TECH 350: DSP

Class IX: The Fourier Transform and its Digital Implementations

CLASS PRESENTATION (10%)

As part of the course that engages with uses of DSP within Music Information Retrieval (MIR) students will be asked to prepare and present a short (5-minute) mini-lecture on a professor-assigned topic: low- and high-level features of music or digital audio techniques (see below). These presentations should include 1) a concise, clear introduction of the topic in plain english (3%), 2) a quantitative description of the topic through class-understandable mathematics and/or code (4%) and 3) a few examples of how the topic is engaged with in the context of DSP applications (for example: within an electronic artist's musical practice) (3%).

Low-level Features:

- Zero-Crossing Rate (ZCR) (rough pitch and brightness estimation)
- f0 Estimation (fundamental pitch analysis)
- Spectral descriptors (centroid, spread, kurtosis, flux, etc.)
- Mel-Frequency Cepstral Coefficients (MFCCs)
- Chromagram (pitch-class detection)

High-level Features:

- Tempo estimation (100BPM or andante, for example)
- Key estimation (Bb or Ab, for example)
- Mood classification (happy, sad, or heroic, for example)

Digital Audio Techniques:

- Karplus-Strong Synthesis (plucked string emulation technique)
- Concatenative Synthesis (intelligent granular synthesis technique)
- Phase-vocoding (flexible time- and frequency-shifting technique)
- Linear Predictive Coding (LPC) (speech compression/processing method)
- Deterministic-plus-Stochastic Model (DSM) (speech analysis/resynthesis method)
- Dynamic Time Warping (DTW) (pre-processing for similarity measurement method)

CLASS PRESENTATION (10%)

As part of the course that engages with uses of DSP within Music Information Retrieval (MIR) students will be asked to prepare and present a short (5-minute) mini-lecture on a professor-assigned topic: low- and high-level features of music or digital audio techniques (see below). These presentations should include 1) a concise, clear introduction of the topic in plain english (3%), 2) a quantitative description of the topic through class-understandable mathematics and/or code (4%) and 3) a few examples of how the topic is engaged with in the context of DSP applications (for example: within an electronic artist's musical practice) (3%).

The Discrete Fourier Transform (DFT)

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i 2\pi k n / N}$$

k = 0, 1, …, N-1

Finding the contribution of a particular signal to a frequency: given a signal x of length N, and a particular frequency k, add up, for each member of x (from 0 to N-1), how much each signal member contributes to frequency k (as a complex number, indicating amplitude (real) and phase (imag.)).

2πk is the speed of the complex sinusoid in Radians/sec.

e^{-i} is a clockwise tracing of the circumference of the circle

n/N is a measure of how far we are in the signal

Then, we simply do this for each k in our series

Also, sorry about the i (instead of j!)

The Inverse DFT

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i 2\pi k n / N}$$

n = 0, 1, …, N-1

Finding the contribution of a particular frequency to a signal: given a set of complex sinusoids X, and a signal x of length N, add up, from k = 0 to N-1, how much each frequency contributes to the value of x at time step n.

2πk is the speed of the complex sinusoid in Radians/sec.

e^i is a counter-clockwise tracing of the circumference of the circle

we divide by N to distribute the full amplitude of the signal over each frequency

Then, we simply do this for each n in our signal

Also, sorry about the i (instead of j!)

The Discrete Fourier Transform

$$X_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n e^{i2\pi k \frac{n}{N}}$$

To find the energy at a particular frequency, spin your signal around a circle at that frequency, and average a bunch of points along that path.

Another potentially useful way to break down the DFT

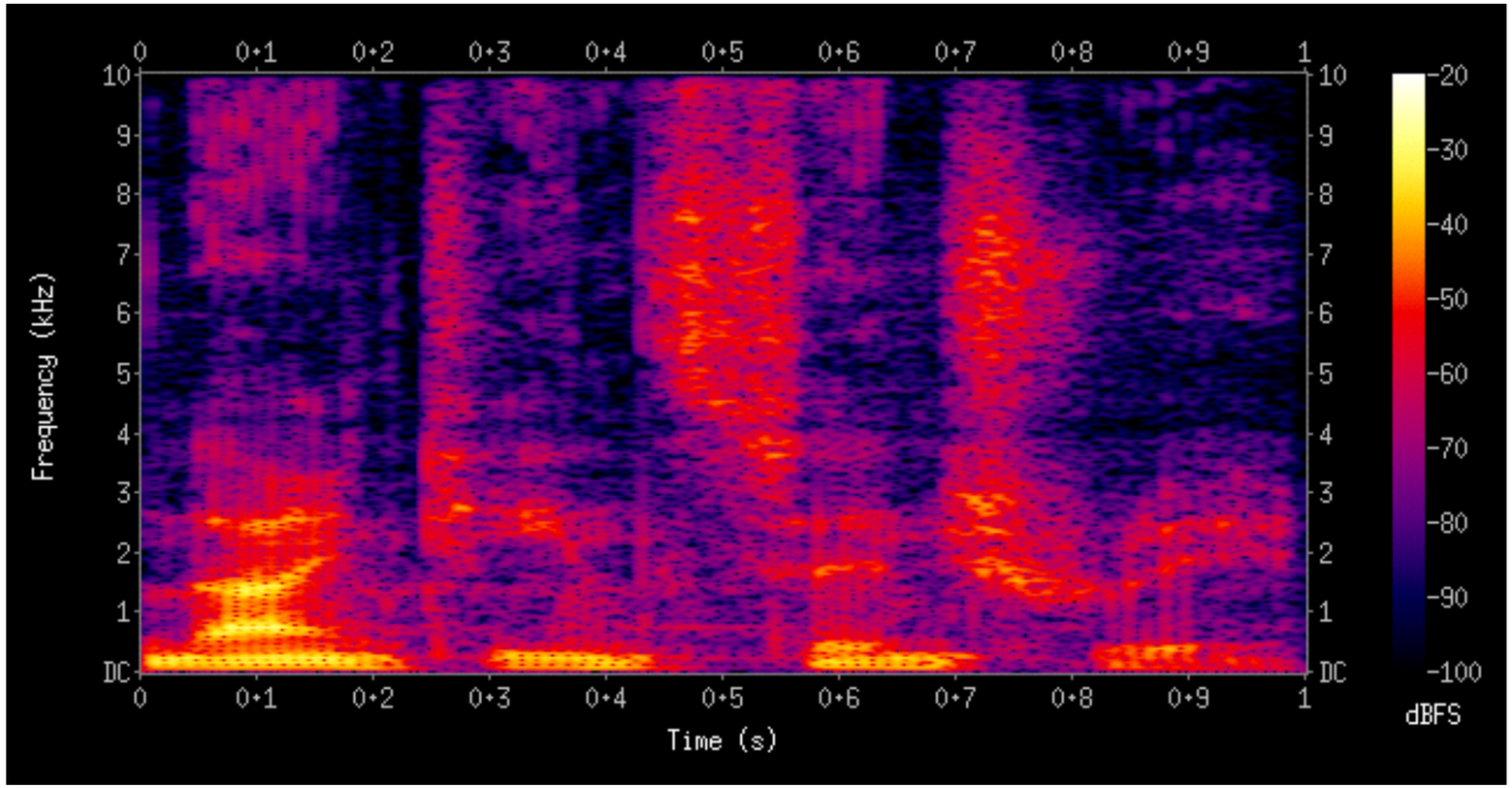
Comparing Fourier Transforms

Time Duration		
Finite	Infinite	
Discrete FT (DFT)	Discrete Time FT (DTFT)	discr.
$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\omega_k n}$	$X(\omega) = \sum_{n=-\infty}^{+\infty} x(n) e^{-j\omega n}$	time
$k = 0, 1, \ldots, N-1$	$\omega \in [-\pi, +\pi)$	n
Fourier Series (FS)	Fourier Transform (FT)	cont.
$X(k) = \frac{1}{P} \int_0^P x(t) e^{-j\omega_k t} dt$	$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt$	time
$k = -\infty, \ldots, +\infty$	$\omega \in (-\infty, +\infty)$	t
discrete freq. k	continuous freq. ω	

Comparison of types of Fourier Transforms

The Short-Time Fourier Transform (STFT)

The Short-Time Fourier Transform: given a signal, separate it into a set of sub-signals (windows), and calculate the DFT on each. This allows us to have information on how the spectrum changes over time (at discrete intervals, called frames), often visualized as a **sonogram or spectrogram**:



The Short-Time Fourier Transform (STFT)

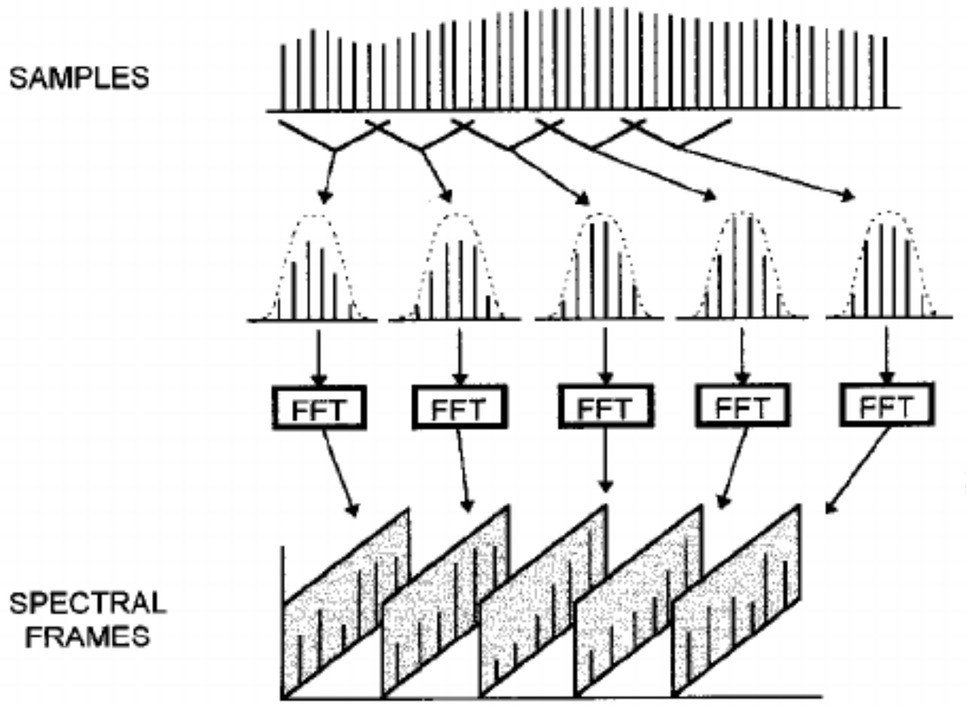
Parameters of the STFT:

Window Size **-** Number of samples in the window

Window Type **-** The type of envelope (windowing function) applied to the digital signal

Window Step (AKA hop size) - How many samples are moved between calculating windows, determining how much the windows ***overlap***

FFT Size **-** the number of FFT bins (fftsize/2) that the spectral energy of the signal is put into. If too few, will get a poor understanding of the spectrum of the sound (leakage). Can oversample, improving frequency resolution, but that becomes computationally slower.



The Short-Time Fourier Transform (STFT)

Choice of windowing function = very important!

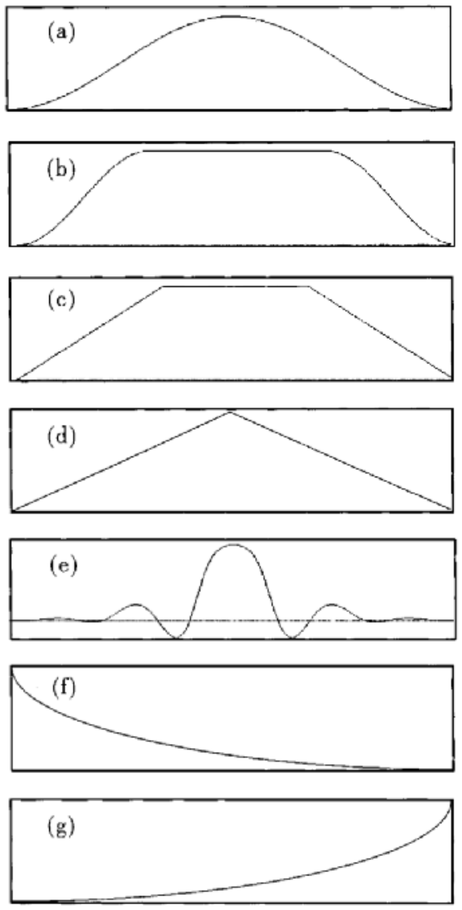
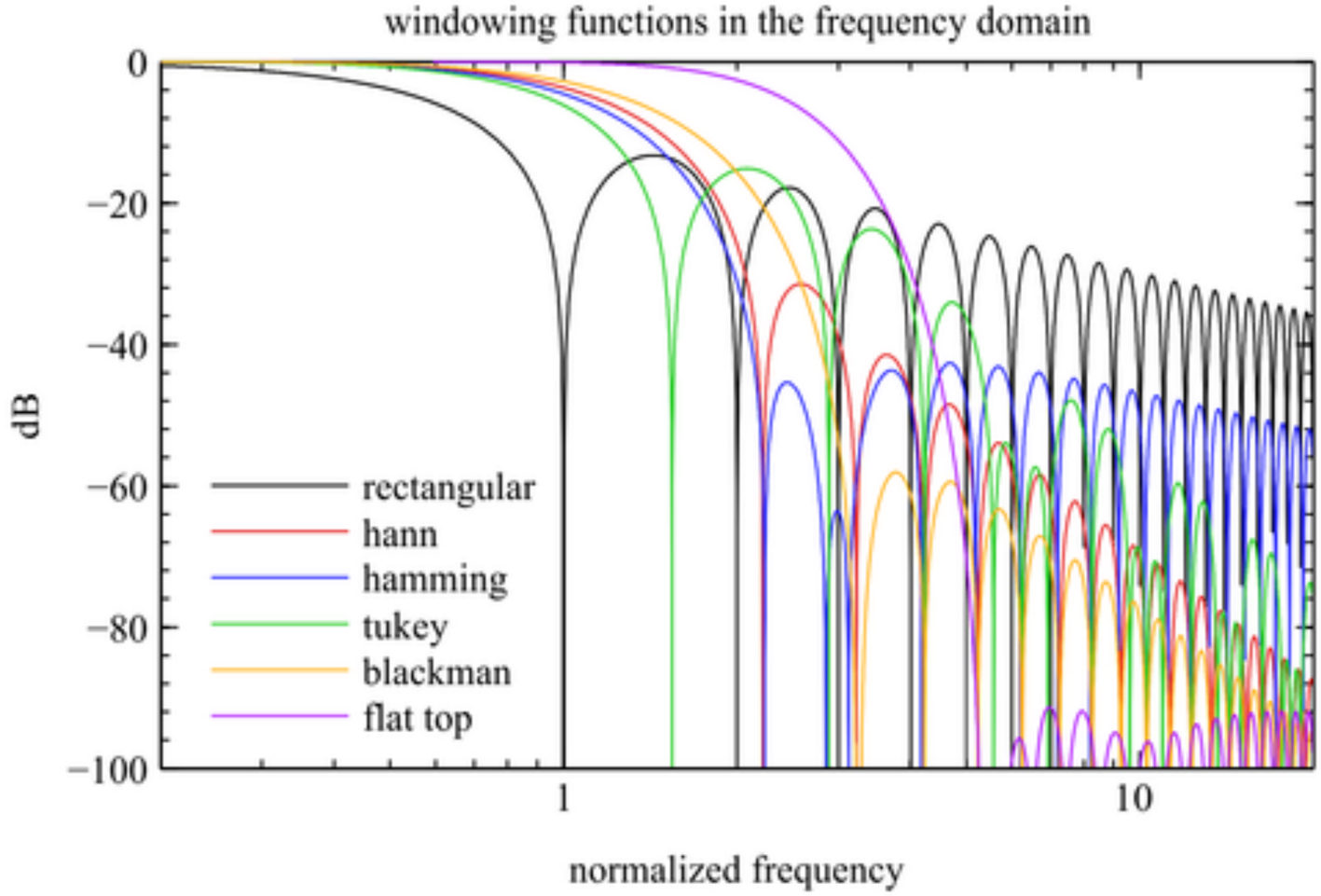


Figure 3.2 Grain envelopes. (a) Gaussian. (b) Quasi-Gaussian. (c) Three-stage line segment. (d) Triangular. (e) Sinc function. (f) Expodec. (g) Rexpodec.



The Short-Time Fourier Transform (STFT)

Most important property of working with the STFT:

Time resolution is inversely proportional to frequency resolution

In other words: The better you can represent transients (attacks), the worse you can represent their frequency content. The better you can represent frequency content, the more smearing of transients you have. **Solution:** Find a happy medium between window size and FFT size and other STFT parameters, depending on the sonic content you're working with (e.g., voice vs. percussion)

The Fast Fourier Transform (FFT)

The Fast Fourier Transform: *a computationally efficient method for computing the DFT of signal* (*O (N log N)* vs. O (N^2)).

Requires **the length of the signal to be a power of 2** (1, 2, 4, … 1024, 2048, 4096, etc.), because the algorithm divides the signal into two parts and calculates the DFT on each of them, recursively.

The FFT is used nearly every time you compute the DFT on a modern computer.

An oft-used (and free) C-based implementation is the "FFTW," the "Fastest Fourier Transform in the West."

FIGURE 12-7
Flow diagram of the FFT. This is based on three steps: (1) decompose an N point time domain signal into N signals each containing a single point, (2) find the spectrum of each of the N point signals (nothing required), and (3) synthesize the N frequency spectra into a single frequency spectrum.