CAN207 Continuous and Discrete Time Signals and Systems

Lecture-1

Introduction to Signals and Systems

Zhao Wang

Zhao.wang@xjtlu.edu.cn

Room EE322

Module Information

- Module Code: CAN207
- Module Title: Continuous and Discrete Time Signals and Systems
- Module Credit: 5 credits
- Module Teachers: Zhao Wang, Qing Liu
  - Office: EE322
  - Tel: 88161421
  - Email: zhao.wang@xjtlu.edu.cn
  - Office hour: Thu. 9:00-11:00, Fri. 13:00-15:00

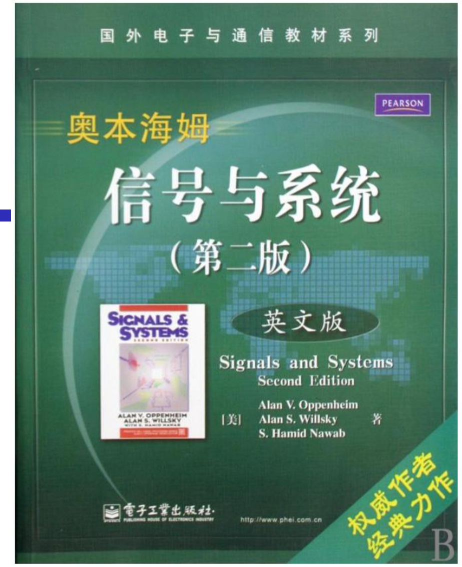
Module Information

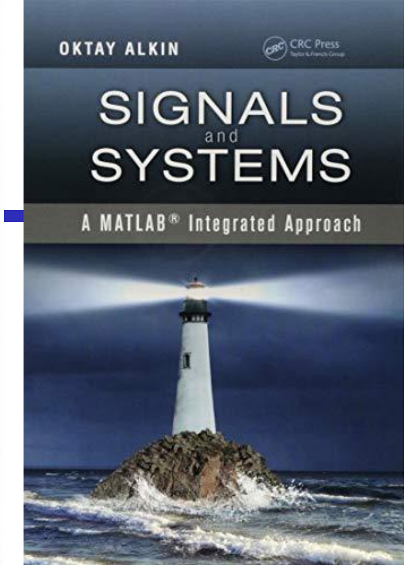
- Two assignments (each one for 15%):
  - One week for completion;
  - General feedback.
- Final Exam (3 hours, 70%)
  - Close book exam, but equation list will be provided;
- Resit Exam (3 hours, 100%)
  - Same as final exam.

Important Notices

- Recording and broadcasting lectures.
  - The lectures will be recorded, and released every week. Please be noticed that the recording are a resource to help study for the exam and not a substitute for attending class. And of course, since all students should be on-site, there won't be broadcasting link provided.
- Attendance policy.
  - The attendance will be recorded, and there will be consequences. Poor attendance could result in losing the opportunity to resit the module if failed in the final. Since our lectures run in two sessions, please make sure you are attending the correct one. If you have conflict timetabling issue, please contact the Registry to move you to another session.
- Week 7.
  - Week 7 will be treated as a non-teaching week, which mean we will not deliver new knowledge than. A tutorial or revision class might be arranged as optional class (no attendance required).
- Academic Integrity.
  - Presenting someone else's ideas as your own is a serious academic offense with serious consequences. There will be no tolerance, and cases detected will be reported to the department and school exam officers. Records of offences will be kept in system for your whole college life. Please be responsible for your own academic career.

Resources





- Learning Mall:
  - Lecture materials and recorded lectures
  - Self-practice problems and answers
  - External links to useful resources

- Notion Course Page:
  - https://furtive-may-686.notion.site/CAN207-Home-Page-e5d42a7a5cf846f783a18c961d20ea7a

- Reference books
  - 1. A.V.Oppenheim, Signals and Systems 2nd, Prentice Hall, 1997;
  - 2. Oktay Alkin, Signals and Systems: A MATLAB® Integrated Approach 1st, CRC Press, 2016.

What's this module about?

- The world is made up of three fundamental elements: ***matter***, ***energy***, and ***information***.
  - **Signals**, which carry both energy and information, are intangible and can take on various forms, particularly in the form of electricity.
  - **Systems** are responsible for processing signals and driving change.

- In the Signals and Systems course, we construct a model of signals that disregards their material aspects and instead focuses on their energy, information, and how they change when affected by a system.
  - The front end -> electrical knowledge, emphasizing energy.
  - The back end -> information theory, focusing on the transmission and processing of information.

Why do we learn this module?

- ***Signals and Systems*** is a crucial foundational module, regardless of the specific discipline.
  - It helps us understand the relationship between an unknown system's input and output, which is essential in studying various problems like semiconductor devices, analog/digital/RF circuits, communication systems, and radar systems.

- Signals and Systems provide us with a basic framework of thinking and theoretical tools that can inspire solutions to specific problems in different disciplines.
  - The module consists of two vital components: signal analysis and signal processing (i.e. systems functions).
  - This course equips us with fundamental mathematical tools that are essential for subsequent courses, while also assigning physical meanings to each mathematical derivation.

Background knowledge?

- **Mathematics foundation**:
  - The fundamental concepts of calculus, including limits, differentiation, and integration, with a focus on the method of integration by parts. Additionally, an understanding of differential equations and difference equations is necessary. Students should also be familiar with trigonometric functions and Euler's formula, as well as basic operations involving complex numbers.

- **Electrical foundation**:
  - A solid understanding of circuit analysis is crucial, including the basic characteristics of R, L, and C components, and the application of Kirchhoff's laws (KCL/KVL) to analyze circuits.

- **Matlab background (optional):**
  - Matlab will be used for practical exercises. Knowing some fundamentals of Matlab will facilitate the understanding of some mathematical equations in this module.

Course outline

- Signals and Systems
- Linear systems
  - LTI-system
  - Impulse response
  - Convolution of signals
- Fourier series
  - Fourier Series
  - Spectrum of a signal
- Fourier transform
  - Fourier transform
  - Properties of Fourier transform
- Laplace transform
  - Laplace transform and ROC
  - Properties of Laplace transform
- System stability
  - Frequency response and transfer function
  - Feedback systems
  - BIBO stability
- Z-transform
  - Z-transform and ROC
  - Properties of Z-transform
  - Inverse Z-transform
- DTFT and DFT
- Discrete system analysis
  - Implementation
  - Zero-pole plot
- Sampling Theory

How to learn this module?

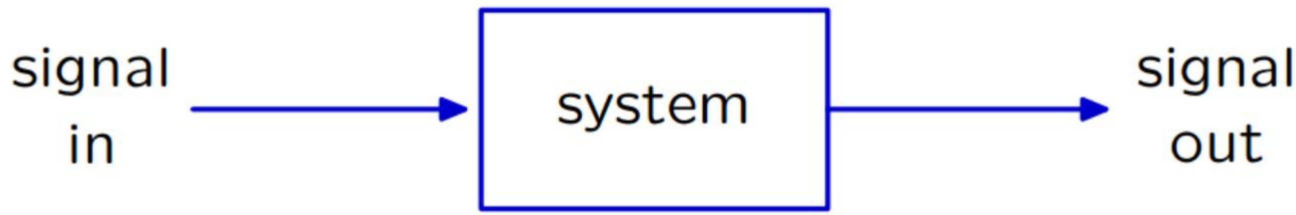
- To effectively approach this course, it is recommended to create a **framework or mind map** of the entire course at the beginning.
  - This will help organize and update the content on a weekly basis, facilitating a comprehensive understanding of the subject matter.
- In-depth study of **basic concepts and derivations** is highly encouraged.
  - Whether covered in class or not, actively deriving concepts on your own will deepen your understanding and enhance your memory retention.
- Regular **practice** is essential, as this course relies heavily on mathematical principles. Without practice, understanding remains superficial.
- **Matlab** will be used for practical exercises.
  - Although not mandatory, completing optional Matlab exercises will aid in comprehension and lay a foundation for future courses (MEC208, CAN202, etc.).

Introduction

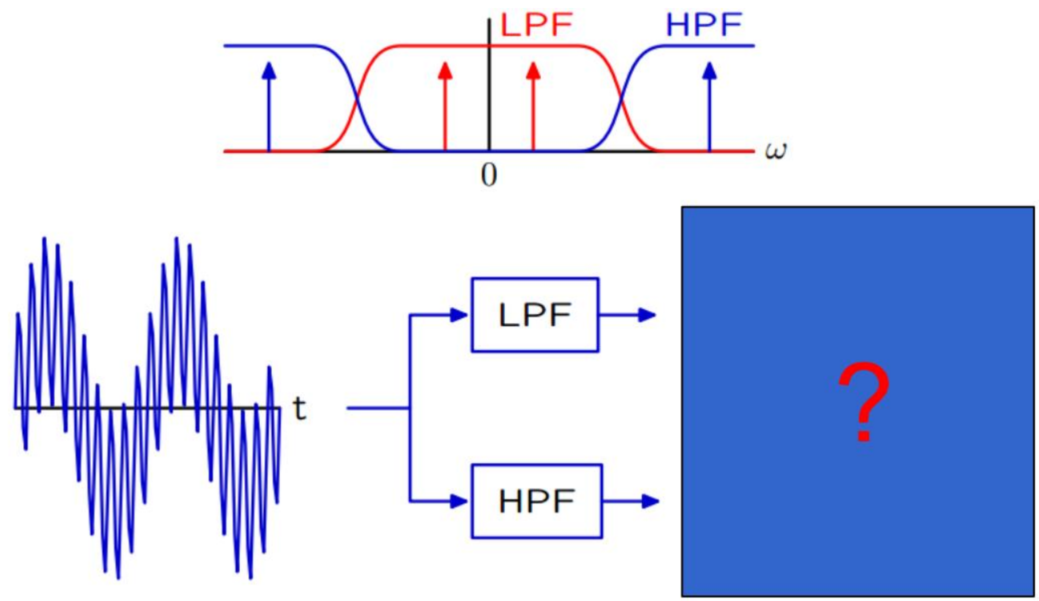
- Signals and Systems Abstraction
- Start from several examples…
- What is a signal?
- What is a system?
- Continuous or Discrete?
- Implementation of the systems

1. Signals and Systems Abstraction

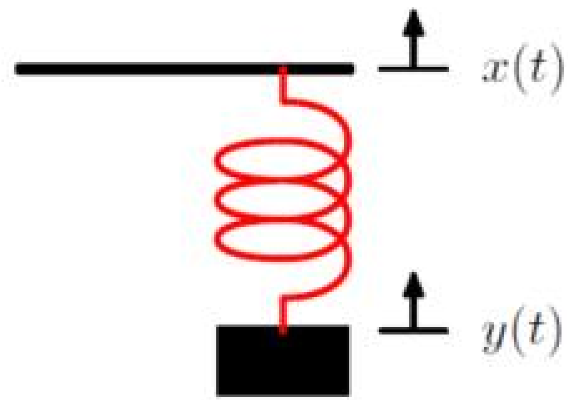
- Describe a system (physical, mathematical, or computational) by the way it transforms an input signal into an output signal.

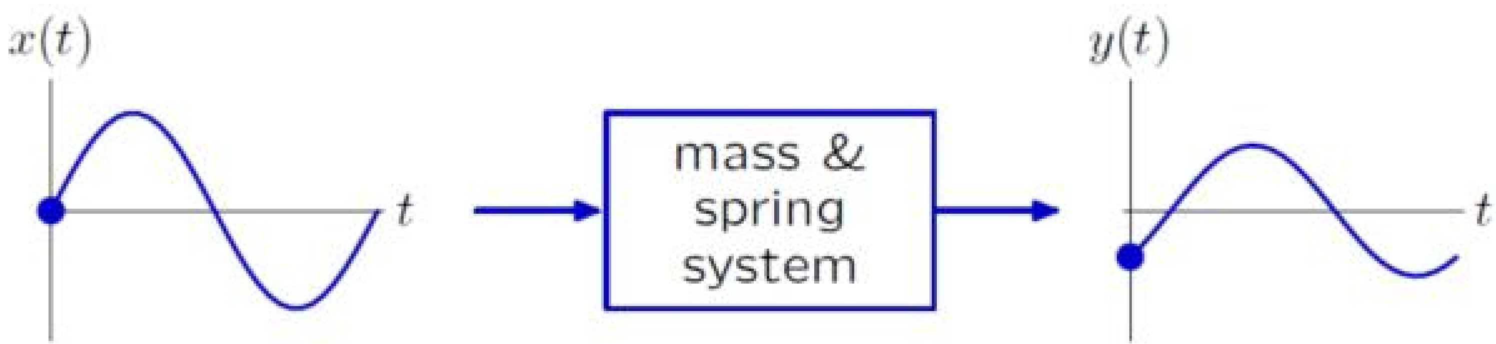


- One example…
  - Systems can be designed to ***selectively*** pass certain frequency bands.
  - Examples: low-pass filter (LPF) and high-pass filter (HPF).

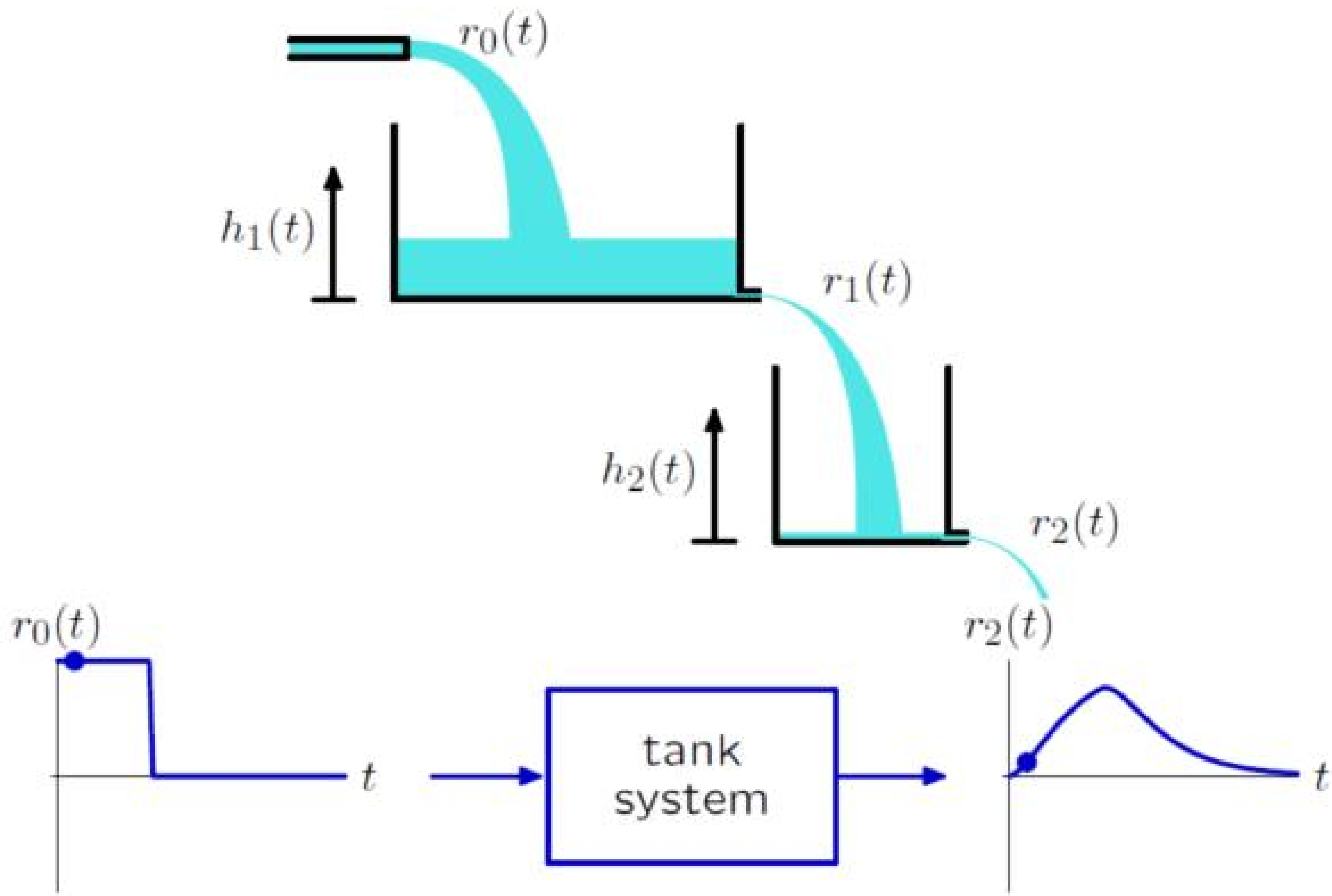


2. Example: Mass and Spring system

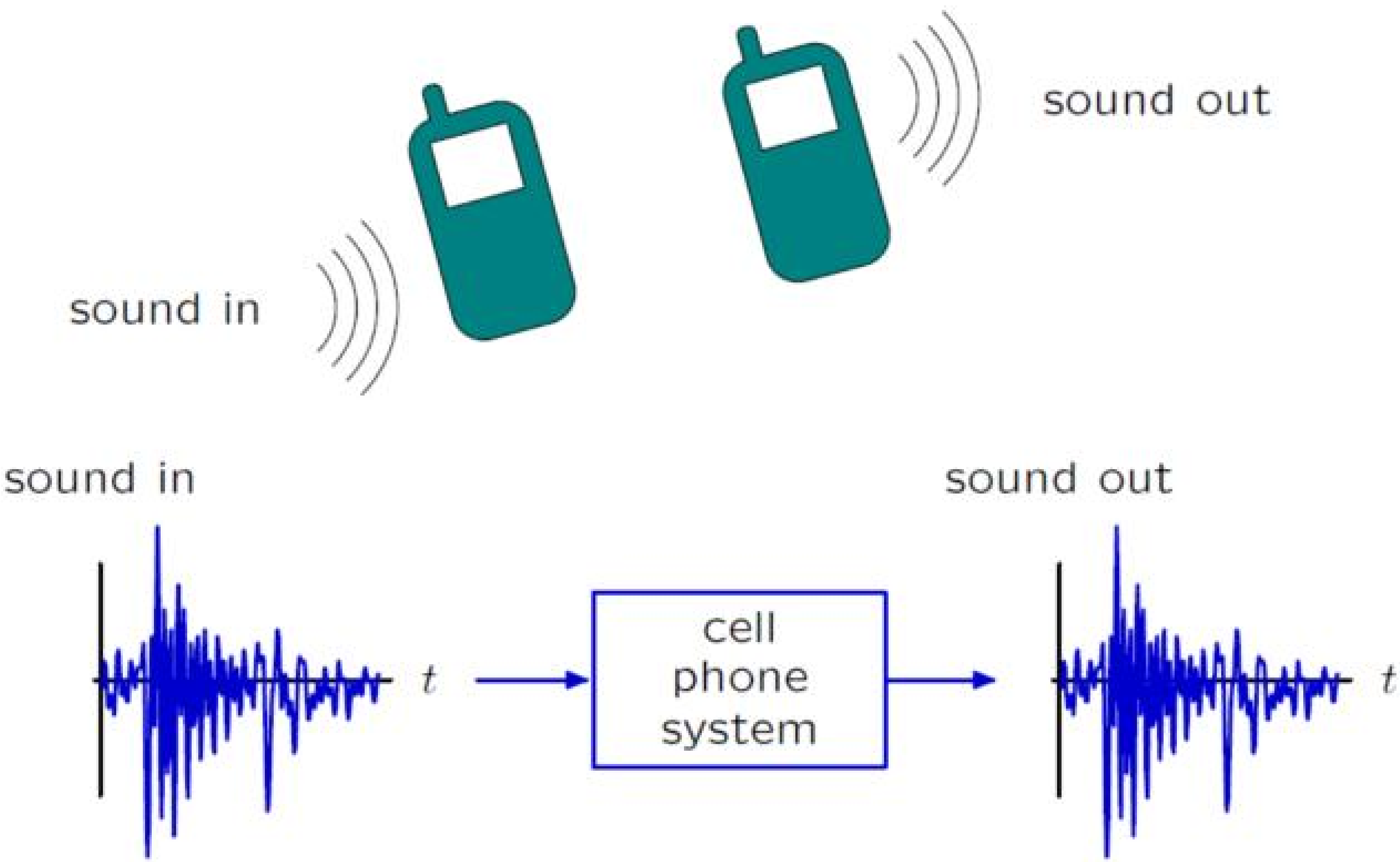




2. Example: Tanks

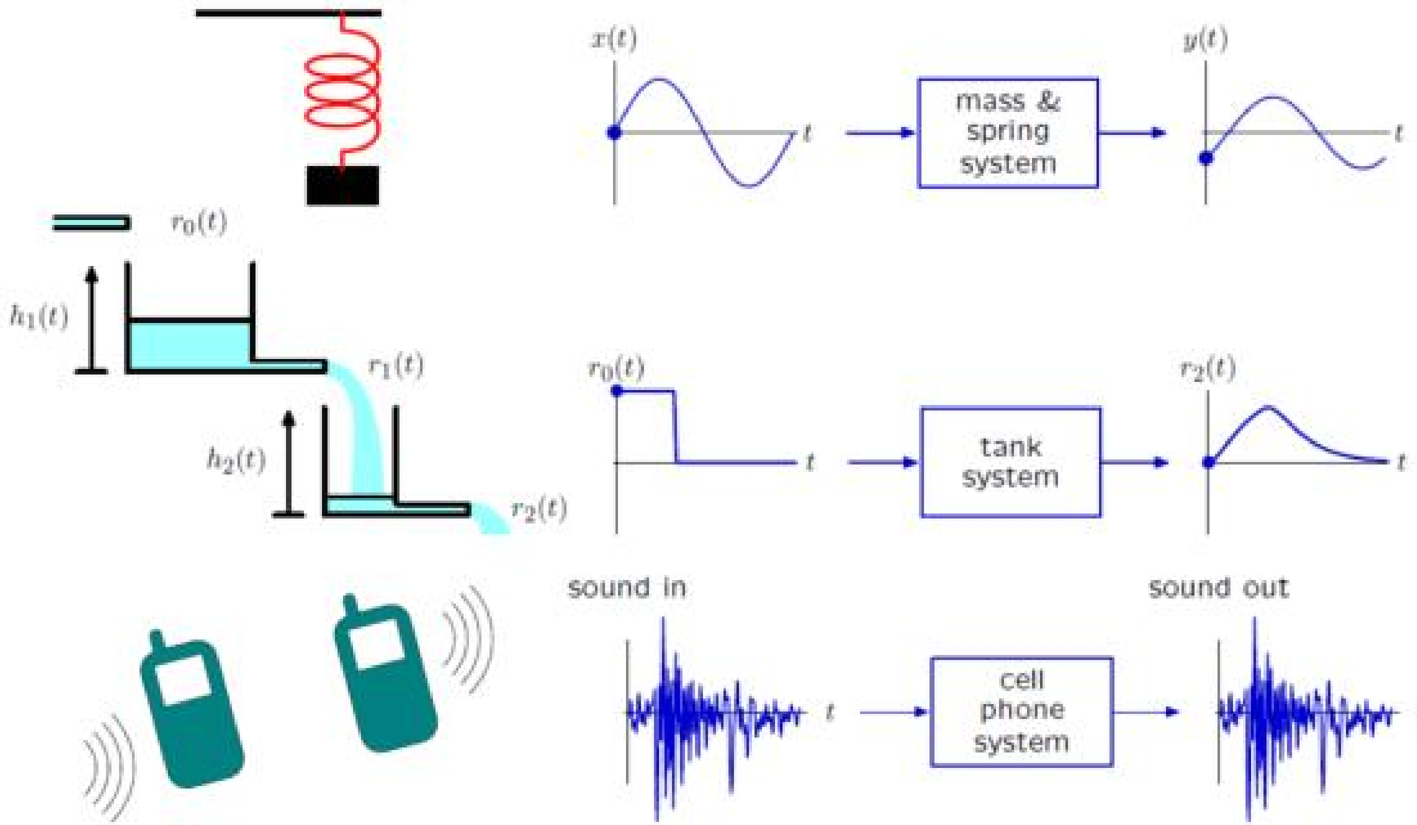


2. Example: Cell Phone System



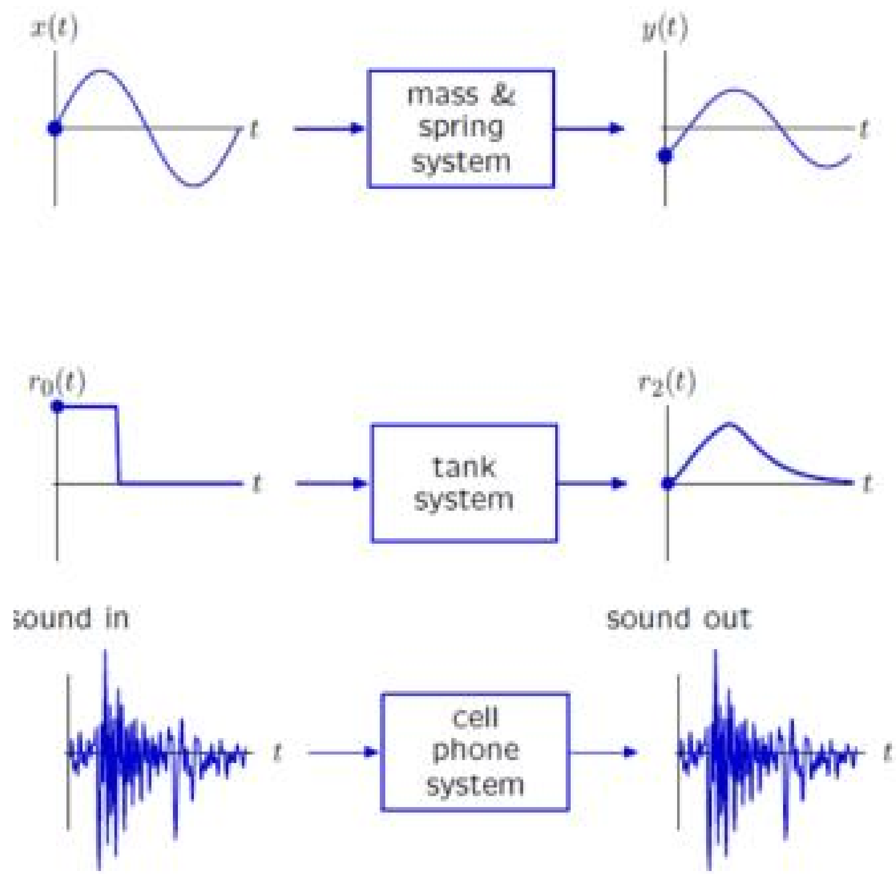
2. Signals and Systems: Widely Applicable

- The Signals and Systems approach has broad application:
  - electrical, mechanical, optical, acoustic, biological, financial, …

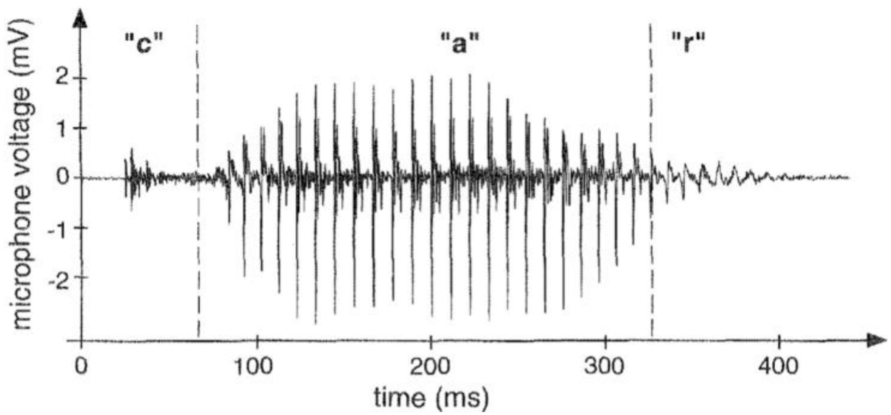


3. What is a **signal**?

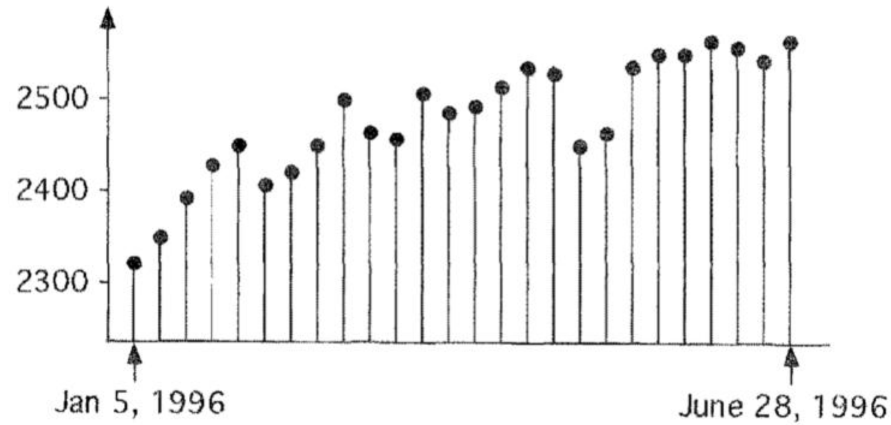
- A signal is a mathematical function or sequence of values that represents information.
  - independent variable = time
  - dependent variable = voltage, flow rate, sound pressure,…
- But not limited to these!



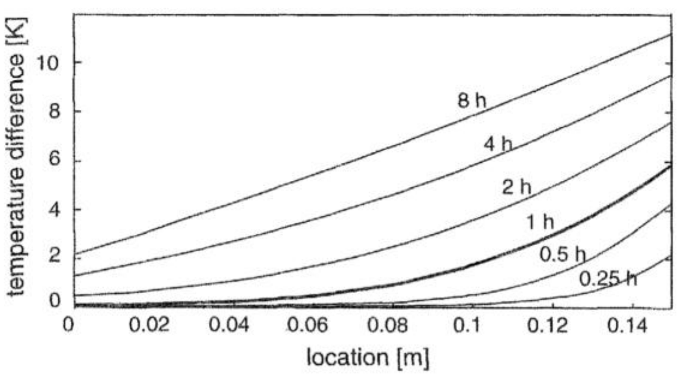
3. Consider the following …



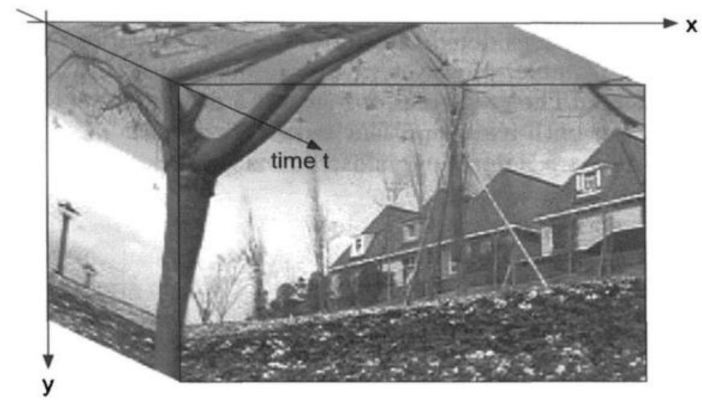
The electrical voltage produced by a microphone



The weekly stock market index



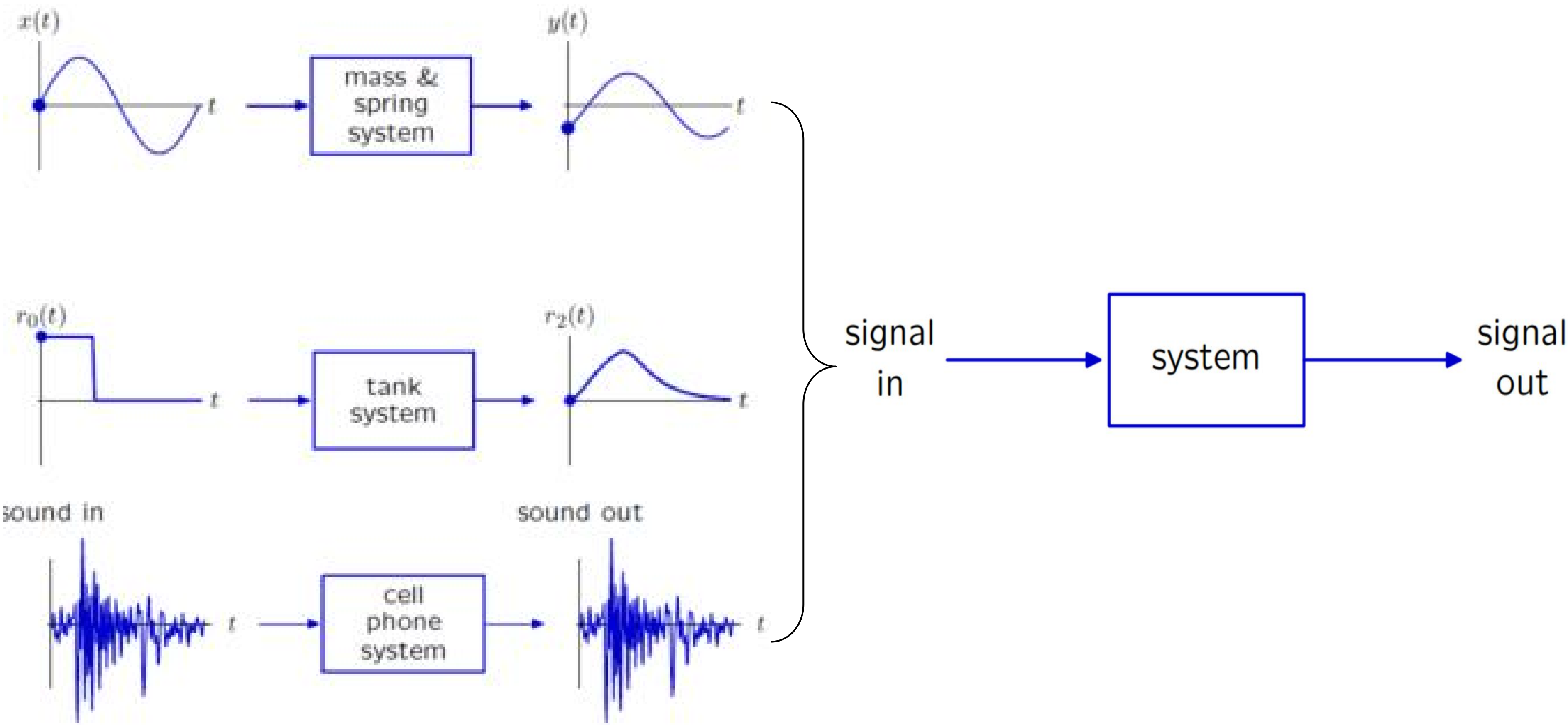
Temperature curves of a house wall, when air temperature outside suddenly rises by 10K



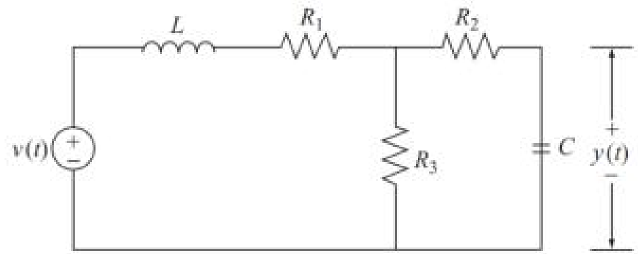
A short video as we move the camera from the left to the right.

4. What is a ***system***?

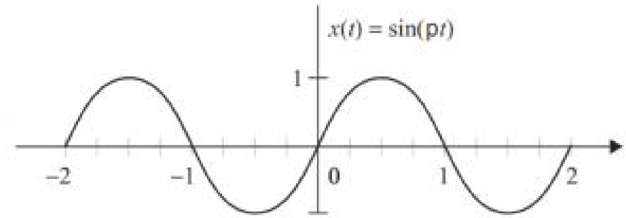
- A system is a process in which input signals are transformed or cause the system to respond in some way.



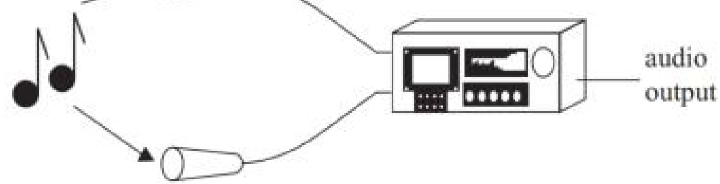
4. And more …



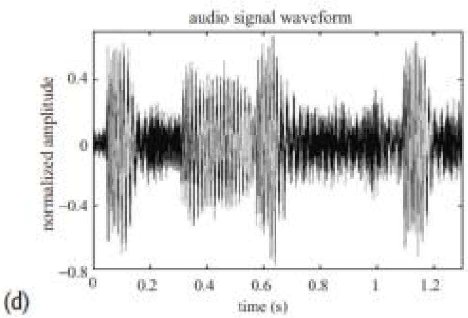
(a)



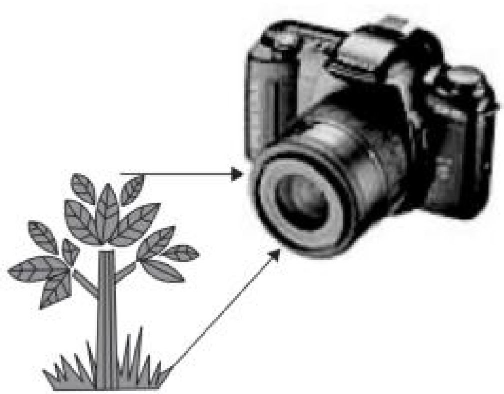
(b)



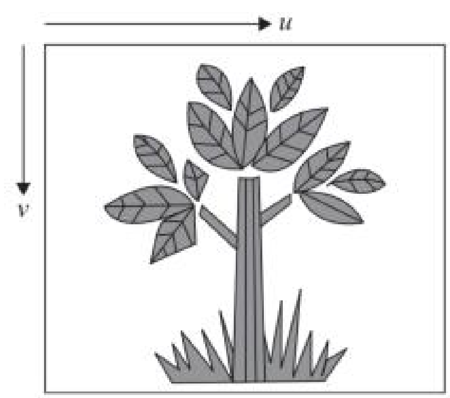
(c)



(d)



(e)



(f)

4. More examples

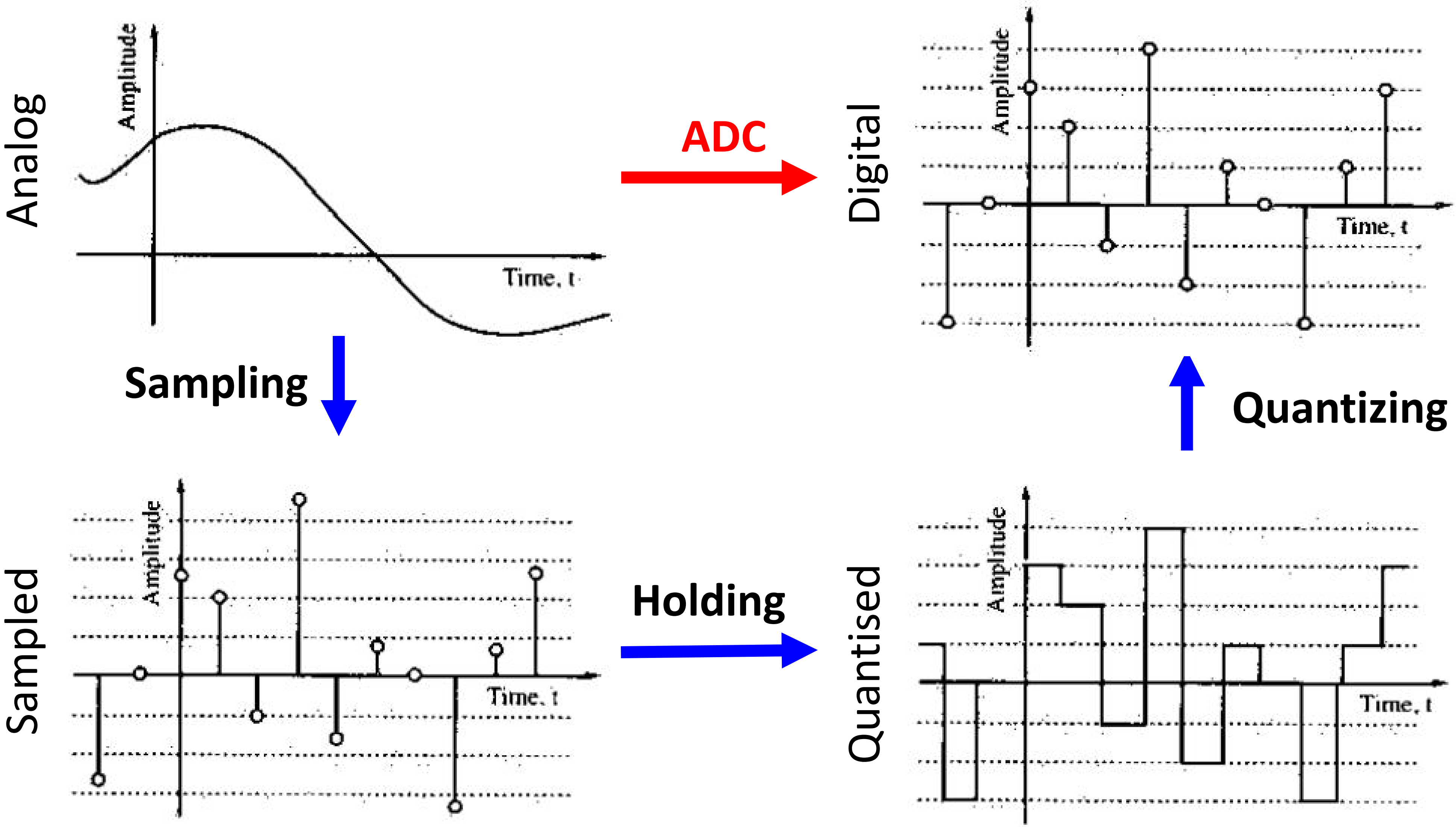
- 1. Compact-Disc (CD) Player
- 2. Software-defined Radio and Cognitive Radio
- 3. Computer-control Systems

  “Signals and Systems using MATLAB”by Luis F. Chaparro, Aydin Akan, ch0.2, pp. 4-8.

- A sound recording system

  “Signals and Systems using transform methods and MATLAB”by M.J.Roberts, , ch1.4, pp. 14-18.

5. Continuous or Discrete?



5. Continuous or Discrete?

- 1. CT and DT Representations
  - a discrete-time signal x[n] and the corresponding continuous-time signal x(t) are related by a sampling process:

    $$x[n] = x(nT_s) = x(t)|_{t=nT_s}$$

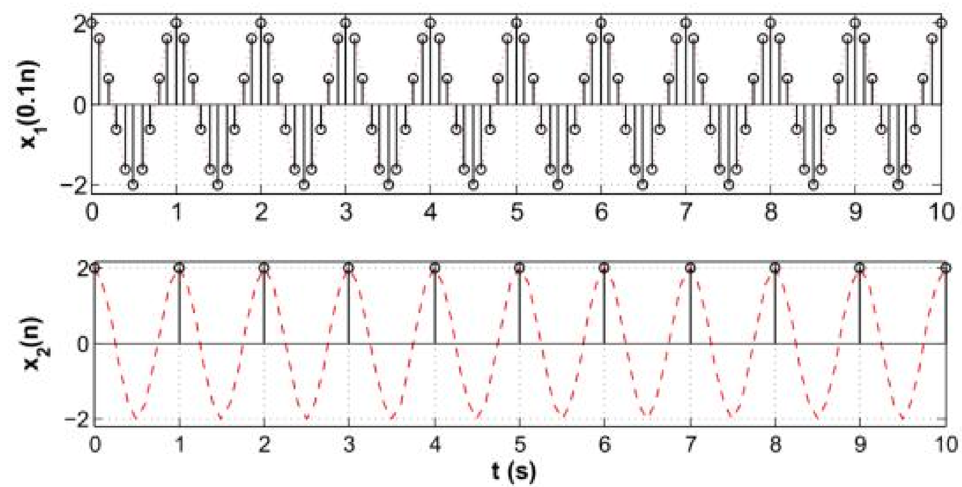
  - the signal $x[n]$ is obtained by sampling $x(t)$ at times $t = nT_s$, where n is an integer and T_s is the sampling period or the time between samples. This results in a sequence:

    $$\{\ldots, x(-T_s), x(0), x(T_s), x(2T_s), \ldots\}$$

  - According to the sampling times, they are:

    $$\{\ldots, x(-1), x(0), x(1), x(2), \ldots\}$$

  - Example: $x(t) = 2\,cos(2\pi t)$:
    - By choosing different sampling period T_s, the DT form of the original CT signal is different.



Ts1 = 0.1 s (top) and Ts2 = 1 s (bottom)

Notice the **similarity** between the DT signal and the CT signal when Ts1 = 0.1 s, while they are very **different** when Ts2 = 1 s, indicating loss of information.

5. Continuous or Discrete?

- 2. Derivatives and Finite Differences
  - the **derivative operator** measures the rate of change of a signal x(t):

$$D[x(t)] = \frac{dx(t)}{dt} = \lim_{h\to 0} \frac{x(t+h) - x(t)}{h}$$

  - the **forward finite-difference operator** measures the change in the signal from one sample to the next:

$$\Delta[x(nT_s)] = x((n+1)T_s) - x(T_s)$$

  - The derivative and the finite-difference operators are clearly not the same:

$$\left.\frac{dx(t)}{dt}\right|_{t=nT_s} = \lim_{T_s\to 0} \frac{\Delta[x(nT_s)]}{T_s}$$

  - Depending on the signal and the chosen value of T_s, the finite-difference operation can be a crude or an accurate approximation to the derivative multiplied by T_s.
  - Actually, when the rate of change of the signal is faster, the difference gets closer to the derivative by making T_s smaller.

5. Continuous or Discrete?

- 3. Integrals and Summations
  - Integration is the opposite of differentiation.
  - Suppose $I(t)$ is the **integration** of a continuous signal $x(t)$ from some time t_0 to t:

$$I(t) = \int_{t_0}^{t} x(\tau)\, d\tau$$

  - The **summation** can be expressed as:

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

  - The derivative of $I(t)$ is

$$\frac{d\, I(t)}{dt} = \lim_{h \to 0} \frac{I(t) - I(t-h)}{h} = \frac{d}{dt} \int_{t_0}^{t} x(\tau)\, d\tau = x(t)$$

  - If we use the derivative operator D[.], then its inverse D^{-1}[.] should be the integration operator, so

$$D[D^{-1}[x(t)]] = x(t)$$

  - Computationally, integration is implemented by sums.
  - The integral can be computed quite accurately using a very small value of T_s.

5. Continuous or Discrete?

- 4. Differential and Difference Equations
  - Linear Constant Coefficient Differential Equation (for CT)

$$\sum_{k=0}^{N} a_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M} b_k \frac{d^k x(t)}{dt^k}$$

  - Linear Constant Coefficient Difference Equation (for DT)

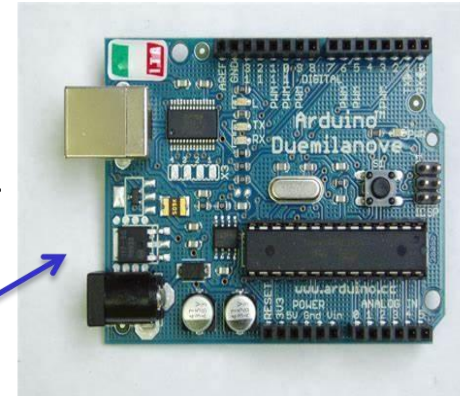
$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]$$

6. Implementations

- CT signals are typically processed using analog systems;
  - resistors, capacitors, inductors and semiconductor electronic components.
- DT signals are sequences of numbers. Processing them requires numerical manipulation of these sequences;
  - simple addition, multiplication, and delay operations.
- A digital signal processing system is represented by a mathematical equation using arithmetic operations.
  - Designing these systems requires the development of an **algorithm** that implements arithmetic operations.
  - A general-purpose computer may be used to develop and test these algorithms.
  - Possible implementations:
    - General-purpose microprocessors (μPs) and micro-controllers (μCs).
    - General-purpose digital signal processors (DSPs).
    - Field-programmable gate arrays (FPGAs).

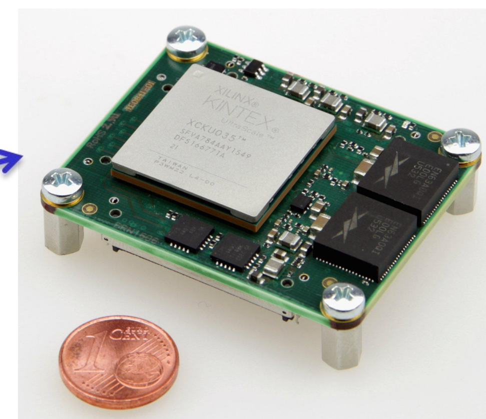
6. Implementations

- MPU: Microprocessor Unit. The core of a computer, contains the actual processor core, possibly some memory (in the form of cache(s)), and interface electronics to more memory and I/O.
- MCU: Microcontroller Unit. A complete computer on a chip, with periherals that are handy to solve the problem at hand.
  - MSP430, Arduino, ARMs, etc.



- DSP: Digital Signal Processor. Essentially an MPU with a core that is optimized to carry out digital signal processing (hardware specially designed for multipilication).
  - TI, Motorola, etc.

- FPGA: Field Programmable Gate Array. Highly configurable, general-purpose integrated circuits (ICs) filled with small programmable digital logic building blocks.
  - Xilinx, altera, etc.



Next …

- A quick (and brief) revision of related Mathematical knowledge, including …
  - 1. Fundamentals
    - Exponential identities
    - Logarithmic identities
  - 2. Derivatives and Integrals
  - 3. Sum formulas
  - 4. Complex numbers & Euler's formula
  - 5. Linear constant-coefficient differential equations
  - 6. Partial fraction expansion