Digital Signal Processing

Guide to AI and Data Courses · Part III: Signals, Images and Audio

Sampling | Linear systems | Fourier and FFT | Filter design | Spectral analysis | Related courses

Sercan KÜLCÜ · Giresun University · Computer Engineering

© Sercan Külcü. All rights reserved. · Interactive HTML version of these slides:
https://sercankulcu.github.io/files/ai_courses/slides/Chapter_16_Digital_Signal_Processing.html

INTRODUCTION

What Question Does It Ask?

> **How are sampled signals analyzed, filtered and transformed?**

Digital signal processing analyzes, filters and transforms audio, vibration, ECG or radio signals that have been **sampled** at regular intervals, using mathematical transforms. The course rests on exact mathematics, not learning; it is the common foundation of the audio, image and communications courses.

At a glance

APPROACH mathematical transforms and filters	**INPUT** sampled signal	**OUTPUT** filtered signal, spectrum
SUCCESS MEASURE signal-to-noise ratio, frequency response	**MATH LOAD** high: complex numbers, Fourier	**DATA NEEDS** no learning; the signal suffices

A Short History

1822
In his book on heat, Fourier proposes writing functions as sums of sinusoids.

1949
Shannon publishes sampling theory: a band-limited signal sampled fast enough is recovered exactly.

1965
Cooley and Tukey publish the fast Fourier transform (FFT); computing spectra becomes practical.

1974
Ahmed, Natarajan and Rao introduce the discrete cosine transform (DCT); JPEG and MP3 build on it.

1975
Oppenheim and Schafer's Digital Signal Processing appears; the field becomes a standard course.

1978
Texas Instruments' Speak & Spell brings single-chip digital speech synthesis into homes.

THE COURSE

Core Concepts

Sampling
Measuring a continuous signal at equal intervals to get a sequence of numbers.

Nyquist criterion
Sample faster than twice the highest frequency.

Aliasing
With undersampling, a high frequency shows up as a low one.

Quantization
Rounding amplitude to a finite number of levels; bit depth.

Convolution
The response of a linear time-invariant system to its input.

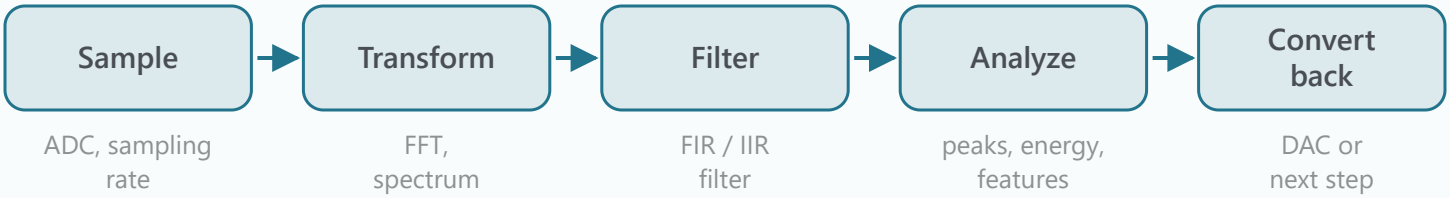
Fourier and FFT
A fast way to split a signal into frequency components.

FIR and IIR filters
Low-pass, high-pass and band-pass digital filters.

z-transform
Poles and zeros of a filter; stability analysis.

THE COURSE

A Typical Workflow



Example: An ECG device samples the heart signal 360 times per second. The spectrum shows 50 Hz mains noise; a notch filter suppresses only that frequency. R peaks are found in the cleaned signal and heart rate is computed from the time between them. No step involves learning.

THE COURSE

What Does the Course Cover?

1. Discrete-time signals and systems
2. Linear time-invariant systems and convolution
3. Sampling, aliasing and reconstruction
4. Discrete-time Fourier transform, frequency response
5. The z-transform: poles, zeros and stability
6. The discrete Fourier transform (DFT) and the FFT
7. Windowing and spectral leakage
8. FIR filter design: the window method
9. IIR filter design: Butterworth, Chebyshev
10. Sample-rate conversion: decimation, interpolation
11. Short-time Fourier transform and spectrograms
12. Quantization noise and fixed-point implementation

Skills you will gain

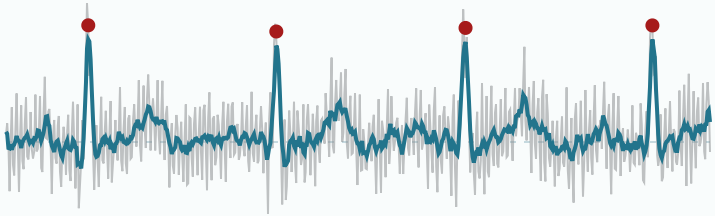
designing filters with scipy.signal | plotting spectra in MATLAB or Octave | reading pole-zero diagrams | choosing a suitable sampling rate | filtering and interpreting a real signal

A typical semester; the order and coverage vary between universities.

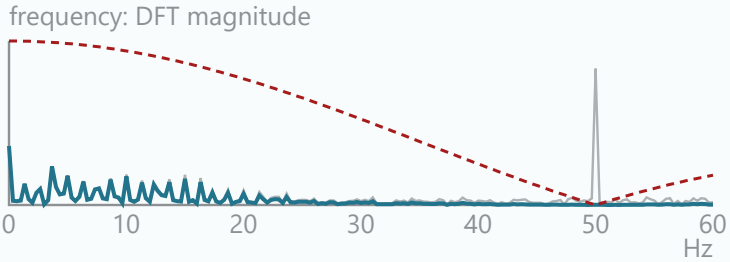
TRY IT

Cleaning a Noisy Pulse

A chest-strap ECG samples at 200 Hz; 50 Hz mains hum and muscle noise have crept into the recording. Change the moving-average window length and watch the signal in both time and frequency.



time (3 s) →



Window M = **4** samples (20 ms)

50 Hz hum **Muscle noise**

Signal/noise: -2.8 dB → **9.1 dB**
R peak height: **91%**
Beats found: 4, heart rate ≈ **75 bpm**

The dashed line is the filter's frequency response: at M = 4 its zero falls exactly on 50 Hz. Larger M means less noise, but the sharp **R peak** gets flattened too.

To separate noise you first look at its **frequency**. The course teaches you to design a filter from its frequency response, not by trial, and to leave the component you want intact.

CONTEXT

Prerequisites

Courses worth knowing before this one, and why they are needed.

Course	Why it is needed
Calculus	integrals, series and complex numbers: the language of the Fourier transform
Differential Equations	behavior and stability of linear systems
Programming with C	to write real-time filters on embedded devices

CONTEXT

Are You Ready for This Course?

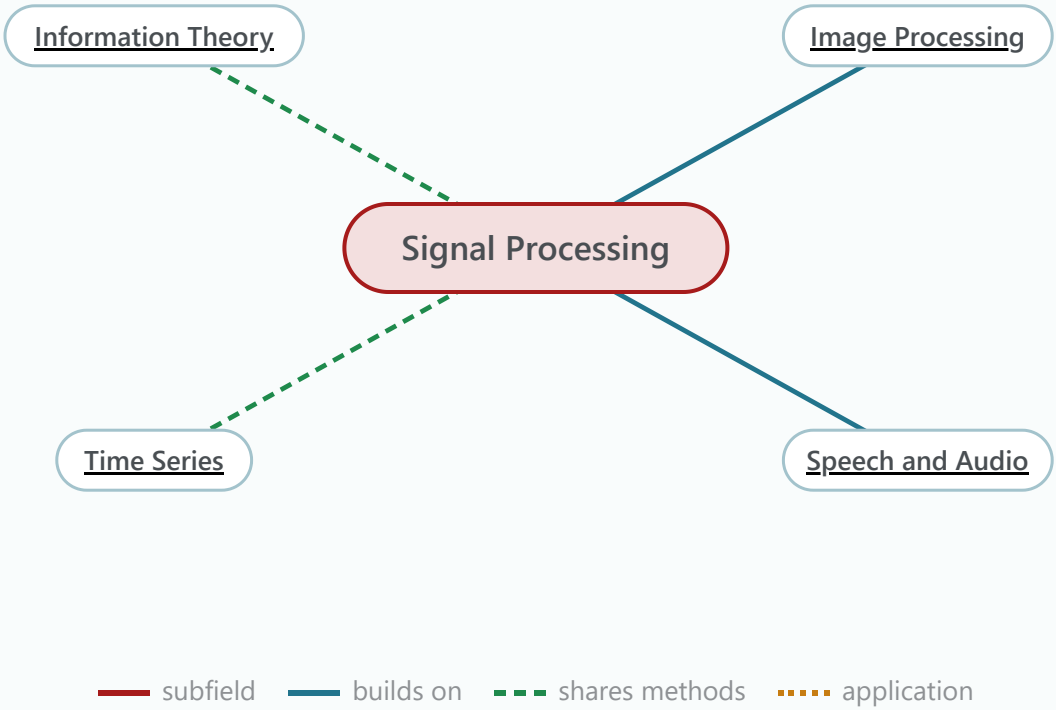
Ask yourself; for every "no", the right-hand column shows where to start.

Ask yourself	If not, start with
Can you write a complex number in polar form and explain Euler's formula?	Complex numbers and Euler's formula from calculus
Can you read the amplitude, frequency and phase of a sinusoid from its plot?	A quick review of trigonometry and sinusoid graphs
Can you evaluate a sum (Σ) and a simple integral?	Series and integrals from calculus
Can you generate an array in Python or MATLAB and plot it?	Intro programming, then a short NumPy and Matplotlib tutorial

If you can say "yes" to most of these, you are ready; one or two gaps can be closed during the semester.

CONTEXT

Neighbouring Courses



OVERLAP

Overlapping Courses

Relation	Course	Shared ground
builds on	Image Processing builds on this course	2D signals, convolution, Fourier transform
builds on	Speech Recognition and Audio Processing builds on this course	spectrogram, MFCC, filters
shares methods	Shares methods with Time Series Analysis and Forecasting	frequency analysis, filtering
shares methods	Shares methods with Information Theory	compression, sampling, channels

Where It Differs (1/2)

Criterion	Signal Processing	Image Processing	Speech and Audio
Core question	How are sampled signals analyzed, filtered and transformed?	How is an image enhanced, transformed and divided into regions?	How are words, speakers and events extracted from a sound wave?
Approach	mathematical transforms and filters	pixel-level transformations	acoustic signal and language model
Input	sampled signal	image	audio waveform
Output	filtered signal, spectrum	processed image or regions	text, speaker, sound event
Success measure	signal-to-noise ratio, frequency response	visual quality, PSNR	word error rate (WER)
Math load	high: complex numbers, Fourier	medium: convolution, Fourier	medium-high
Data needs	no learning; the signal suffices	little; learning not required	many hours of recordings

- **Image Processing:** Signal processing works with one-dimensional, time-dependent signals; image processing carries the same tools to a 2D pixel grid.
- **Speech and Audio:** Signal processing provides general tools; speech recognition combines them with language models and learning to output words.

DIFFERENCES

Where It Differs (2/2)

Criterion	Signal Processing	Time Series	Information Theory
Core question	How are sampled signals analyzed, filtered and transformed?	How do we find structure in measurements over time and forecast the future?	How is information measured, and how is it compressed and transmitted with the least loss?
Approach	mathematical transforms and filters	time-dependent statistical models	measuring information with probability
Input	sampled signal	timestamped measurements	source and channel model
Output	filtered signal, spectrum	forecast, trend, seasonality	bounds, codes, compression rate
Success measure	signal-to-noise ratio, frequency response	forecast error (MAE, RMSE)	bits, channel capacity
Math load	high: complex numbers, Fourier	medium-high	high: probability
Data needs	no learning; the signal suffices	long history	none

- **Time Series:** Both look at time-dependent data; time series analysis needs forecasting and statistics, signal processing needs frequency and filters.
- **Information Theory:** Information theory gives the measure of information and the limits of compression; signal processing gives the algorithms that act on the signal.

PRACTICE

Applications

Audio and music
Equalizers, noise suppression, MP3 compression.

Communications
Modems, Wi-Fi, 4G/5G modulation.

Biomedical
Filtering ECG and EEG, finding peaks.

Radar and sonar
Measuring range and speed from echoes.

Vibration monitoring
Detecting machine faults from the spectrum.

Image and video
Transforms in JPEG and video coding.

PRACTICE

Risks and Ethics

Aliasing
Without a low-pass filter before sampling, high frequencies show up as fake low ones and cannot be undone.

Unstable IIR filters
A pole outside the unit circle makes the output blow up; coefficient rounding in fixed point can push it there.

Spectral leakage
An FFT taken without a suitable window shows frequency components that are not there and leads to wrong diagnoses.

Phase distortion
The nonlinear phase of an IIR filter distorts waveforms such as ECG, which can affect a diagnosis.

In depth: Explainable AI and AI Ethics.

PRACTICE

Tools and Datasets

Tools and libraries

MATLAB | GNU Octave | NumPy | SciPy (scipy.signal) | librosa | GNU Radio

Sample signals and datasets

MIT-BIH Arrhythmia (ECG) | PhysioNet | TIMIT (speech) | CWRU bearing vibration data | Synthetic sines and noise

In class a **synthetic signal** is often enough: summing known frequencies and adding noise shows clearly what a filter does.

PRACTICE

Common Misconceptions

Common belief	In fact
A higher sampling rate is always better.	Beyond the Nyquist limit you gain no information; only data and computation grow.
The FFT is a different transform from the Fourier transform.	The FFT is an algorithm that computes the discrete Fourier transform (DFT) fast; the result is the same.
A filter tells noise from signal and removes it.	A filter separates by frequency; if noise shares the signal's band, the filter cannot separate it.
Deep learning has made signal processing obsolete.	Audio and sensor models usually take a spectrogram or a filtered signal as input.

NEXT STEPS

After This Course

Courses that build on or extend this one

Image Processing | Speech Recognition and Audio Processing

Related careers

DSP engineer | Audio software engineer | Communications engineer | Embedded systems engineer | Biomedical engineer

Digital signal processing is the gateway to the third part: extended to two dimensions it becomes image processing, combined with language models it becomes speech recognition. Time series analysis and information theory also use its tools. On the hardware side, you can move on to embedded AI and communications.

NEXT STEPS

Further Reading

- Oppenheim and Schafer, *Discrete-Time Signal Processing*, 3rd ed., Pearson, 2010.
- Proakis and Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th ed., Pearson, 2007.
- Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, 1997.
- Cooley and Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", *Mathematics of Computation* 19(90), 1965.

The End

Chapter 16: Digital Signal Processing

← Previous chapter · Map of the field · Next chapter →