Digital Signal Processing

Instructor: Ninh Khánh Duy

Embedded System Division, IT Faculty,

Danang University of Science and Technology

Email: ninhkhanhduy@gmail.com

Facebook: Ninh Khánh-Duy

Goals

- **Knowledge:**
  - Tools such as convolution, Z-transform (ZT), Discrete-Time Fourier Transform (DTFT) and Discrete Fourier Transform (DFT)
  - Digital signal and system analysis and its apps

- **Skills:**
  - Matlab (or Python) programing
  - Teamwork: 4 students/group for mini-projects
  - Writing & presenting report

Grading

- **Exercise: 25%**
- **Midterm: 25%**
- **Final exam: 50%**

Exercise and Midterm are computer-based tests using MATLAB or Python

References

- PDF files shared via Gdrive:
  - Bài giảng Xử lý tín hiệu số
  - Book: Digital signal processing - Principles, algorithms, and applications, Proakis & Manolakis, 1995 (& its solution)
  - Matlab & Python books
- Books at Resource Information Center:
  - Xử lý tín hiệu số, Quách Tuấn Ngọc, 1999
  - Xử lý tín hiệu và lọc số, Nguyễn Quốc Trung, 2006

Signal processing programming

- Matlab: easy, but commercial software
  - Self-studying Matlab: https://www.mathworks.com/help/matlab/index.html
  - Signal Processing Toolbox (chỉ cần cài SP toolbox và Matlab): https://www.mathworks.com/help/signal/
- Python is not as easy as Matlab, but open-sourced
  - Self-studying Python & its libraries: http://cs231n.github.io/python-numpy-tutorial/
  - Signal processing with SciPy: https://docs.scipy.org/doc/scipy/reference/signal.html
  - Setting up Python environment with Anaconda: https://machinelearningmastery.com/setup-python-environment-machine-learning-deep-learning-anaconda/

Schedule

- Chapter 1: Introduction to Digital Signal Processing
- Chapter 2: Discrete-time Signals and Systems
- Chapter 3: Z-transforms and LTI system analysis
- Chapter 4: Discrete-time Fourier Transform (DTFT)
- Chapter 5: Discrete Fourier Transform (DFT)
- Mini-projects (intended): Week 7-8, Week 14-15

Chapter 1: Introduction to Digital Signal Processing

Outline:

1. Signals
2. Digital Signal Processing (DSP) systems
3. Why DSP?
4. DSP applications

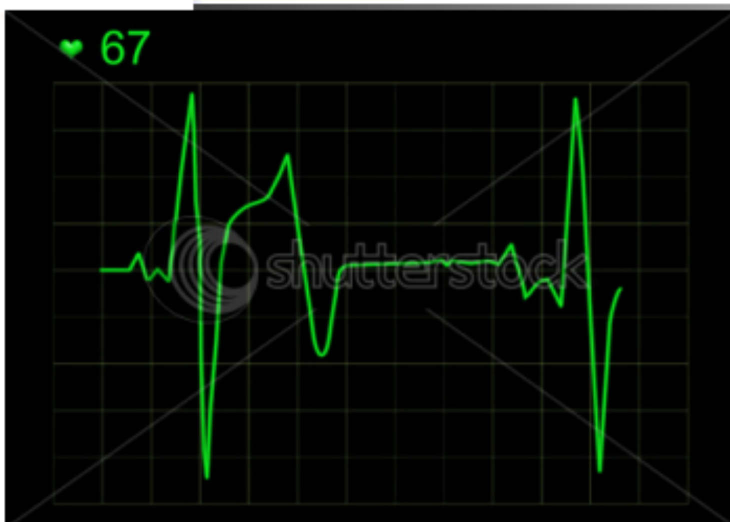
Signals

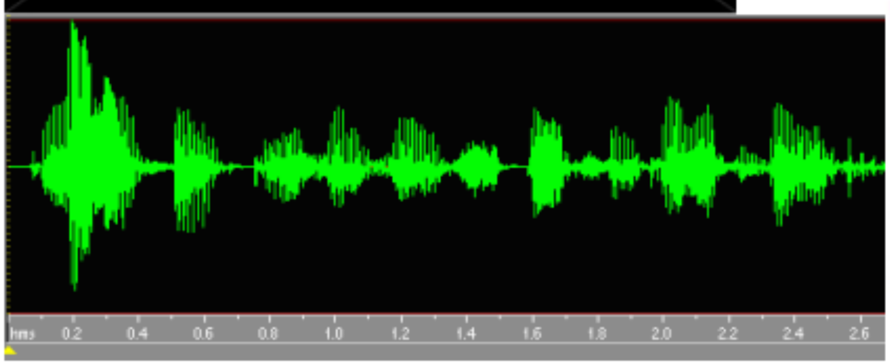
- Physics: Physical representation of information
- Math: Function of independent variables such as time, distance, position, temperature
- Objects that convey information
- **Examples:**
  - 1D signal: speech, music, biosensor…
  - 2D signal: image
  - 3D signal: video (2D image + time)

D = dimensional

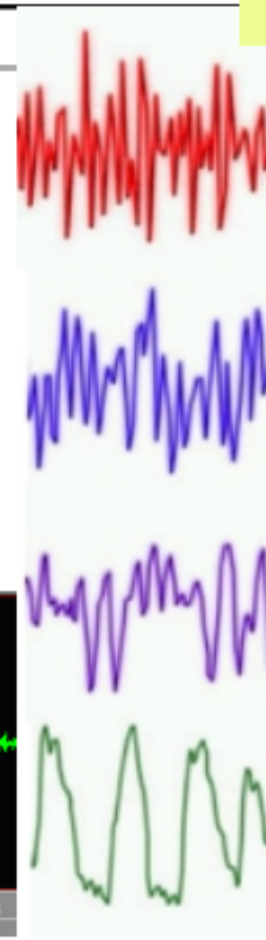
1D signals: f(t)

ECG





EEG



Beta 15-30 Hz

Awake, normal alert consciousness

Alpha 9-14 Hz

Relaxed, calm, meditation, creative visualisation

Theta 4-8 Hz

Deep relaxation and meditation, problem solving

Delta 1-3 Hz

Deep, dreamless sleep

2D image signals: f(x,y)

Grey image

Color image

3D video signals f(x,y,t)

1st step in DSP: signal acquisition

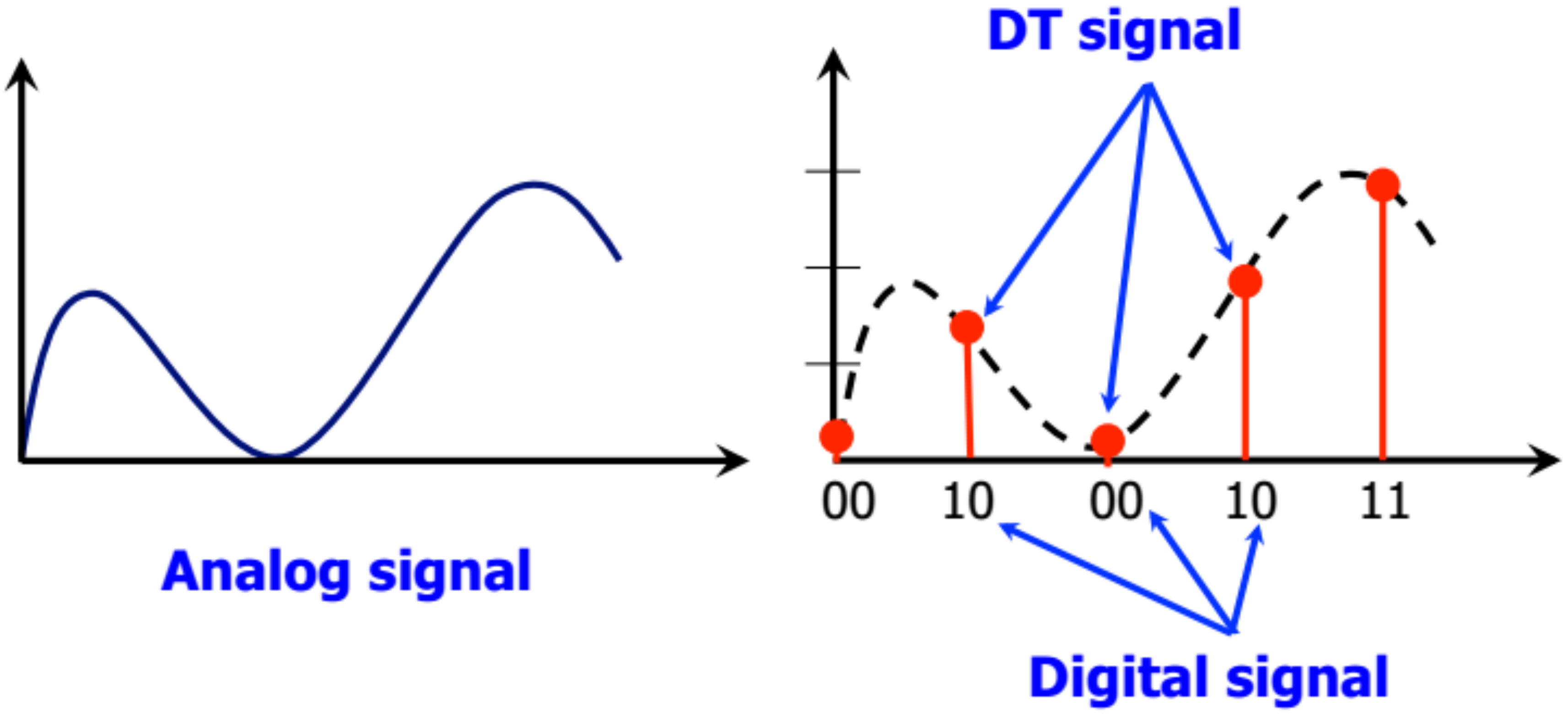
- We need devices for converting non-measurable quantities into measurable ones (often, electrical signals)
- Transducer:
  - Microphone: audio wave → voltage signal
  - Sensors

Only 1D signal, dimension is time

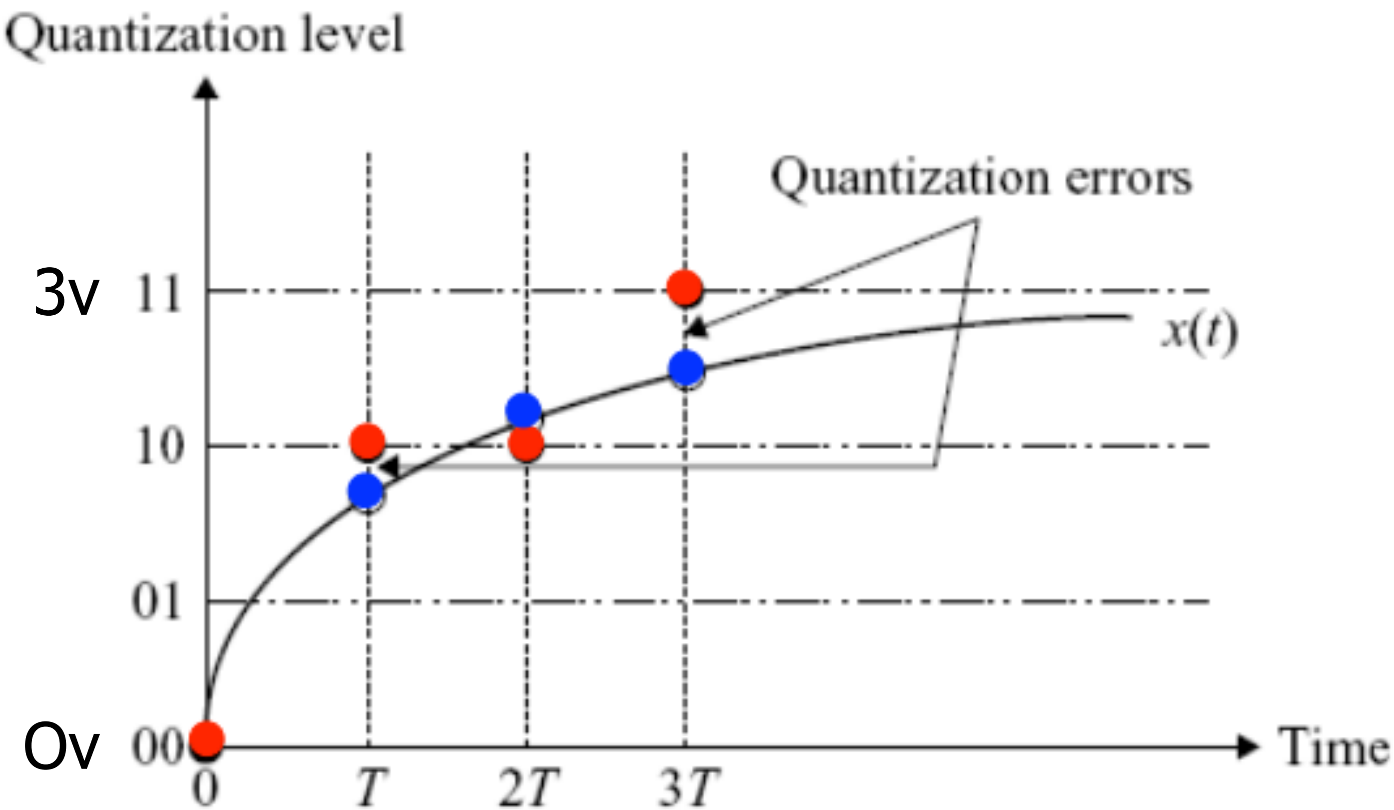
Discrete-time (DT) signal vs. continuous-time (CT) signal

- **Continuous-time signal:**
  - its magnitudes (amplitudes/values) define for a continuous duration of time
  - sound, voice…
- **Discrete-time signal:**
  - define only for discrete points in time (hourly, every second, …)
  - an image in computer, a recorded audio file
  - amplitude could be discrete or continuous
  - if the amplitude is also discrete, the signal is digital

Analog signal vs. digital signal



Analog signal vs. digital signal



Quantization

T: time interval between two consecutive samples

2^{nd} step in DSP: Digitalization of signals

- Process converting contiguous signal to digital one
- Include 3 steps:
  - Discretizing time axis (Sampling: lấy mẫu)
    - Sampling period (chu kỳ lấy mẫu T)
  - Discretizing amplitude axis (Quantization: lượng tử hoá)
    - Precision/Resolution (number of quantization levels = 2^N, N: number of bits to store a sample value)
  - Coding: convert quantization levels into binary codes

Chapter 1: Introduction to Digital Signal Processing

Outline:

1. Signals
2. Digital Signal Processing (DSP) systems
3. Why DSP?
4. DSP applications

What is Digital Signal Processing?

- Represent a signal by a sequence of numbers (called a "discrete-time signal" or "digital signal")
- Modify this sequence of numbers by a computing process to change or extract information from the original signal
- The "computing process" is a system that converts one digital signal into another — it is a "discrete-time system" or "digital system"

An example of DSP

- From a discrete-time input signal:
  x[n] = **{ 1, 2, 4, -9, 5, 3 }**
- Create a discrete-time output signal:
  y[n] = **{ 1/3, 1, 7/3, -1, 0, -1/3, 8/3, 1 }**
- What is the relation between input and output

	1	2	4	-9	5	3			x[n]
+		1	2	4	-9	5	3		x[n-1]
			1	2	4	-9	5	3	x[n-2]
	1	3	7	-3	0	-1	8	3	3y[n]

--> System: y[n]=1/3(x[n]+x[n-1]+x[n-2])

DSP implementation

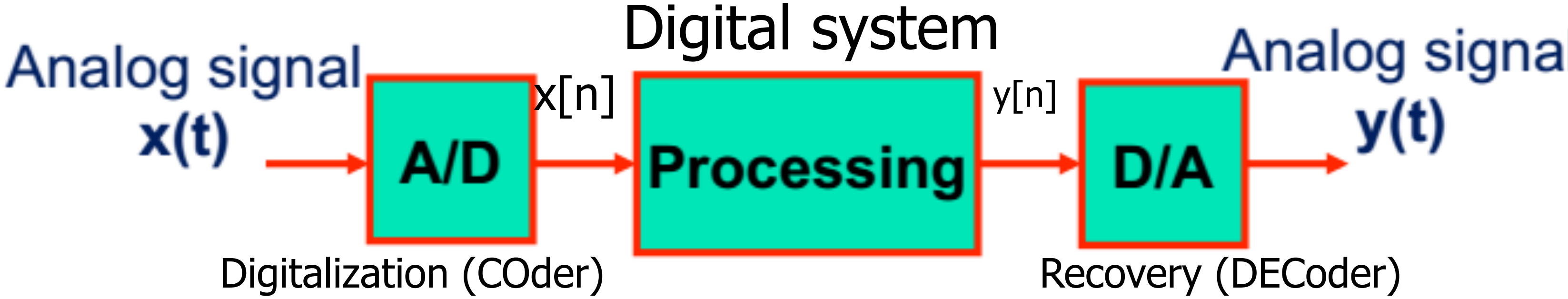
- Use basic operations of addition, multiplication and delay (shift)
- Combine these operations to accomplish processing: a discrete-time input signal → another discrete-time output signal

Signal processing system

Analog system

Analog signal processing

Digital system



Digital signal processing

Digital processing of analog signals

DSP implementation options

- Special-purpose (custom) chips: application-specific integrated circuits (ASIC)
- Field-programmable gate arrays (FPGA)
- General-purpose microprocessors or microcontrollers (µP/µC)
- General-purpose digital signal processors (DSP processors)
- DSP processors with application-specific hardware (HW) accelerators

DSP implementation options

	ASIC	FPGA	$\mu P/\mu C$	DSP processor	DSP processors with HW accelerators
Flexibility	None	Limited	High	High	Medium
Design time	Long	Medium	Short	Short	Short
Power consumption	Low	Low–medium	Medium–high	Low–medium	Low–medium
Performance	High	High	Low–medium	Medium–high	High
Development cost	High	Medium	Low	Low	Low
Production cost	Low	Low–medium	Medium–high	Low–medium	Medium

Chapter 1: Introduction to Digital Signal Processing

Outline:

1. Signals
2. Digital Signal Processing (DSP)
3. Why DSP?
4. DSP applications

Advantages of DSP

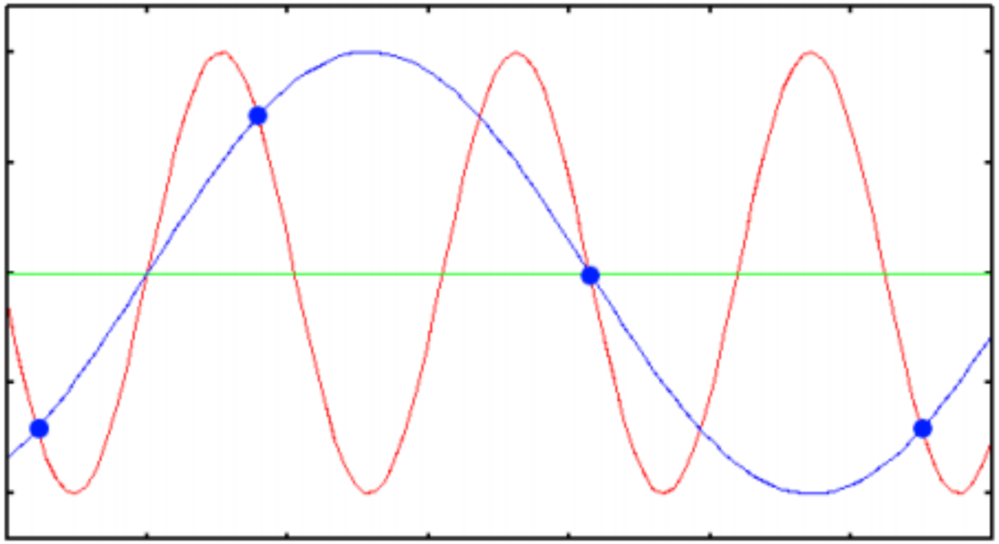
- Flexible: re-programming ability
- More reliable
- Smaller, lighter → less power
- Easy to use, to develop and test (by using the assistant tools)
- Suitable to sophisticated applications

Limitations of DSP

- **Aliasing in sampling**: taking samples at intervals and don't know what happens in between

→ can't distinguish higher and lower frequencies

→ to avoid: **sampling theory**

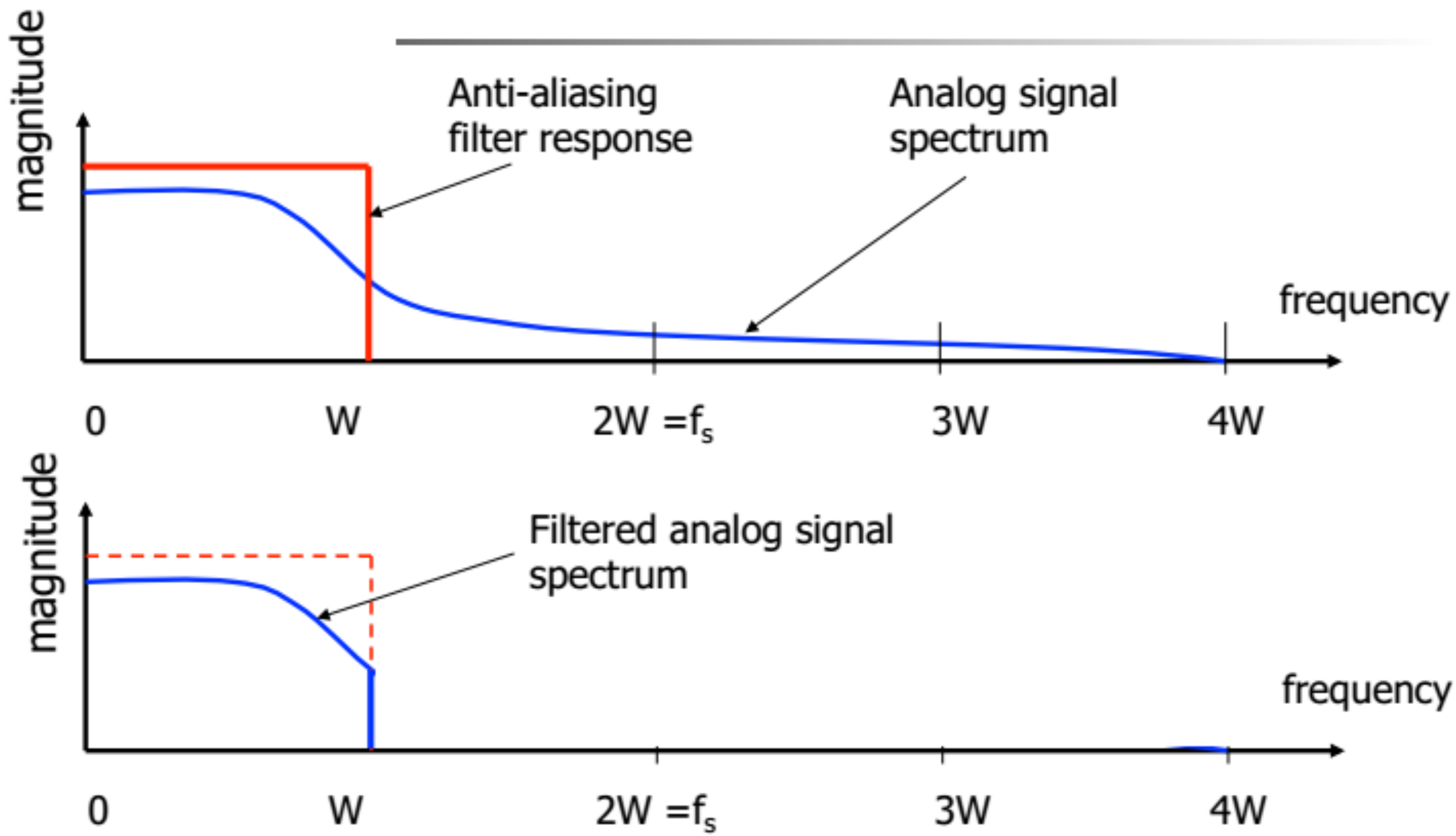


Nyquist/Shannon sampling theory

- Goal: To guarantee that an analog signal can be perfectly recovered from its sample values (discrete signal)
- Theory: an analog signal with maximum frequency of *B* (Hz) must be sampled *at least 2B times per second*
- Nyquist sampling rate: *minimum sampling frequency* ($f_{s\,min} = 2B$ *Hz)* fs = 1/T (T: sampling interval)
- Nyquist frequency ($f_s/2$): max. freq. can be recovered
- To remove all signal elements above the Nyquist frequency → anti-aliasing filter

Eg.: audio signals has B=22kHz → fs min=2x22kHz=44kHz=44k samples/second

Anti-aliasing filter



Some examples of sampling frequency

- Speech coding/compression ITU G.711, G.729, G.723.1 (telephony)

  $f_s = 8$ kHz → $T = 1/8000$ s $= 125\mu s$

- Audio CDs:

  $f_s = 44.1$ kHz → $T = 1/44100$ s $= 22.676\mu s$

- Audio hi-fi, e.g., MPEG-2, AAC, MP3:

  $f_s = 48$ kHz → $T = 1/48\ 000$ s $= 20.833\mu s$

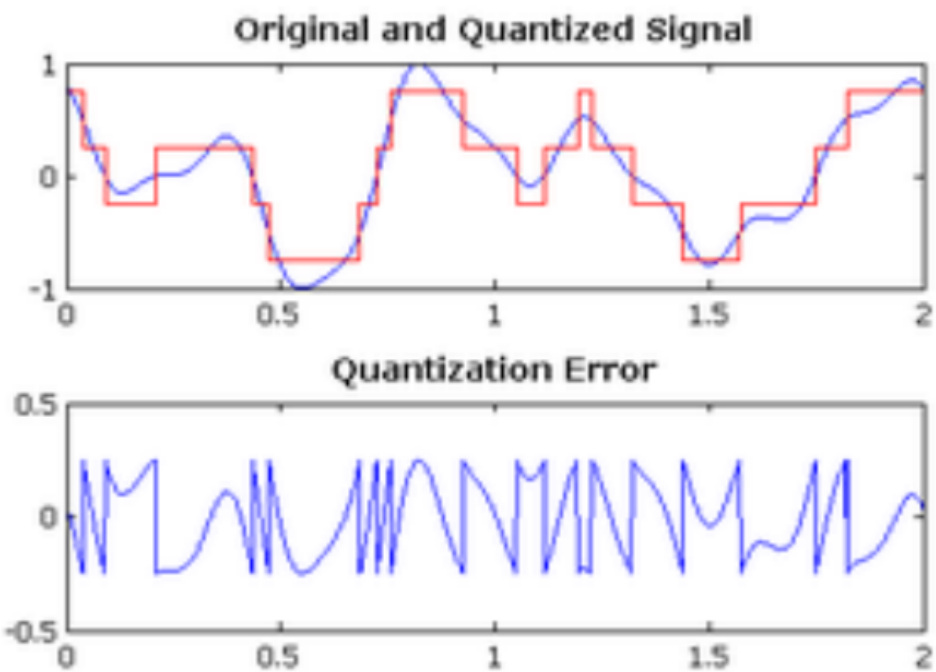
Homework: Record your voice with the same sentence using Audacity software with 3 fs: 8kHz, 16kHz, 44.1kHz. After that, playback three signals & give your remarks about their quality.

Limitations of DSP

- **Quantization error**: due to the limited number of bit available

→ smoothly varying signal represented by stepped waveform

→ limited precision in data storage and arithmetic



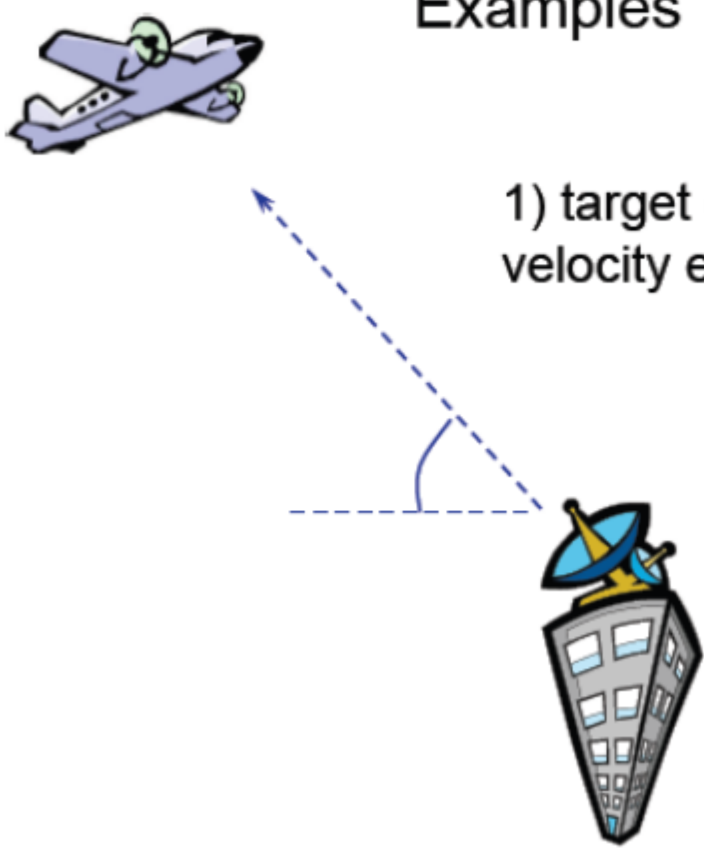
Chapter 1: Introduction to Digital Signal Processing

Outline:

1. Signals
2. Digital Signal Processing (DSP)
3. Why DSP?
4. DSP applications

DSP applications - radar

Examples



1) target detection – position and velocity estimation

2) tracking

DSP applications - biomedical

- Analysis of biomedical signals, diagnosis, patient monitoring, preventive health care

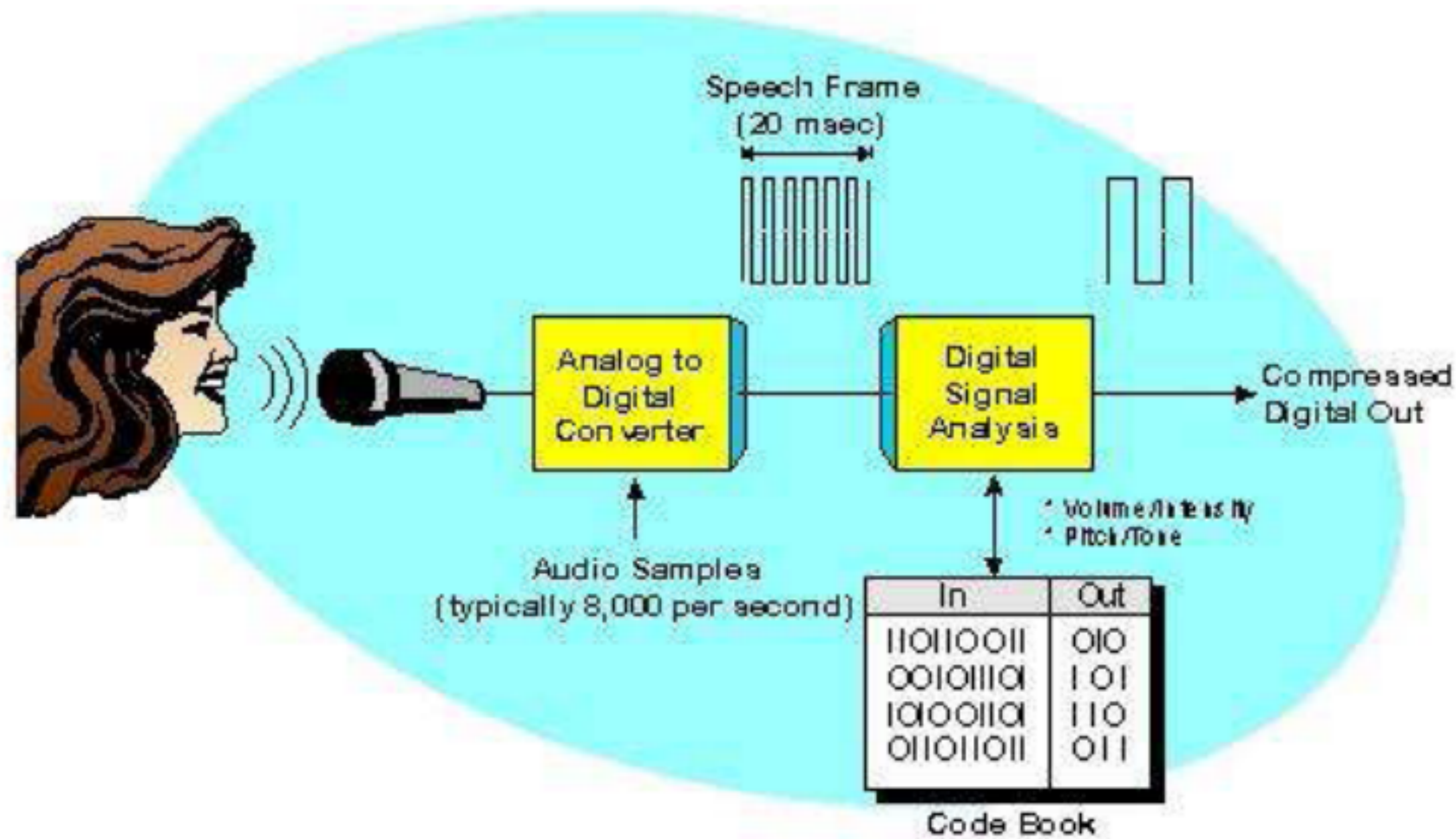
Examples:

1) electrocardiogram (ECG) signal – provides doctor with information about the condition of the patient's heart

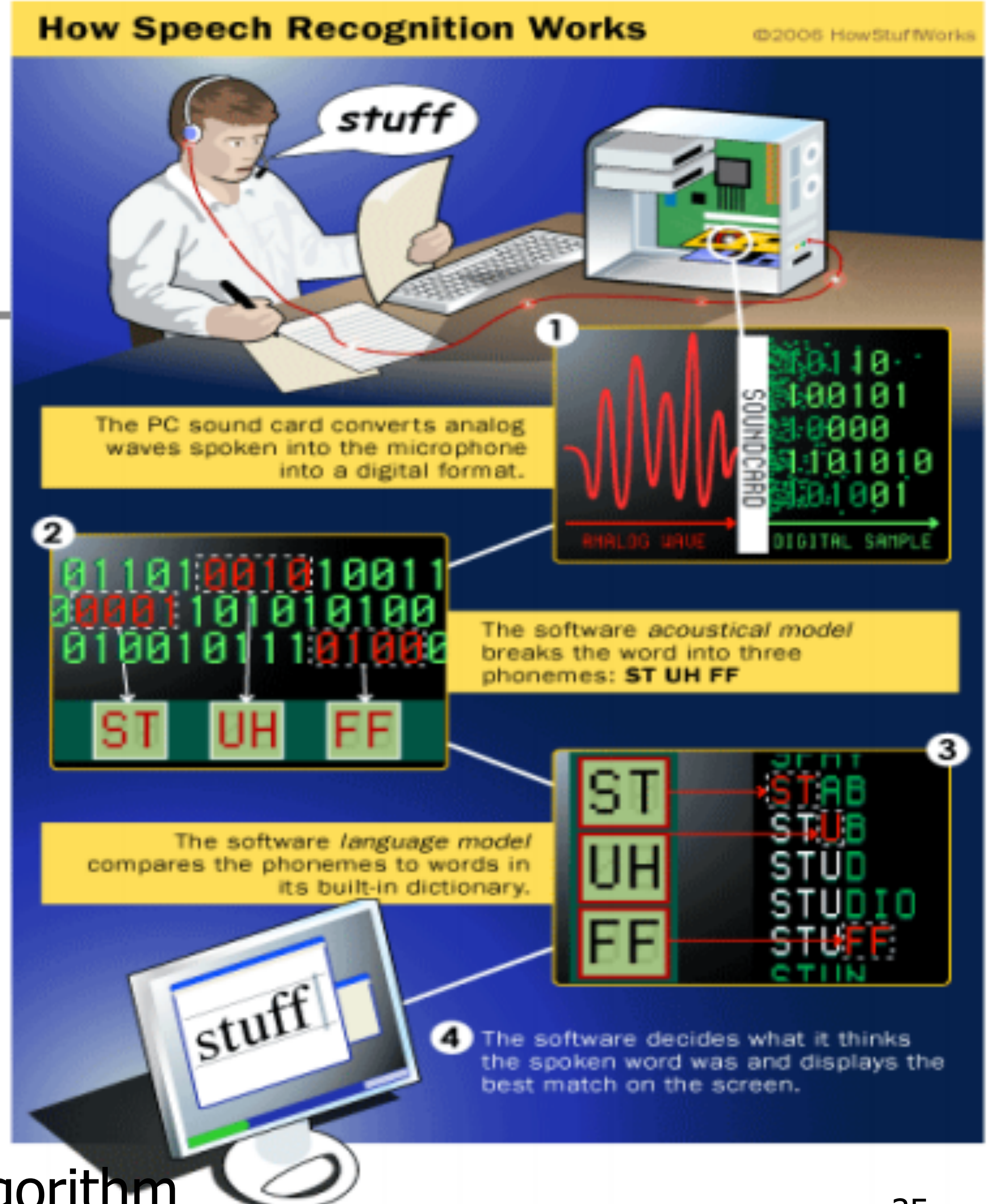
2) electroencephalogram (EEG) signal – provides Information about the activity of the brain

DSP applications – speech compression

(reduce much storage w/ little distortion)



DSP applications – speech recognition (speech-to-text)



Recogniton =
Feature extraction + AI algorithm

DSP applications - communication

- **Digital telephony**: transmission of information in digital form via telephone lines, mobile phone

MODEM = MOdulation (D/A) + DEModution (A/D)

DSP applications – image processing

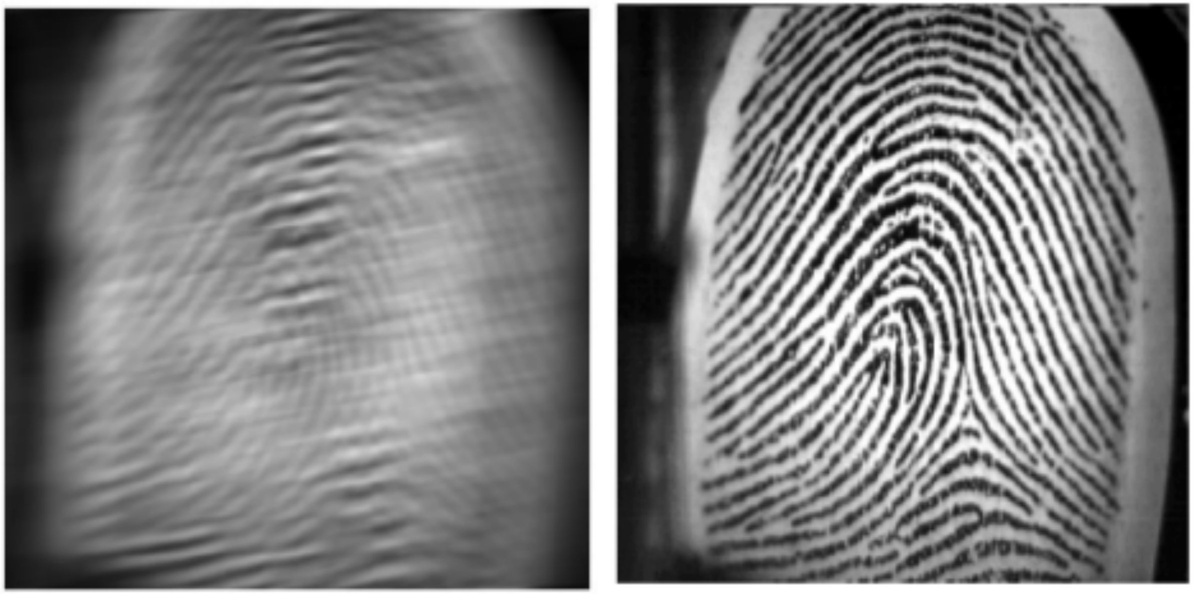
- Image compression: reducing the redundancy in the image data

487x414 pixels,
Uncompressed, 600471 Bytes,24 bpp

487x414 pixels
41174 Bytes, 1.63 bpp, CR=14.7

DSP applications – image processing

- **Image restoration (enhancement)**: reconstruct a degraded image using a priori knowledge of the degradation phenomenon



DSP applications - music

Recording, encoding, storing

Playback

Manipulation/mixing

DSP applications - noise removal