ECE 420

Aiden Sirotkine

Fall 2025

Contents

Chapter 1

ECE 310 Overview

1.1 Sampling

1.1.1 Shannon-Nyquist Theorem:

Proves that discrete samples can perfectly reconstruct a continuous signal.

Reconstruction
Up-Down Sampling
z-transform
CTFT
DTFT
DFT (FFT)
Yea its literally all fourier transforms lmao

1.1.2 Discrete LTI System

Convolution

Impulse Response
Frequency Response

1.1.3 Filters

Digital Filters

FIR vs IIR
Linear Phase

1.2 310 vs 420

- How do we get the data?
  ECE 310 = offline (batched)
  ECE 420 = online (stream)
  We use buffering techniques
  overlapped added
  windowing
- Who Computes?
  ECE 310 = Computer code
  ECE 420 = Mobile app/phone
  RUN-TIME IS IMPORTANT
  time-domain processes and FFT'S

1.3 Skillsets

Android app dev
C++ and Java.
Do not need to know crazy circuit shenanigans
You do not need a fancy UI for the DSP final project.
The DSP algorithm is more important.

1.4 Signals

IMU signals = 1d signals that are acceleration + Gyro
Audio signal is also 1d in the 20 - 20,000 Hz range
Image signals are 2d signals with visible light and video.

1.5 Android Device

You can loan a tablet
Need a specific OS
Needs gradle compiler.

1.6 Code

Need Python and need Android Studio

1.7 Basic Practice

1. Develop and Test DSP algorithms in hihg-level languages (Python)
2. Port tested algorithms into Android platform (C++, Java)

1.7.1 Attendance Quiz

There will be a quiz and you answer the question in the quiz and that gives you the attendance credit.

1.8 Labs

1. Lab Quiz (PrairieTest)
2. Demo Lab
3. Office Hours

Prelab (Individual)
Quiz (Individual) Need laptop and ID
Lab Demo (Group). Groups are randomized every week.
This class should be very chill I'll be so fr I'm very glad I picked this class.

Chapter 2

Sampling

Take amplitude at various points in time.

You then reconstruct the continuous signal using just the sampled points.

2.1 CTFT

$$X_a(\Omega) = \int_{\infty}^{\infty} x_a(t)e^{j\Omega t}dt$$

Scale by Ts and make periodic every 2π to get

2.1.1 DTFT

$$X(\omega) = \sum_{\infty}^{\infty} x[n]e^{j\omega n}$$

To avoid overlap,

$$B < \frac{\pi}{T_s} \qquad f < \frac{1}{2T_s}$$

Where T_s is the sampling period so $1/T_s$ is the sampling rate, and B is the angular speed

2.1.2 Nyquist rate

$$F_s = \frac{1}{T_s} > 2f$$

2.1.3 Audio Nyquist Rate

The band max is 20kHz, so the Nyquist rate is

$$\frac{1}{T_s} = 2f = 40Khz$$

So the sampling rate is 40kHz

2.2 Digital Filtering

In Lab 2, we're going to want to take out certain frequencies from our entire noise space.

We can try to use a continuous-time bandstop filter. We could use an RLC circuit, but that has all sorts of consequences.

Instead, we can use a digital filter.

Digital Filters are equivalent to analog filters IF we have Nyquist rate sampling.

2.2.1 Digital Filter

Big silly equation that's in the lecture slides

$$y[n] = (b_0 x[n] + b_1 x[n-1] + \ldots + b_K x[n-K]) -$$
$$(a_0 y[n] + a_1 y[n-1] + \ldots + a_L y[n-L])$$

FIR, if no feedback (L=0)
IIR, if feedback (L $\neq$ 0)

2.2.2 Large N

- CLose to desired response
- Sharper transition
- Less ripples

  BUT

- More computation/memory
- Longer Delay
- (for IIR) possible worse performance

2.2.3 FIR and Convolution

$$y[n] = (b_0 x[n] + b_1 x[n-1] + \ldots + b_K x[n-K])$$
$$y[n] = \sum_{k=0}^{K} b_k x[n-k]$$

2.2.4 Batch vs Block Process

h = filter x = batch samples h*x = ideal output

makes an N-size output

We can have multiple convolution functions, and they can cause discontinuities if we just add them together.

2.2.5 Convolution by Circular Buffer

Challege 1: Block Processing

Audio samples come as a buffer,

which means discontinuities between buffers

Solution: Use a buffer (as a global variable) to store the samples from the previous buffer.

something something more words from the lecture slides.

$$y[n] = \sum_{k=0}^{K} h[k]x[n-k]$$

Assume $K = 2$ and $h[n], y[n] = 0$

Consider a Circular Buffer 0, 1, 2

idk fix later

2.3 OpenSL ES

Open Sound Library for Embedded Systems

Use default sampling rate (48kHz)

The library gives you weird 8 bit sampling

Use a bitwise operature to turn the 8 bit sampled data into the original 16 bit data.

Chapter 3

Spectral Analysis

Lab2 and Quiz 2 on digital filtering and Audio notch filtering.

Spectral analysis give you signal in the time domain.

You can also see signal in the frequency domain.

It shows the relative distribution of signal ”energy” in a different basis

The most common choice is the Fourier basis (frequency)

The magnitude/log, phase and other post-processing are possible.

3.1 Fourier Transforms

Turns time domain into frequency domain or vice versa.

3.1.1 Continuous Time, Continuous Frequency

You use a CTFT, $X_A(\Omega)$

$$X_a(\Omega) = \int_{-\infty}^{\infty} x_a(t) e^{-j\Omega t} dt \qquad \Omega = 2\pi f$$

3.1.2 Discrete Time, Continuous Frequency

You use a DTFT $X(\omega)$

$$X(\omega) = \sum_{-\infty}^{\infty} x[n] e^{-j\omega n} dt \qquad \Omega = 2\pi f$$

3.1.3 Discrete Time and Discrete Frequency

DFT X[k]

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

3.1.4 Continuous Time and Discrete Frequency

Fourier Series $\{a_k\}$

3.2 CTFT vs DTFT vs DFT

- CTFT use an integral and everything is continuous
- DTFT takes in a discrete input and you use a discrete sum, but you get a continuous output
- DFT's exist because you cant integrate or sum to infinity on a computer.

  The relation between all of them is

  $$\frac{f}{F_s} = \frac{\omega}{2\pi} = \frac{k}{N}$$

3.2.1 Consequences to DFT Truncation

In order to use DFT, the length of the input samples must be **finite**

3.3 Time-Windowing

Duration and bandwidth are inverse.

3.3.1 Rectangular Window

$$W_R = \begin{cases} 1 : 0 < t < T \\ 0 : \text{ otherwise} \end{cases}$$

It's just a step function.

If we have a function of just

$$x_a(t) = A\cos(\omega_1 t)$$

Then when we take our DFT, we get

$$X_a(j\Omega) = A\pi\delta(\Omega - \omega_1) + A\pi\delta(\Omega + \omega_1)$$

So we take our Discrete Fourier Transform to get

$$\tilde{x_a}(t) = w_R(t)x_a(t) = \frac{1}{2}Aw_R(t)e^{j\omega_1 t} + \frac{1}{2}Aw_R(t)e^{-j\omega_1 t}$$

$$\tilde{X_a}(\Omega) = w_R(t)x_a(t) = \frac{1}{2}Aw_R(t)(\Omega - \omega_1) + \frac{1}{2}Aw_R(t)(\Omega + \omega_1)$$

3.3.2 Hamming Window

Instead of a step function, it's more of a bump.

There are not artifacts on the side, but the main lobe is more wide.

3.4 Zero-Padding

When the window length L is less than the DFT length N, you add $N - L$ zeros to the end of the sequence. However, it **only** increases the resolution of the DFT, **not the DTFT**.

If you want an actually sharper frequency-domain image, you have to increase the length of the DTFT.

3.5 STFT

So far, we've assumed that signals are periodic and stationary. If this is not true, we have to change our Fourier transform parameters.

A STFT can cut out a segment from a signal and move the window with a shift m

$$X(\Omega, t) = \int_{-\infty}^{\infty} w(t-\tau)x(t)e^{-j\Omega t}\, dt$$

$$X(k, m] = \sum_{n=0}^{N-1} w[n-m]x[n]e^{-j2\pi kn/N}$$

3.6 Uncertainty Principle

Time resolution and frequency resolution cannot be improved simultaneously in the spectrum

3.7 Spectrogram

Magnitude of STFT.

Every timestamp, you perform an STFT to get a 2d plot of frequency over time.

3.7.1 Resolution vs Window Size

Larger windows mean finer frequency resolution. Smaller windows mean.

A Hamming window is a very good option for a spectrogram.

Rectangular windows are very noisy with prominent side-lobes. They are not good for spectrograms.

Chapter 4

Source-Filter Model

Excitation Generator $\rightarrow$ LTI System h(t) (Linear and Time Invariant).

The excitation parameters are

- amplitude (loudness)
- frequency (pitch)
- phase/delay
- Type (voiced, unvoiced, silenced)

The parameters of an LTI system are

- IMPULSE RESPONSE
- resonance frequency

In the frequency space, you can just multiply the generated signal and the LTI system's frequency response.

An LTI system **cannot** create new frequencies in the output.

4.0.1 Speech

Given speech, you can see multiple different syllables, but it has very dynamically changing frequencies and amplitude.

Speech contains an envelope of frequencies, and human speech is contained in a very narrow bandwidth ($< 2000Hz$)

The sampling rate for speech is usually only 4kHz or 8kHz because humans are not very high pitched.

4.1 Characterization of Frames

- Voiced Sounds
- Unvoiced Sounds (P)
- Silence/noise (no active speech)

4.2 Pitch Detection Algorithm

Figure out if sound is voiced or unvoiced.

If voiced, find the frequency. Voiced signals are touder and more sustained. Unvoiced signals are more abrupt.

Pitch calculation is found with autocorrelation

$$R_{xx}[l] = \frac{\sum_{n=0}^{N-1} x[n]x^*[n-l]}{\sum_{n=0}^{N-1} |x[n]|^2}$$

Auto-correlation is an N length vector that spans the possible available lags.

l is defined by a circular shift

$$x^*[< n - l >_N]$$

So the lag wraps around if you're using a negative number (this is already done in python).

The way you figure out the frequency is with good old unit analysis because lag is in samples and sampling rate is samples per second.

$$fs * \frac{1}{l} = \frac{samples}{s} * \frac{1}{samples} = \frac{1}{s} = Hz$$

4.3 Complexity

big O notation.

the autocorrelation function is $O(n^2)$ Because finding the autocorrelation for a single lag is $O(n)$, and then

It is very similar to a circular convolution

$$y[l] = \sum_{n=0}^{N-1} x[n]h[l-n]$$

Convolution involves flipping the out-of-phase portion

The reason we use circular convolution is because of now DFT's work

The autocorrelation can be written as

$$X[k]X^*[k]$$

4.4 Uncertainty Principle

Time resolution and frequency resolution cannot be improved simultaneously in the spectrum.

4.5 Time-Windowing

Effect of time-windowing: Blurring and spreading the original spectrum. To improve the frequency resolution, use a longer time window.

Rectangular window means higher sidelobes. Increase T and decrease $\Delta\Omega$.

4.6 Challenges of Pitch Detection