ELEC 221 Lecture 01
Overview and intro to signals and systems

Thursday 8 September 2022

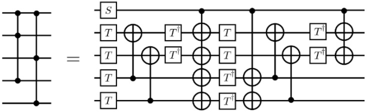
Olivia Di Matteo – KAIS 3043 – `olivia@ece.ubc.ca`

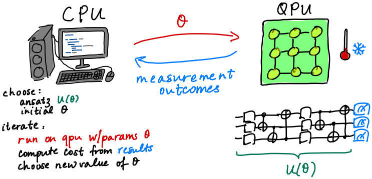
I have a physics background and I work on the **software and algorithms** side of **quantum computing**.

```python
import pennylane as qml

dev = qml.device("default.qubit", wires=3)

@qml.qnode(dev, diff_method="parameter-shift")
@qml.compile()
def circuit(data, params):
    qml.AngleEmbedding(data, wires=range(3))
    qml.RY(params[0], wires=0)
    qml.RX(params[1], wires=1)
    qml.RZ(params[2], wires=2)
    qml.CNOT(wires=[0, 1])
    qml.CNOT(wires=[1, 2])
    qml.CNOT(wires=[2, 0])
    return qml.expval(qml.PauliZ(0))
```





(Find me on GitHub as `glassnotes`.)

Intros

Who are you?

Go to `menti.com`...

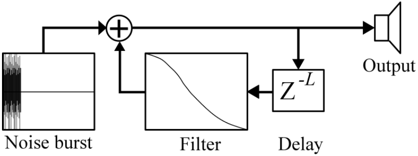
Course learning outcomes

A broad overview of processing and analysis of signals and systems, grounded in real-world examples, to equip you with a core set of tools you can use to explore them further.

- Express continuous-time and discrete-time signals as functions, and characterize how linear, time-invariant systems respond to them
- Apply Fourier analysis (as well as more generalized transforms), sampling, and interpolation techniques to characterize and reconstruct signals
- Manipulate signals and systems using filters, modulation, and feedback

By the end of the course, you will be able to do the above both **mathematically**, and **computationally** using Python and NumPy.

Signals and systems are EVERYWHERE: audio processing



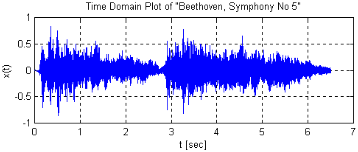
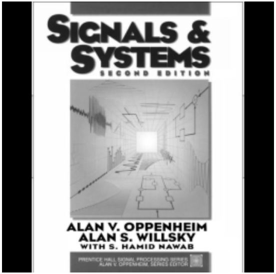
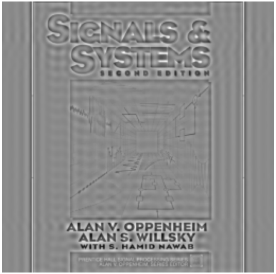


Image credits:
`https://www.uberchord.com/blog/guitar-effects-pedalboard-order/`
`https://en.wikipedia.org/wiki/File:Karplus-strong-schematic.svg`
`https://stats.stackexchange.com/questions/72386/find-music-tempo-with-a-audio-signal`

Signals and systems are EVERYWHERE: image processing

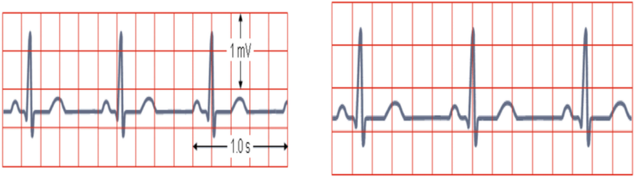




Signals and systems are EVERYWHERE: finance

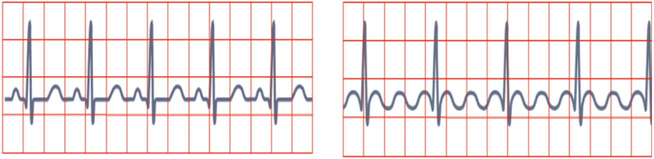
Image credit: google.com market summary

Signals and systems are EVERYWHERE: medicine



(a) sinus rhythm, normal complex, Heart rate-60-100/min

(b) bradycardia, normal complex, hear rate <60



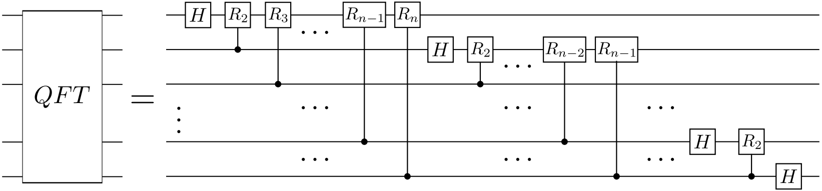
(c) tachycardia, normal complex, heart rate >100/minute

(d) Atrial flutter, the irregular response of ventricular, rapid flutter waves

Image credit: Thirugnanam, Mythili & Pasupuleti, Megana. (2021). Cardiomyopathy-induced arrhythmia classification and pre-fall alert generation using Convolutional Neural Network and Long Short-Term Memory model. Evolutionary Intelligence. 14. 10.1007/s12065-020-00454-0.

Signals and systems are EVERYWHERE: quantum computing

You need the **quantum Fourier transform** for Shor's algorithm (the one that breaks RSA!).



Signals and systems are EVERYWHERE: quantum computing

The outputs of circuits used in, e.g., **quantum machine learning** can be expressed as **Fourier series** of the input parameters.

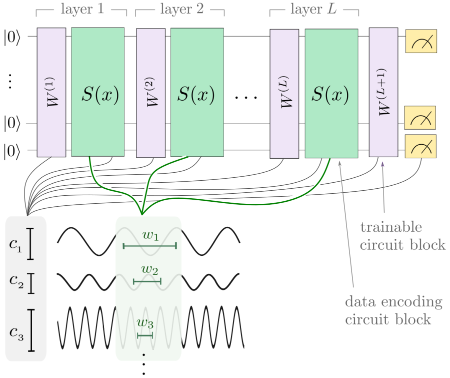
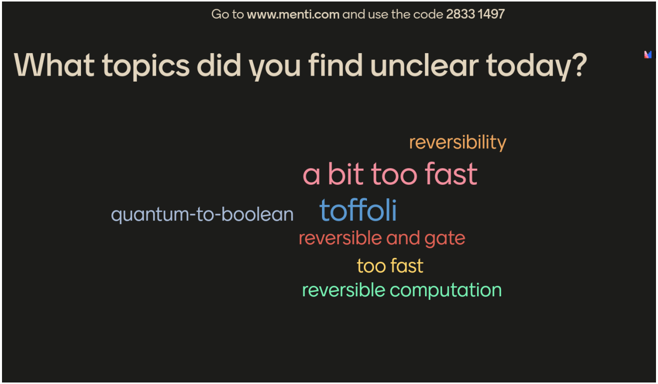


Image credit: Schuld, M., Sweke, R., and Meyer, J. J., "Effect of data encoding on the expressive power of variational quantum-machine-learning models", Physical Review A, vol. 103, no. 3, 2021.

Signals and systems are EVERYWHERE: teaching

Your feedback in the class is an important signal for me.



I will respond to your feedback by adjusting the pace of the course, going over things again, etc.

Course overview

We will roughly follow Oppenheim's Signals and Systems 2nd ed.

We will explore topics using a mix of **math** and **programming**, with lots of visualizations.

I will often **live code** (in Python) to demonstrate concepts. Bring your laptops to follow along!

All lecture slides/demos will be posted on the course GitHub:

`https://github.com/glassnotes/ELEC-221`

Course overview: assignments and grading

Five graded components:

- 20% Assignments (8)
- 10% In-class quizzes (10)
- 15% Midterm 1
- 15% Midterm 2
- 40% Final exam

Course overview: assignments and grading

20% assignments (8)

- Distributed and submitted on PrairieLearn

Course overview: assignments and grading

20% assignments (8)

- Distributed and submitted on PrairieLearn
- Short but frequent

Course overview: assignments and grading

20% assignments (8)

- Distributed and submitted on PrairieLearn
- Short but frequent
- Assignments 1, 3, 5, 7 will be mathematical

Course overview: assignments and grading

20% assignments (8)

- Distributed and submitted on PrairieLearn
- Short but frequent
- Assignments 1, 3, 5, 7 will be mathematical
- 2, 4, 6, 8 will be computational (in Python)

Course overview: assignments and grading

20% assignments (8)

- Distributed and submitted on PrairieLearn
- Short but frequent
- Assignments 1, 3, 5, 7 will be mathematical
- 2, 4, 6, 8 will be computational (in Python)
- **You may consult with your classmates, however:**

Course overview: assignments and grading

20% assignments (8)

- Distributed and submitted on PrairieLearn
- Short but frequent
- Assignments 1, 3, 5, 7 will be mathematical
- 2, 4, 6, 8 will be computational (in Python)
- **You may consult with your classmates, however:**
    - What you submit must be your own work

Course overview: assignments and grading

20% assignments (8)

- Distributed and submitted on PrairieLearn
- Short but frequent
- Assignments 1, 3, 5, 7 will be mathematical
- 2, 4, 6, 8 will be computational (in Python)
- **You may consult with your classmates, however:**
  - What you submit must be your own work
  - You **must** include a statement of contributions (e.g., *"I used Stack Overflow (link) for Q1, checked answers with classmate X for Q2-3, and classmate X explained Z which helped me solve Q3"*)

Course overview: assignments and grading

10% quizzes (10)

- Done on PrairieLearn in first 10 minutes of Tuesday classes

Course overview: assignments and grading

10% quizzes (10)

- Done on PrairieLearn in first 10 minutes of Tuesday classes
- 1-3 (quantitative) multiple choice questions based on material from previous week

Course overview: assignments and grading

10% quizzes (10)

- Done on PrairieLearn in first 10 minutes of Tuesday classes
- 1-3 (quantitative) multiple choice questions based on material from previous week
- Open-book, but individual

Course overview: assignments and grading

10% quizzes (10)

- Done on PrairieLearn in first 10 minutes of Tuesday classes
- 1-3 (quantitative) multiple choice questions based on material from previous week
- Open-book, but individual

10% quizzes (10)

- Done on PrairieLearn in first 10 minutes of Tuesday classes
- 1-3 (quantitative) multiple choice questions based on material from previous week
- Open-book, but individual

First one on Tuesday.

Course overview: assignments and grading

15% + 15% midterms

Midterm 1: Thursday 13 Oct. (during class)

- Individual exam
- Covers material up to that point in the course

Midterm 2: Monday 14 Nov. (during tutorial)

- Two-stage exam: do exam alone first, then do same exam again in small groups
- Grade will be a weighted average of individual (85%) and group (15%) part
- Not cumulative

Course overview: assignments and grading

40% final

- Individual exam, scheduled during the regular exam period

40% final

- Individual exam, scheduled during the regular exam period
- **Cumulative**, but more emphasis on later parts

Course overview: logistics

	Office hours
Olivia Di Matteo	F 09:00-10:00 (open-door afternoons; by appointment)
(TA) Ezequiel Hernandez	M 10:00-11:00
(TA) Jiaming Cheng	T 11:00-12:00
(TA) Sarthak Panda	W 13:00-14:00
(TA) Mohammad Taha Askari	Th 11:00-12:00

TAs will also run a tutorial on M from 17:30-19:30. First one is on **Monday 19 Sept.** (not next week!).

(See syllabus in Canvas for TA personal details)

Today

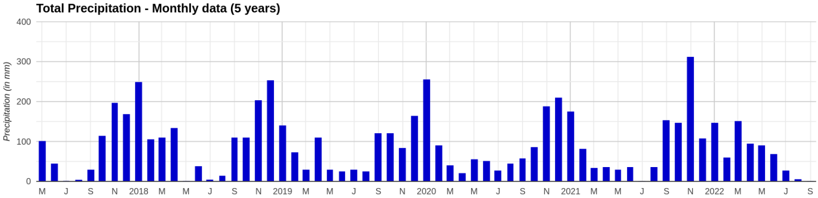
Learning outcomes:

- Define signals and systems, and provide real-world examples of each
- Express continuous-time and discrete-time signals as functions mathematically and in Python
- Apply time shifts, time scaling, and time reversal transforms to a signal
- Identify whether a system has the following properties: linear, causal, memoryless, time invariant, stable, invertible

Signals

Signals are "patterns of variations" that contain information about the behaviour of a phenomenon.

Example: total precipitation in Vancouver vs. time

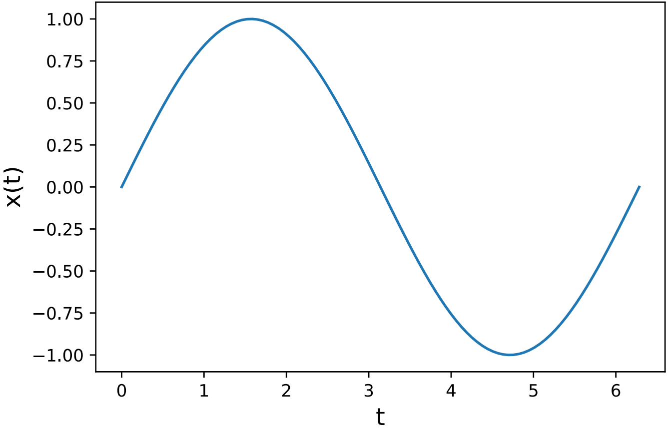


Mathematically, they are functions of one or more independent variables (often, but not always, time).

Image generated from: `https://vancouver.weatherstats.ca/charts/precipitation-monthly.html`

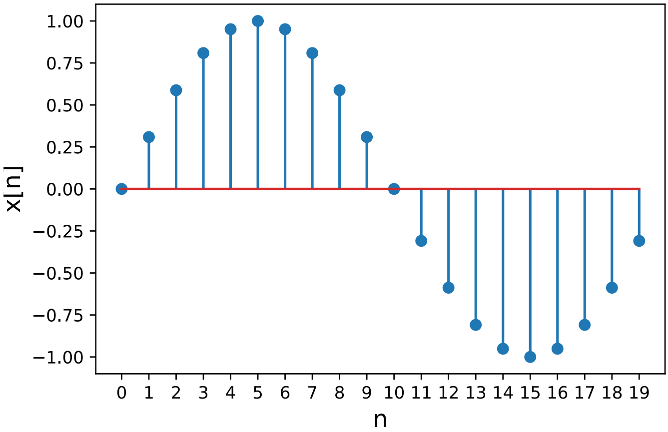
Continuous-time (CT) signals

Notation: $x(t)$, where t can be any real number.



Discrete-time (DT) signals

Notation: $x[n]$, where n is strictly an integer (positive or negative).



CT vs. DT signals

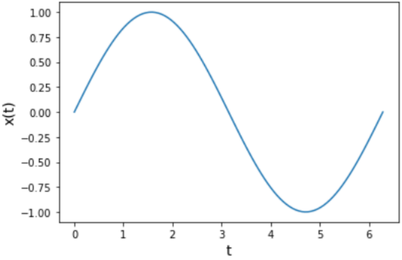
Proliferation of digital signal processing tools means dealing with CT (in theory) signals involves discretization of the signal by sampling it at many points.

```
def our_signal(t):
    return np.sin(t)

t = np.linspace(0, 2*np.pi, 100)

x_t = [our_signal(time) for time in t]

plt.plot(t, x_t)
```



We will revisit this later in the course.

CT vs. DT signals

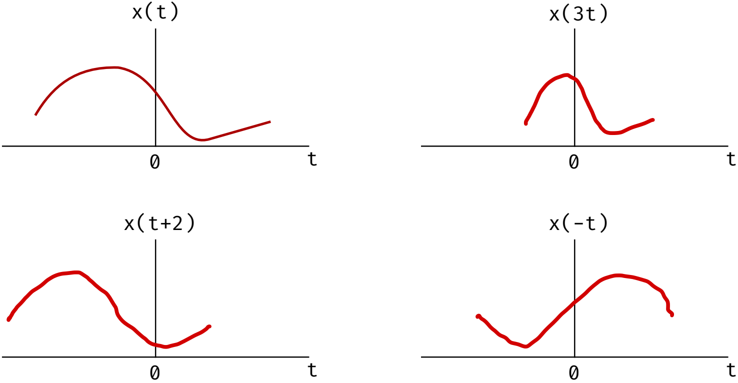
In this class we will be

- Transforming signals
- Synthesizing and analyzing signals
- Determining how systems respond to them
- Designing systems to manipulate signals in specific ways

$x(t)$

Let's start with a few simple examples:

- Time shift: $x(t) \to x(t - t_0)$
- Time scaling: $x(t) \to x(at)$
- Time reversal: $x(t) \to x(-t)$



We will implement some DT signals as Python functions, then apply transformations to them.

- Time shift: $x[n] \to x[n - n_0]$
- Time scaling: $x[n] \to x[an]$
- Time reversal: $x[n] \to x[-n]$

A CT signal is periodic with period T if for all t,

$$x(t+T) = x(t)$$

A DT signal is periodic with period N if for all n,

$$x[n+N] = x[n]$$

The smallest value of T (or N) for which this holds is the **fundamental period**.

Consequences: time reversal and symmetry

A CT signal is **odd** if for all t

$$x(-t) = -x(t)$$

and **even** if

$$x(-t) = x(t)$$

A DT signal is **odd** if for all n

$$x[-n] = -x[n]$$

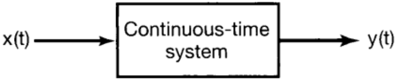
and **even** if

$$x[-n] = x[n]$$

Systems respond to signals and produce some desired behaviour, or other signals as outputs.

A CT system sends

$$x(t) \longrightarrow y(t)$$



A DT system sends

$$x[n] \longrightarrow y[n]$$

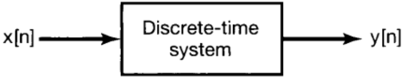


Image credits: Signals and Systems 2nd ed., Oppenheim

Often systems are represented as block diagrams. We can combine multiple systems in series or parallel.

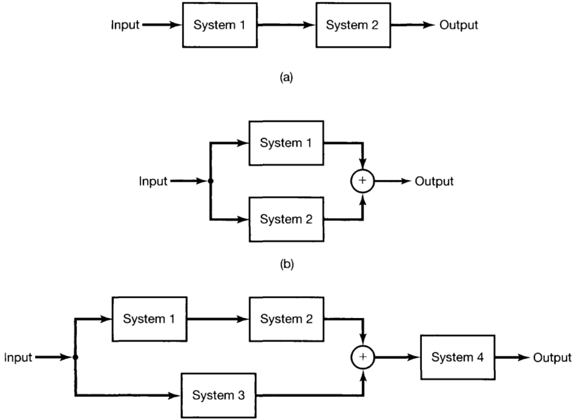


Image credits: Signals and Systems 2nd ed., Oppenheim

Properties of systems

1. Memory
2. Invertibility
3. Causality
4. Stability
5. **Time invariance**
6. **Linearity**

The last two are the most important for this course.

A system is **memoryless** if it the output at each time depends only on the input *at the same time*.

(✓) Output voltage after current goes through a resistor (R)

$$V(t) = R\,I(t)$$

(X) Output voltage after current $I(t)$ goes through capacitor (C)

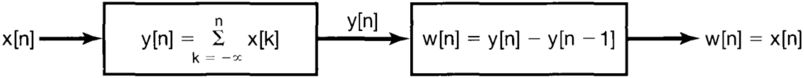
$$V(t) = \frac{1}{C}\int_{-\infty}^{t} I(\tau)\,d\tau$$

(X) A delay system

$$y[n] = x[n-1]$$

A system is **invertible** if distinct inputs lead to distinct outputs.

(✓) Accumulator



(X) The following system is not invertible:

$$y(t) = x^2(t)$$

Image credits: Signals and Systems 2nd ed., Oppenheim

A system is **causal** if the output at *any time* depends only on the input at the present time or in the past.

($\checkmark$) Output voltage after current $I(t)$ goes through capacitor (C)

$$V(t) = \frac{1}{C}\int_{-\infty}^{t} I(\tau)\,d\tau$$

(X) A moving average

$$y[n] = \frac{1}{2M+1}\sum_{k=-M}^{M} x[n-k]$$

(X) Time reversal

$$y[n] = x[-n]$$

A system is **stable** if small changes in input do not cause the output to diverge (bounded inputs lead to bounded outputs).

(✓) Moving average of a bounded function, $x[n] \leq B$

$$y[n] = \frac{1}{2M+1} \sum_{k=-M}^{M} x[n-k]$$

(X) Compound interest in a bank account

$$y[n] = 1{,}01\, y[n-1] + x[n]$$

Properties of systems: time invariance

A system is **time invariant** if time shifts in the input lead to identical time shifts in the output, i.e.,

$$x(t) \rightarrow y(t) \qquad x(t-t_0) \rightarrow y(t-t_0)$$

(✓) This system is time invariant:

$$y(t) = x(t+1) - x(t-1)$$

Why?

$$x_1(t) \rightarrow y_1(t) = x_1(t+1) - x_1(t-1) \qquad y_1(t-t_0) = x(t-t_0+1) - x_1(t-t_0-1)$$

$$x_2(t) = x_1(t-t_0) \rightarrow y_2 = x_2(t+1) - x_2(t-1)$$
$$= x_1(t-t_0+1) - x_1(t-t_0-1)$$

(X) This system is not time invariant:

$$y(t) = \cos(3t)x(t)$$

Why?

$$x_1(t) \to y_1(t) = \cos(3t)x_1(t)$$

$$y_1(t-t_0) = \cos(3(t-t_0))x_1(t-t_0)$$

$$x_2(t) = x_1(t-t_0) \to y_2(t) = \cos(3t)x_2(t) = \cos(3t)x_1(t-t_0)$$

A **linear** system $x(t) \to y(t)$ sends

(Additivity) $x_1(t) + x_2(t) \to y_1(t) + y_2(t)$

(Homogeneity) $a x_1(t) \to a y_1(t)$

$x_1(t), x_2(t)$ $\quad x_1(t) + x_2(t) \longrightarrow y_1(t) + y_2(t)$

(Additivity)

Consequently, a linear system sends

$$a x_1(t) + b x_2(t) \longrightarrow a y_1(t) + b y_2(t)$$

for arbitrary coefficients a, b (which may be complex).

(X) The following system is *not linear*:

$$x[n] \to y[n] = x[n] + 1$$

Why? $x_1[n]$, $x_2[n]$

$$x_1[n] \to y_1[n] = (x_1[n] + 1)\,a$$

$$x_2[n] = a\,x_1[n] \Rightarrow y_2[n] = x_2[n] + 1$$

$$= a\,x_1[n] + 1$$

non linear

$$x_3[n] = x_1[n] + x_2[n]$$

Live code 2: let's implement a linear system

Recap

Today's learning outcomes were:

- Define signals and systems, and provide real-world examples of each
- Express continuous-time and discrete-time signals as functions mathematically and in Python
- Apply time shifts, time scaling, and time reversal transforms to a signal
- Identify whether a system has the following properties: linear, causal, memoryless, time invariant, stable, invertible

What topics did you find unclear today?

For next time

Content:

- Quiz #1 (topic: system properties)
- Linear time-invariant (LTI) systems, unit impulse, and impulse response

Action items:

1. Follow instructions on Canvas to get set up with PrairieLearn and Piazza
2. NO TUTORIAL ON MONDAY

Recommended reading: Oppenheim Chapter 1.1-1.2, 1.5-1.6