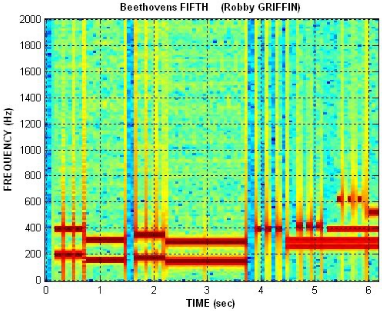
COMS20011 – Data-Driven Computer Science



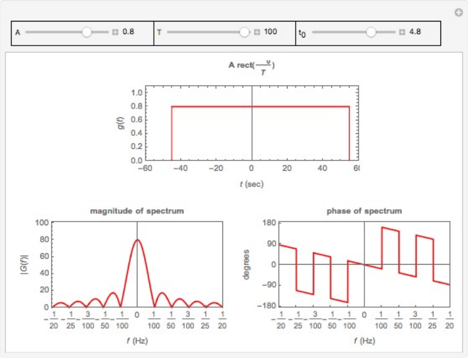
http://dspfirst.gatech.edu/chapters/03spect/demos/spectrog/real/

Lecture Video MM08 – 1D Fourier Transform

March 2022

Majid Mirmehdi

Next in DDCS



https://demonstrations.wolfram.com/RectangularPulseAndItsFourierTransform/

Feature Selection and Extraction

- Signal basics and Fourier Series
- **1D** and 2D **Fourier Transform**
- Another look at features
- Convolutions

Frequency Analysis for Feature Extraction & more...

- The aim of processing a signal using Fourier analysis is to *manipulate the spectrum of a signal* rather than manipulating the signal itself. Example: *simple compression*

- Functions that are ***not periodic*** can also be expressed as the integral of sines and/or cosines weighted by a coefficient. In this case we have the Fourier transform.

1D Fourier Transform

The Fourier Transform of a single variable continuous function $f(x)$ is:

$$F(u) = \int_{-\infty}^{\infty} f(x) \quad e^{-j2\pi ux} dx$$

Conversely, given $F(u)$, we can obtain $f(x)$ by means of the *inverse* Fourier Transform:

$$f(x) = \int_{-\infty}^{\infty} F(u) \quad e^{j2\pi ux} du$$

These two equations are also known as the Fourier Transform Pair.

Note, they constitute a lossless representation of data.

1D Fourier Transform: discrete form

The Fourier Transform of a discrete function of one variable, $f(x),\ x=0,1,2\ldots,N\text{-}1$ is:

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\, e^{\frac{-j2\pi ux}{N}} \qquad \text{for } u = 0,1,2,\ldots,N-1.$$

Conversely, given $F(u)$, we can obtain $f(x)$ by means of the *inverse* Fourier Transform:

$$f(x) = \sum_{u=0}^{N-1} F(u)\, e^{\frac{j2\pi ux}{N}} \qquad \text{for } x = 0,1,2,\ldots,N-1.$$

These two equations are also known as the Fourier Transform Pair.

Note, they constitute a lossless representation of data.

1D Fourier Transform

The concept of the frequency domain follows from *Euler's Formula*:

$$e^{-j\theta} = \cos\theta - j\sin\theta$$

Thus, each term of the Fourier Transform is composed of the sum of *all* values of the function $f(x)$ multiplied by sines and cosines of various frequencies:

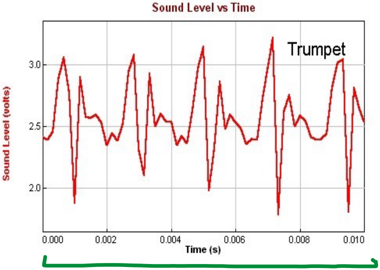
$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\left[\cos\left(\frac{2\pi ux}{N}\right) - j\sin\left(\frac{2\pi ux}{N}\right)\right]$$

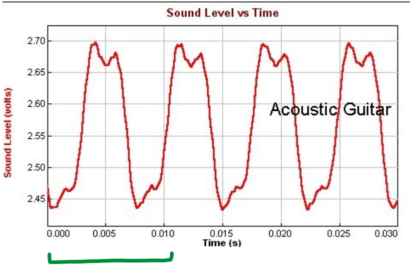
for $u = 0,1,2,\ldots,N-1.$

We have transformed from a time domain to a frequency domain representation.

Example: low and high frequencies

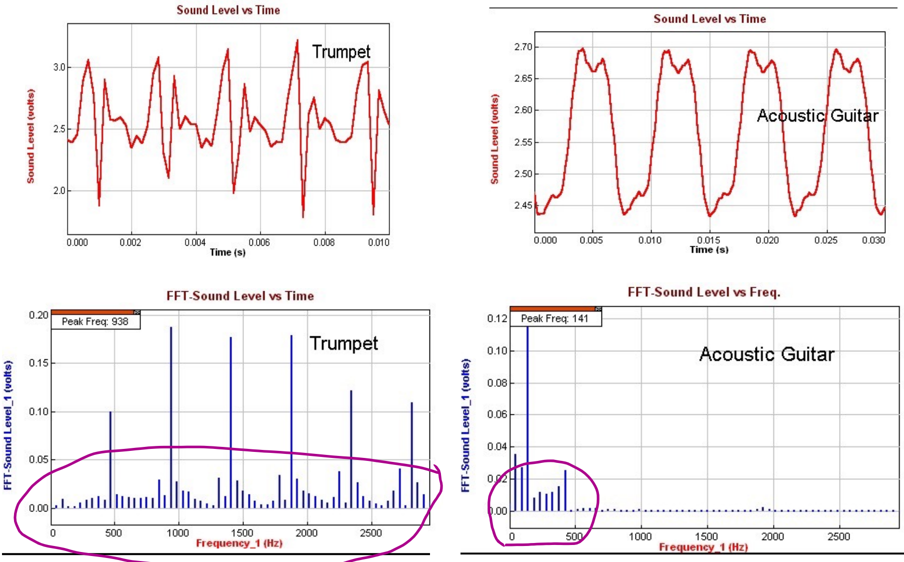
Characteristics of sound in audio signals:





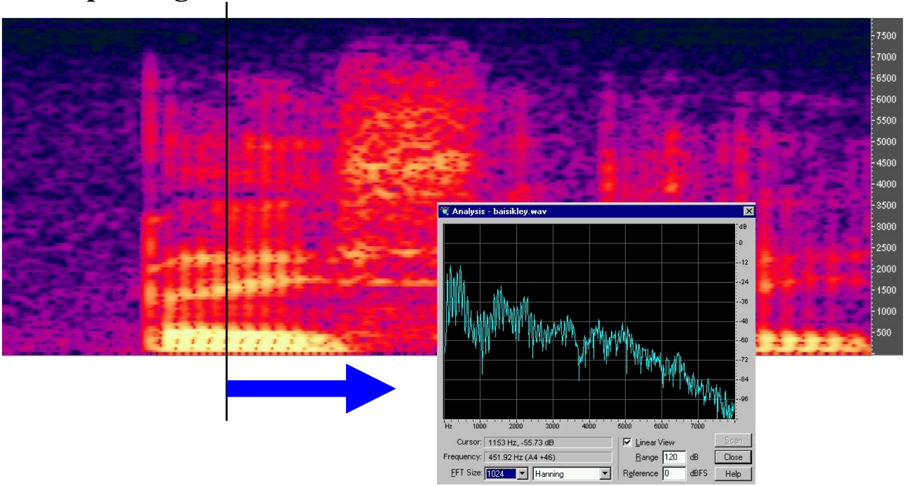
Example: low and high frequencies

Characteristics of sound in audio signals:



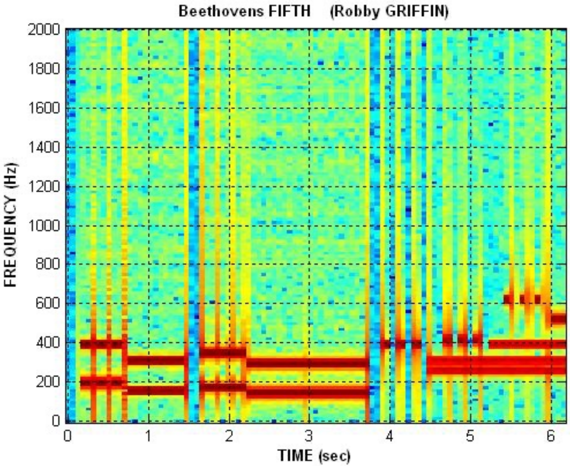
Example: Acoustic Data Analysis

Spectrogram



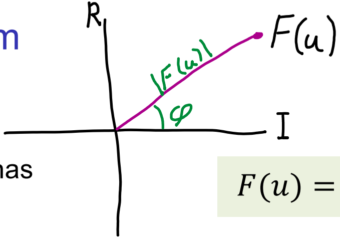
Can you match the sound to the frequencies?

If not, ask me to explain in our Q&A session!



http://dspfirst.gatech.edu/chapters/03spect/demos/spectrog/real/

1D Fourier Transform



F(u) is a complex number & has real and imaginary parts:

$$F(u) = R(u) + jI(u)$$

Magnitude or *spectrum* of the FT:

$$|F(u)| = \sqrt{R^2(u) + I^2(u)}$$

Phase angle or phase spectrum:

$$\varphi(u) = \tan^{-1}\frac{I(u)}{R(u)}$$

Expressing *F(u)* in polar coordinates:

$$F(u) = |F(u)|e^{j\varphi(u)}$$

Very Simple Application Example

Naïve approach: 🤣

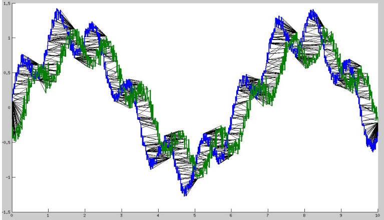
$$E = \sum_{\forall n} (x(n) - y(n))^2$$

Automatic speech recognition between two speech utterances $x(n)$ and $y(n)$:

Problems with this approach?

$$\begin{cases} x(n) = K\, y(n), \text{ yet } E \neq 0 \\ \quad \textit{K being a scaling parameter} \\ x(n) = y(n-m), \text{ yet } E \neq 0 \\ \quad \textit{m causing a delay shift} \end{cases}$$

One solution could be Dynamic Time Warping? (recall from earlier lecture)



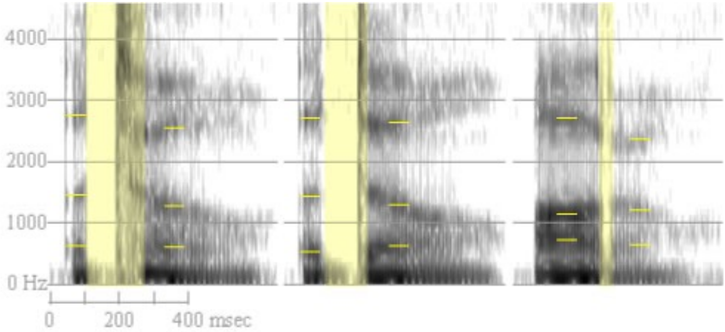
Use Frequency Domain Features!

- Take the Fourier transform of both utterances to get $X(u)$ and $Y(u)$.
- Then consider the Euclidean distance between their magnitude spectrums: $|X(u)|$ and $|Y(u)|$:

$$d_E = \sum_{\forall u} (|X(u)| - |Y(u)|)^2$$

Use Frequency Domain Features!

Can still be a difficult task even in the frequency domain

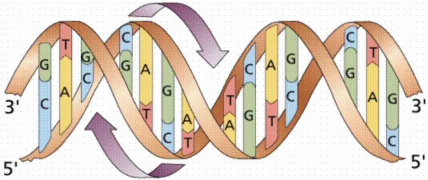


a tot *a dot* *otto*

DNA Sequence Example

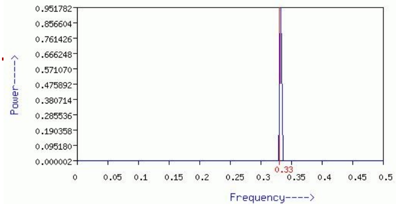
- The analysis of correlations in DNA sequences is used to identify protein coding genes in genomic DNA.
- Locating and characterizing repeats and periodic clusters provides certain information about the structural and functional characteristics of the molecule.
- DNA sequences are represented by letters, A, C, G or T, and –

- e.g. ACAATG-GCCATAAT-ATGTGAAC--GCTCA…



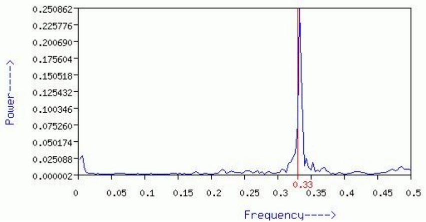
DNA Sequence Example

Consider the periodic sequence A--A--A--A--..... where blanks can be filled randomly by A, C, G or T. This shows a periodicity of 3.



The spectral density of such a sequence is significantly non-zero only at one frequency (0.33) which corresponds to the perfect periodicity of base A $(1/0.333 = 3.0)$.

Destroy the perfect repetition by randomly replacing the A's with all letters…



DNA Sequence Analysis

The computation of Fourier and other linear transforms of *symbolic data* is a big problem.

The simplest solution is to map each symbol to a number. The difficulty with this approach is the dependence on the particular labeling adopted.

Consider, for example, the following symbolic periodic sequence:
s = (ATAGACATAGACATAGAC . . .).

The mapping:
A → 1,
T → 0,
G → 0,
C → 0,
→ Period = 2

The mapping:
A → 1,
T → 2,
G → 3,
C → 4,
→ Period = 6

This clearly shows that some of the relevant harmonic structure can be exposed by the symbolic-to-numeric labelling.

Frequency Spectrum – self-study

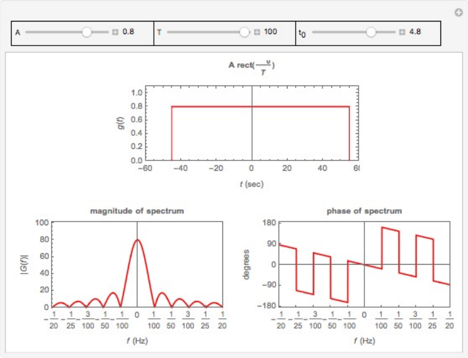
$$|F(u)| = \sqrt{R^2(u) + I^2(u)}$$

- Distribution of $|F(u)|$ → frequency spectrum of signal.
- Features often extracted from the Power Spectrum:

$$PS(f) = |F(u)|^2$$

- Slowly changing signals → spectrum concentrated around low frequencies.
- Rapidly changing signals → spectrum concentrated around high frequencies.
- Also bandlimited signals → frequency content confined within some frequency band.

Next in DDCS



https://demonstrations.wolfram.com/RectangularPulseAndItsFourierTransform/

Feature Selection and Extraction

- Signal basics and **Fourier Series**
- **1D** and **2D Fourier Transform**
- Another look at features
- Convolutions