Audio Signal Processing Basics

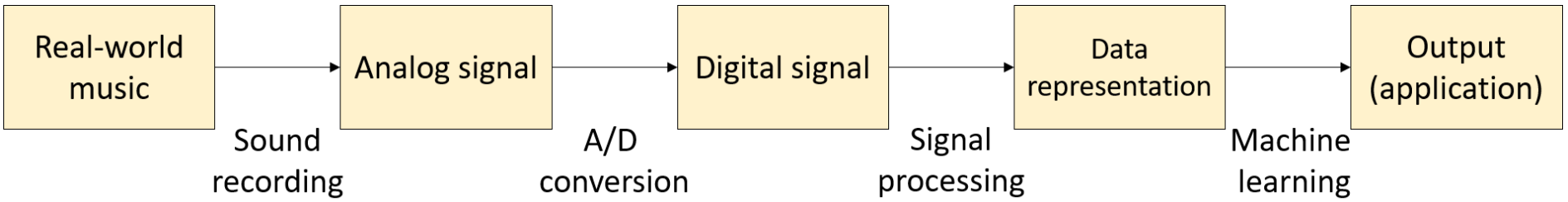
Li Su

Institute of Information Science
Academia Sinica

lisu@iis.sinica.edu.tw

February 24, 2019

Prologue: the road of a music AI system



- Knowing how the data come from is very important.
- What is frequency? what is a spectrum?
- What is the sampling theorem?
- How the Fourier transform works?
- Filter, linear filter, nonlinear filter: filter everywhere?

Important methodology in designing audio data representation:

- Time-frequency analysis
- Digital filters and filterbanks
- Wavelet transforms

Overview

Types of signals

Can you tell the difference among the following terms?

- Continuous (連續)
- Analog (類比)
- Discrete (離散)
- Digital (數位)

Roughly speaking,

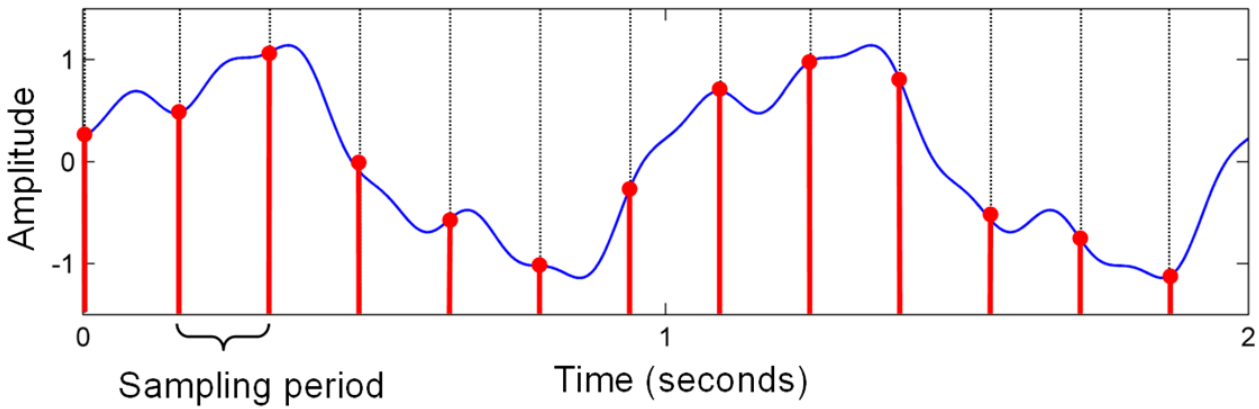
- What we perceive: continuous and analog
- What we process: discrete and digital

Sampling

- Sampling: make a *continuous*-time signal be *discrete*-time
- A discrete-time signal $x[n]$, $n \in \mathbb{Z}$, sampled from $f(t)$, $t \in \mathbb{R}$, by a sampling period T

$$x(n) = f(nT) \tag{1}$$

- Sampling frequency $f_s = \frac{1}{T}$, we have $n = tf_s$
- Audio signals: $f_s = 8$ kHz, 16 kHz, 22.05 kHz, 44.1 kHz, etc.

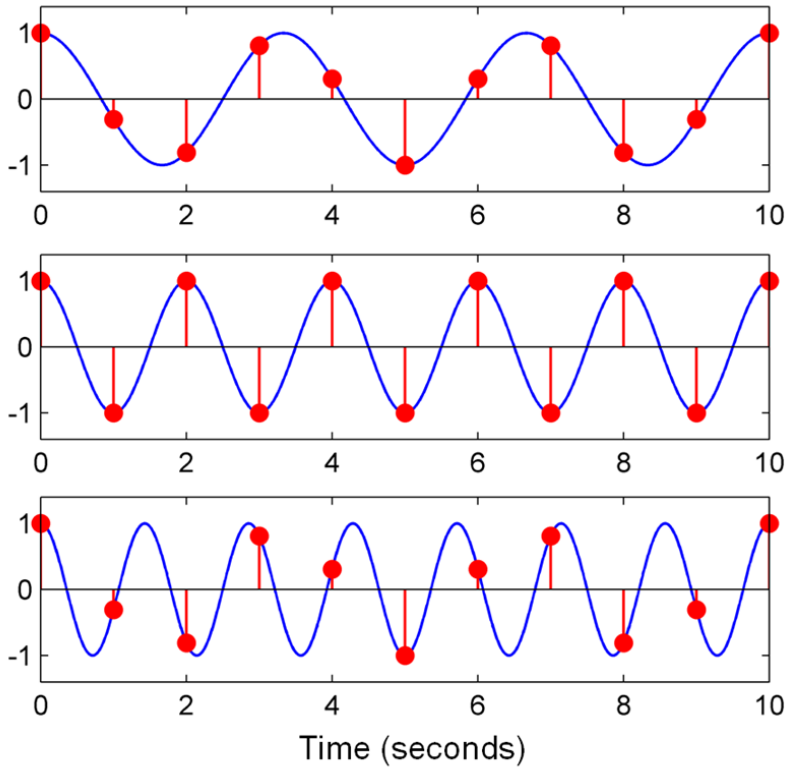


From: M. Müller, *Fundamentals of Music Processing*, Chapter 2, Springer 2015

- $T =?$
- $f_s =?$

Aliasing

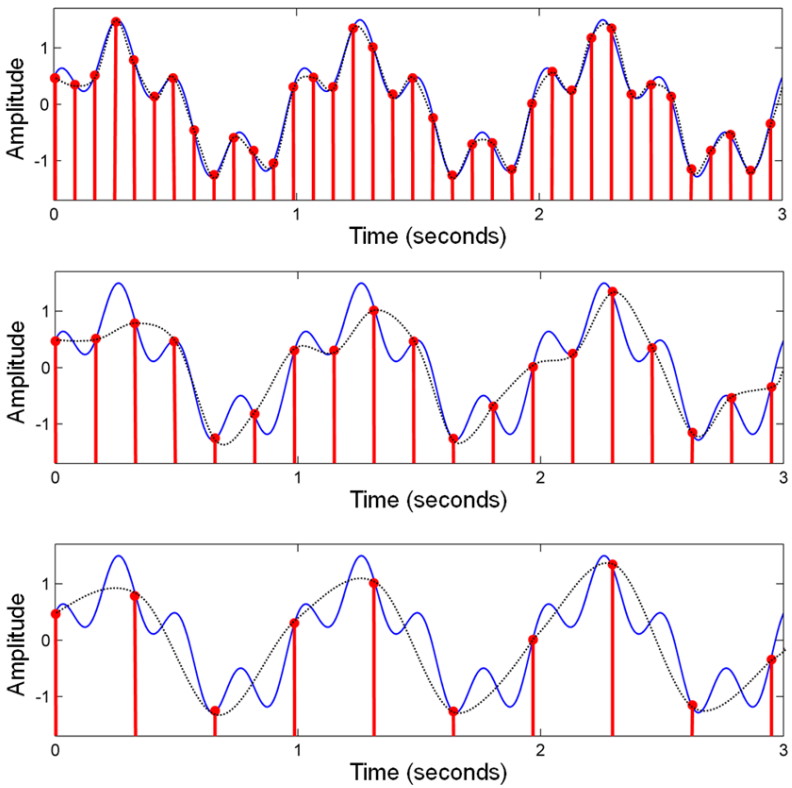
- What do we lost when performing sampling?
- Limited sampling rate implies limited information of the spectrum
- **Nyquist-Shannon sampling theorem**: if the highest frequency of a signal $f(t)$ is f_h, then the signal can be perfectly reconstructed by a sampled signal with sampling rate of at least $2f_h$
- If sampling rate smaller than $2f_h$: aliasing



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 2, Springer 2015

Aliasing effects in image, video and sound

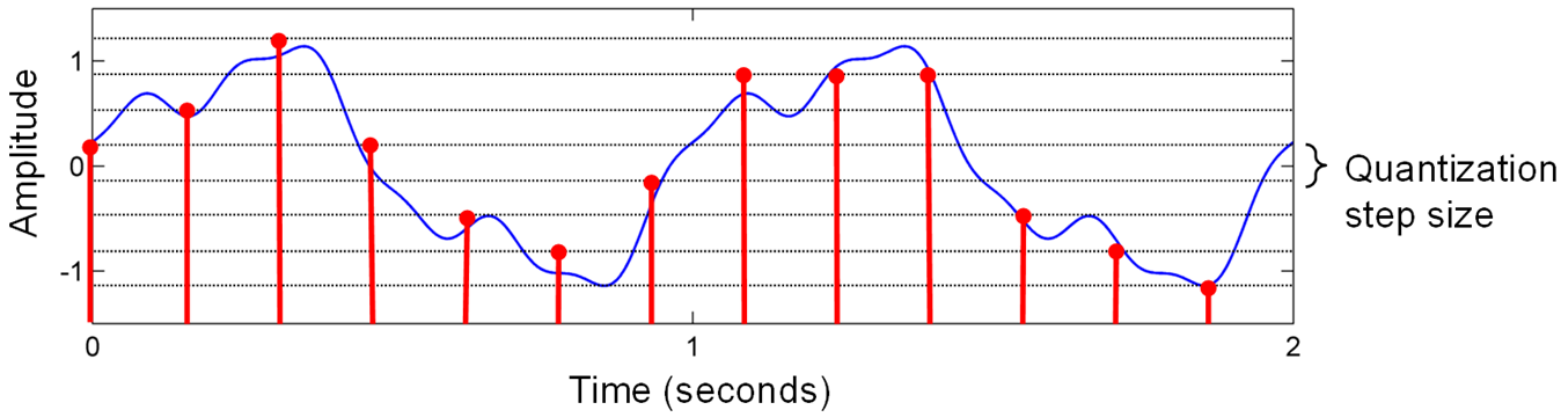
▸ Video example and ▸ Audio example



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 2, Springer 2015

Quantization

- Quantization: sampling in amplitude
- Quantization – make a *analog*-value signal *digital*
- Bit depth: number of bits of information in each sample
- Example: bit depth $= 4$
- Audio signals: 16-bit, 24-bit or more
- For an audio signal with 2-channel, 16-bit and sampling rate 44.1 kHz, what is its bit rate?



From: M. Mueller, *Fundamentals of Music Processing,* Chapter 2, Springer 2015

Basic format of audio signals

Waveform Audio File (.WAV) format

- *Discrete* and *digital* audio signals
- Pulse-code modulation
- Contains 'header' + 'chunk' + 'data'
- Sampling rate (f_s)
- Bit depth: number of bit per sample
- Number of channel:

Other audio file formats

- .AIFF, .MP3, .OGG, etc.
- Free software: ffmpeg

Fourier transform

"Mathematics compares the most diverse phenomena and discovers the secret analogies that unite them."

– Joseph Fourier (1768 – 1830)

Frequency and spectrum

- Signal model: a signal $f(t) \in \mathbb{R}$ contains components of different frequencies
- For physical meaning, the term "frequency" is defined in terms of sinusoidal function $\cos(\omega t)$ and $\sin(\omega t)$, where ω is the *angular frequency* (rad/s)
- The spectrum shows the intensity and phase of the sinusoidal functions composing $f(t)$
- Fourier analysis facilitates this model
- Fourier analysis is a classic method of retrieving the spectrum of a signal

Fourier series

- A periodic signal $f(t) \in \mathbb{R}$ can be represented in terms of an infinite sum of sines and cosines.

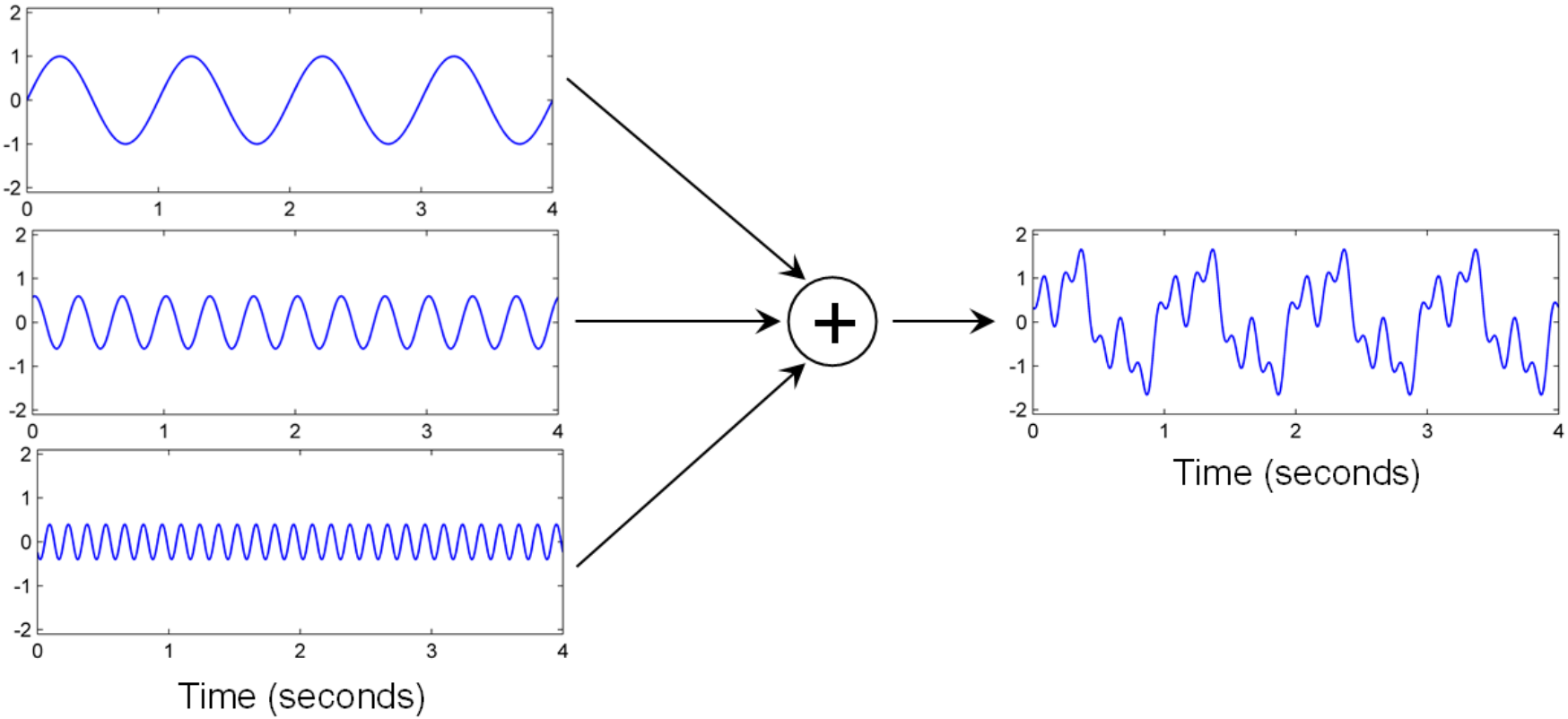
$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(\omega_n t) + \sum_{n=1}^{\infty} b_n \sin(\omega_n t) \tag{2}$$

$$a_0 = \int_{-\pi}^{\pi} f(t)dt, \quad a_n = \int_{-\pi}^{\pi} f(t)\cos(\omega_n t)dt, \quad b_n = \int_{-\pi}^{\pi} f(t)\sin(\omega_n t)dt$$

- The frequencies of these sines and cosines are constructed by a *harmonic series*, containing a fundamental frequency (i.e., ω_0) and its integer multiples (harmonics); $\omega_n = n\omega_0$.
- These sines and cosines form an orthogonal basis, and the Fourier coefficients are the projection of $f(t)$ on the basis.

Illustration

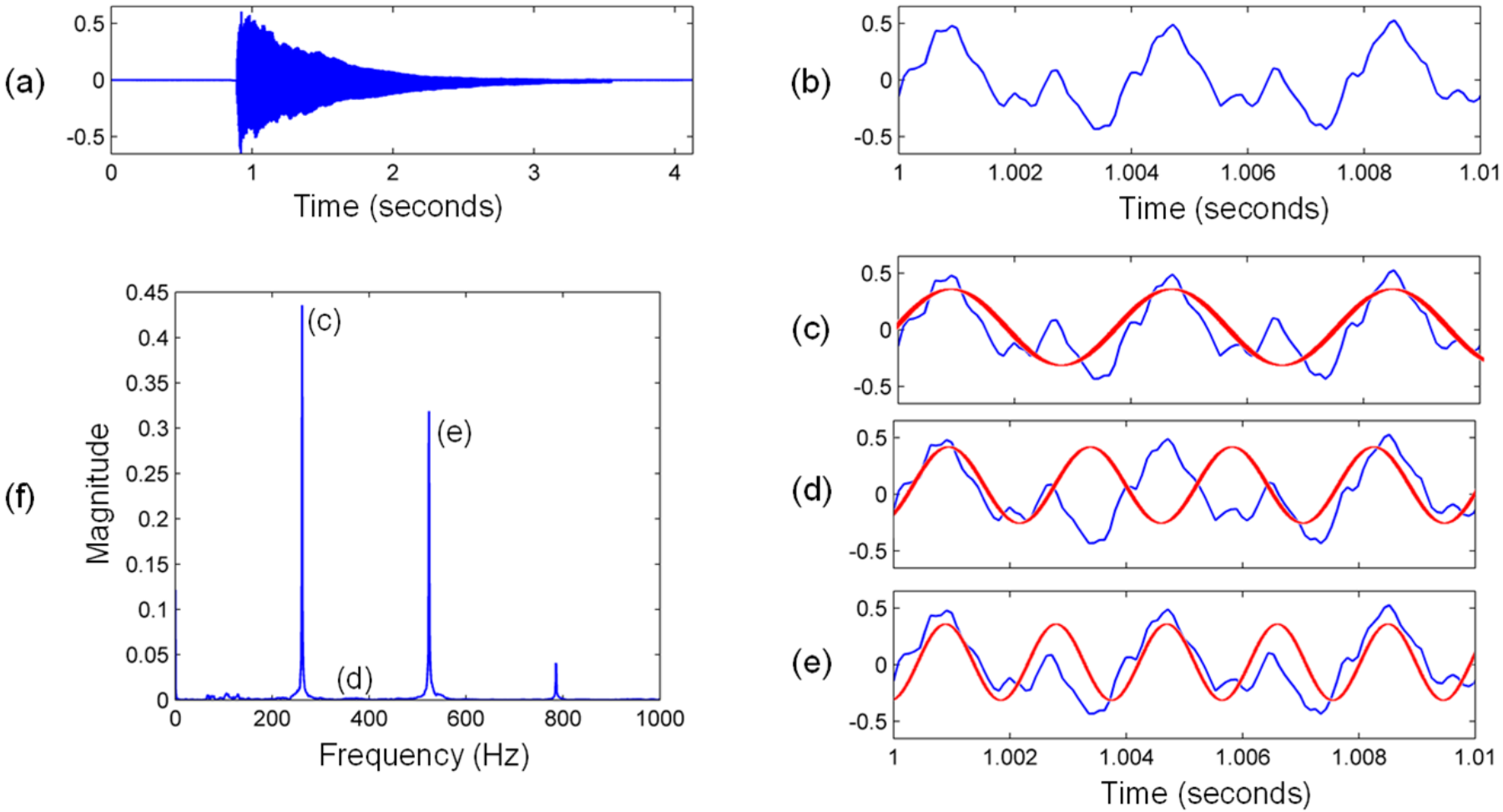
A synthetic viewpoint:



From: M. Mueller, "*Fundamentals of Music Processing*," Chapter 2, Springer 2015

Example: a note C4 played on a piano

- Fundamental frequency $f_0 = 261.6$ Hz

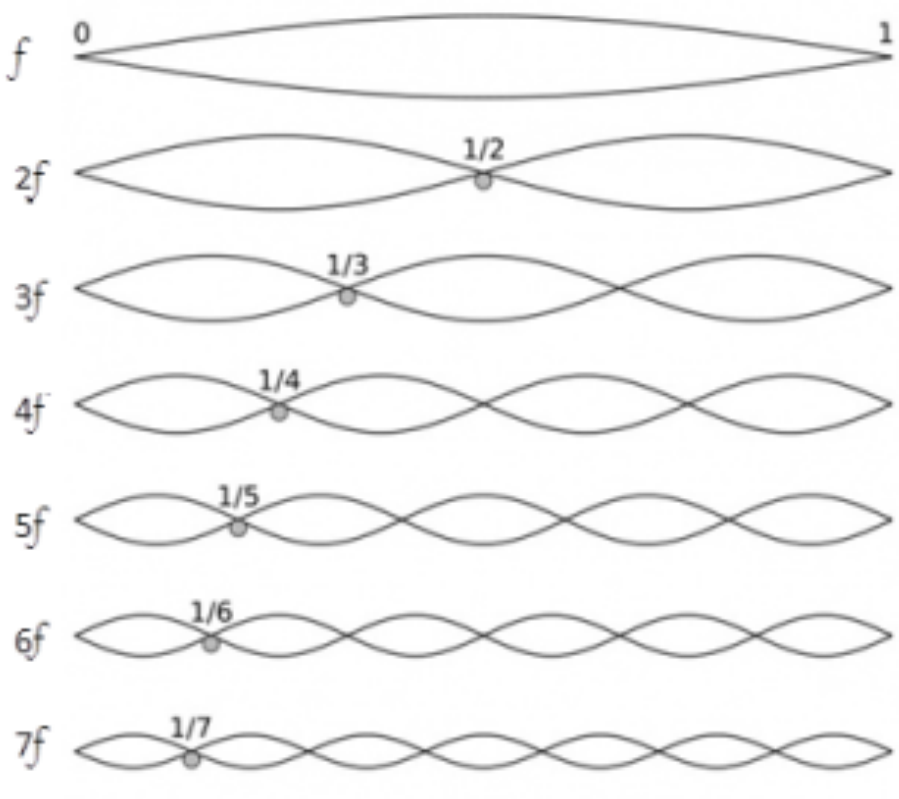


From: M. Mueller, *Fundamentals of Music Processing*, Chapter 2, Springer 2015

Physical interpretation

For a bounded vibrating string with length L:

- Fundamental mode with wavelength $\lambda_0 = \frac{L}{2}$
- Fundamental frequency $f_0 = \frac{2v}{L}$
- 1st harmonic frequency $2f_0$
- 2nd harmonic frequency $3f_0$
- Pitched musical signal $\rightarrow$ f_0+ higher order harmonics $\rightarrow$ periodic



Fourier transform

The Fourier transform is a generalization of the Fourier series, by changing the sum to integral.

- A signal $f(t) \in \mathbb{R}$ can be represented as

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \,, \tag{3}$$

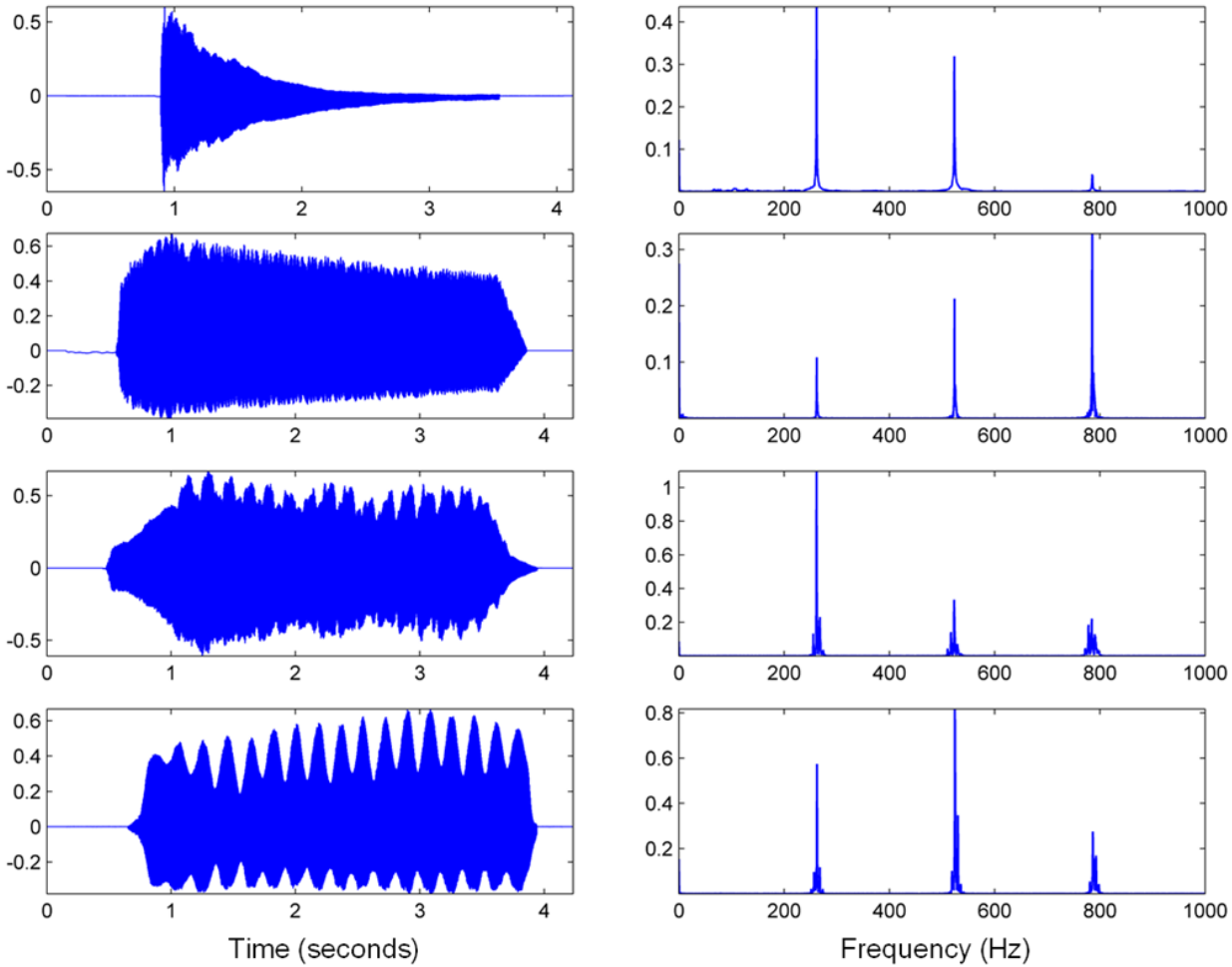
- The inverse Fourier transform

$$f(\omega) = \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \,. \tag{4}$$

- Fourier series: only for analyzing *periodic* signals
- Fourier transform: suitable for analyzing all (including non-periodic) signals

More examples

(a) Piano (b) Trumpet (c) Violin (d) Flute



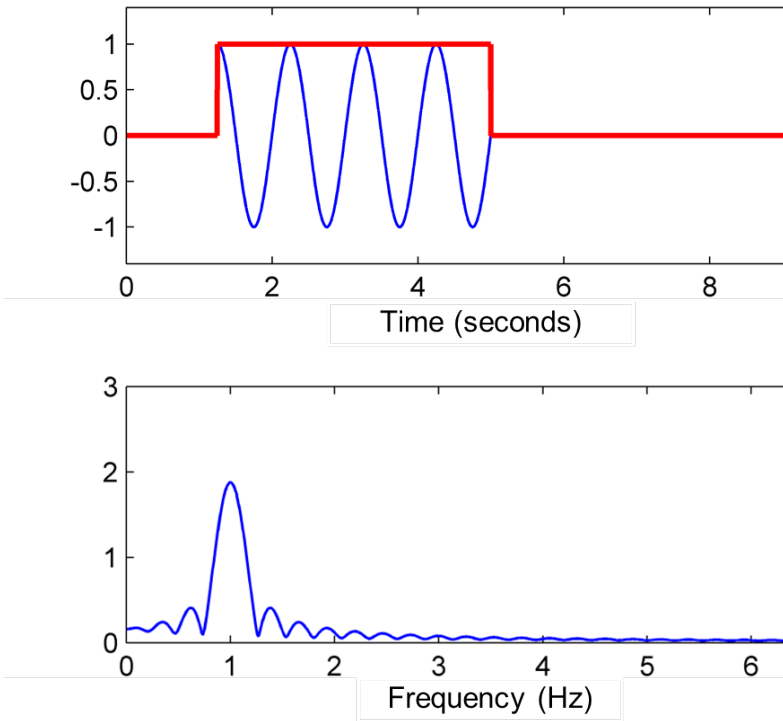
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 2, Springer 2015

Spectral leakage

- We are never able to measure a signal of infinite length
- Fourier transform on a *finite support* $[T_A, T_B]$:

$$F(\omega) = \int_{T_A}^{T_B} f(t) e^{-j\omega t} dt$$

- Spectral leakage: the spectrum of a sinusoidal function is never a impulse
- Heisenberg's uncertainty: better localization in time implies worse localization in frequency



From: M. Mueller, *Fundamentals of Music Processing*, 2, Springer 2015

Discrete Fourier Transform (DFT)

For a discrete-time signal (usually sampled from a continuous-time signal) $\vec{x} \in \mathbb{R}^N$, $\mathbf{x} = [x(0), x(1), \ldots, x(N-1)]$

$$X(k) = \sum_{0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}} \tag{5}$$

Let $\mathbf{X} \in \mathbb{R}^N$, $\mathbf{X} = [X(0), X(1), \ldots, X(N-1)]$ and $\omega = -\frac{j2\pi}{N}$

$$\mathbf{X} = \mathbf{F}\mathbf{x}, \quad \mathbf{F} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix} \tag{6}$$

Interpretation of DFT (1)

- Frequency resolution $= \frac{f_s}{N}$ (the distance between two bins)
- When N is even:

$$X(0) = \sum_{n=0}^{N-1} x(n) \qquad f = 0$$

$$X(1) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi n}{N}} = \sum_{n=0}^{N-1} x(n) e^{-j2\pi\left(\frac{f_s}{N} t\right)} \qquad f = \frac{f_s}{N}$$

$$\vdots$$

$$X\left(\frac{N}{2} - 1\right) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi n}{N} \cdot \left(\frac{N}{2} - 1\right)} \qquad f = \frac{f_s}{2} - \frac{f_s}{N}$$

Interpretation of DFT (2)

$$X\left(\frac{N}{2}\right) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi n}{N}\cdot\left(\frac{N}{2}\right)} \qquad f = \frac{f_s}{2}$$

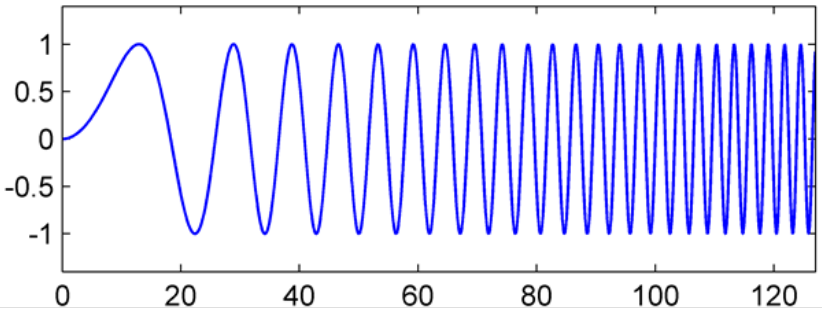
$$X\left(\frac{N}{2}+1\right) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi n}{N}\cdot\left(\frac{N}{2}-1\right)} \qquad f = -\frac{f_s}{2} + \frac{f_s}{N}$$

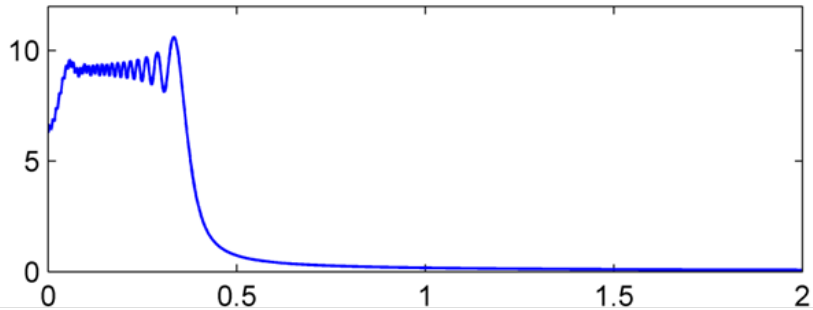
$$\vdots$$

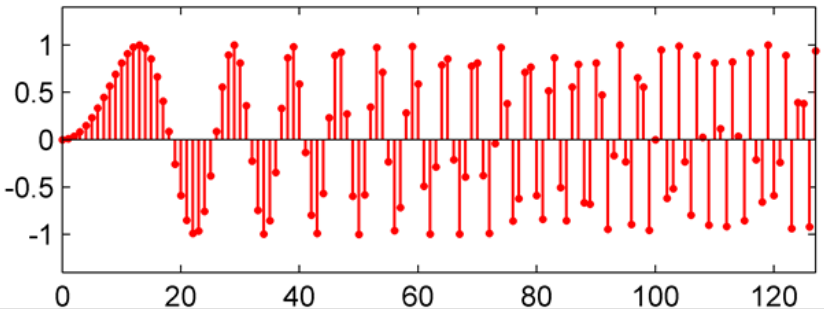
$$X(N-1) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi n}{N}\cdot\left(\frac{N}{2}-1\right)} \qquad f = -\frac{f_s}{N}$$

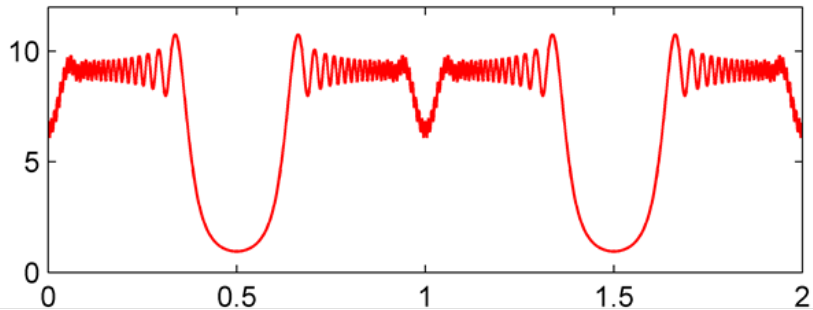
Example (1): normal case

A *chirp* signal $f(t) := \sin(0.003\pi t^2)$ for $t \geq 0$





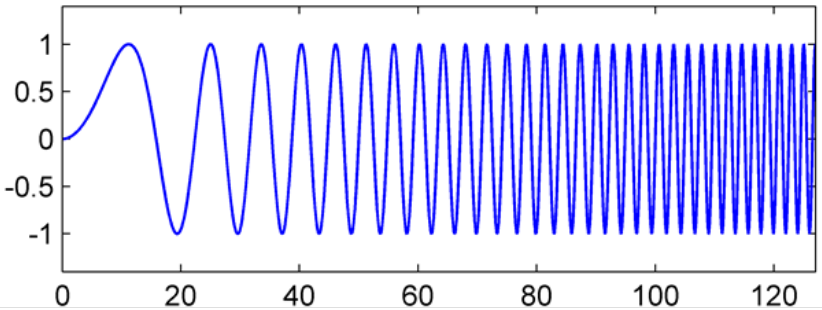


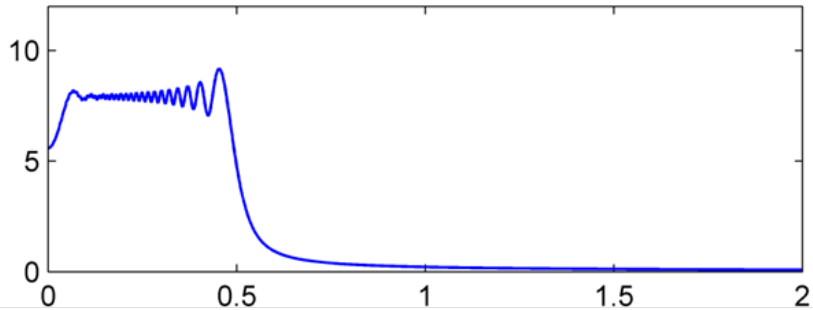


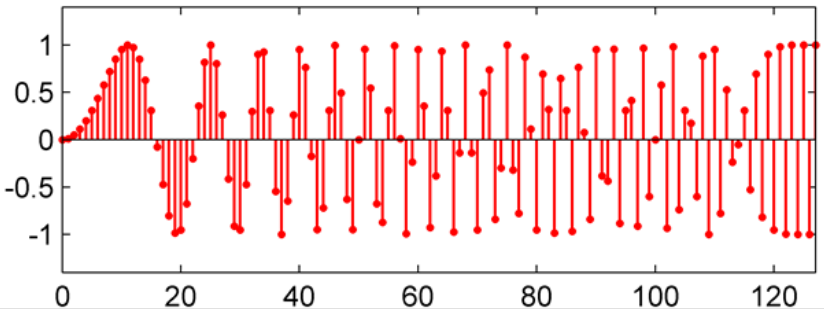
From: M. Mueller, *Fundamentals of Music Processing,* Chapter 2, Springer 2015

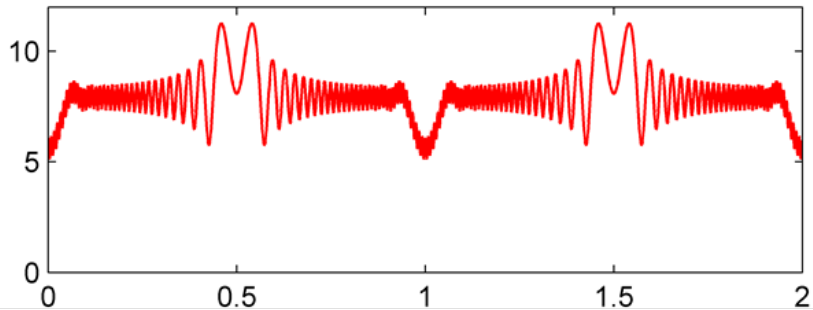
Example (2): aliasing

A *chirp* signal $f(t) := \sin(0.004\pi t^2)$ for $t \geq 0$



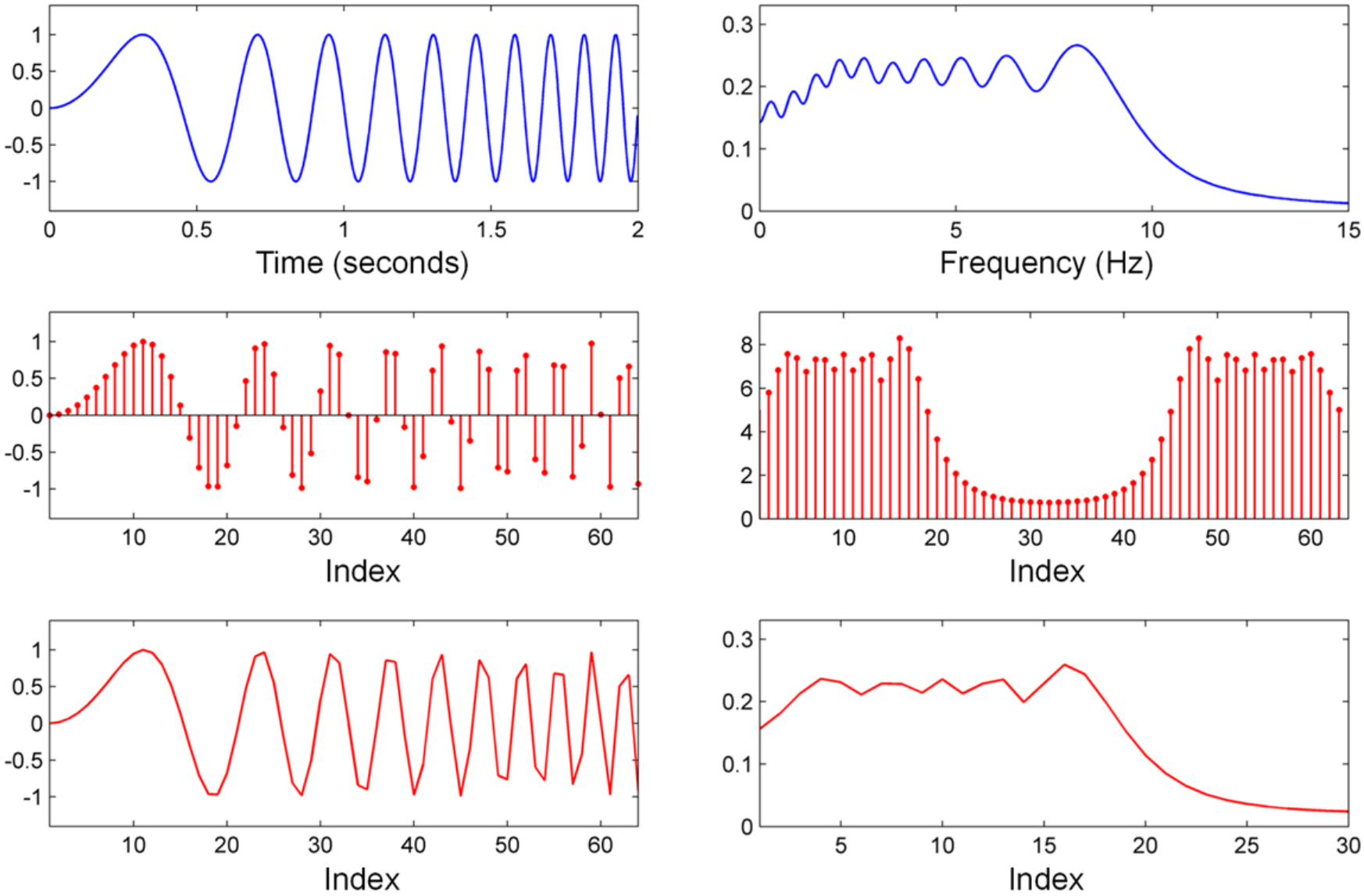






From: M. Mueller, *Fundamentals of Music Processing*, Chapter 2, Springer 2015

Example (3): approximation property



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 2, Springer 2015

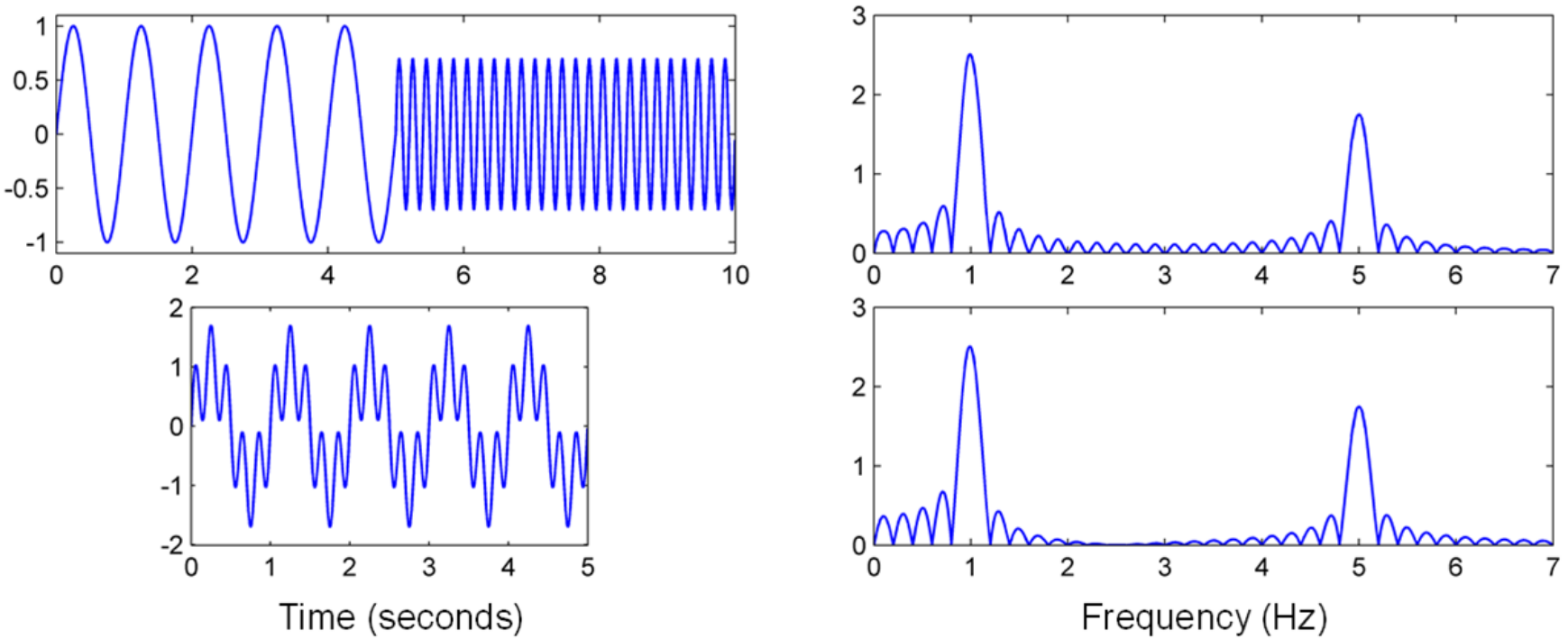
Non-stationarity of signals

- Fourier analysis assumes that the amplitude/frequency/phase of a signal do not change over time
- But this never happens in real world
- Music is non-stationary: onset, offset, pitch change, etc.

$$f(t) = \sum_m A_m(t) \cos(\omega_m(t)t + \phi_m(t)) \tag{7}$$

- How to model the temporal dynamics of a non-stationary signal?
- How to model time and frequency information at the same time?

Example



From: M. Mueller, *Fundamentals of Music Processing,* Chapter 2, Springer 2015

Short-Time Fourier Transform (STFT)

- Dennis Gabor (1900 – 1979)
- Nobel Prize in Physics (1971)
- Known for Holography, Time-frequency analysis, etc.

– Google Doodle on June 5, 2010

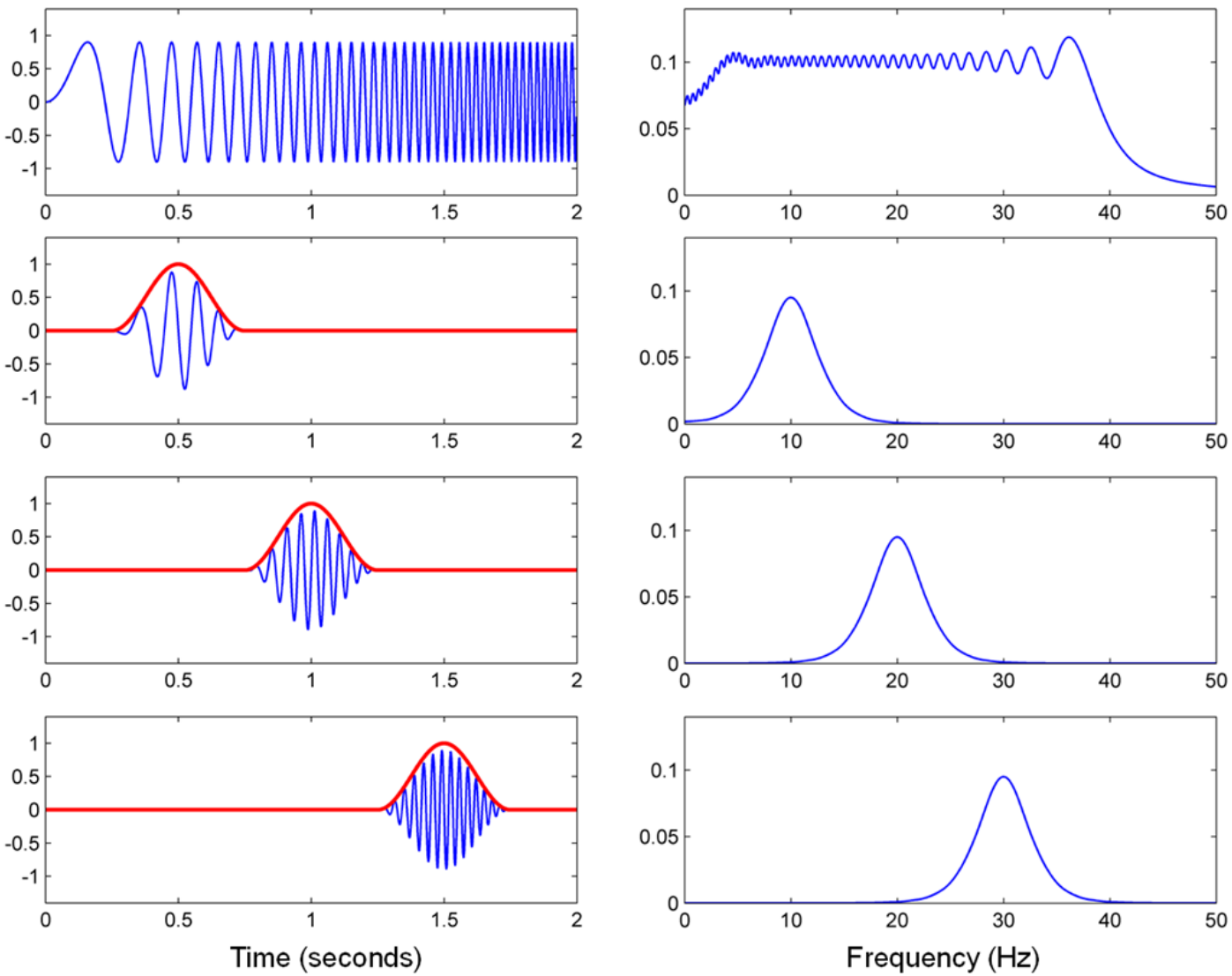
Short-Time Fourier Transform (STFT)

- Short-time Fourier transform (STFT): capture *local* information through a sliding window function $h(t)$

$$S_x^{(h)}(t,\omega) = \int x(\tau)h(\tau - t)e^{-j\omega\tau}d\tau, \tag{8}$$

- Also have magnitude and phase
- Spectrogram: $|S_x^{(h)}|^2$

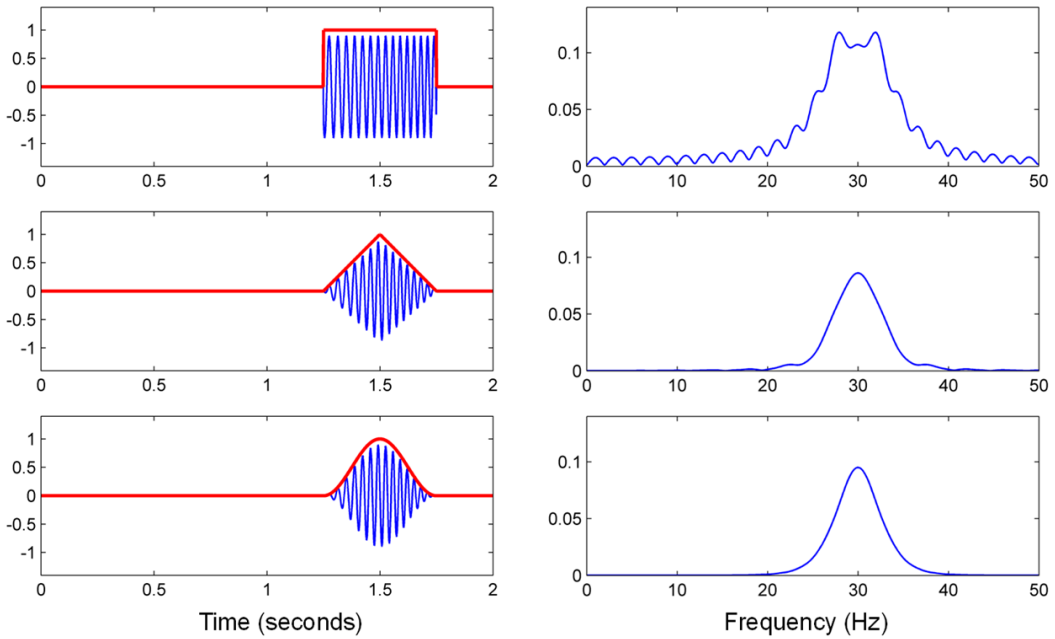
Sliding window and short-time spectrum



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 2, Springer 2015

Window function

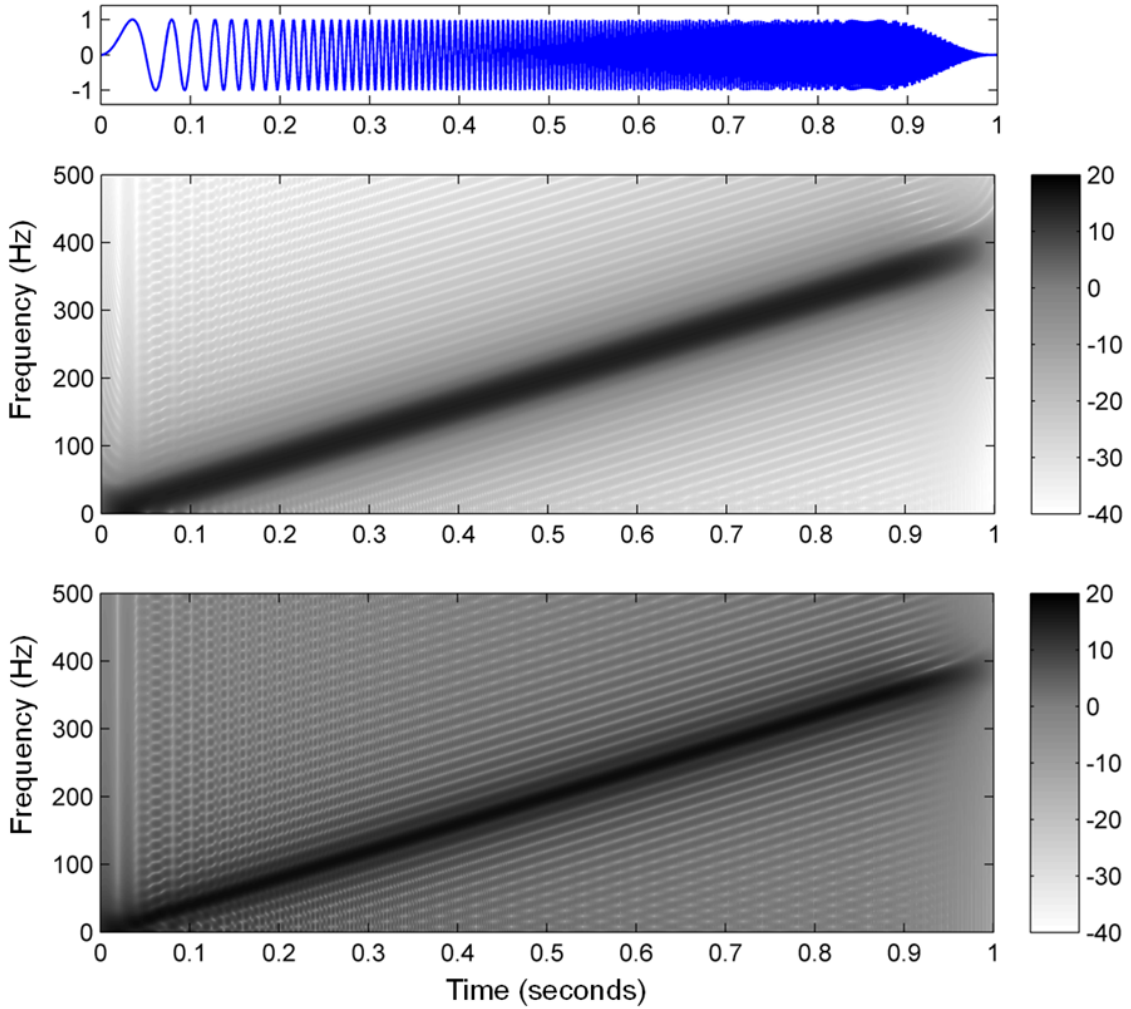
- For better temporal continuity, better control of spectral leakage and less ripple artifacts (than using rectangular window)
- Hann window: $h(t) = \frac{1}{2} + \frac{1}{2}\cos\left(\frac{2\pi n}{N}\right), \quad n = -\frac{N}{2}, \ldots, \frac{N}{2}$
- Comparison of rectangular, triangular and Hann windows:



From: M. Mueller, *Fundamentals of Music Processing,* Chapter 2, Springer 2015

Illustration (1): window type

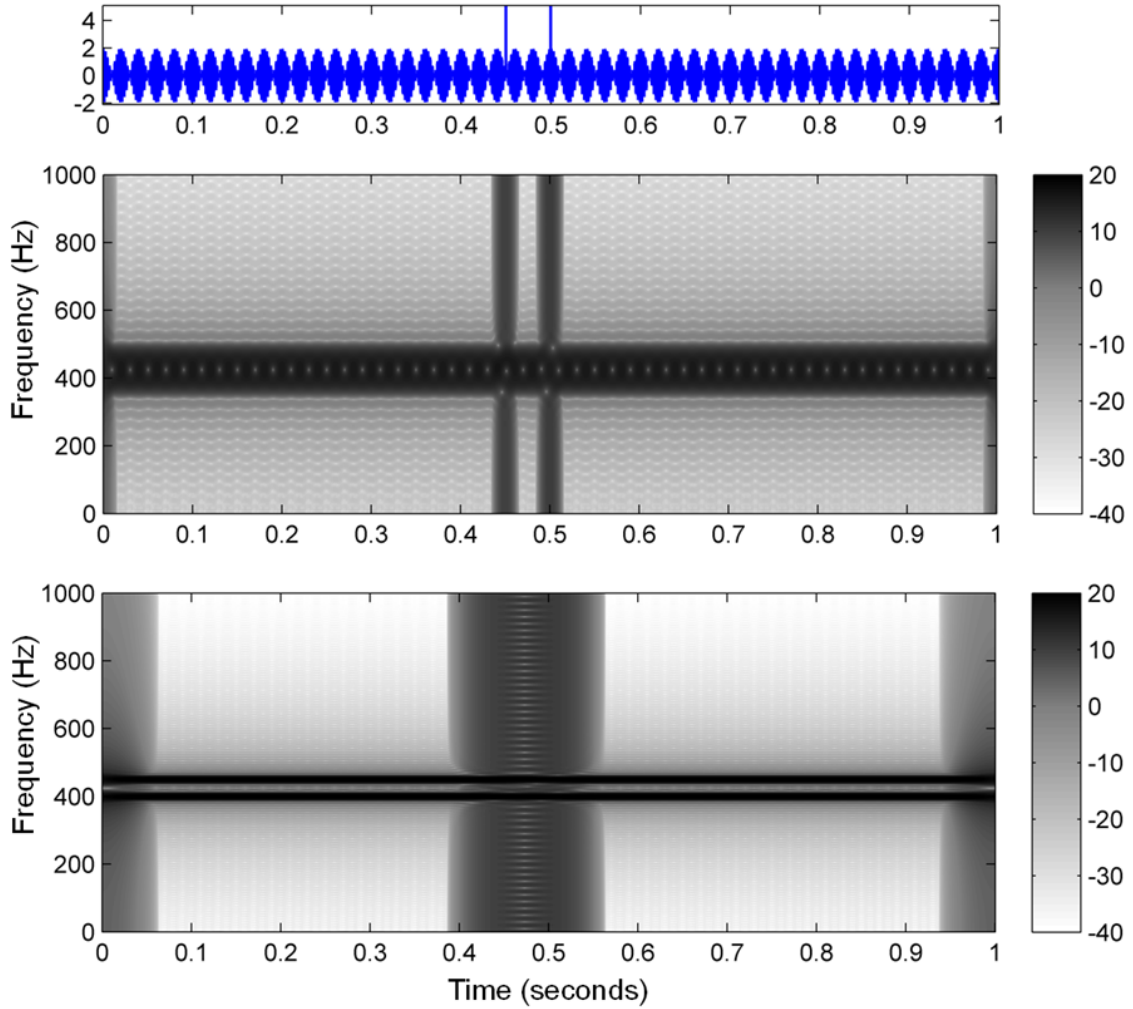
Hann window (up) and rectangular window (bottom)



From: M. Mueller, *Fundamentals of Music Processing,* Chapter 2, Springer 2015

Illustration (2): window length

Short Hann window (32 ms, up) and long Hann window (128 ms, bottom)

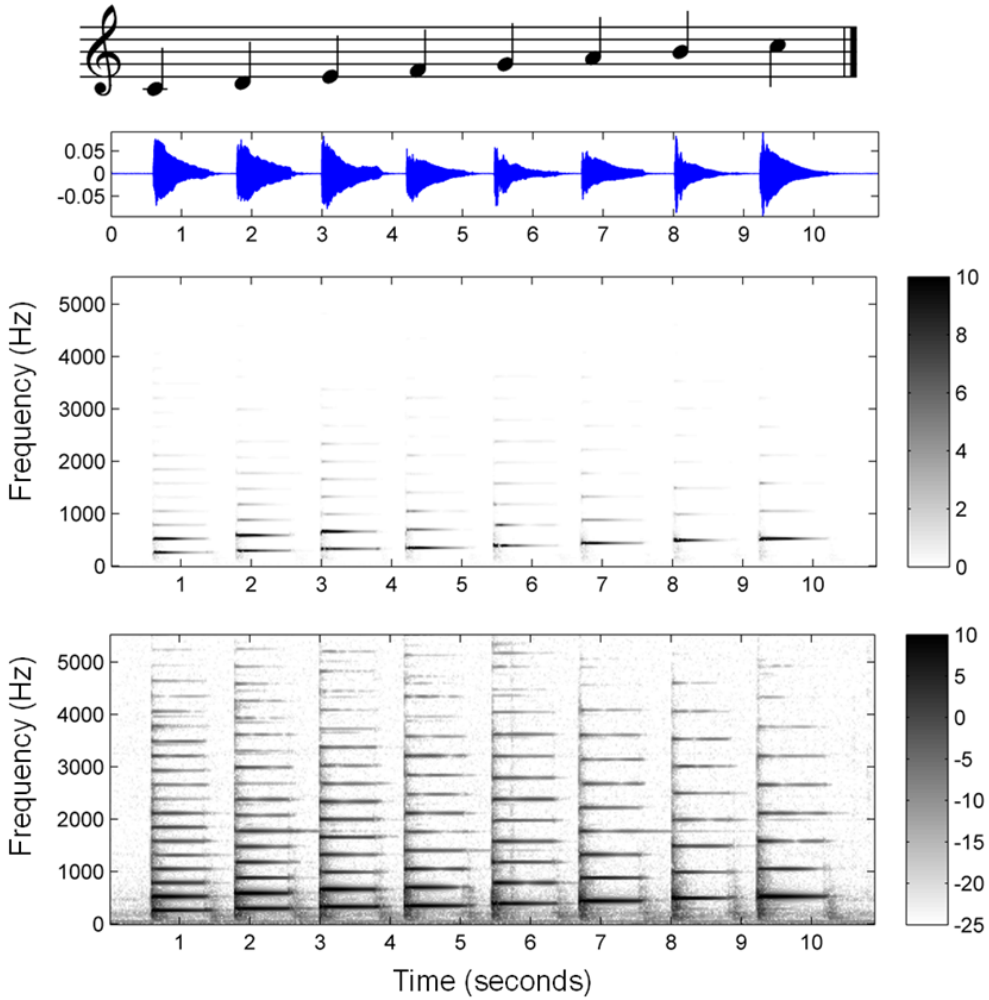


Window size is a tradeoff between resolution of time and frequency

From: M. Mueller, *Fundamentals of Music Processing,* Chapter 2, Springer 2015

Illustration (3): scaling

Spectrogram and dB-scaled spectrogram of a scale played with the piano

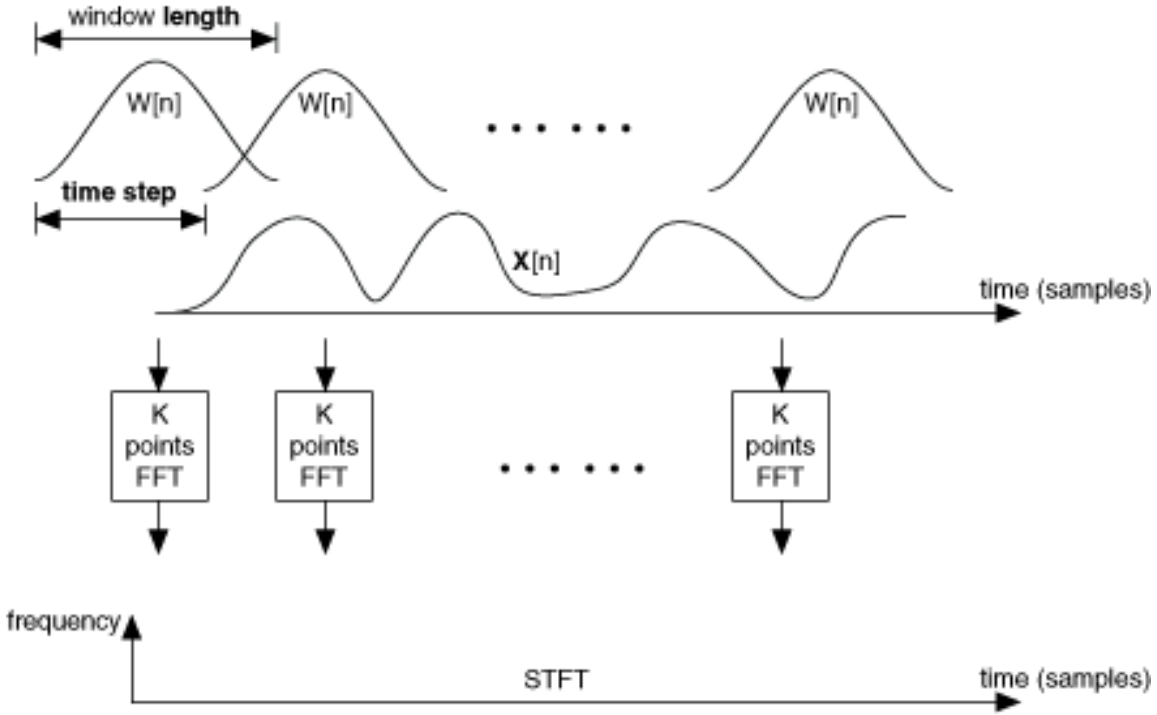


From: M. Mueller, *Fundamentals of Music Processing,* Chapter 2, Springer 2015

Implementation of STFT (1)

Python librosa

- Slice the signal into *frames* of segments
- Multiply the short segments by a window function
- Do FFT for each segment!



From: `http://zone.ni.com/reference/en-XX/help/371361J-01/lvanls/stft_spectrogram_core/`

Implementation of STFT (2)

- Discrete STFT:

$$X(n,k) = \sum_{m=0}^{N-1} x(m+nH)h(m)e^{-\frac{j2\pi km}{N}} \tag{9}$$

- H: time step (hop size)
- The index k corresponds to the frequency $f(k) := \frac{kf_s}{N}$
- The index n corresponds to the time $t(n) := \frac{nH}{f_s}$
- Example: for an audio signal sampled at $f_s = 44.1$ kHz, use window size $N = 4096$ samples and hop size $H = 1024$ samples. Could you specify the time-frequency grid in the STFT representation?

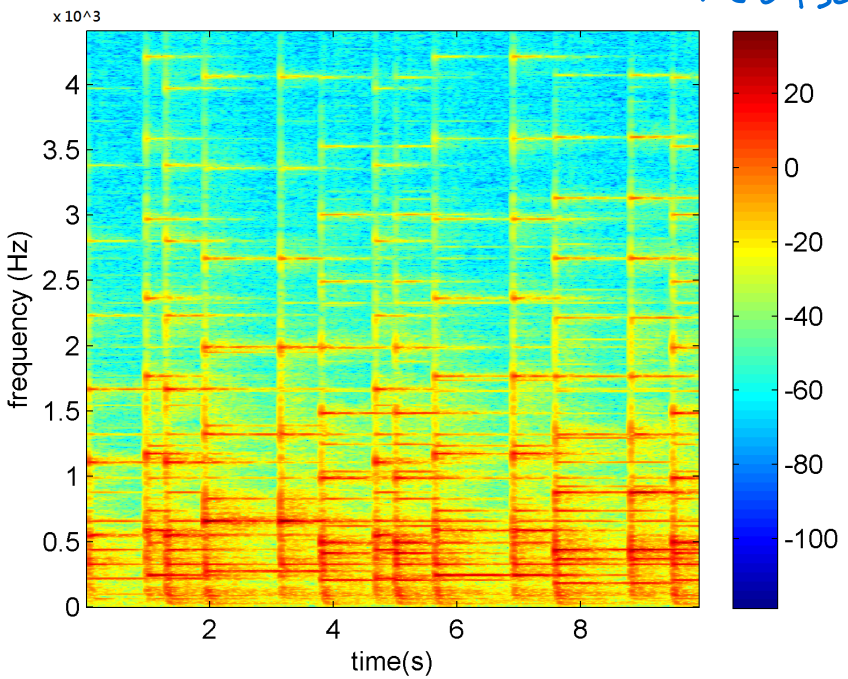
Example (1): piano solo

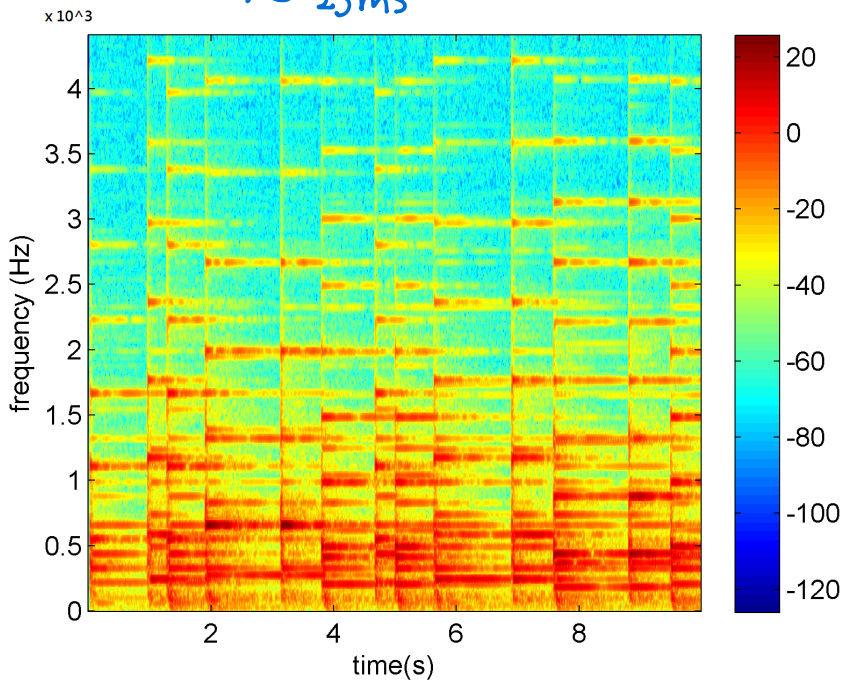
window size (#sample)

$f_s = 44.1$ kHz, $H = 441$, $N = 4096$ (left) and $N = 1024$ (right)

$N = 4096$: ~0.1 sec

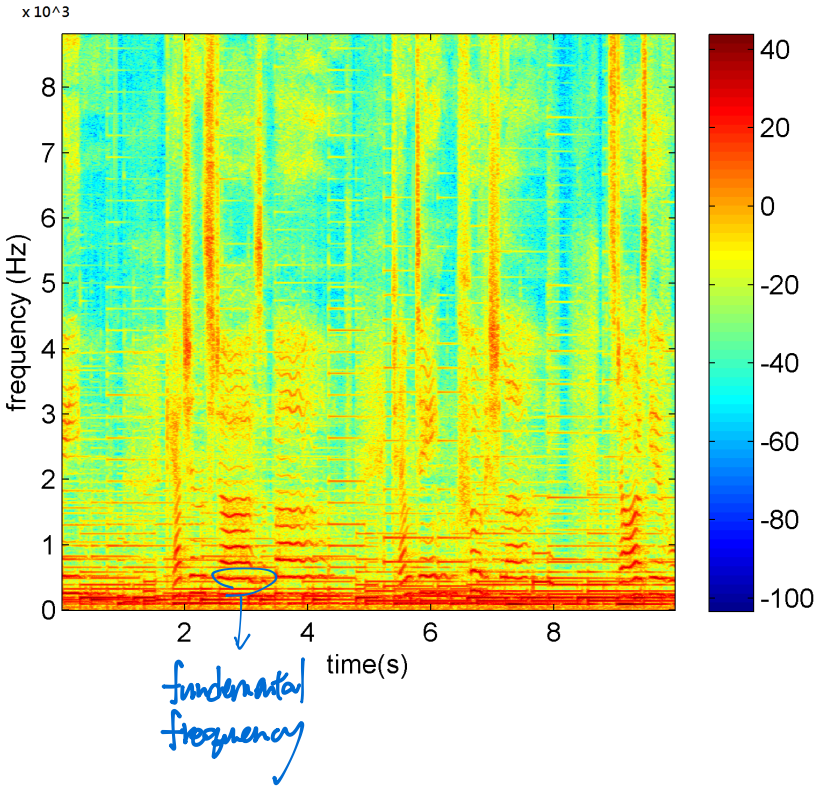
$N = 1024$: ~23 ms

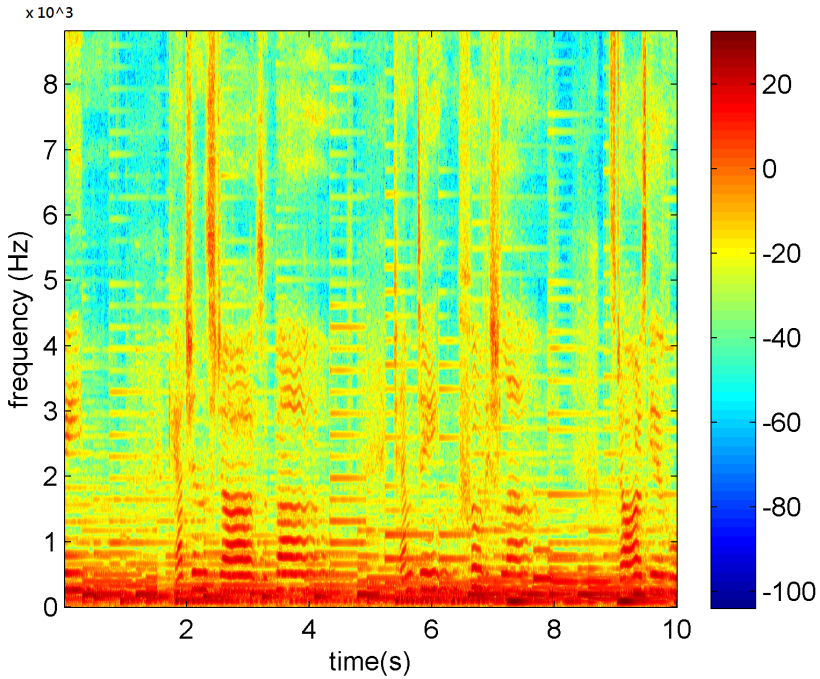




Example (2): voice

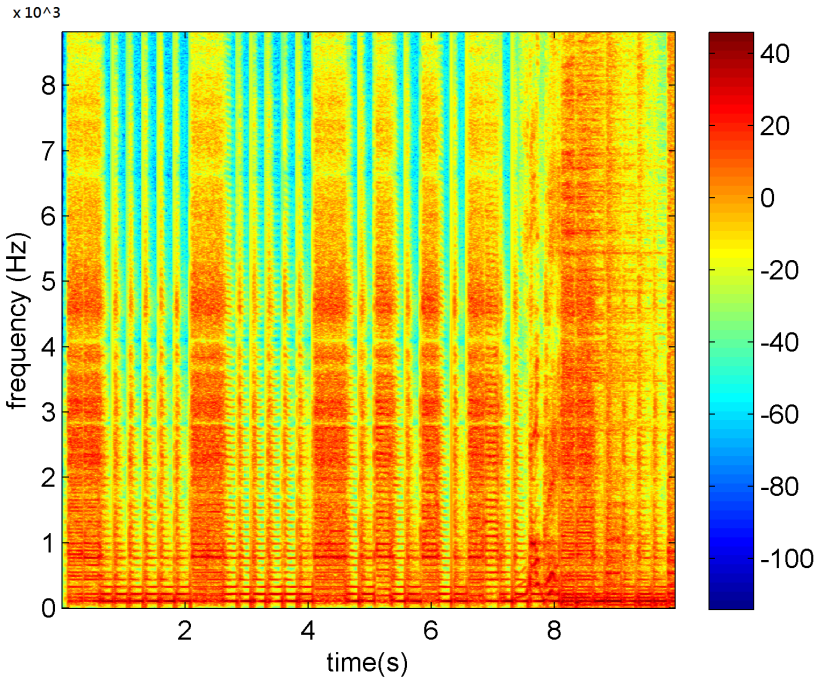
$f_s = 44.1$ kHz, $H = 441$, $N = 4096$ (left) and $N = 1024$ (right)

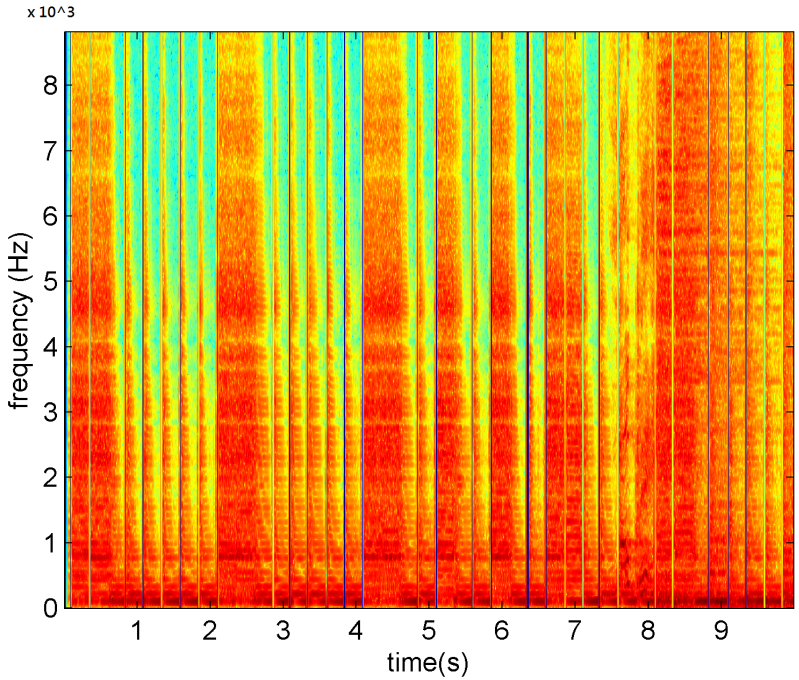




Example (3): rock

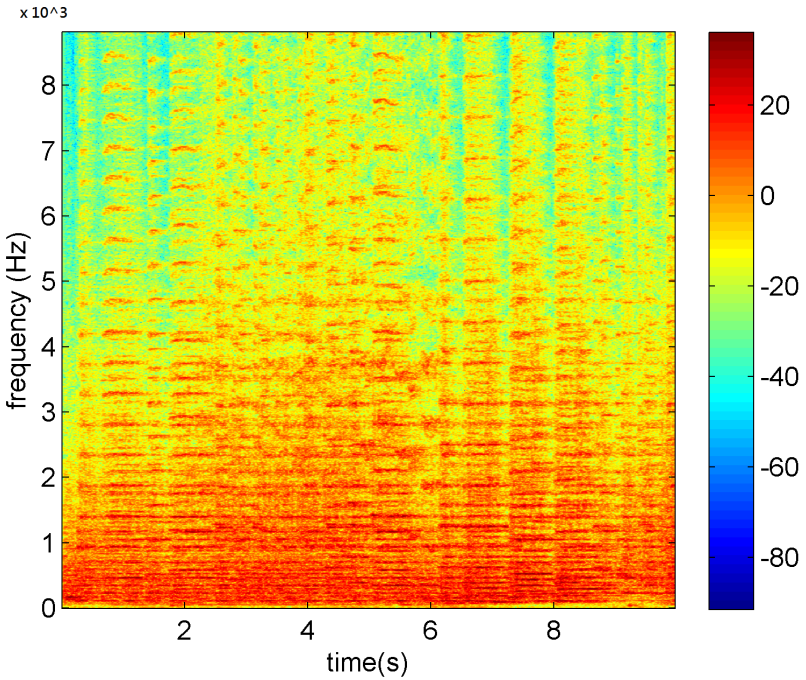
$f_s = 44.1$ kHz, $H = 441$, $N = 4096$ (left) and $N = 1024$ (right)

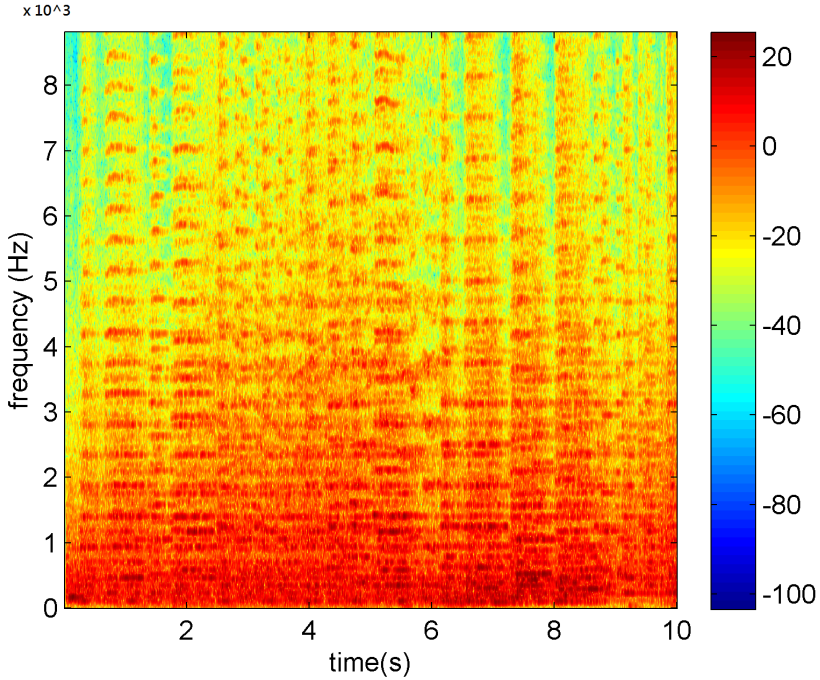




Example (4): symphony

$f_s = 44.1$ kHz, $H = 441$, $N = 4096$ (left) and $N = 1024$ (right)





Discussion

海森堡不確定性原則，outcome 和測量 (window size) 有關

- **Gabor-Heisenberg uncertainty principle**: it is impossible to get a distribution perfectly localized in both the time and the frequency domains.
- Δt: variance in the time domain
- Δf: variance in the frequency domain

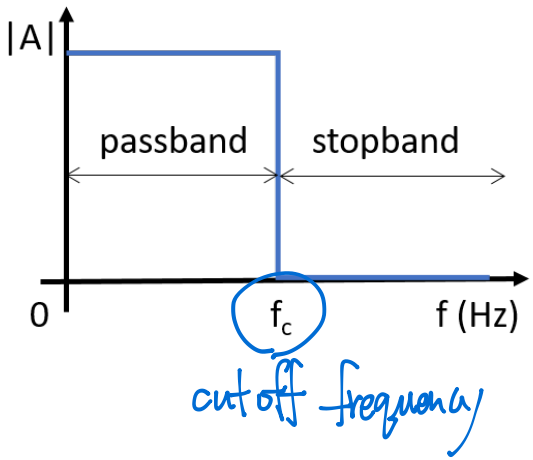
$$\Delta t \Delta f \geq C, \quad C > 0 \tag{10}$$

- Increase window size: $\Delta t \uparrow$, $\Delta f \downarrow$
- Decrease window size: $\Delta t \downarrow$, $\Delta f \uparrow$

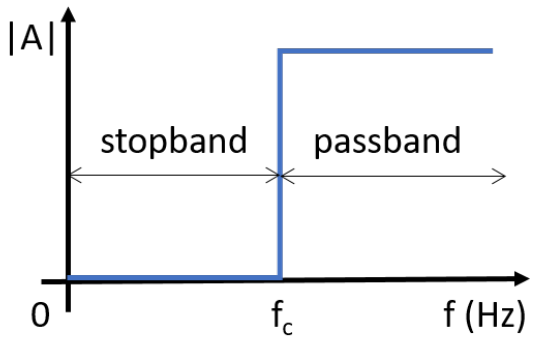
Basics of filters

- A filter performs mathematical operation that reduces or enhances certain elements of a signal in a spectrum.
- Basics: *linear*, *time-invariant* filters
- Low-pass filters (LPF), high-pass filters (HPF), band-pass filters (BPF)
- Ideal filters:

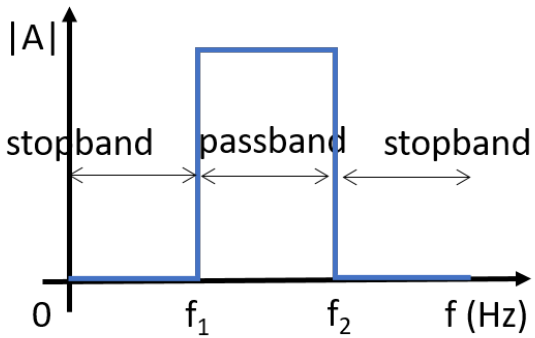
Low-pass filter (LPF)



High-pass filter (HPF)



Band-pass filter (BPF)



Two ways implementing a filter

Multiplication in the frequency domain

- The spectrum of an input signal $X[k]$ FFT
- The *frequency response* of the filter $H[k]$
- The spectrum of the output signal $Y[k] := X[k]H[k]$ iFFT

Convolution in the time domain 直接在 time signal 上對 filter 做 convolution

- An input signal $x[n]$
- The *impulse response* of the filter $h[n]$
- The spectrum of the output signal $y[n] := x[n] * h[n]$
- *Circular convolution*: x and h are of finite length N. $\hat{x}$ and $\hat{h}$ are *periodic extensions* of x and h. The circular convolution of x and h is

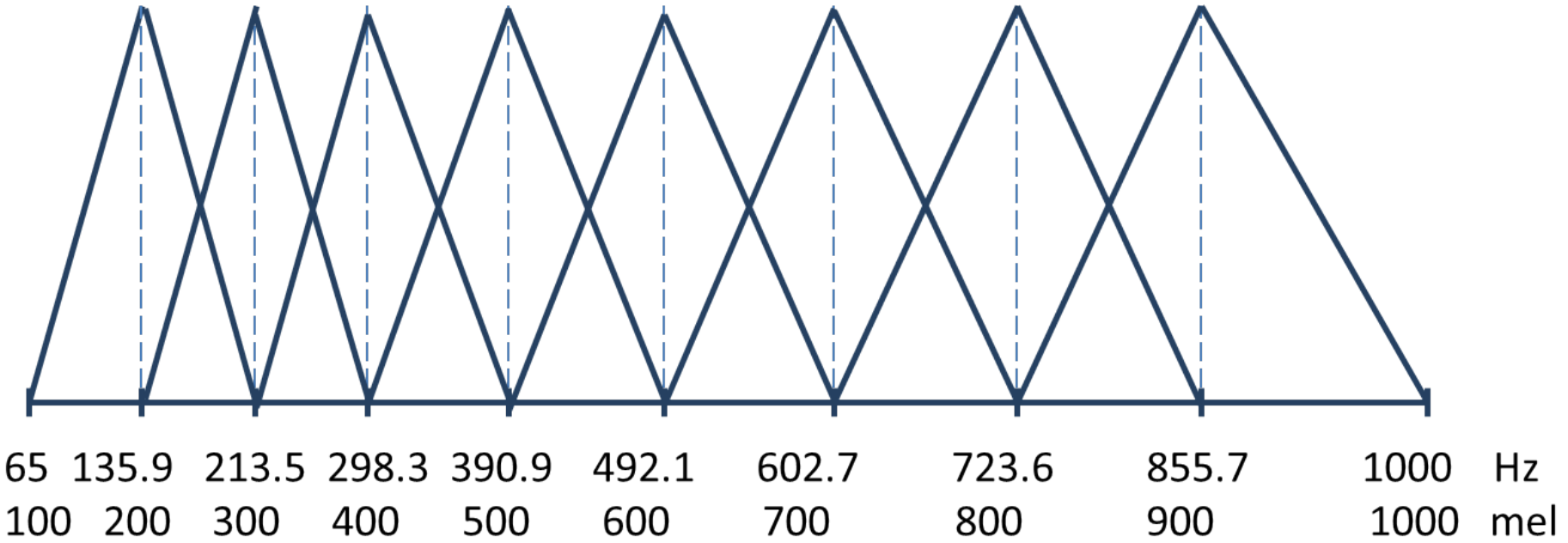
moving-average filter (time) = lowpass filter (frequency)

first-order-difference filter (time) = highpass filter (frequency)

$$x[n] * h[n] = \sum_{q=0}^{N-1} x[q]y[n-q] \tag{11}$$

Filterbanks

- A series of filters operating on different *bands* (e.g., perceptual scales)
- Multiplication in the frequency domain
- Examples: mel-frequency filterbank, log-frequency filterbank (a.k.a. constant-Q filterbank)
- Note: the frequency of a pitch with MIDI number p is $f = 440 \times 2^{\frac{p-69}{12}}$



Some issues about filters

Multiplication in the frequency domain

- Pros: straightforward in processing the full spectrum
- Cons: slow (需要FFT和iFFT)
- Usually used in *frame-wise* processing

Convolution in the time domain

- Pros: fast
- Cons: inflexible in processing the full spectrum (每個頻帶都要設計一個filter)
- Usually used the processing of specific bands, e.g., use an LPF to preserve DC and slow-varying components

When using a filter to process a signal, notice that:

- There is no *ideal* frequency selectivity (FFT和iFFT有uncertainty)
- There is always a *delay* between the input and the output
- Usually this delay is frequency-dependent (except for the case of *linear-phase* filters)

(以moving-average為例，n-k, n, n+k → n的output一定要等k)

Examples of some basic digital filters

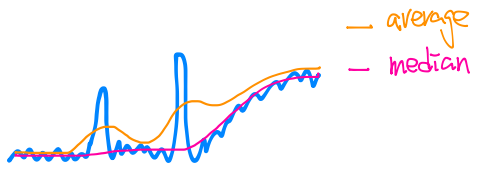
- Input signal: $x[n]$; output (filtered) signal: $y[n]$; n: the time index
- A K-term **moving-average filter**:

$$y[n] = \frac{1}{K}\left(x[n] + x[n-1] + \cdots + x[n-K+1]\right) \tag{12}$$

- How does $y[n]$ sounds? Is this filter a LPF or a HPF?
- A "first-order-difference" filter:

$$y[n] = x[n] - x[n-1] \tag{13}$$

- In this filter a LPF or a HPF?
- A K-term **median filter**: 濾雜波 (非線性, hard for NN & backpropogation)



$$y[n] = \text{median}\left(x[n], x[n-1], \cdots, x[n-K+1]\right) \tag{14}$$

- What is the characteristics of the median filter?

Filters and latency

- Another way to represent a moving-average filter:

$$y[n] = \frac{1}{2K+1} \sum_{k=-K}^{K} x[n+k] \tag{15}$$

- The output at time n is produced by the input from $n-K$ to $n+K$
- In other words, the output at time n is known later than when the input at $n+K$ is known
- The *latency* (e.g., delay time) of this filter is K

General form of digital filters

Finite-impulse-response (FIR) filters

$$y[n] := \sum_{k=1}^{K} h[k]x[n-k] \tag{16}$$

train/design this

- Convolution of $x[n]$ and $h[n]$

Infinite-impulse-response (IIR) filters

$$y[n] := \sum_{k=1}^{K} h[k]y[n-k] \tag{17}$$

Auto regression model 自迴歸

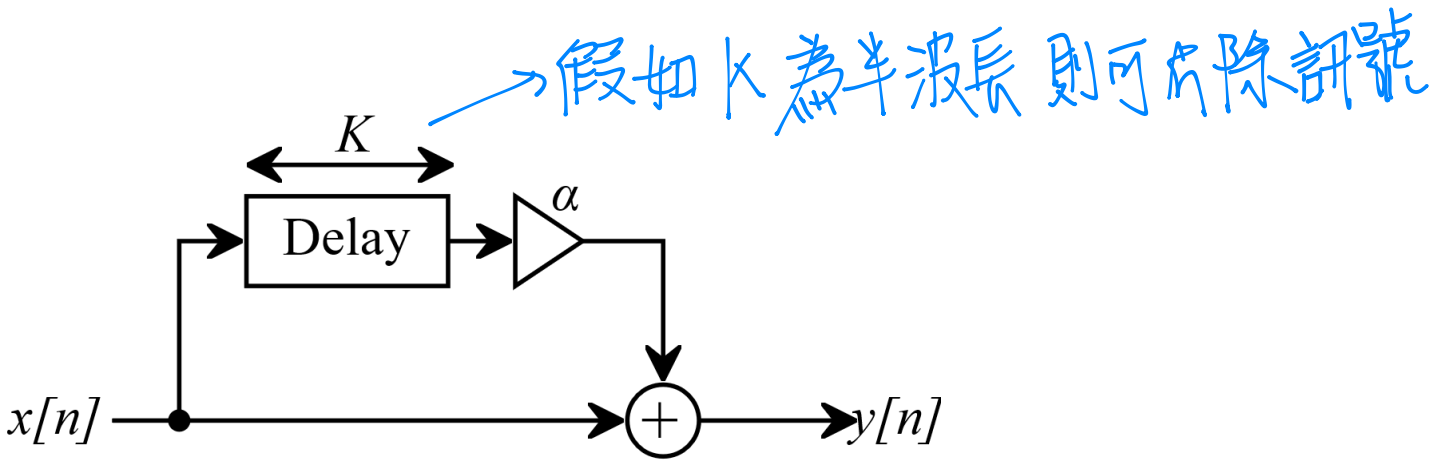
- Autoregressive model
- Generative model
- What is the latency of a FIR or an IIR filter?

Comb filters

- A comb filter adds a delayed version of a signal to itself
- Constructive and destructive interference at certain frequencies

$$y[n] = x[n] + \alpha x[n - K + 1] \tag{18}$$

- Example: if $\alpha = -1$, total destructive interference at frequency f_s/K, $2f_s/K$, $\cdots$



Audio effects based on comb filters

- How about adding several different delayed versions of a signal to itself? 把訊號delay一些再加上去，常用來使音色飽滿
- Flanging, phasing, chorus effects, ...
- ..."flanging" can be credited in part to John Lennon, According to Mark Lewisohn's fine book "THE BEATLES RECORDING SESSIONS", Lennon just hated singing the tedious "multi-tracks" so prevalent in Beatles recordings prior to 1966. In response to this, on April 6, 1966, Ken Townsend, Abbey Road Studios' Tech Engineer had a brilliant idea. During the mixing process, the output of the vocal track was recorded on another open reel machine and then combined with the original track to produce a "multed" sound. (`http://www.mikekonopka.com/page21.html`)
- More details:`https://ccrma.stanford.edu/~jos/pasp/Time_Varying_Delay_Effects.html`

Advanced topics in digital filters

Read them on-line!

- Julius O Smith III, "Mathematics of the Discrete Fourier Transform (DFT)," `https://www.dsprelated.com/freebooks/mdft/`
- Julius O Smith III, "Introduction to Digital Filters," `https://www.dsprelated.com/freebooks/filters/`

Constant-Q transform

- Balance the human perceptual scaling and Gabor-Heisenberg uncertainty
- Use long windows for low-frequency bands, short windows for high-frequency bands, such that $\Delta t \Delta f$ is a constant throughout the time-frequency plane
- Reference: Schörkhuber *et al.*, "A Matlab Toolbox for Efficient Perfect Reconstruction Time-Frequency Transforms with Log-Frequency Resolution" `https://www.cs.tut.fi/sgn/arg/CQT/schoerkhuber-aes-2014.pdf`

$$X[k,n] = \sum_{m=0}^{N} x[m] a_k^*[n-m] \tag{19}$$

$$a_k[m] = w_k[m] e^{\frac{i2\pi m f_k}{f_s}}, \quad m \in \mathbb{Z} \tag{20}$$

$$f_k = f_0 2^{\frac{k}{b}}, \quad k = 0, 1, \cdots, K-1 \tag{21}$$