the Fourier transform and applications

Michael Klingbeil
michael@klingbeil.com

CMC2005

analyzing complex waveforms

- Jean-Baptise Joseph Fourier ~1822
- Any periodic complex waveform can be represented as a sum of *harmonically* related sinusoids each with a particular amplitude (and phase).
- The Fourier transform takes a waveform and computes the exact amplitudes of the sinusoids that comprise the waveform.
- The transform is (theoretically) lossless.

definition of the Fourier transform

- **forward transform**

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-2\pi i f t} dt$$

- **inverse transform**

$$x(t) = \int_{-\infty}^{+\infty} X(f) e^{2\pi i f t} df$$

what does this mean?

forward transform

- **beginning with the forward transform:**

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-2\pi i f t} dt$$

- $x(t)$ **is our time domain audio signal**
- **we are multiplying it by** $e^{-2\pi ift}$ **(to be explained)**
- 2π **and** i **are constants**
- f **is a value which corresponds to frequency**
- **We are integrating (adding up) over all time values (from $-\infty$ to $+\infty$) at some specifically chosen frequency value** f

complex exponentials

- What about $e^{2\pi i f t}$?
- e is the base of the natural logarithm
  - $e \approx 2.718281828459045\ldots$
- i is the complex number
  - $i = \sqrt{-1}$

- Euler's identity

$$e^{i\theta} = \cos\theta + i\sin\theta$$

- follows from the series expansions for e^x $\sin$ and $\cos$

computing fourier transforms

- **the transforms are defined for…**
  - continuous (non-sampled) functions of time $x(t)$
  - signals of infinite length (from $-\infty$ to $+\infty$)

- **computers work with…**
  - discrete sampled waveforms
  - finite length signals

Discrete Fourier Transform

Definition:

$$\begin{aligned} DFT(k) &= \frac{1}{N}\sum_{n=0}^{N} x(n)e^{-2\pi ik\frac{n}{N}} \quad k = 0, 1, 2, \ldots, N-1 \\ &= \frac{1}{N}\sum_{n=0}^{N} x(n)[\cos 2\pi k\frac{n}{N} - i\sin 2\pi k\frac{n}{N}] \end{aligned}$$

Discrete Fourier Transform

- **takes a sampled signal *x(n)* of *N* samples**
- **for each frequency value *k* of *N* discrete frequencies:**
  - **pointwise multiply waveform samples by a cosine wave at frequency *k* and adds up the results**
  - **pointwise multiply waveform samples by a sine wave at frequency *k* and adds up the results**

$$\begin{aligned} DFT(k) &= \frac{1}{N}\sum_{n=0}^{N} x(n)[\cos 2\pi k\frac{n}{N} - i\sin 2\pi k\frac{n}{N}] \\ &= \frac{1}{N}\sum_{n=0}^{N} x(n)\cos 2\pi k\frac{n}{N} - i\frac{1}{N}\sum_{n=0}^{N} x(n)\sin 2\pi k\frac{n}{N} \\ &= a_k + ib_k \end{aligned}$$

real and imaginary components

- The DFT gives us two values per frequency:
  - real
  - imaginary

$$DFT(k) = a_k + ib_k \qquad k = 0, 1, 2, \ldots, N-1$$

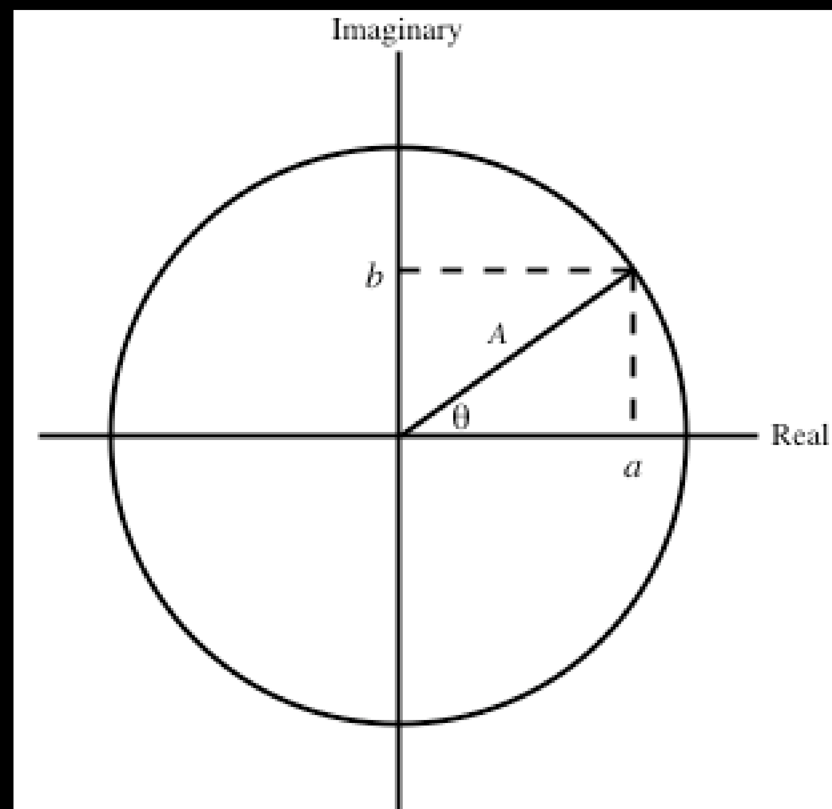
- the DFT outputs *complex numbers*
- all cosine sums are real valued
- all sine sums are imaginary
  - (multiplied by the imaginary number i)

Intuitive Interpretation

- start with a waveform to be analyzed
- choose a set of reference sine and cosine waves at discrete frequencies
- “compare” the waveform to each reference sine and cosine wave by multiplying them point by point and adding up the values
- portions of the waveform that are like the reference sinusoid will result in larger sums
- the reference sinusoid will “resonate” with waveform components that are close in frequency and phase

interpreting complex values

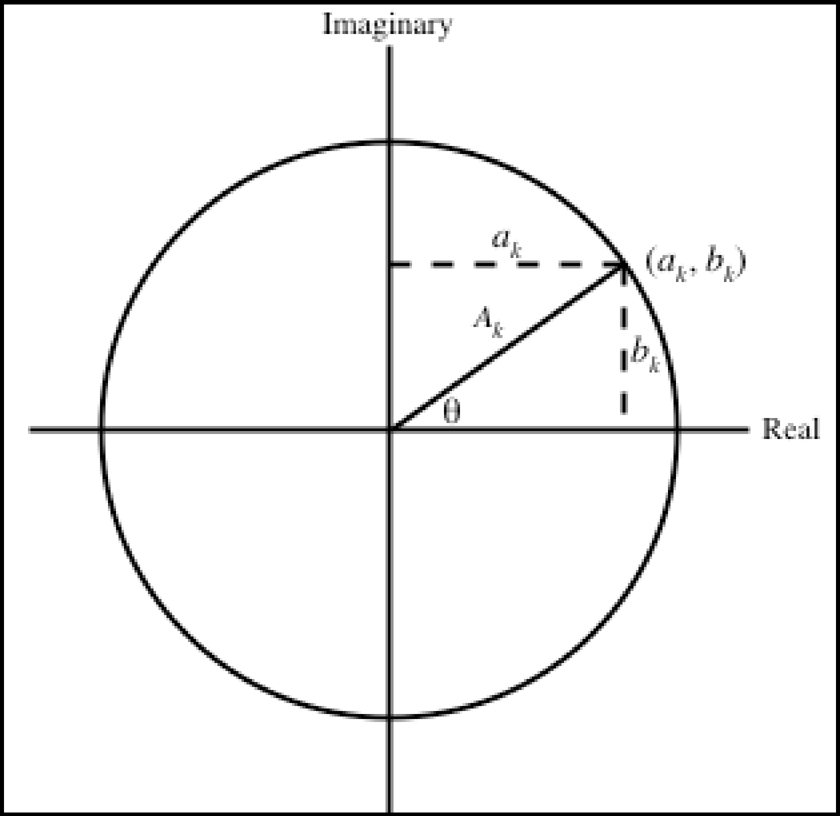
- both real and imaginary components represent the same frequency $DFT(k) = a_k + ib_k$
- but relative to different phases $cos(\theta) = sin(\theta + \pi/2)$
- cosine and sine components represent the instantaneous position of a *complex sinusoid*
- this position can be graphed on the complex plane



magnitude…

- the *magnitude* (amplitude) of a complex sinusoid is the distance from the origin to the complex point (a_k, b_k)

$$A_k = \sqrt{a_k^2 + b_k^2}$$



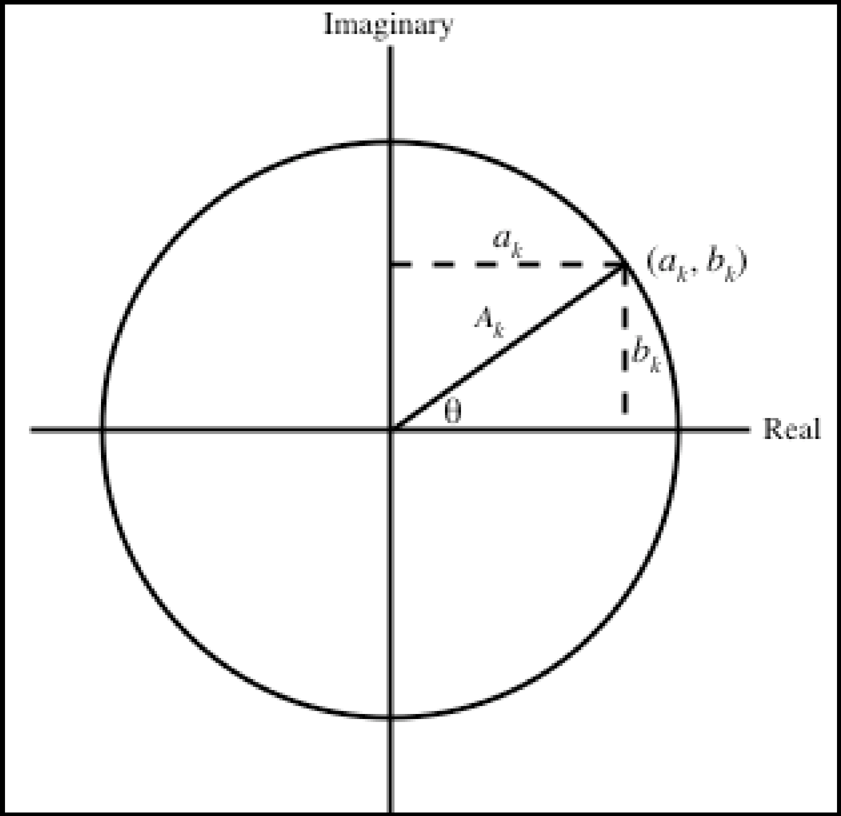
and phase

- the phase is the angle of rotation (θ) of the complex point
- recall the definition of the tangent function:

$$\tan\theta = \frac{b_k}{a_k}$$

- so…

$$\theta = \tan^{-1}\frac{b_k}{a_k}$$



- where $\tan^{-1}$ is the inverse tangent function

DFT summary

- **Input: sampled signal $x(n)$ of length N**
- **Output: N pairs of values**

$$a_k, b_k \text{ with } k = 0 \ldots N-1$$

- **there are k periods of the sinusoid in the space of N samples**
  - **i.e. for $k = 2$, there are 2 cycles per N samples**
  - **for $k = 13$, there are 13 cycles per N samples, and so on**

- **each value of k corresponds to a *frequency bin***
- **the spectrum is discretely sampled at each frequency bin**

Fast Fourier Transform (FFT)

- DFT in its direct form is slow to compute
- FFT is an optimized DFT where N is restricted to powers of 2 ($N = 2^p$ for some positive integer p)

- typical values of N for audio work at a sampling rate of 44100
  - 512
  - 1024
  - 2048
  - 4096
  - 8192

FFT summary

- Input: sampled signal $x(n)$ of length N where $N = 2^p$
- Output: N pairs of values a_k, b_k for each frequency bin k
- Magnitude at frequency bin k

$$A_k = \sqrt{a_k^2 + b_k^2}$$

- Phase at frequency bin k

$$\theta_k = \tan^{-1} \frac{b_k}{a_k}$$

Example

- for sampling rate 44100 and FFT size N = 1024
- for frequency bin k there are k cycles per 1024 samples
- the frequency in hertz at bin k is given by

$$\mathrm{Hz}_k = k\frac{44100}{1024}$$

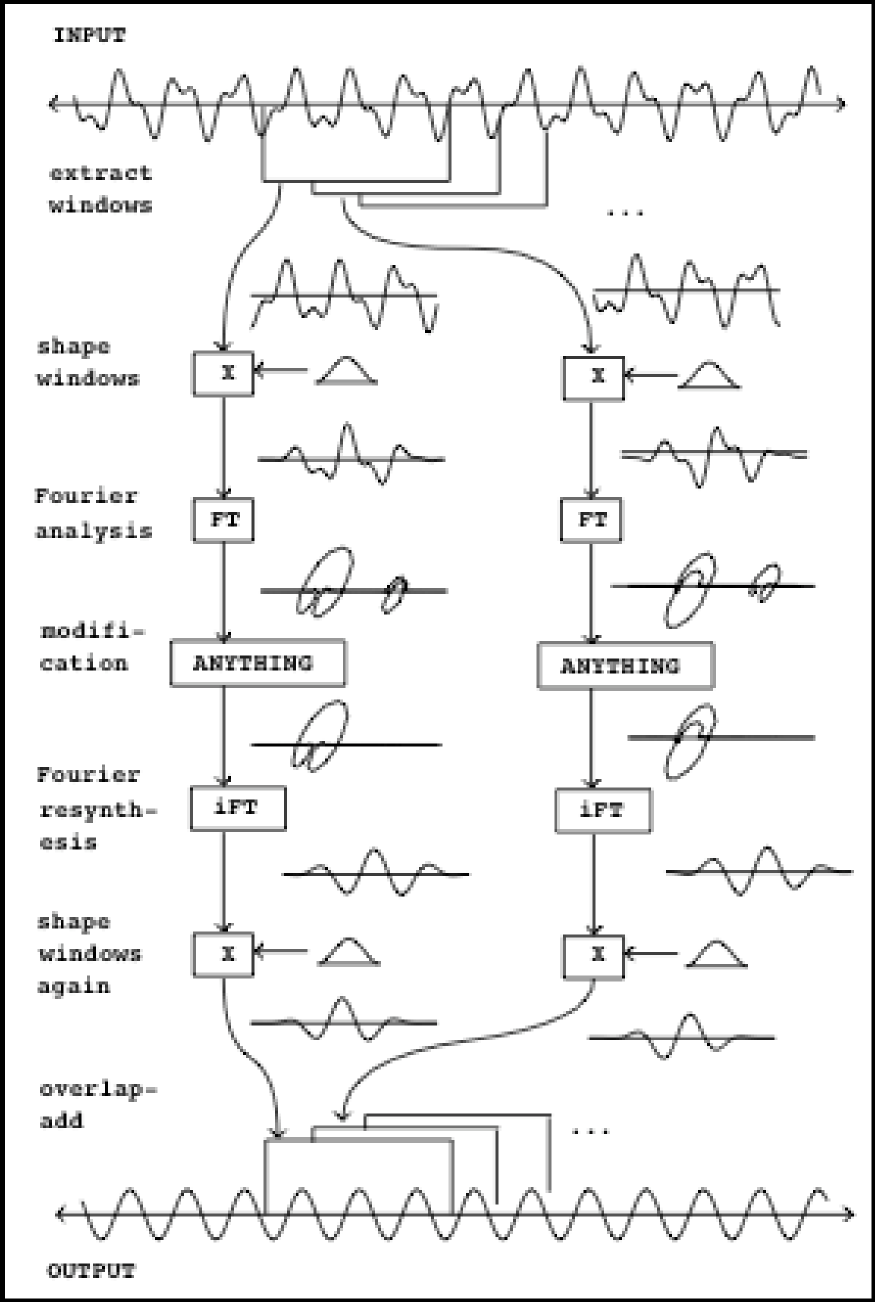
- the spectrum is sampled at intervals of 43.06 Hz
- Example: at bin 2 we have a = 0.345, b = –0.213
- The magnitude and phase corresponding to a cosine wave at 86.133 Hz are

$$\mathrm{Magnitude} = \sqrt{0.345^2 + -0.213^2} = 0.4054553$$

$$\mathrm{Phase} = \tan^{-1}\frac{-0.213}{0.345}$$

real spectrum symmetry

overlap-add analysis/resynthesis



from Miller Puckette, *Theory and Techniques of Electronic Music* p. 268

overlapping analysis

- each input signal block is smoothed by a window function $w(n)$
- this reduces the discontinuity at the block boundary at the expense of some frequency resolution
- each successive windowed block is overlapped in time (overlap factor)
- each analyzed windowed block is called a *frame*
- the amount of time between each frame is called the hop size H
- example: $N = 2048$ samples, overlap 4x
  $H = 512$ samples

DFT applications: the phase vocoder

- each bin of the FFT samples the frequency spectrum on a relatively coarse grid
  - (43.06 Hz in the case of SR=44100, FFT size=1024)

- Idea: use the time varying phase values to improve the frequency estimates
- compare the phase θ_k in bin k at frame n to to the corresponding phase at frame $n{-}1$
- the change in phase has a correspondence to a change in frequency

phase deviation

- change in phase per unit time is a frequency measurement

phase deviation: examples

- consider analyzing a cosine of 258.39844 Hz
- SR = 44100 samples/sec.
- FFT size N = 1024, overlap 4x, hop size H = 256
- the bin spacing for this FFT is 43.06 Hz
- the cosine is aligned at the center of bin 6
- let the phase in bin 6 at frame n be 0.0
- what is the expected phase in bin 6 at frame n+1?

phase deviation: examples

- convert 258.39844 Hz to radians per hop

$$\frac{258.39844 \text{ cycles}}{1 \text{ sec.}} \cdot \frac{2\pi \text{ radians}}{1 \text{ cycles}} \cdot \frac{1 \text{ sec.}}{44100 \text{ samples}} \cdot \frac{256 \text{ samples}}{1 \text{ hop}} = \frac{3\pi \text{ radians}}{1 \text{ hop}}$$

- so phase will increase a distance of 3π radians every analysis hop

phase deviation: examples

- now consider the same situation but with a cosine at 280 Hz
- what will happen to the phase in bin 6?

$$\frac{280 \text{ cycles}}{1 \text{ sec.}} \cdot \frac{2\pi \text{ radians}}{1 \text{ cycles}} \cdot \frac{1 \text{ sec.}}{44100 \text{ samples}} \cdot \frac{256 \text{ samples}}{1 \text{ hop}} = \frac{3.2507937\pi \text{ radians}}{1 \text{ hop}}$$

- the sinusoid is at a higher frequency so its phase is increasing faster
- the phase is running 0.25079π radians per hop faster

phase vocoder frequency estimation

- at frame n subtract the expected phase of a sinusoid perfectly centered on the analysis bin from the actual phase
- expected phase is computed based on the previous phase in frame n-1
- the difference is the *phase deviation*
- in our example the phase deviation was 0.25079π
- converting to cycles per second:

$$\frac{0.2507937\pi \text{ radians}}{1 \text{ hop}} \cdot \frac{1 \text{ cyc.}}{2\pi \text{ rad.}} \cdot \frac{1 \text{ hop}}{256 \text{ samp.}} \cdot \frac{44100 \text{ samp.}}{1 \text{ sec.}} = 21.6015663 \text{ Hz}$$

phase vocoder frequency estimation

- a frequency deviation of 21.6015663 Hz is added to the center frequency of bin 6
- 258.39844 Hz + 21.6015663 Hz = 280 Hz
- summary:
- computing phase deviations allow us to find the actual frequency of the analyzed sinusoids

phase vocoder applications:

- resynthesize at a different rate to time compress/exapnd without changing the pitch
- adjust the frequencies to transpose without changing the duration
- resynthesis can be done with oscillators (slow)
- resynthesis can be done with inverse FFTs (fast)
- the subjective quality of the resynthesis is largely dependant on how “well” the time varying phases are managed and updated