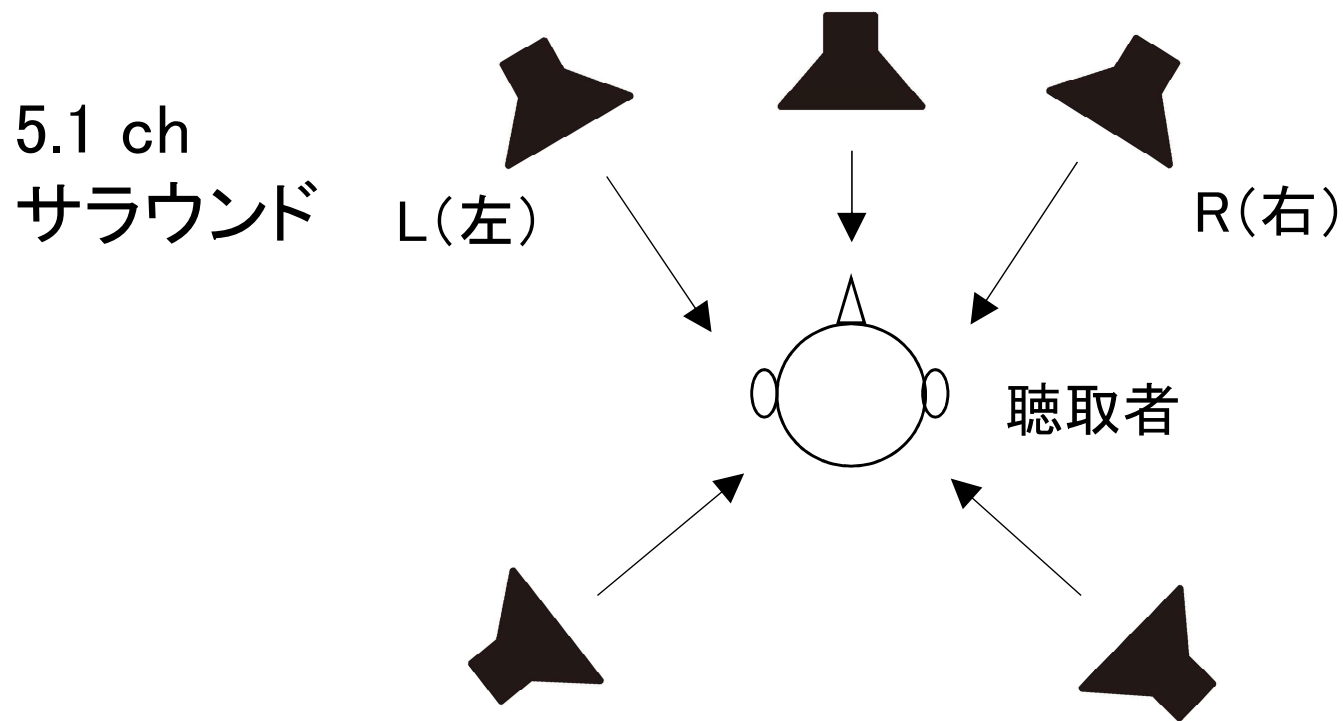
メディアの規格

- 音データは「モノラル」が基本
 - 1つのスピーカーを使って音を再生(1チャンネル)
- 2つ以上のスピーカーを使って音を再生する「マルチチャンネル」のものもある
 - 音楽CDは2チャンネルの「ステレオ」



メディアの規格

- 音データは「モノラル」が基本
 - 1つのスピーカーを使って音を再生(1チャンネル)
- 2つ以上のスピーカーを使って音を再生する「マルチチャンネル」のものもある
 - 音楽CDは2チャンネルの「ステレオ」



メディアの規格

- 音楽CDの規格

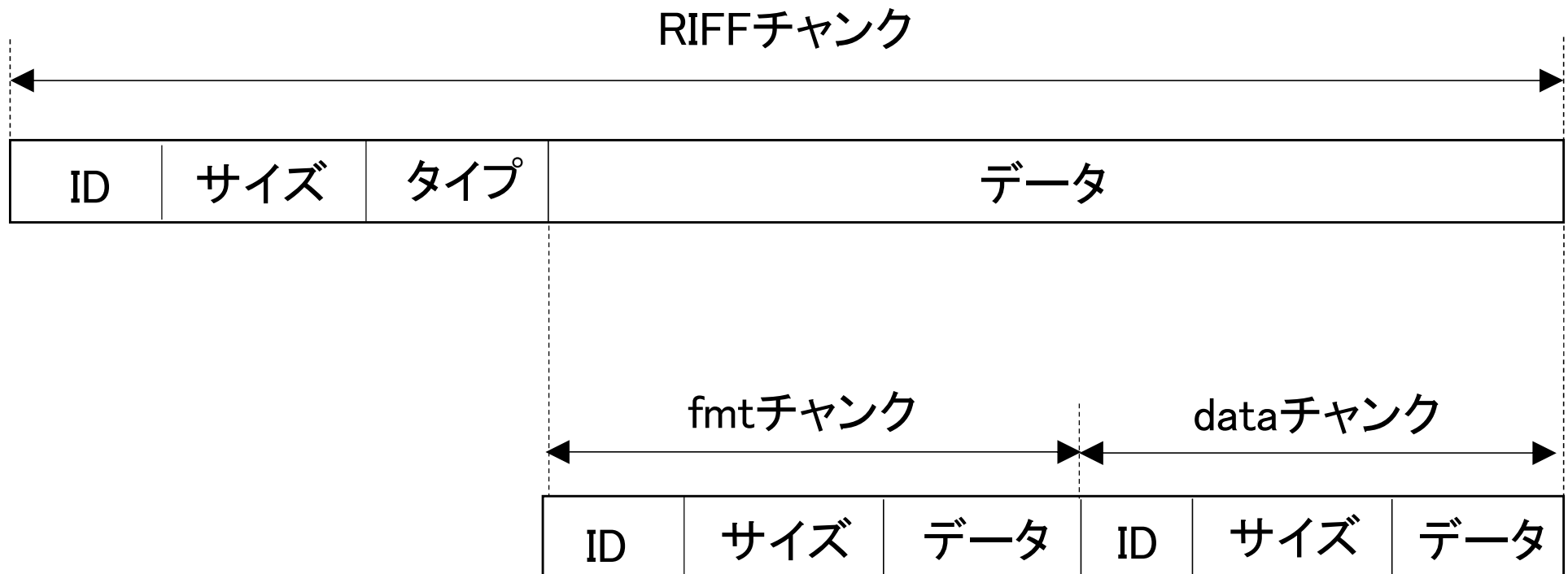
- サンプリング周波数: 16kHz
- 量子化ビット数: 16ビット
- チャンネル数: 2(ステレオ)

- DVD-Audioの規格

- サンプリング周波数: 192kHz
- 量子化ビット数: 24ビット

WAVEファイル

- 音データをコンピュータに記録するための標準のファイルフォーマットの1つ
⇒主にWindowsで用いられる(当然Macでも)
- 基本構造



WAVEファイルのパラメータ

• RIFFチャンク

パラメータ	サイズ (byte)	内容
riff_chunk_ID	4	'R' 'I' 'F' 'F'
riff_chunk_size	4	36 + data_chunk_size
file_format_type	4	'W' 'A' 'V' 'E'

※data_chunk_sizeはdataチャンクのサイズ

⇒riff_chunk_size以外は固定値

WAVEファイルのパラメータ

- fmtチャンク: 音データに関する情報を記述

パラメータ	サイズ (byte)	内容
fmt_format_type	4	'f' 'm' 't' ''
fmt_chunk_size	4	16
wave_format_type	2	PCMは1
channel	4	モノラルは1、ステレオは2
sample_per_sec	4	標本化周波数
bytes_per_sec	4	$\text{block_size} * \text{sample_per_sec}$
block_size	2	$\text{bits_per_sample} * \text{channel} / 8$
bits_per_sample	2	量子化ビット数

WAVEファイルのパラメータ

- dataチャンク: 音データそのものを記録

パラメータ	サイズ (byte)	内容
data_chunk_ID	4	'd' 'a' 't' 'a'
data_chunk_size	4	音データの長さ * channel
data	data_chunk_size	音データ

データの記録順序

- モノラルは先頭から順番に
 - $s(0), s(1), s(2), s(3), \dots$
- ステレオは先頭から順番かつ左右交互に
 - $s_L(0), s_R(0), s_L(1), s_R(1), s_L(2), s_R(2), \dots$

今日の講義内容まとめ

- アナログ信号とデジタル信号
 - アナログ信号: 振幅値も時間も連続的
 - デジタル信号: 振幅値も時間も離散的
 - デジタル信号の2つの離散化
 - 標本化: 時間方向に離散化
 - 量子化: 振幅方向に離散化
- サンプリング周波数、量子化ビット数、ダイナミックレンジ
- 音楽CDの規格
 - サンプリング周波数16kHz, 量子化ビット数16bit
 - 2チャンネル(ステレオ)
 - WAVEファイルの仕様

第2回
サウンドメディア論
および演習 演習編

Moodle

- コースを作りました
「サウンドメディア論及び演習2018前期」
- 登録キーは「sound-media」
- ファイルをアップロードしておきました
 - 講義資料(第2回講義資料.pdf)
 - C言語のプログラム(20180413.zip)
- アップロードされてなかったり、ダウンロードできない場合はこちら
 - 講義資料 <https://goo.gl/iM823P>
 - プログラム <https://goo.gl/ZAbAxK>

演習の準備

1. Moodleより、ex1.zipをダウンロードし、適当なフォルダの下に解凍 ⇒ ex1/ フォルダが出現
2. ex1フォルダに移動
 - /Users/tamamori/sound/20180413/ex1 ならば,
\$ cd ~/sound/ex1
3. 解凍して出てきたex1_1.cをコンパイル
\$ gcc ex1_1.c -o ex1_1
4. プログラムを実行
\$./ex1_1
5. 実行するとb1.wavが出力される

演習の準備

6. a1.wavとb1.wavを聞き比べてみる

⇒ターミナルからafplayコマンドを使う

```
$ afplay a1.wav
```

```
$ afplay b1.wav
```

同じ音になっていることを確認

プログラムの中身を見してみる

- ex1_1.cはMONO_PCM型の構造体によってモノラルの音データを取り扱っている
- MONO_PCM構造体 (wave.hを開いてみよう)
 - メンバ変数
 - fs : 標本化周波数 (int)
 - bits : 量子化ビット数 (int)
 - length : 音データの長さ (int)
 - s 音データ (double *)

プログラムの中身を見してみる

- ex1_1.cの処理の流れ

1. wave_read_16bit_mono関数を使ってa1.wavから読み取ったデータをpcm0構造体に格納
2. pcm1構造体に音データをコピー
3. wave_write_16bit_mono関数を使ってb1.wavに書き出す

演習

1. ex1_2.cをコンパイルし、実行せよ。

```
$ gcc ex1_2.c -o ex1_2
```

```
$ ./ex1_2
```

2. a2.wavと出力されたb2.wavを聞き比べてみよ。

```
$ afplay a2.wav
```

```
$ afplay b2.wav
```

※ステレオ音声データ用のSTEREO_PCM構造体や、
wave_read_16bit_stereo()関数が使われている

練習

1. ex1_2.cをコンパイルし、実行せよ。

```
$ gcc ex1_2.c -o ex1_2
```

```
$ ./ex1_2
```

2. a2.wavと出力されたb2.wavを聞き比べてみよ。

```
$ afplay a2.wav
```

```
$ afplay b2.wav
```

※ステレオ音声データ用のSTEREO_PCM構造体や、
wave_read_16bit_stereo()関数が使われている

デジタル信号の正弦波

$$s(n) = a \sin \left(\frac{2\pi f_0 n}{F_s} \right) \\ (0 \leq n \leq N - 1)$$

a : 振幅

f_0 : 正弦波の周波数
(基本周波数)

N : データの長さ

F_s : サンプリング周波数

練習

ex2_1.cは正弦波を作成するプログラムである。

1. ターミナルからex2フォルダに移動せよ

```
$ cd ../ex2
```

2. ex2_1.cをコンパイルし、実行せよ

```
$ gcc ex2_1.c -o ex2_1
```

```
$ ./ex2_1
```

3. ex2_1.wavを再生し、聞いてみよ

```
$ afplay ex2_1.wav
```

ex2_1.cの解説

$$s(n) = a \sin \left(\frac{2\pi f_0 n}{F_s} \right)$$

F_s を44.1kHz, a を0.1, f_0 を500Hz

N を44100として正弦波を作成するプログラム

⇒音の長さは $1/44100 \text{ Hz} \times 44100 = 1 \text{ 秒}$

練習

1. ex2_2.cをコンパイルし、実行せよ

```
$ gcc ex2_2.c -o ex2_2
```

```
$ ./ex2_2
```

2. ex2_2.wavを再生し、聞いてみよ

```
$ afplay ex2_2.wav
```

練習

- ex2_3.c : gnuplotによる波形表示機能を追加

1. ex2_3.cをコンパイルし、実行する

※最初は標本化周波数 30Hz, データ長30,
基本周波数1.0Hzとなっている。

2. グラフが表示されることを確認せよ

練習

- ex2_4.c : gnuplotによるグラフの画像ファイル保存の機能を追加
 1. ex2_4.cをコンパイルし、実行する
 2. グラフが出力されることを確認せよ
 2. `$ open ex2_4.png` をターミナルに打ち込み、PNG画像を開き、gnuplotのグラフと同じものが出力されていることを確認せよ。

以降、gnuplotの出力結果を保存するプログラムを書く際にはex2_4.cを参考にするとよい

演習課題1

- 以下を行え

1. ex1_1.cをex1_3.cとしてコピー
2. 出力ファイルの標本化周波数が8000となるよう、ex1_3.cを修正する。入力ファイル名はsample_ja.wav、出力ファイル名はsample_ja_8kHz.wavとせよ。
3. コンパイルして実行し、sample_ja_8kHz.wavを聞いてみる。sample_ja.wavとは違って聞こえるはずである。その理由を答えなさい。

演習課題2

- 以下を行え

1. ex2_1.c を ex2_5.cとしてコピーせよ
2. ex2_5.cを編集し、実行せよ。

ただし、標本化周波数およびデータ長は学籍番号、振幅は小数点以下を自分の誕生日の日付とする。基本周波数は自分の学籍番号に基づく(下記参照)。出力ファイル名はex2_5.wavとする。

例: 4月13日生まれで学籍番号K16114ならば、
標本化周波数とデータ長は16114, 振幅は0.13, 基本周波数は1611.4

演習課題3

- 以下を行え

1. ex2_4.cをex2_6.cとしてコピーせよ
2. ex2_6.cを編集し、基本周波数を0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0など、色々と変えて実行し出力結果を確認せよ。振幅の値は何でも良い。それぞれの実行結果の意味を考えなさい(異なる結果になるのはなぜか?)
3. 自分の学籍番号の下二桁を基本周波数の小数第1位と第2位としてプログラムを実行せよ。整数部は1とする。出力画像ファイル名はex2_6.pngとする。

例: 学籍番号 K16113ならば、1.13

提出するもの

- 演習課題1

- ex1_3.c
- sample_ja_8kHz.wav
- sample_ja_8kHz.wavが違って聞こえる理由を書いたテキストファイル

- 演習課題2

- ex2_5.c
- ex2_5.wav

- 演習課題3

- ex2_6.c
- ex2_6.png

提出方法

1. フォルダを作成

フォルダ名は「学籍番号_今日の日付」とする

例: 学籍番号がK123456で4月13日ならば
「K123456_0413」を作成する

2. 提出するファイルをその中に入れる

3. Finder上でフォルダをCtrl+左クリックし、圧縮ファイル(zip)を作成する

4. Moodleにアップロードして課題提出

連絡先

- 課題提出に関して何か不具合や問題点などありましたら、

akira-tamamori@aitech.ac.jp

までお願いします。