Discrete signals

Alejandro Ribeiro
Dept. of Electrical and Systems Engineering
University of Pennsylvania
aribeiro@seas.upenn.edu
http://www.seas.upenn.edu/~aribeiro/

February 12, 2015

Discrete signals

Discrete signals

- Discrete and finite time index $n = 0, 1, \ldots, N-1 = [0, N-1]$.
- Discrete signal x is a function mapping $[0, N-1]$ to real value $x(n)$

$$x : [0, N-1] \to \mathbb{R}$$

- The values that the signal takes at time index n is $x(n)$
- Sometimes it will make sense to talk about complex signals

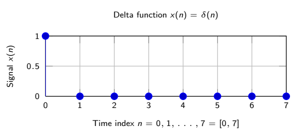
$$x : [0, N-1] \to \mathbb{C}$$

- The values $x(t) = x_R(t) + j\,x_I(t)$ are complex numbers
- Space of signals = space of N-dimensional vectors $\mathbb{R}^N$ or $\mathbb{C}^N$

Deltas (impulses, spikes)

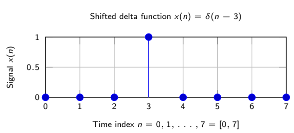
- The discrete delta function $\delta(n)$ is a spike at (initial) time $n = 0$

$$\delta(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{else} \end{cases}$$



- The shifted delta function $\delta(n - n_0)$ has a spike at time $n = n_0$

$$\delta(n - n_0) = \begin{cases} 1 & \text{if } n = n_0 \\ 0 & \text{else} \end{cases}$$

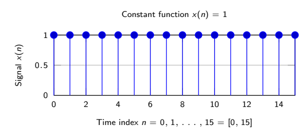


- This is not a new definition, just a time shift

Constants and square pulses

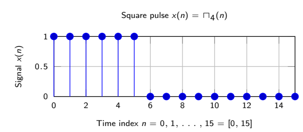
- A constant function $x(n)$ has the same value c for all n

$$x(n) = c, \quad \text{for all } n$$



- A square pulse of width M, $\sqcap_M(n)$, equals one for the first M values

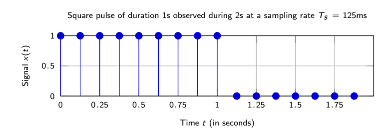
$$\sqcap_M(n) = \begin{cases} 1 & \text{if } 0 \le n < M \\ 0 & \text{if } M \le n \end{cases}$$



- Can consider shifted pulses $\sqcap_M(n - n_0)$, with $n_0 < N - M$

Units: Sampling time and signal duration

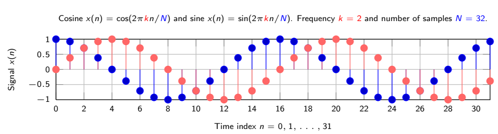
- Sampling time T_s $\Rightarrow$ Time elapsed between indexes n and $n+1$
  - $\Rightarrow$ Sampling frequency $f_s := 1/T_s$
- Time index n represents actual time $t = nT_s$



- Signal duration $T = NT_s$ $\Rightarrow$ Time length of signal
  - $\Rightarrow$ The last sample is "held" during T_s time units

Discrete cosines and sines

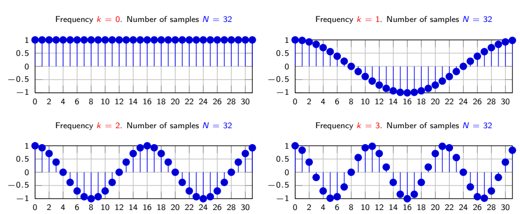
- For a signal of duration N define (assume N is even):
  - $\Rightarrow$ Discrete cosine of discrete frequency k $\Rightarrow$ $x(n) = \cos(2\pi kn/N)$
  - $\Rightarrow$ Discrete sine of discrete frequency k $\Rightarrow$ $x(n) = \sin(2\pi kn/N)$



- Frequency k is discrete. I.e., $k = 0, 1, 2, \ldots$
  - $\Rightarrow$ Have an integer number of complete oscillations

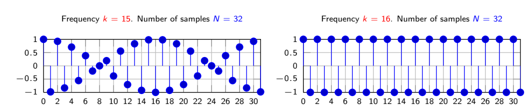
Cosines of different frequencies (1 of 2)

- Discrete frequency $k = 0$ is a constant
- Discrete frequency $k = 1$ is a complete oscillation
- Frequency $k = 2$ is two oscillations, for $k = 3$ three oscillations ...



Cosines of different frequencies (2 of 2)

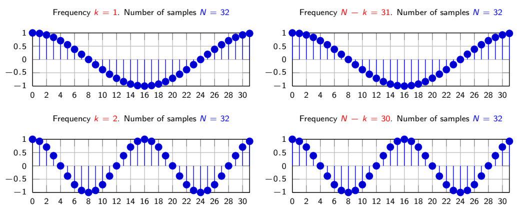
- Frequency k represents k complete oscillations
- Although for large k the oscillations may be difficult to see



- Do note that we can't have more than $N/2$ oscillations
  - $\Rightarrow$ Indeed $1 \to -1 \to 1, \to -1, \ldots$
  - $\Rightarrow$ Frequency $N/2$ is the last one with physical meaning
- Larger frequencies replicate frequencies between $k = 0$ and $k = N/2$

Duplicated frequencies

- Frequencies k and $N-k$ represent the same cosine



- Actually, if $k + l = \dot{N}$, cosines of frequencies k and l are equivalent
- Not true for sines, but almost. The signals have opposite signs

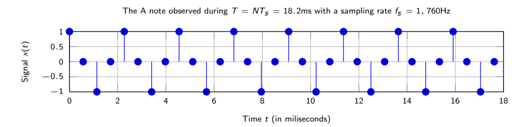
Discrete frequencies and actual frequencies

- What is the discrete frequency k of a cosine of frequency f_0?
- Depends on sampling time T_s, frequency $f_s = \frac{1}{T_s}$, duration $T = NT_s$
- Period of discrete cosine of frequency k is T/k (k oscillations)
- Thus, regular frequency of said cosine is $\Rightarrow f_0 = \frac{k}{T} = \frac{k}{NT_s} = \frac{k}{N} f_s$
- A cosine of frequency f_0 has discrete frequency $k = (f_0/f_s)N$
- Only frequencies up to $N/2 \leftrightarrow f_s/2$ have physical meaning
- Sampling frequency f_s $\Rightarrow$ Cosines up to frequency $f_0 = f_s/2$

Use of units example

- Generate $N = 32$ samples of an A note with sampling frequency $f_s = 1,760$Hz
- The frequency of an A note is $f_0 = 440$Hz. This entails a discrete frequency

$$k = \frac{f_0}{f_s} N = \frac{440\text{Hz}}{1,760\text{Hz}} 32 = 8$$



- Alternatively $\Rightarrow x(n) = \cos\left[2\pi k n/N\right] = \cos\left[2\pi (f_0/f_s) N n/N\right]$
- Which simplifies to $\Rightarrow x(n) = \cos\left[2\pi (f_0/f_s) n\right] = \cos\left[2\pi f_0 (nT_s)\right]$

Noninteger frequencies

- The frequency k need not be integer but it's not a discrete cosine
  - $\Rightarrow$ Sampled cosine $\Rightarrow x(n) = \cos(2\pi k n/N)$
  - $\Rightarrow$ Sampled sine $\Rightarrow x(n) = \sin(2\pi k n/N)$
- Discrete sine and cosine have complete oscillations
- Sampled sine and cosine may have incomplete oscillations
- Discrete sine and cosine are used to define Fourier transforms (later)

Inner products and energy

Inner product

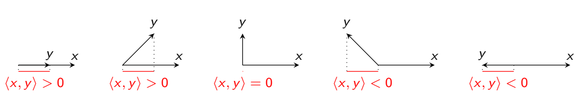
- Given two signals x and y define the inner product of x and y as

$$\langle x, y\rangle := \sum_{n=0}^{N-1} x(n) y^*(n)$$
$$= \sum_{n=0}^{N-1} x_R(n) y_R(n) + \sum_{n=0}^{N-1} x_I(n) y_I(n) + j \sum_{n=0}^{N-1} x_I(n) y_R(n) - j \sum_{n=0}^{N-1} x_R(n) y_I(n)$$

- Inner product between vectors x and y, just with different notation
- Inner product is a linear operations $\Rightarrow \langle x, y+z\rangle = \langle x, y\rangle + \langle x, z\rangle$
- Reversing order equals conjugation $\Rightarrow \langle y, x\rangle = \langle x, y\rangle^*$

Inner product interpretation

- Signal inner product has same intuition as vector inner product
  - $\Rightarrow$ Inner product $\langle x, y\rangle$ is the projection of y on x
  - $\Rightarrow$ The value of $\langle x, y\rangle$ is how much of y falls in x direction
- E.g., if $\langle x, y\rangle = 0$ the signals are orthogonal. They are "unrelated"



Norm and energy

- Following the algebra analogies, define the norm of signal x as

$$\|x\| := \left[\sum_{n=0}^{N-1} |x(n)|^2\right]^{1/2} = \left[\sum_{n=0}^{N-1} |x_R(n)|^2 + \sum_{n=0}^{N-1} |x_I(n)|^2\right]^{1/2}$$

- More important, define the energy of the signal as the norm squared

$$\|x\|^2 := \sum_{n=0}^{N-1} |x(n)|^2 = \sum_{n=0}^{N-1} |x_R(n)|^2 + \sum_{n=0}^{N-1} |x_I(n)|^2$$

- For complex numbers $x(n)x^*(n) = |x_R(n)|^2 + |x_I(n)|^2 = |x(n)|^2$
- Thus, we can write the energy as $\Rightarrow \|x\|^2 = \langle x, x\rangle$

Cauchy Schwarz inequality

- The largest an inner product can be is when the vectors are collinear

$$-\|x\|\,\|y\| \leq \langle x, y\rangle \leq \|x\|\,\|y\|$$

- Or in terms of energy $\Rightarrow \langle x, y\rangle^2 \leq \|x\|^2\,\|y\|^2$
- If you are the sort of person that prefers explicit expressions

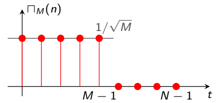
$$\sum_{n=0}^{N-1} x(n) y^*(n) \leq \left[\sum_{n=0}^{N-1} |x(n)|^2\right]\left[\sum_{n=0}^{N-1} |y(n)|^2\right]$$

- The equalities hold if and only if x and y are collinear

Example: Square pulse of unit energy

- The unit energy square pulse is the signal $\sqcap_M(n)$ that takes values

$$\sqcap_M(n) = \frac{1}{\sqrt{M}} \quad \text{if } 0 \le n < M \qquad \sqcap_M(n) = 0 \quad \text{if } M \le n$$



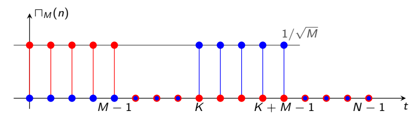
- To compute energy of the pulse we just evaluate the definition

$$\|\sqcap_M\|^2 := \sum_{n=0}^{N-1} |\sqcap_M(n)|^2 = \sum_{n=0}^{M-1} \left|(1/\sqrt{M})\right|^2 = \frac{M}{M} = 1$$

- Indeed, the unit energy square pulse has unit energy
- If the height of the pulse is 1 instead of $1/\sqrt{M}$, the energy is M.

Shifted pulses

- To shift a pulse we modify the argument $\Rightarrow \sqcap_M(n-K)$
  $\Rightarrow$ The pulse is now centered at K ($n = K$ is as $n = 0$ before)



- Inner product of two pulses with disjoint support ($K \ge M$)

$$\langle \sqcap_M(n), \sqcap_M(n-K)\rangle := \sum_{n=0}^{N-1} \sqcap_M(n)\, \sqcap_M(n-K) = 0$$

- The signals are orthogonal, and indeed, "unrelated" to each other

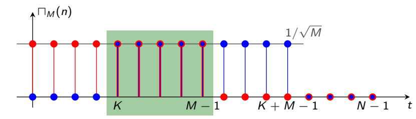
Overlapping shifted pulses

- Inner product of two pulses with overlapping support ($K < M$)

$$\langle \sqcap_M(n), \sqcap_M(n-K)\rangle := \sum_{n=0}^{N-1} \sqcap_M(n)\, \sqcap_M(n-K)$$

- The signals overlap between K and $M-1$. Thus

$$\langle \sqcap_M(n), \sqcap_M(n-K)\rangle = \sum_{n=K}^{M-1} \left(1/\sqrt{M}\right)\left(1/\sqrt{M}\right) = \frac{M-K}{M} = 1 - \frac{K}{M}$$



- Inner product is proportional to the relative overlap
  $\Rightarrow$ which is, indeed, how much the signals are "related" to each other

Discrete complex exponentials

Discrete Complex exponentials

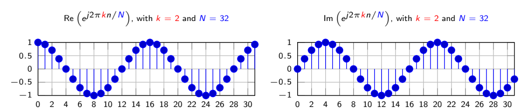
- Discrete complex exponential of discrete frequency k and duration N

$$e_{kN}(n) = \frac{1}{\sqrt{N}} e^{j2\pi kn/N} = \frac{1}{\sqrt{N}} \exp(j2\pi kn/N)$$

- The complex exponential is explicitly given by

$$e^{j2\pi kn/N} = \cos(2\pi kn/N) + j \sin(2\pi kn/N)$$

- Real part is a discrete cosine and imaginary part a discrete sine



Properties

[P1] For frequency $k = 0$, the exponential $e_{kN}(n) = e_{0N}(n)$ is a constant

$$e_{kN}(n) = \frac{1}{\sqrt{N}} = \frac{1}{\sqrt{N}}\,1$$

[P2] For frequency $k = N$, the exponential $e_{kN}(n) = e_{NN}(n)$ is a constant

$$e_{NN}(n) = \frac{e^{j2\pi Nn/N}}{\sqrt{N}} = \frac{(e^{j2\pi})^n}{\sqrt{N}} = \frac{(1)^n}{\sqrt{N}} = \frac{1}{\sqrt{N}}$$

- Actually, true for any frequency $k \in \dot{N}$ (multiple of N)

[P3] For $k = N/2$, the exponential $e_{kN}(n) = e_{N/2N}(n) = (-1)^n/\sqrt{N}$

$$e_{N/2N}(n) = \frac{e^{j2\pi(N/2)n/N}}{\sqrt{N}} = \frac{(e^{j\pi})^n}{\sqrt{N}} = \frac{(-1)^n}{\sqrt{N}}$$

- The fastest possible oscillation with N samples

That $e^{j2\pi} = 1$ follows from $e^{j\pi} = -1$, which follows from $e^{j\pi} + 1 = 0$. The latter relates the five most important constants in mathematics and proves god's existence.

Equivalent frequencies

Theorem
If $k - l = N$ the signals $e_{kN}(n)$ and $e_{lN}(n)$ coincide for all n, i.e.,

$$e_{kN}(n) = \frac{e^{j2\pi kn/N}}{\sqrt{N}} = \frac{e^{j2\pi ln/N}}{\sqrt{N}} = e_{lN}(n)$$

- Exponentials with frequencies k and l are equivalent if $k - l = N$

Canonical frequency sets

ARI

- Exponentials with frequencies that are N apart are equivalent

$-N$,	$-N+1$,	...,	-1
0,	1,	...,	$N-1$
N,	$N+1$,	...,	$2N-1$

- Suffice to look at N consecutive frequencies, e.g., $k = 0, 1, \ldots N-1$
- Another canonical choice is to make $k = 0$ the center frequency

$-N/2+1$,	...,	-1,	0,	...,	$N/2$
$N/2+1$,	...,	$N-1$,	N,	...,	$3N/2$

- With N even (as usual) use $N/2$ positive and $N/2-1$ negative
- From one canonical set to the other $\Rightarrow$ Chop and shift

Proof of equivalence

Proof.

- We prove by showing that $e_{kN}(n)/e_{lN}(n) = 1$. Indeed,

$$\frac{e_{kN}(n)}{e_{lN}(n)} = \frac{e^{j2\pi kn/N}}{e^{-j2\pi ln/N}} = e^{j2\pi(k-l)n/N}$$

- But since we have that $k - l = N$ the above simplifies to

$$\frac{e_{kN}(n)}{e_{lN}(n)} = e^{j2\pi Nn/N} = \left[e^{j2\pi}\right]^n = 1^n = 1 \qquad \square$$

Orthogonality

Theorem
Complex exponentials with nonequivalent frequencies are orthogonal. I.e.

$$\langle e_{kN}, e_{lN} \rangle = 0$$

when $k - l < N$. *E.g., when* $k = 0, \ldots N-1$, *or* $k = -N/2+1, \ldots, N/2$.

- Signals of canonical sets are "unrelated." Different rates of change
- Also note that the energy is $\|e_{kN}\|^2 = \langle e_{kN}, e_{kN} \rangle = 1$
- Exponentials with frequencies $k = 0, 1, \ldots, N-1$ are orthonormal

$$\langle e_{kN}, e_{lN} \rangle = \delta(l-k)$$

- They are an orthonormal basis of signal space with N samples

Proof of orthogonality

Proof.

- Use definitions of inner product and discrete complex exponential to write

$$\langle e_{kN}, e_{lN} \rangle = \sum_{n=0}^{N-1} e_{kN}(n) e_{lN}^*(n) = \sum_{n=0}^{N-1} \frac{e^{j2\pi kn/N}}{\sqrt{N}} \frac{e^{-j2\pi ln/N}}{\sqrt{N}}$$

- Regroup terms to write as geometric series

$$\langle e_{kN}, e_{lN} \rangle = \frac{1}{N} \sum_{n=0}^{N-1} e^{j2\pi(k-l)n/N} = \frac{1}{N} \sum_{n=0}^{N-1} \left[e^{j2\pi(k-l)/N} \right]^n$$

- Geometric series with basis a sums to $\sum_{n=0}^{N-1} a^n = (1-a^N)/(1-a)$. Thus,

$$\langle e_{kN}, e_{lN} \rangle = \frac{1}{N} \frac{1 - \left[e^{j2\pi(k-l)/N} \right]^N}{1 - e^{j2\pi(k-l)/N}} = \frac{1}{N} \frac{1-1}{1 - e^{j2\pi(k-l)/N}} = 0$$

- Completed proof by noting $\left[e^{j2\pi(k-l)/N} \right]^N = e^{j2\pi(k-l)} = \left[e^{j2\pi} \right]^{(k-l)} = 1$ □

Canonical frequency sets

ARI

- Exponentials with frequencies that are N apart are equivalent

$-N$,	$-N+1$,	$\ldots$,	-1
0,	1,	$\ldots$,	$N-1$
N,	$N+1$,	$\ldots$,	$2N-1$

- Suffice to look at N consecutive frequencies, e.g., $k = 0, 1, \ldots N-1$
- Another canonical choice is to make $k = 0$ the center frequency

$-N/2+1$,	$\ldots$,	-1,	0,	$\ldots$,	$N/2$
$N/2+1$,	$\ldots$,	$N-1$,	N,	$\ldots$,	$3N/2$

- With N even (as usual) use $N/2$ positive and $N/2-1$ negative
- From one canonical set to the other $\Rightarrow$ Chop and shift

Conjugate frequencies

Theorem
Opposite frequencies k *and* $-k$ *yield conjugate signals:* $e_{-kN} = e_{kN}^*(n)$

Proof.

- Just use the definitions to write the chain of equalities

$$e_{-kN}(n) = \frac{e^{j2\pi(-k)n/N}}{\sqrt{N}} = \frac{e^{-j2\pi kn/N}}{\sqrt{N}} = \left[\frac{e^{j2\pi kn/N}}{\sqrt{N}} \right]^* = e_{kN}^*(n) \qquad \square$$

- Opposite frequencies $\Rightarrow$ Same real part. Opposite imaginary part
  $\Rightarrow$ The cosine is the same, the sine changes sign

Physical meaning

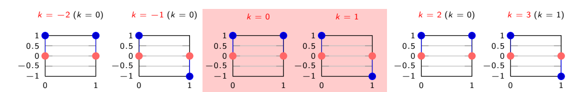
- Of the N canonical frequencies, only $N/2+1$ are distinct.

0,	1,	$\ldots$,	$N/2-1$	$N/2$
	-1,	$\ldots$,	$-N/2+1$	
	$N-1$,	$\ldots$,	$N/2+1$	

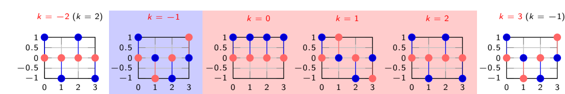
- Frequencies 0 and $N/2$ have no counterpart. Others have conjugates
- Canonical set $-N/2+1, \ldots, -1, 0, 1, \ldots, N/2$ easier to interpret
- Reasonable $\Rightarrow$ Can't have more than $N/2$ oscillations in N samples
- With sampling frequency f_s and signal duration $T = NT_s = N/f_s$
  $\Rightarrow$ Discrete frequency k $\Rightarrow$ frequency $f_0 = \dfrac{k}{T} = \dfrac{k}{NT_s} = \dfrac{k}{N} f_s$
- Frequencies from 0 to $N/2 \leftrightarrow f_s/2$ have physical meaning
  $\Rightarrow$ Negative frequencies are conjugates of the positive frequencies

Complex exponentials for $N = 2$ and $N = 4$

- When $N = 2$ only $k = 0$ and $k = 1$ represent distinct signals



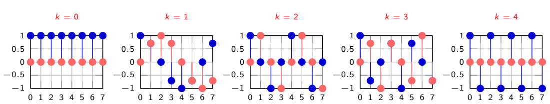
- The signals are real, they have no imaginary parts
- When $N = 4$, $k = 0, 1, 2$ are distinct. $k = -1$ is conjugate of $k = 1$



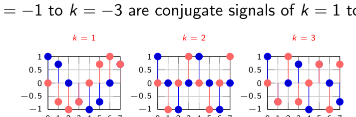
- Can also use $k = 3$ as canonical instead of $k = -1$ – conjugate of $k = 1$

Complex exponentials for $N = 8$

- Frequencies from $k = 1$ to $k = 4$ represent distinct signals



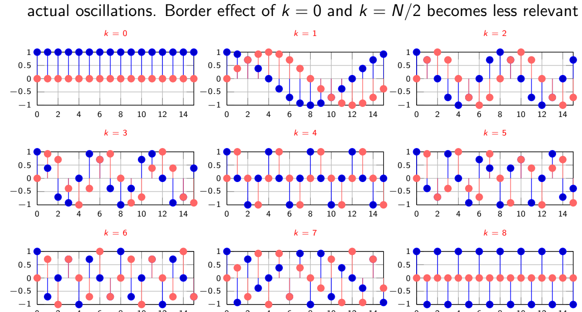
- Frequencies $k = -1$ to $k = -3$ are conjugate signals of $k = 1$ to $k = 3$



- All other frequencies represent one of the signals above

Complex exponentials for $N = 16$

- There are 9 distinct frequencies and 7 conjugates (not shown). Some look like actual oscillations. Border effect of $k = 0$ and $k = N/2$ becomes less relevant



Discrete Fourier transform

Alejandro Ribeiro
Dept. of Electrical and Systems Engineering
University of Pennsylvania
aribeiro@seas.upenn.edu
http://www.seas.upenn.edu/users/~aribeiro/

January 21, 2015

Discrete Fourier transform

Definition of discrete Fourier transform (DFT)

- Signal x of duration N with elements $x(n)$ for $n = 0, \ldots, N-1$
- X is the discrete Fourier transform (DFT) of x if for all $k \in \mathbb{Z}$

$$X(k) := \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) \exp(-j2\pi kn/N)$$

- We write $X = \mathcal{F}(x)$. All values of X depend on all values of x
- The argument k of the DFT is referred to as frequency
- DFT is complex even if signal is real $\Rightarrow X(k) = X_R(k) + jX_I(k)$
  - $\Rightarrow$ It is customary to focus on magnitude

$$\left|X(k)\right| = \left[X_R^2(k) + X_I^2(k)\right]^{1/2} = \left[X(k)X^*(k)\right]^{1/2}$$

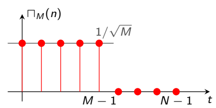
DFT elements as inner products

- Discrete complex exponential (freq. k) $\Rightarrow e_{-kN}(n) = \frac{1}{\sqrt{N}} e^{-j2\pi kn/N}$
- Can rewrite DFT as $\Rightarrow X(k) = \sum_{n=0}^{N-1} x(n) e_{-kN}(n) = \sum_{n=0}^{N-1} x(n) e_{kN}^*(n)$
- And from the definition of inner product $\Rightarrow X(k) = \langle x, e_{kN} \rangle$
- DFT element $X(k)$ $\Rightarrow$ inner product of $x(n)$ with $e_{kN}(n)$
  - $\Rightarrow$ Projection of $x(n)$ onto complex exponential of frequency k
  - $\Rightarrow$ How much of the signal x is an oscillation of frequency k

DFT of a square pulse (derivation)

- The unit energy square pulse is the signal $\sqcap_M(n)$ that takes values

$$\sqcap_M(n) = \frac{1}{\sqrt{M}} \quad \text{if } 0 \le n < M$$
$$\sqcap_M(n) = 0 \quad \text{if } M \le n$$



- Since only the first $M-1$ elements of $\sqcap_M(n)$ are not null, the DFT is

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \sqcap_M(n) e^{-j2\pi kn/N} = \frac{1}{\sqrt{N}} \sum_{n=0}^{M-1} \frac{1}{\sqrt{M}} e^{-j2\pi kn/N}$$

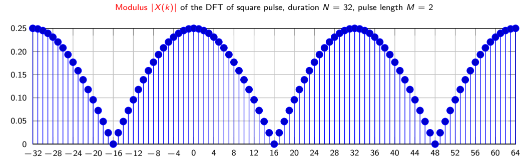
- $X(k)$ = sum of first M components of exponential of frequency $-k$
- Can reduce to simpler expression but who cares? $\Rightarrow$ It's just a sum

DFT of a square pulse (illustration)

- Square pulse of length $M = 2$ and overall signal duration $N = 32$

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{1} \frac{1}{\sqrt{2}} e^{-j2\pi kn/N} = \frac{1}{\sqrt{2N}} \left(1 + e^{-j2\pi k/N}\right)$$

- E.g., $X(k) = \frac{2}{\sqrt{2N}}$ at $k = 0, \pm N, \ldots$ and $X(k) = 0$ at $k = 0, \pm 3N/2, \ldots$



- This DFT is periodic with period N $\Rightarrow$ true in general

Periodicity of the DFT

- Consider frequencies k and $k + N$. The DFT at $k + N$ is

$$X(k+N) := \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi(k+N)n/N}$$

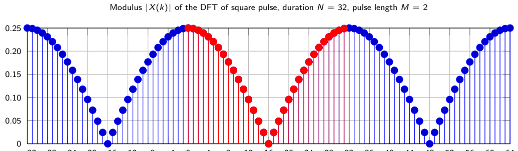
- Complex exponentials of freqs. k and $k + N$ are equivalent. Then

$$X(k+N) := \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} = X(k)$$

- DFT values N apart are equivalent $\Rightarrow$ DFT has period N
- Suffices to look at N consecutive frequencies $\Rightarrow$ canonical sets
  - $\Rightarrow$ Computation $\Rightarrow k \in [0, N-1]$
  - $\Rightarrow$ Interpretation $\Rightarrow k \in [-N/2, N/2]$ (actually, $N+1$ freqs.)
  - $\Rightarrow$ Related by chop and shift $\Rightarrow [-N/2, -1] \sim [N/2, N-1]$

Canonical set $k \in [0, N-1]$

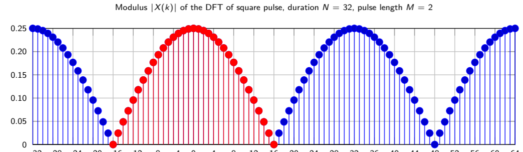
- DFT of the square pulse highlighting frequencies $k \in [0, N-1]$



- Frequencies larger than $N/2$ have no clear physical meaning

Canonical set $k \in [-N/2, N/2]$

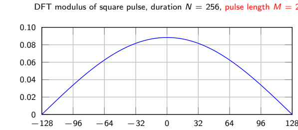
- DFT of the square pulse highlighting frequencies $k \in [-N/2, N/2]$
- Negative freq. $-k$ has the same interpretation as positive freq. k
- One redundant element $\Rightarrow X(-N/2) = X(N/2)$. Just convenient

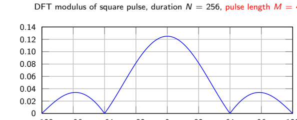


- Obtain frequencies $k \in [-N/2, -1]$ from frequencies $[N/2, N-1]$

Pulses of different length

- The DFT X gives information on how fast the signal x changes

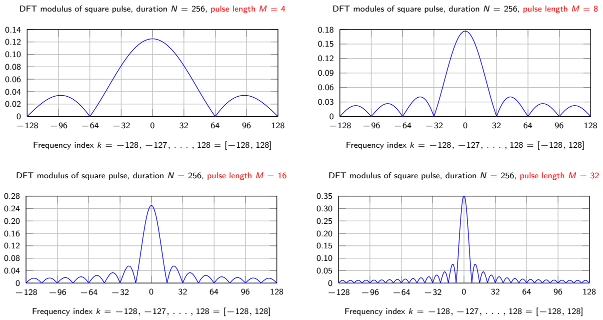




- For length $M = 2$ have weight at high frequencies
- Length $M = 4$ concentrates weight at lower frequencies
- Pulse of length $M = 2$ changes more than a pulse of length $M = 4$

More DFTs of pulses of different length

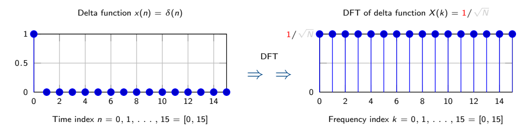
- The lengthier the pulse the less it changes $\Rightarrow$ DFT concentrates at zero freq.



DFT of a delta function

- The delta function is $\delta(0) = 1$ and $\delta(n) = 0$, else. Then, the DFT is

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \delta(n) e^{-j2\pi kn/N} = \frac{1}{\sqrt{N}} \delta(0) e^{-j2\pi k0/N} = \frac{1}{\sqrt{N}}$$



- Only the N values $k \in [0, 15]$ shown. DFT defined for all k but periodic
- Observe that the energy is conserved $\|X\|^2 = \|\delta\|^2 = 1$

DFT of a shifted delta function

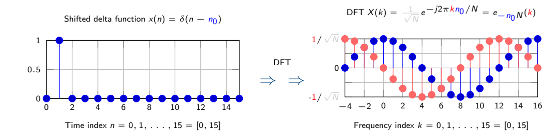
- For shifted delta $\delta(n_0 - n_0) = 1$ and $\delta(n - n_0) = 0$ otherwise. Thus

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \delta(n - n_0) e^{-j2\pi kn/N} = \frac{1}{\sqrt{N}} \delta(n_0 - n_0) e^{-j2\pi kn_0/N}$$

- Of course $\delta(n_0 - n_0) = \delta(0) = 1$, implying that

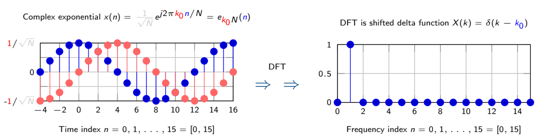
$$X(k) = \frac{1}{\sqrt{N}} e^{-j2\pi kn_0/N} = e_{-n_0N}(k)$$

- Complex exponential of frequency $-n_0$ (below, $N = 16$ and $n_0 = 1$)



DFT of a complex exponential

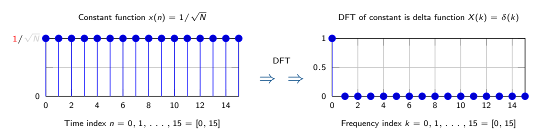
- Complex exponential of freq. k_0 $\Rightarrow x(n) = \frac{1}{\sqrt{N}} e^{j2\pi k_0 n/N} = e_{k_0N}(n)$
- Use inner product form of DFT definition $\Rightarrow X(k) = \langle e_{k_0N}, e_{kN} \rangle$
- Orthonormality of complex exponentials $\Rightarrow \langle e_{k_0N}, e_{kN} \rangle = \delta(k - k_0)$



- DFT of exponential $e_{k_0N}(n)$ is shifted delta $X(k) = \delta(k - k_0)$

DFT of a constant

- Constant function $x(n) = 1/\sqrt{N}$ (it has unit energy) and $k = 0$
  $\Rightarrow$ Complex exponential with frequency $k_0 = 0$ $\Rightarrow x(n) = e_{0N}$
- Use inner product form of DFT definition $\Rightarrow X(k) = \langle e_{0N}, e_{kN} \rangle$
- Complex exponential orthonormality $\Rightarrow \langle e_{0N}, e_{kN} \rangle = \delta(k - 0) = \delta(k)$



- DFT of constant $x(n) = 1/\sqrt{N}$ is delta function $X(k) = \delta(k)$

Observations

- DFT of a signal captures its rate of change
- Signals that change faster have more DFT weight at high frequencies

- DFT conserves energy (all have unit energy in our examples)
- Energy of DFT $X = \mathcal{F}(x)$ is the same as energy of the signal x
- Indeed, an important property we will show

- Duality of signal - transform pairs (signals and DFTs come in pairs)
- DFT of delta is a constant. DFT of constant is a delta
- DFT of exponential is shifted delta. DFT of shifted delta is exponential
- Indeed, a fact that follows from the form of the inverse DFT

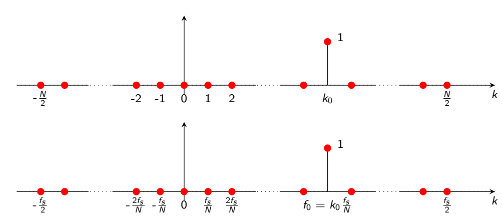
Units of the DFT

Units

- Sampling time T_s, sampling frequency f_s, signal duration $T = NT_s$
- Discrete frequency k $\Rightarrow$ k oscillations in time NT_s = Period NT_s/k
- Discrete frequency k equivalent to real frequency $f_k = \dfrac{k}{NT_s} = k\dfrac{f_s}{N}$
- In particular, $k = N/2$ equivalent to $\Rightarrow f_{N/2} = \dfrac{N/2f_s}{N} = \dfrac{f_s}{2}$
- Set of frequencies $k \in [-N/2, N/2]$ equivalent to real frequencies ...
  $\Rightarrow$ That lie between $-f_s/2$ and $f_s/2$
  $\Rightarrow$ Are spaced by f_s/N (difference between frequencies f_k and f_{k+1})
- Interval width given by sampling frequency. Resolution given by N

Units in DFT of a discrete complex exponential

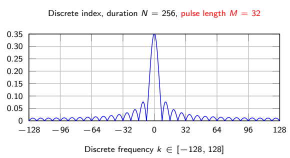
- Complex exponential of frequency $f_0 = k_0 f_s/N$
  $\Rightarrow$ Discrete frequency k_0 and DFT $\Rightarrow X(k) = \delta(k - k_0)$
- But frequency k_0 corresponds to frequency f_0 $\Rightarrow X(f) = \delta(f - f_0)$

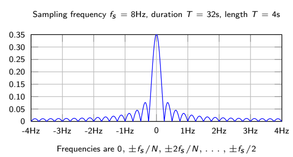


- True only when frequency $f_0 = (k_0/N)f_s$ is a multiple of f_s/N

Units in DFT of a square pulse

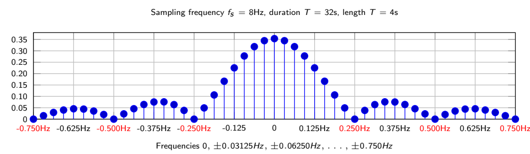
- Square pulse of length $T_0 = 4$s observed during a total of $T = 32$s.
- Sampled every $T_s = 125$ms $\Rightarrow$ Sample frequency $f_s = 8$Hz
- Total number of samples $\Rightarrow N = T/T_s = 256$
- Maximum frequency $k = N/2 = 128 \leftrightarrow f_k = f_{N/2} = f_s/2 = 4Hz$
- Fequency resolution $f_s/N = 8\text{Hz}/256 = 0.03125\text{Hz}$





Units in DFT of a square pulse

- Interval between freqs. $\Rightarrow f_s/N = 8\text{Hz}/256 = 1/32 = 0.03125\text{Hz}$
  $\Rightarrow$ 32 equally spaced frees for each 1Hz interval = 8 every 0.125 Hz.



- Zeros of DFT are at frequencies 0.250Hz, 0.500 Hz, 0.750 Hz, ...
  $\Rightarrow$ Thus, zeros are at frequencies are $1/T_0, 2/T_0, 3/T_0, \ldots$
- Most (a lot) of the DFT energy is between freqs. $-1/T_0$ and $1/T_0$

DFT inverse

Definition of DFT inverse

- Given a Fourier transform X, the inverse (i)DFT $x = \mathcal{F}^{-1}(X)$ is

$$x(n) := \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) \exp(j2\pi kn/N)$$

- Same as DFT but for sign in the exponent (also, sum over k, not n)
- Any summation over N consecutive frequencies works as well. E.g.,

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} X(k) e^{j2\pi kn/N}$$

- Because for a DFT X we know that it must be $X(k+N) = X(k)$

iDFT is, indeed, the inverse of the DFT

Theorem

The inverse DFT of the DFT of x is the signal $x \Rightarrow \mathcal{F}^{-1}[\mathcal{F}(x)] = x$

- Every signal x can be written as a sum of complex exponentials

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} = \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} X(k) e^{j2\pi kn/N}$$

- Coefficient multiplying $e^{j2\pi kn/N}$ is $X(k)$ = kth element of DFT of x

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

Proof of DFT inverse formula

Proof.

- Let $X = \mathcal{F}(x)$ be the DFT of x. Let $\tilde{x} = \mathcal{F}^{-1}(X)$ be the iDFT of X.
  $\Rightarrow$ We want to show that $\tilde{x} \equiv x$
- From the definition of the iDFT of $X \Rightarrow \tilde{x}(\tilde{n}) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{j2\pi k\tilde{n}/N}$
- From the definition of the DFT of $x \Rightarrow X(k) := \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$
- Substituting expression for $X(k)$ into expression for $\tilde{x}(\tilde{n})$ yields

$$\tilde{x}(\tilde{n}) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left[\frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \right] e^{j2\pi k\tilde{n}/N}$$

Proof of DFT inverse formula

Proof.

- Exchange summation order to sum first over k and then over n

$$\tilde{x}(\tilde{n}) = \sum_{n=0}^{N-1} x(n) \left[\sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} e^{j2\pi k\tilde{n}/N} \frac{1}{\sqrt{N}} e^{-j2\pi kn/N} \right]$$

- Pulled $x(n)$ out because it doesn't depend on k
- Innermost sum is the inner product between $e_{\tilde{n}N}$ and e_{nN}. Orthonormality:

$$\sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} e^{j2\pi k\tilde{n}/N} \frac{1}{\sqrt{N}} e^{-j2\pi kn/N} = \delta(\tilde{n} - n)$$

- Reducing to $\Rightarrow \tilde{x}(\tilde{n}) = \sum_{n=0}^{N-1} x(n)\delta(\tilde{n} - n) = x(\tilde{n})$
- Last equation is true because only term $n = \tilde{n}$ is not null in the sum □

Inverse DFT as inner product

- Discrete complex exponential (freq. n) $\Rightarrow e_{nN}(k) = \frac{1}{\sqrt{N}} e^{j2\pi kn/N}$
- Rewrite iDFT as $\Rightarrow x(n) = \sum_{k=0}^{N-1} X(k) e_{nN}(k) = \sum_{k=0}^{N-1} X(k) e^{*}_{-nN}(k)$
- And from the definition of inner product $\Rightarrow x(n) = \langle X, e_{nN} \rangle$
- iDFT element $X(k) \Rightarrow$ inner product of $X(k)$ with $e_{-nN}(k)$
- Different from DFT, this is not the most useful interpretation

Inverse DFT as successive approximations

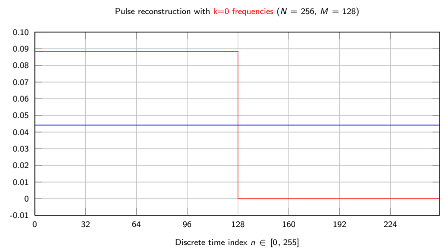
- Signal as sum of exponentials $\Rightarrow x(n) = \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} X(k) e^{j2\pi kn/N}$
- Expand the sum inside out from $k = 0$ to $k = \pm 1$, to $k = \pm 2$, ...

$$\begin{array}{llllll}
x(n) = & X(0) & e^{j2\pi 0n/N} & & & \text{constant} \\
& +X(1) & e^{j2\pi 1n/N} & +X(-1) & e^{-j2\pi 1n/N} & \text{single oscillation} \\
& +X(2) & e^{j2\pi 2n/N} & +X(-2) & e^{-j2\pi 2n/N} & \text{double oscillation} \\
& \vdots & \vdots & \vdots & \vdots & \vdots \\
& +X\left(\frac{N}{2}-1\right) & e^{j2\pi\left(\frac{N}{2}-1\right)n/N} & +X\left(-\frac{N}{2}+1\right) & e^{-j2\pi\left(\frac{N}{2}-1\right)n/N} & \left(\frac{N}{2}-1\right) - \text{oscillation} \\
& +X\left(\frac{N}{2}\right) & e^{j2\pi\left(\frac{N}{2}\right)n/N} & & & \frac{N}{2} - \text{oscillation}
\end{array}$$

- Start with slow variations and progress on to add faster variations

Reconstruction of square pulse

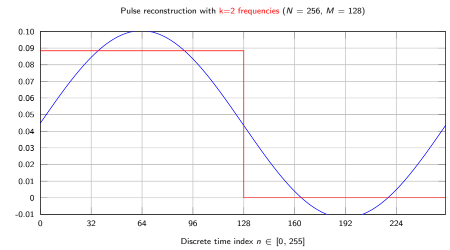
- Consider square pulse of duration $N = 256$ and length $M = 128$
- Reconstruct with frequency $k = 0$ only (DC component)



- Bound to be not very good $\Rightarrow$ Just the average signal value

Reconstruction of square pulse

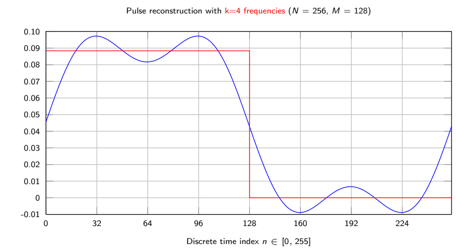
- Consider square pulse of duration $N = 256$ and length $M = 128$
- Reconstruct with frequencies $k = 0$, $k = \pm 1$, and $k = \pm 2$



- Not too bad, sort of looks like a pulse $\Rightarrow$ only 3 frequencies

Reconstruction of square pulse

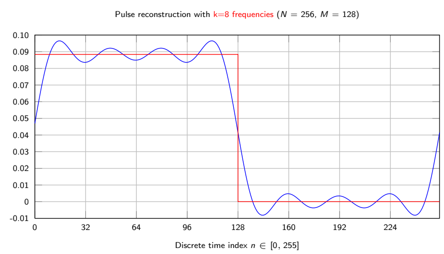
- Consider square pulse of duration $N = 256$ and length $M = 128$
- Reconstruct with frequencies up to $k = 4$



- Starts to look like a good approximation

Reconstruction of square pulse

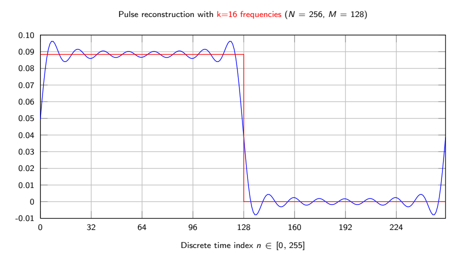
- Consider square pulse of duration $N = 256$ and length $M = 128$
- Reconstruct with frequencies up to $k = 8$



- Good approximation of the $N = 256$ values with 9 DFT coefficients

Reconstruction of square pulse

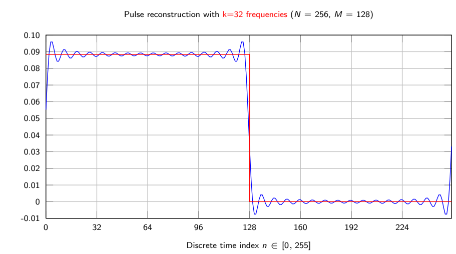
- Consider square pulse of duration $N = 256$ and length $M = 128$
- Reconstruct with frequencies up to $k = 16$



- Compression $\Rightarrow$ Store $k + 1 = 17$ DFT values instead of $N = 128$ samples

Reconstruction of square pulse

- Consider square pulse of duration $N = 256$ and length $M = 128$
- Reconstruct with frequencies up to $k = 32$



- Can tradeoff less compression for better signal accuracy

Spectrum (re)shaping

(1) Start with a signal x with elements $x(n)$. Compute DFT X as

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

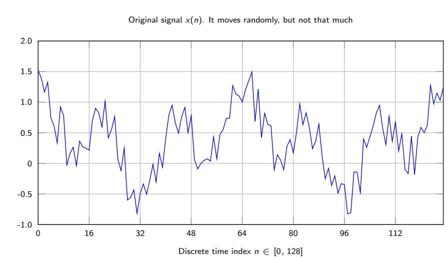
(2) (Re)shape spectrum $\Rightarrow$ Transform DFT X into DFT Y

(3) With DFT Y available, recover signal y with inverse DFT

$$y(n) := \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} Y(k) e^{j2\pi kn/N}$$

Spectrum reshaping to remove noise

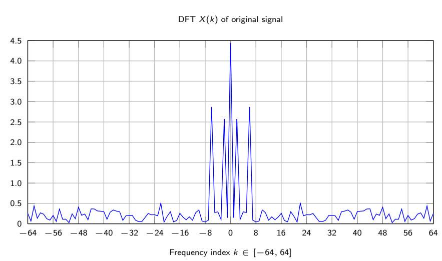
- An application of spectrum reshaping is to clean a noisy signal
- Signal with some underlying trend (good) and some noise (bad)



- Which is which? $\Rightarrow$ Not clear $\Rightarrow$ Let's look at the spectrum (DFT)

Spectrum reshaping to remove noise

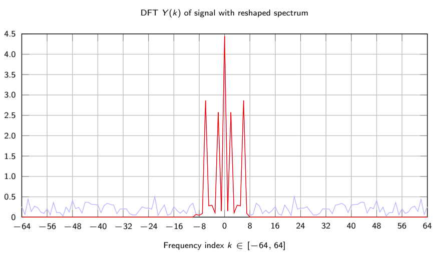
- An application of spectrum reshaping is to clean a noisy signal
- Now the trend (spikes) is clearly separated from the noise (the floor)



- How do we remove the noise? $\Rightarrow$ Reshape the spectrum

Spectrum reshaping to remove noise

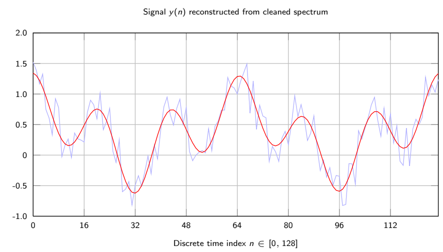
- An application of spectrum reshaping is to clean a noisy signal
- Remove freqs. larger than 8 $\Rightarrow Y(k) = 0$ for $k > 8$, $Y(k) = X(k)$ else



- How do we recover the trend? $\Rightarrow$ Inverse DFT

Spectrum reshaping to remove noise

- An application of spectrum reshaping is to clean a noisy signal
- Inverse DFT of reshaped specturm $Y(k)$ yields cleaned signal $y(n)$



- The trend now is clearly visible. Noise has been removed

Properties of the DFT

Three important properties of DFTs

- DFTs of real signals (no imaginary part) are conjugate symmetric

$$X(-k) = X^*(k)$$

- Signals of unit energy have transforms of unit energy
- More generically, the DFT preserves energy (Parseval's theorem)

$$\sum_{n=0}^{N-1} |x(n)|^2 = \|x\|^2 = \|X\|^2 = \sum_{k=0}^{N-1} |X(k)|^2$$

- The DFT operator is a linear operator

$$\mathcal{F}(ax + by) = a\mathcal{F}(x) + b\mathcal{F}(y)$$

Symmetry

Theorem

The DFT $X = \mathcal{F}(x)$ of a real signal x is conjugate symmetric

$$X(-k) = X^*(k)$$

- Can recover all DFT components from those with freqs. $k \in [0, N/2]$
- What about components with freqs. $k \in [-N/2, -1]$?
  $\Rightarrow$ Conjugates of those with freqs $k \in [0, N/2]$
- Other elements are equivalent to one in $[-N/2, N/2]$ (periodicity)

Proof of symmetry property

Proof.

- Write the DFT $X(-k)$ using its definition

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

- When the signal is real, its conjugate is itself $\Rightarrow x(n) = x^*(n)$
- Conjugating a complex exponential $\Rightarrow$ changing the exponent's sign
- Can then rewrite $\Rightarrow X(-k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x^*(n) \left(e^{-j2\pi kn/N}\right)^*$
- Sum and multiplication can change order with conjugation

$$X(-k) = \left[\frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}\right]^* = X^*(k) \qquad \square$$

Energy conservation

Theorem (Parseval)

Let $X = \mathcal{F}(x)$ be the DFT of signal x. The energies of x and X are the same, i.e.,

$$\sum_{n=0}^{N-1} |x(n)|^2 = \|x\|^2 = \|X\|^2 = \sum_{k=0}^{N-1} |X(k)|^2$$

- In energy of DFT, any set of consecutive freqs. would do. E.g.,

$$\|X\|^2 = \sum_{k=0}^{N-1} |X(k)|^2 = \sum_{k=-N/2+1}^{N/2} |X(k)|^2$$

Proof of Parseval's Theorem

Proof.

- From the definition of the energy of $X \Rightarrow \|X\|^2 = \sum_{k=0}^{N-1} X(k)X^*(k)$
- From the definition of the DFT of $x \Rightarrow X(k) := \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$
- Substitute expression for $X(k)$ into one for $\|X\|^2$ (observe conjugation)

$$\|X\|^2 = \sum_{k=0}^{N-1} \left[\frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}\right] \left[\frac{1}{\sqrt{N}} \sum_{\tilde{n}=0}^{N-1} x^*(\tilde{n}) e^{+j2\pi k\tilde{n}/N}\right]$$

Proof of Parseval's Theorem

Proof.

- Distribute product and exchange order of summations $\Rightarrow$ sum over k first

$$\|X\|^2 = \sum_{n=0}^{N-1} \sum_{\tilde{n}=0}^{N-1} x(n) x^*(\tilde{n}) \left[\sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} e^{-j2\pi kn/N} \frac{1}{\sqrt{N}} e^{+j2\pi k\tilde{n}/N}\right]$$

- Pulled $x(n)$ and $x^*(\tilde{n})$ out because they don't depend on k
- Innermost sum is the inner product between $e_{\tilde{n}N}$ and e_{nN}. Orthonormality:

$$\sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} e^{-j2\pi kn/N} \frac{1}{\sqrt{N}} e^{+j2\pi k\tilde{n}/N} = \langle e_{\tilde{n}N}, e_{nN} \rangle = \delta(\tilde{n} - n)$$

- Thus $\Rightarrow \|X\|^2 = \sum_{n=0}^{N-1} \sum_{\tilde{n}=0}^{N-1} x(n) x^*(\tilde{n}) \delta(\tilde{n} - n) = \sum_{n=0}^{N-1} x(n) x^*(n) = \|x\|^2$
- True because only terms $n = \tilde{n}$ are not null in the sum $\square$

Linearity

Theorem
The DFT of a linear combination of signals is the linear combination of the respective DFTs of the individual signals,

$$\mathcal{F}(ax+by) = a\mathcal{F}(x) + b\mathcal{F}(y).$$

- In particular...
  - $\Rightarrow$ Adding signals ($z = x + y$) $\Rightarrow$ Adding DFTs ($Z = X + Y$)
  - $\Rightarrow$ Scaling signals($y = ax$) $\Rightarrow$ Scaling DFTs ($Y = aX$)

Proof of Linearity

Proof.

- Let $Z := \mathcal{F}(ax+by)$. From the definition of the DFT we have

$$Z(k) = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}\Big[ax(n) + by(n)\Big]e^{-j2\pi kn/N}$$

- Expand the product, reorder terms, identify the DFTs of x and y

$$Z(k) = \frac{a}{\sqrt{N}}\sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} + \frac{b}{\sqrt{N}}\sum_{n=0}^{N-1} y(n)e^{-j2\pi kn/N}$$

- First sum is DFT $X = \mathcal{F}(x)$. Second sum is DFT $Y = \mathcal{F}(y)$

$$Z(k) = aX(k) + bY(k) \qquad \square$$

DFT of a discrete cosine

- DFT of discrete cosine of freq. k_0 $\Rightarrow x(n) = \frac{1}{\sqrt{N}}\cos(2\pi k_0 n/N)$
- Can write cosine as a sum of discrete complex exponentials

$$x(n) = \frac{1}{2\sqrt{N}}\Big[e^{j2\pi k_0 n/N} + e^{-j2\pi k_0 n/N}\Big] = \frac{1}{2}\Big[e_{k_0N}(n) + e_{-k_0N}(n)\Big]$$

- From linearity of DFTs $\Rightarrow X = \mathcal{F}(x) = \frac{1}{2}\Big[\mathcal{F}(e_{k_0N}) + \mathcal{F}(e_{-k_0N})\Big]$
- DFT of complex exponential e_{kN} is delta function $\delta(k-k_0)$. Then

$$X(k) = \frac{1}{2}\Big[\delta(k-k_0) + \delta(k+k_0)\Big]$$

- A pair of deltas at positive and negative frequency k_0

DFT of a discrete sine

- DFT of discrete sine of freq. k_0 $\Rightarrow x(n) = \frac{1}{\sqrt{N}}\sin(2\pi k_0 n/N)$
- Can write sine as a difference of discrete complex exponentials

$$x(n) = \frac{1}{2j\sqrt{N}}\Big[e^{j2\pi k_0 n/N} - e^{-j2\pi k_0 n/N}\Big] = \frac{-j}{2}\Big[e_{k_0N}(n) - e_{-k_0N}(n)\Big]$$

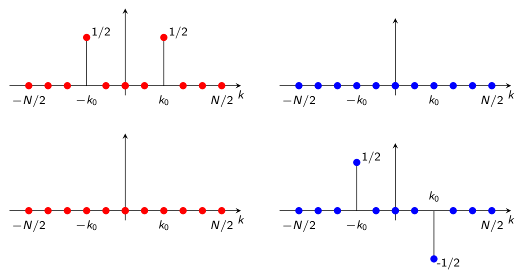
- From linearity of DFTs $\Rightarrow X = \mathcal{F}(x) = \frac{j}{2}\Big[\mathcal{F}(e_{-k_0N}) - \mathcal{F}(e_{k_0N})\Big]$
- DFT of complex exponential e_{kN} is delta function $\delta(k-k_0)$. Then

$$X(k) = \frac{j}{2}\Big[\delta(k+k_0) - \delta(k-k_0)\Big]$$

- Pair of opposite complex deltas at positive and negative frequency k_0

DFT of discrete cosine and discrete sine

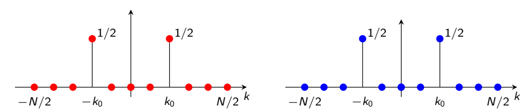
- Cosine has real part only (top). Sine has imaginary part only (bottom)



- Cosine is symmetric around $k = 0$. Sine is antisymmetric around $k = 0$.

DFT of discrete cosine and discrete sine (more)

- Real and imaginary parts are different but the moduli are the same



- Cosine and sine are essentially the same signal (shifted versions)
  - $\Rightarrow$ The moduli of their DFTs are identical
  - $\Rightarrow$ Phase difference captured by phase of complex number $X(\pm k_0)$

Fourier transforms

Alejandro Ribeiro
Dept. of Electrical and Systems Engineering
University of Pennsylvania
aribeiro@seas.upenn.edu
http://www.seas.upenn.edu/users/~aribeiro/

February 10, 2015

Discrete signals and DFT

- Fourier analysis of discrete signals $x : [0, N-1] \to \mathbb{C}$ $\Rightarrow$ DFT, iDFT
- Good (and quick) computational tool
  - $\Rightarrow$ Signal analysis $\Rightarrow$ pattern discovery, frequency components
  - $\Rightarrow$ Signal processing $\Rightarrow$ compression, noise removal
- Two important limitations
  - $\Rightarrow$ Time is neither discrete nor finite (not always, at least)
  - $\Rightarrow$ Properties and interpretations are easier in continuous time
- Fourier analysis of continuous signals $\Rightarrow$ Fourier transform (FT)

Continuous time signals

To infinity and beyond

- We have been dealing with discrete signals $x : [0, N-1] \to \mathbb{C}$
- To infinity $\Rightarrow$ Let number of samples go to infinity
  - $\Rightarrow$ Discrete time signal $x : \mathbb{Z} \to \mathbb{C}$
  - $\Rightarrow$ Values $x(n)$ for $n = \ldots, -1, 0, 1, \ldots$
- And beyond $\Rightarrow$ Fill in the gaps between samples
  - $\Rightarrow$ Continuous time signal $x : \mathbb{R} \to \mathbb{C}$
  - $\Rightarrow$ Values $x(t)$ for t any real number in $(-\infty, +\infty)$
- Let's begin by studying continuous time signals

Continuous time signals

- Continuous time variable $t \in \mathbb{R}$.
- Continuous time signal x is a function that maps t to real value $x(t)$

$$x : \mathbb{R} \to \mathbb{R}$$

- The values that the signal takes at time t is $x(t)$
- It will make sense to talk about complex signals (as in discrete case)

$$x : \mathbb{R} \to \mathbb{C}$$

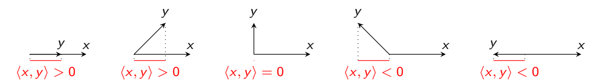
- where the values $x(t) = x_R(t) + j\,x_I(t)$ are complex numbers

Inner product

- Given two signals x and y define the inner product of x and y as

$$\langle x, y \rangle := \int_{-\infty}^{\infty} x(t) y^*(t) dt$$

- Akin to inner product of discrete signals $\Rightarrow \langle x, y \rangle = \sum_{n=0}^{N} x(n) y(n)$



- But we have infinite number of components. To infinity and beyond
- Intuition holds $\Rightarrow \langle x, y \rangle$ is how much of y falls in x direction
- E.g., if $\langle x, y \rangle = 0$ the signals are orthogonal. They are "unrelated"

Norm and energy

- As for regular (finite dimensional) signals define the norm of signal x

$$\|x\| := \left[\int_{-\infty}^{\infty} |x(t)|^2 dt\right]^{1/2} = \left[\int_{-\infty}^{\infty} |x_R(t)|^2 dt + \int_{-\infty}^{\infty} |x_I(t)|^2 dt\right]^{1/2}$$

- More important, define the energy of the signal as the norm squared

$$\|x\|^2 := \int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |x_R(t)|^2 dt + \int_{-\infty}^{\infty} |x_I(t)|^2 dt$$

- For complex numbers $x(t)x^*(t) = |x_R(t)|^2 + |x_I(t)|^2 = |x(t)|^2$
- Thus, we can write the energy as $\Rightarrow \|x\|^2 = \langle x, x \rangle$
- Energy might be infinite. When energy is finite we write $\|x\|^2 < \infty$

Cauchy Schwarz inequality

- The largest an inner product can be is when the vectors are collinear

$$-\|x\|\,\|y\| \le \langle x, y \rangle \le \|x\|\,\|y\|$$

- Or in terms of energy $\Rightarrow \langle x, y \rangle^2 \le \|x\|^2\,\|y\|^2$
- If you are the sort of person that prefers explicit expressions

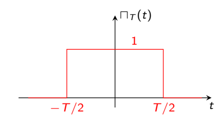
$$\int_{-\infty}^{\infty} x(t) y^*(t) dt \le \left[\int_{-\infty}^{\infty} |x(t)|^2 dt\right]\left[\int_{-\infty}^{\infty} |y(t)|^2 dt\right]$$

- The equalities hold if and only if x and y are collinear

Example: Square pulse

- The square pulse is the signal $\sqcap_T(t)$ that takes values

$$\sqcap_T(t) = 1 \quad \text{for } -\frac{T}{2} \le t < \frac{T}{2}$$
$$\sqcap_T(t) = 0 \quad \text{otherwise}$$



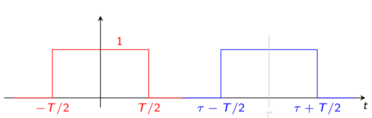
- To compute energy of the pulse we just evaluate the definition

$$\|\sqcap_T(t)\|^2 := \int_{-\infty}^{\infty} |\sqcap_T(t)|^2 dt = \int_{-T/2}^{T/2} |1|^2\, dt = T$$

- Energy proportional to pulse duration (duh!)
- Can normalize energy dividing by $\sqrt{T}$. But we rather not.

Shifted pulses (1 of 2)

- To shift a pulse we modify the argument $\Rightarrow \sqcap_T(t-\tau)$
  - $\Rightarrow$ The pulse is now centered at τ ($t = \tau$ is as $t = 0$ before)



- Inner product of two pulses with disjoint support ($\tau > T$)

$$\langle \sqcap_T(t), \sqcap_T(t-\tau) \rangle := \int_{-\infty}^{\infty} \sqcap_T(t) \sqcap_T(t-\tau) = 0$$

- The signals are orthogonal, and indeed, "unrelated" to each other

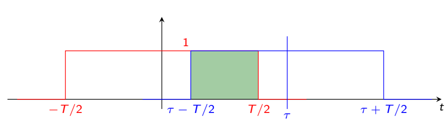
Shifted pulses (2 of 2)

- Inner product of two pulses with overlapping support ($\tau > T$)

$$\langle \sqcap_T(t), \sqcap_T(t-\tau) \rangle := \int_{-\infty}^{\infty} \sqcap_T(t) \sqcap_T(t-\tau)$$

- The signals overlap between $\tau - T/2$ and $T/2$. Thus

$$\langle \sqcap_T(t), \sqcap_T(t-\tau) \rangle = \int_{\tau-T/2}^{T/2} (1)(1) dt = \frac{T}{2} - \left(\tau - \frac{T}{2}\right) = T - \tau$$



- Inner product is proportional to the relative overlap
  - $\Rightarrow$ which is, indeed, how much the signals are "related" to each other

Complex exponentials

- Inner product and energy are indefinite integrals $\Rightarrow$ need not exist
- Complex exponential of frequency f is e_f with $e_f(t) = e^{j2\pi ft}$
- Since they have unit modulus ($|e_f(t)| = |e^{j2\pi ft}| = 1$), their energy is

$$\|e_f\|^2 := \int_{-\infty}^{\infty} |e_f(t)|^2 dt = \int_{-\infty}^{\infty} 1 dt = \infty$$

- Inner product of complex exponentials not defined ("keeps oscillating")

$$\langle e_f, e_g \rangle := \int_{-\infty}^{\infty} e_f(t) e_g^*(t) dt = \int_{-\infty}^{\infty} e^{j2\pi ft} e^{-j2\pi gt} dt = \int_{-\infty}^{\infty} e^{j2\pi(f-g)t} dt \Rightarrow \nexists$$

- This is a problem because we can't talk about orthogonality
  - $\Rightarrow$ Still, a complex exponential is much more like itself than another

Fourier transform

Definition of Fourier transform

- The Fourier transform of x is the function $X : \mathbb{R} \to \mathbb{C}$ with values

$$X(f) := \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt$$

- We write $X = \mathcal{F}(x)$. All values of X depend on all values of x
- Integral need not exist $\Rightarrow$ Not all signals have a Fourier transform
- The argument f of the Fourier transform is referred to as frequency
- Or, define e_f with values $e_f(t) = e^{j2\pi f t}$ to write as inner product

$$X(f) = \langle x, e_f \rangle = \int_{-\infty}^{\infty} x(t) e_f^*(t) dt$$

- Both, time and frequency are real $\Rightarrow$ domain is infinite and dense
  $\Rightarrow$ This is an analytical tool, not a computational tool (as the DFT)

Example: Fourier transform of a square pulse

- Since pulse is not null only when $T/2 \leq t \leq T/2$ we reduce $X(f)$ to

$$X(f) := \int_{-\infty}^{\infty} \sqcap_T(t) e^{-j2\pi f t} dt = \int_{-T/2}^{T/2} e^{-j2\pi f t} dt$$

- For $f \neq 0$, the primitive of $e^{-j2\pi f t}$ is $(-1/j2\pi f) e^{-j2\pi f t}$, which yields

$$X(f) = \left[\frac{-e^{-j2\pi f T/2}}{j2\pi f} - \frac{-e^{+j2\pi f T/2}}{j2\pi f} \right] = \frac{\sin(\pi f T)}{\pi f}$$

- Where we used $e^{j\pi f T} - e^{-j\pi f T} = 2j \sin(\pi f T)$
- For $f = 0$ we have $e^{-j2\pi f t} = 1$ and $X(f)$ reduces to $\Rightarrow X(f) = T$

The sinc function

- Transform is important enough to justify definition of sinc function

$$\text{sinc}(u) = \frac{\sin(u)}{u} \quad \text{for } u \neq 0$$
$$\text{sinc}(u) = 1 \quad \text{for } u = 0$$

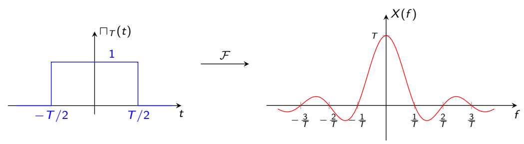
- Value at origin, $\text{sinc}(0) = 1$, makes the function continuous
- With this definition and $f \neq 0$ we can write the pulse transform as

$$X(f) = \frac{\sin(\pi f T)}{\pi f} = T \frac{\sin(\pi f T)}{\pi f T} = T \text{sinc}(\pi f T)$$

- Which is also true for $f = 0$ because $X(0) = T\text{sinc}(\pi 0 T) = T$

The pulse and its transform

- Fourier transform of pulse of width T is sinc with null crossings $\frac{k}{T}$

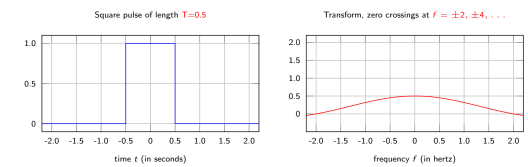


- Most of the Fourier Transform energy is between $-1/T$ and $1/T$

$$\int_{-1/T}^{1/T} \left| X(f) \right|^2 df = \int_{-1/T}^{1/T} \left| T\text{sinc}(\pi f T) \right|^2 df \approx 0.90T = 0.90 \| \sqcap_T(t) \|^2$$

Pulses of different width

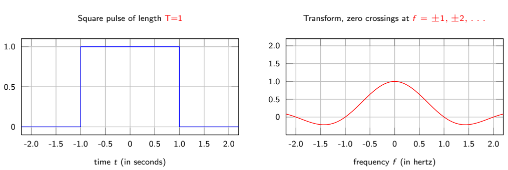
- Transforms of wider pulses are more concentrated around $f = 0$



- Consistent with interpretation that shorter pulses are faster varying

Pulses of different width

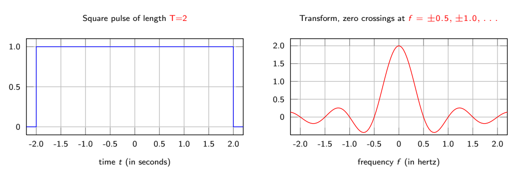
- Transforms of wider pulses are more concentrated around $f = 0$



- Consistent with interpretation that shorter pulses are faster varying

Pulses of different width

- Transforms of wider pulses are more concentrated around $f = 0$



- Consistent with interpretation that shorter pulses are faster varying

The Fourier transform and the DFT

- Let's compute a Fourier transform by approximating the integral
- Use samples spaced by T_s time units

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \approx T_s \sum_{-\infty}^{\infty} x(nT_s) e^{-j2\pi f n T_s}$$

- Still not computable $\Rightarrow$ consider only N samples from 0 to $N - 1$

$$X(f) \approx T_s \sum_{k=0}^{N-1} x(nT_s) e^{-j2\pi f n T_s}$$

- This is true for all frequencies. Consider frequencies $f = (k/N) f_s$

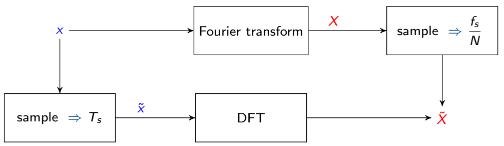
$$X\left(\frac{k}{N} f_s\right) \approx T_s \sum_{k=0}^{N-1} x(nT_s) e^{-j2\pi (k/N) f_s n T_s} = T_s \sum_{k=0}^{N-1} x(nT_s) e^{-j2\pi k n/N}$$

- Definition of the DFT of a discrete signal (up to constants)

DFT as approximation of Fourier transform

- Define $\tilde{x}$ with $\tilde{x}(n) = x(nT_s)$. The DFT of $\tilde{X} = \mathcal{F}(\tilde{x})$ has components

$$\tilde{X}(k) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \tilde{x}(n) e^{-j2\pi kn/N} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x(nT_s) e^{-j2\pi kn/N} = \frac{1}{T_s\sqrt{N}} X\left(\frac{k}{N} f_s\right)$$



- Can then aproximate Fourier transform as $\Rightarrow X\left(\frac{k}{N} f_s\right) \approx T_s\sqrt{N}\tilde{X}(k)$
- Approximation becomes equality at infinity an beyond ($N \to \infty,\ T_s \to 0$)

Fourier transform of a complex exponential

- Complex exponential of frequency f_0 $\Rightarrow e_{f_0}(t) = e^{j2\pi f_0 t}$
- Use inner product form to write the components of $X = \mathcal{F}(e_{f_0})$ as

$$X(f) = \langle x, e_f \rangle = \langle e_{f_0}, e_f \rangle$$

- We've seen that $\langle e_{f_0}, e_f \rangle = \infty$ if $f = f_0$ and oscillates ($\nexists$) if $f \neq f_0$
- The complex exponential does not have a Fourier transform
  - $\Rightarrow$ Happens because energy of complex exponentials is not finite
- Truncate to $T/2 \leq t \leq T/2$ $\Rightarrow$ multiply by square pulse $\sqcap_T(t)$

$$\tilde{e}_{f_0 T}(t) := e_{f_0}(t) \sqcap_T(t) = e^{j2\pi f_0 t} \sqcap_T(t)$$

Fourier transform of a complex exponential

- Truncated exponential not null only when $T/2 \leq t \leq T/2$ (pulse)
- Then, the Fourier transform $\tilde{X}_T(f) := \mathcal{F}(\tilde{e}_{f_0 T})$ is given by

$$\tilde{X}(f) := \int_{-\infty}^{\infty} e^{j2\pi f_0 t} \sqcap_T(t) e^{-j2\pi f t} dt = \int_{-T/2}^{T/2} e^{j2\pi f_0 t} e^{-j2\pi f t} dt = \int_{-T/2}^{T/2} e^{-j2\pi (f-f_0) t} dt$$

- Same as pulse transform, except for frequency shift in exponent
- For $f \neq f_0$, primitive of $e^{-j2\pi f t}$ is $(-1/j2\pi(f - f_0))e^{-j2\pi(f-f_0)t}$. Thus

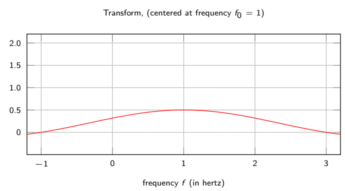
$$\tilde{X}(f) = \left[\frac{-e^{-j2\pi(f-f_0)T/2}}{j2\pi(f-f_0)} - \frac{-e^{+j2\pi(f-f_0)T/2}}{j2\pi(f-f_0)}\right] = \frac{\sin(\pi(f-f_0)T)}{\pi(f-f_0)}$$

- For $f = f_0$ we have $e^{-j2\pi(f-f_0)t} = 1$ and $\tilde{X}(f)$ reduces to $\Rightarrow \tilde{X}(f) = T$

Shifted sinc

- Fourier transform of truncated complex exponential is shifted sinc

$$\tilde{X}(f) = T\text{sinc}(\pi(f - f_0)T)$$

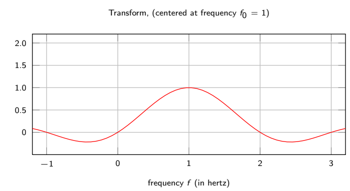


- As $T \to \infty$ truncated exponential approaches exponential
  - $\Rightarrow$ And shifted sinc becomes infinitely tall $\Rightarrow$ delta function

Shifted sinc

- Fourier transform of truncated complex exponential is shifted sinc

$$\tilde{X}(f) = T\text{sinc}(\pi(f - f_0)T)$$

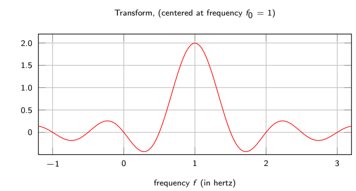


- As $T \to \infty$ truncated exponential approaches exponential
  - $\Rightarrow$ And shifted sinc becomes infinitely tall $\Rightarrow$ delta function

Shifted sinc

- Fourier transform of truncated complex exponential is shifted sinc

$$\tilde{X}(f) = T\text{sinc}(\pi(f - f_0)T)$$



- As $T \to \infty$ truncated exponential approaches exponential
  - $\Rightarrow$ And shifted sinc becomes infinitely tall $\Rightarrow$ delta function

Inverse Fourier transform

Inverse Fourier transform

- Given a transform X, the inverse Fourier transform is defined as

$$x(t) := \int_{-\infty}^{\infty} X(f) e^{j2\pi f t}\, df$$

- We denote the inverse transform as $x = \mathcal{F}^{-1}(X)$
- Sign in the exponent changes with respect to Fourier transform
- Can write as inner product $\Rightarrow x(t) = \langle X, e_{-t} \rangle$ $(e_{-t}(f) = e^{-j2\pi f t})$
- As in the case of the iDFT, this is not the most useful interpretation

Indeed, the inverse of the Fourier transform

Theorem
The inverse Fourier transform $\tilde{x}$ of the Fourier transform X of a given signal x is the given signal x

$$\tilde{x} = \mathcal{F}^{-1}(X) = \mathcal{F}^{-1}[\mathcal{F}(x)] = x$$

- Signals with Fourier transforms can be written as sums of oscillations

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi f t}\, df \approx (\Delta f) \sum_{n=\infty}^{\infty} X(f_n) e^{j2\pi f_n t}$$

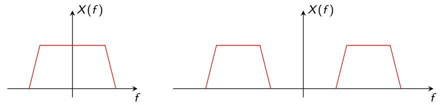
- This is conceptual, not literal (as was the case in discrete signals)

Frequency decomposition of a signal

- $X(f)$ determines the density of frequency f in the signal $x(t)$

$$x(t) \approx \sum_{n=\infty}^{\infty} (\Delta f) X(f_n) e^{j2\pi f_n t}$$

- It represents relative contribution (as opposed to absolute)



- Signal on left accumulates mass at low frequencies (changes slowly)
- Signal on right accumulates mass at high frequencies (changes fast)

Proof of inverse Fourier transform

Proof.

- We want to show $\Rightarrow \tilde{x} = \mathcal{F}^{-1}(X) = \mathcal{F}^{-1}[\mathcal{F}(x)] = x$. Use definitions
- From definition of inverse transform of $X \Rightarrow \tilde{x}(\tilde{t}) := \int_{-\infty}^{\infty} X(f) e^{j2\pi f\tilde{t}} df$
- From definition of transform of $x \Rightarrow X(f) := \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$
- Substituting expression for $X(f)$ into expression for $\tilde{x}(\tilde{t})$ yields

$$\tilde{x}(\tilde{t}) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \right] e^{j2\pi f\tilde{t}} df$$

- Repeating steps done for DFT and iDFT with integrals instead of sums

Proof of inverse Fourier transform

Proof.

- Exchange integration order to integrate first over f and then over t

$$\tilde{x}(\tilde{t}) = \int_{-\infty}^{\infty} x(t) \left[\int_{-\infty}^{\infty} e^{j2\pi f\tilde{t}} e^{-j2\pi ft} df \right] dt$$

- Pulled $x(t)$ out because it doesn't depend on k
- Innermost integral is the inner product between $e_{\tilde{t}}$ and e_t.

$$\int_{-\infty}^{\infty} e^{j2\pi f\tilde{t}} e^{-j2\pi ft} df = \langle e_{\tilde{t}}, e_t \rangle$$

- Up until now we repeated same steps we did for DFT and iDFT
- But we encounter a problem $\Rightarrow \langle e_{\tilde{t}}, e_t \rangle$ does not exist (infinity, oscillates)
- To exchange integration order, all integrals have to exist. But one doesn't
  - $\Rightarrow$ It is mathematically incorrect to interchange the order of integration

Proof of inverse Fourier transform

Proof.

- Replace infinite summation boundaries with finite summation boundaries

$$\tilde{x}(\tilde{t}) \overset{F\to\infty}{=} \int_{-\infty}^{\infty} x(t) \left[\int_{-F/2}^{F/2} e^{j2\pi f\tilde{t}} e^{-j2\pi ft} df \right] dt$$

- Eventually, we need to take $F \to \infty$, but not yet.
- All integrals exist now. Innermost one is a sinc (truncated exponential)

$$\int_{-F/2}^{F/2} e^{j2\pi f\tilde{t}} e^{-j2\pi ft} df = F\text{sinc}(\pi(t-\tilde{t})F)$$

- Substitute sinc for innermost integral on previous expression

$$\tilde{x}(\tilde{t}) \overset{F\to\infty}{=} \int_{-\infty}^{\infty} x(t) \left[F\text{sinc}(\pi(t-\tilde{t})F) \right] dt$$

Proof of inverse Fourier transform

Proof.

- take the limit formally $\Rightarrow \tilde{x}(\tilde{t}) = \lim_{F\to\infty} \int_{-\infty}^{\infty} x(t) \left[F\text{sinc}(\pi(t-\tilde{t})F) \right] dt$
- The sinc function is centered at time $t = \tilde{t}$
- The sinc becomes infinitely tall and thin as we take $F \to \infty$
- Can then take $x(\tilde{t})$ outside of the integral (only "meaningful" value)

$$\tilde{x}(\tilde{t}) = \lim_{F\to\infty} x(\tilde{t}) \int_{-\infty}^{\infty} F\text{sinc}(\pi(t-\tilde{t})F) dt$$

- The sinc function has unit integral $\Rightarrow \int_{-\infty}^{\infty} F\text{sinc}(\pi(t-\tilde{t})F) = 1$
- We then have $\tilde{x}(\tilde{t}) = x(\tilde{t})$ and $\tilde{x} = x$ as we wanted to show □

Fourier transform pairs

- Symmetry between transform and inverse $\Rightarrow$ Transform pairs
- Interpret given function z as signal. Fourier transform $X = \mathcal{F}(z)$ is

$$X(f) = \int_{-\infty}^{\infty} z(t) e^{-j2\pi ft} dt$$

- Conjugate z and interpet z^* as a transform. Inverse $x = \mathcal{F}^{-1}(z^*)$ is

$$x(t) = \int_{-\infty}^{\infty} z^*(f) e^{j2\pi ft} df = \left[\int_{-\infty}^{\infty} z(f) e^{-j2\pi ft} df \right]^*$$

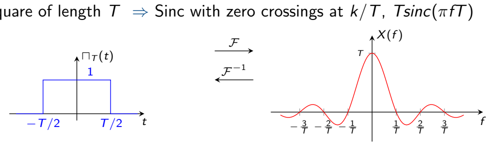
- Same integrals except for switch of integration index and argument

$$X(f) = x^*(t), \qquad \text{when } f = t$$

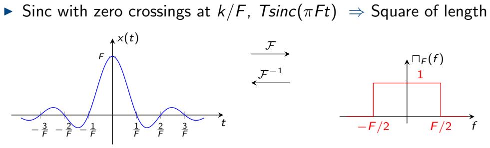
- X is transform of z and z is transform of $X^* \equiv x^*$ $\Rightarrow$ They are a pair
  - $\Rightarrow$ Conjugation unnecessary when signal and transform are real

The square pulse – sinc Fourier transform pair

- Square of length T $\Rightarrow$ Sinc with zero crossings at k/T, $T\text{sinc}(\pi fT)$



- Sinc with zero crossings at k/F, $T\text{sinc}(\pi Ft)$ $\Rightarrow$ Square of length F



- Transform of sinc pulse is difficult to compute through direct operation

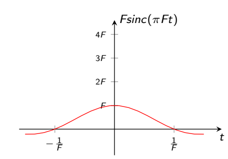
Delta function

Sequence of progressively taller sinc pulses

- Define the continuous time delta function as the limit of a sinc pulse

$$\delta(t) := \lim_{F\to\infty} F\text{sinc}(\pi Ft)$$

- Limit is $\delta(t) = \infty$ for $t = 0$
- But does not exist for other t
  - $\Rightarrow$ Oscillates between $\pm 1/\pi t$

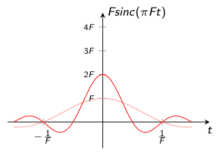


Sequence of progressively taller sinc pulses

- Define the continuous time delta function as the limit of a sinc pulse

$$\delta(t) := \lim_{F\to\infty} Fsinc(\pi F t)$$

- Limit is $\delta(t) = \infty$ for $t = 0$
- But does not exist for other t
  - $\Rightarrow$ Oscillates between $\pm 1/\pi t$

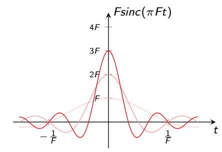


Sequence of progressively taller sinc pulses

- Define the continuous time delta function as the limit of a sinc pulse

$$\delta(t) := \lim_{F\to\infty} Fsinc(\pi F t)$$

- Limit is $\delta(t) = \infty$ for $t = 0$
- But does not exist for other t
  - $\Rightarrow$ Oscillates between $\pm 1/\pi t$

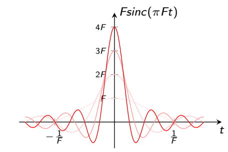


Sequence of progressively taller sinc pulses

- Define the continuous time delta function as the limit of a sinc pulse

$$\delta(t) := \lim_{F\to\infty} Fsinc(\pi F t)$$

- Limit is $\delta(t) = \infty$ for $t = 0$
- But does not exist for other t
  - $\Rightarrow$ Oscillates between $\pm 1/\pi t$

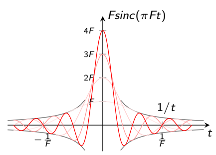


Sequence of progressively taller sinc pulses

- Define the continuous time delta function as the limit of a sinc pulse

$$\delta(t) := \lim_{F\to\infty} Fsinc(\pi F t)$$

- Limit is $\delta(t) = \infty$ for $t = 0$
- But does not exist for other t
  - $\Rightarrow$ Oscillates between $\pm 1/\pi t$

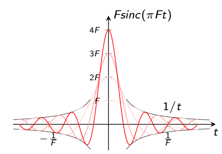


Sequence of progressively taller sinc pulses

- Define the continuous time delta function as the limit of a sinc pulse

$$\delta(t) := \lim_{F\to\infty} Fsinc(\pi F t)$$

- Limit is $\delta(t) = \infty$ for $t = 0$
- But does not exist for other t
  - $\Rightarrow$ Oscillates between $\pm 1/\pi t$



- On second thought, maybe we should use a different definition
- Intuitively, we want to say that the delta function is
  - $\Rightarrow$ Infinity for $t = 0$ $\Rightarrow$ $\delta(t) = \infty$ for $t = 0$
  - $\Rightarrow$ Null for all other t $\Rightarrow$ $\delta(t) = 0$ for $t \neq 0$
- But the question is what can we say mathematically? $\Rightarrow$ Integrate

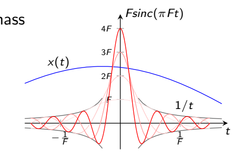
Limit of inner products

- Integrate the product of a signal with a sinc that is thin and tall
  - $\Rightarrow$ Recovers the value of the signal at time $t = 0$
- Since $x(0)$ multiplies most of sinc mass

$$\int_{-\infty}^{\infty} x(t) Fsinc(\pi F t) dt \approx x(0)$$

- Can write formally as

$$\lim_{F\to\infty} \int_{-\infty}^{\infty} x(t) Fsinc(\pi F t) dt = x(0)$$



- Observe that integral is the inner product of x with sinc. Then

$$\lim_{F\to\infty} \langle x, Fsinc(\pi F t)\rangle = x(0)$$

- Inner product of a signal with arbitrarily tall sinc is its value at zero

Delta function

- Define delta function as the entity δ that has this property. I.e., if

$$\langle x, \delta\rangle = x(0)$$

- for any signal x, we say that δ is a delta function
- In terms of integrals we write $\Rightarrow \int_{-\infty}^{\infty} x(t)\delta(t) dt = x(0)$
- Is the delta function a function? $\Rightarrow$ Of course not
- We say that δ is a distribution or generalized function
- Abstract entity without meaning until we pass through an integral
  - $\Rightarrow$ Can't observe directly, but can observe its effect on other signals
- Can define orthogonality and transforms of complex exponentials

Generalized orthogonality

Orthogonality of complex exponentials

- Consider complex exponentials of frequencies f and g
  - $\Rightarrow$ Frequency f $\Rightarrow e_f(t) = e^{j2\pi ft}$. Frequency g $\Rightarrow e_g(t) = e^{j2\pi gt}$
- We define their inner product $\langle e_f, e_g\rangle$ as the delta function $\delta(f-g)$

$$\langle e_f, e_g\rangle = \delta(f-g)$$

- This is a definition, not a derivation. We are accepting it to be true.
- If it is a definition: Does it make sense? What's its meaning?

It makes sense

- Complex exponentials don't have a mutual inner product.
- But truncated exponentials $e_{f,T}$ and e_{gT} do have a mutual product
  $\Rightarrow$ Multiply by $\sqcap_T$. Make signal null for $t > T/2$ and $t < T/2$
- Can write inner product of truncated signals as

$$\langle e_{fT}, e_{gT}\rangle := \int_{-T/2}^{T/2} e_f(t)e_g^*(t)dt = \int_{-T/2}^{T/2} e^{j2\pi ft}e^{-j2\pi gt}dt = \int_{-T/2}^{T/2} e^{j2\pi(f-g)t}dt$$

- Integral above resolves to a sinc with zero crossings at k/T

$$\langle e_{fT}, e_{gT}\rangle = T\text{sinc}[\pi(f-g)T]$$

- As $T \to \infty$ truncated signals approach non-truncated counterparts...
- ...and the sinc limit is our first attempt at defining $\delta(f-g)$
- Definition didn't work. But we are looking for sense, not meaning

What does it mean?

- Delta function is not observable directly, only after integration
- For an arbitrary given signal $X(f)$ we must have

$$\int_{-\infty}^{\infty} X(f)\langle e_{fT}, e_{gT}\rangle df = \int_{-\infty}^{\infty} X(f)\delta(f-g)df = X(g)$$

- Equivalently, we can write in terms of integrals

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} X(f)e^{j2\pi ft}e^{-j2\pi gt}\,dt\,df = X(g)$$

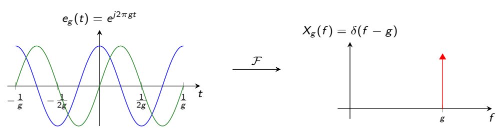
- OK, fine, but really, stop messing and tell us what it means
  $\Rightarrow$ When $f = g \Rightarrow \langle e_f, e_f\rangle = \infty$. When $f \neq g \Rightarrow \langle e_f, e_g\rangle = 0$
- Can use for intuitive reasoning, but not for mathematical derivations

Generalized Fourier transforms

Fourier transform of complex exponential

- Again, we can define, not derive, the Fourier transform of e_g
- Denote as $X_g := \mathcal{F}(e_g)$ the transform of e_g. We define X_g as

$$X_g(f) = \delta(f-g)$$



- We draw delta functions with an arrow pointing to the sky

It makes sense and it has meaning

- Does it make sense to have $X_g(f) = \delta(f-g)$
- Yes $\Rightarrow$ Transform definition consistent with orthogonality definition

$$X_g(f) = \langle e_g, e_f\rangle = \delta(f-g)$$

- Yes $\Rightarrow$ Definition is consistent with definition of inverse transform

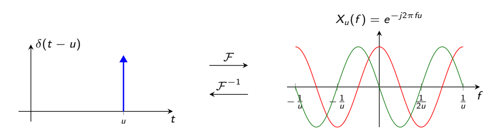
$$e_g(t) = \int_{-\infty}^{\infty} X_g(f)e^{j2\pi ft}df = \int_{-\infty}^{\infty}\delta(f-g)e^{j2\pi ft}df = e^{j2\pi gt}$$

- Making $X_g(f) = \delta(f-g)$ maintains Fourier analysis coherence
- Definition has clear, albeit, disappointingly trivial meaning
- Exponential of freq. g can be written as exponential of freq. g

Fourier transform of a shifted delta function

- Denote as X_u the transform of the shifted delta function $\delta(t-u)$
- This one we can compute $\Rightarrow$ Complex exponential of frequency u

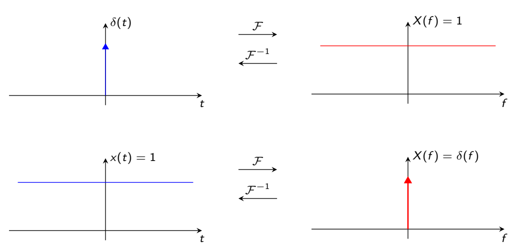
$$X_u(f) = \int_{-\infty}^{\infty}\delta(t-u)e^{-j2\pi ft}dt = e^{-j2\pi fu} = e_{-u}(f)$$



- It is the inverse we need to define as a delta function centered at u

The delta – constant transform pair

- When frequencies are null we have constants and unshifted deltas
- Transform of $x(t) = \delta(t) \Rightarrow X(f) = 1$. Transform of $x(t) = 1 \Rightarrow X(f) = \delta(f)$



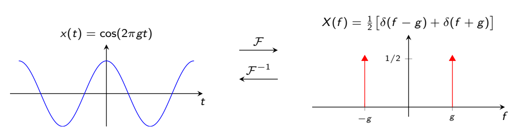
Fourier transform of a cosine

- To find Fourier transform of cosine write as difference of exponentials

$$\cos(2\pi gt) = \frac{1}{2}\left[e^{j2\pi gt} + e^{-j2\pi gt}\right]$$

- Since Fourier is a linear operator we transform each of the summands

$$X(f) = \frac{1}{2}\left[\delta(f-g) + \delta(f+g)\right]$$



- Pair of deltas of "height 1/2" at (opposite) frequencies $\pm g$

Properties of the Fourier transform

Three properties we already studied for the DFT

- Fourier transform is conjugate symmetric, linear, and conserves energy
- Transforms of real signals satisfy $\Rightarrow X(-k) = X^*(k)$
- Linearity $\Rightarrow \mathcal{F}(ax+by) = a\mathcal{F}(x) + b\mathcal{F}(y)$
- Energy $\Rightarrow \int_{-\infty}^{\infty} |x(t)|^2 dt = \|x\|^2 = \|X\|^2 = \int_{-\infty}^{\infty} |X(f)|^2 df$
- Not surprising, Fourier transform and DFT are conceptually identical
- Properties follow from properties of inner products and orthogonality
- Both transforms are projections on complex exponentials (inner product)
- And both project onto sets of orthogonal signals

Symmetry

Theorem
The Fourier transform $X = \mathcal{F}(x)$ of a real signal x is conjugate symmetric

$$X(-f) = X^*(f)$$

- For real signals only positive half of spectrum carries information
- Conjugate symmetry implies that $X(-f)$ and $X^*(f)$ are such that...
  - $\Rightarrow$ Real parts are equal $\Rightarrow \text{Re}(X(f)) = \text{Re}(X(-f))$
  - $\Rightarrow$ Imaginary parts are opposites $\Rightarrow \text{Im}(X(f)) = \text{Im}(X(-f))$
  - $\Rightarrow$ Moduli are equal $\Rightarrow |X(f)| = |X(-f)|$

Proof of symmetry property

Proof.

- Write the Fourier transform $X(-k)$ using its definition

$$X(-f) := \int_{-\infty}^{\infty} x(t) e^{-j2\pi(-f)t} dt$$

- When the signal is real, its conjugate is itself $\Rightarrow x(n) = x^*(n)$
- Conjugating a complex exponential $\Rightarrow$ changing the exponent's sign
- Can then rewrite $\Rightarrow X(-f) := \int_{-\infty}^{\infty} x^*(t) \left(e^{-j2\pi ft}\right)^* dt$
- Integration and multiplication can change order with conjugation

$$X(-f) = \left[\int_{-\infty}^{\infty} x^*(t) \left(e^{-j2\pi ft}\right)^* dt\right]^* = X^*(f) \qquad \square$$

Linearity

Theorem
The Fourier transform of a linear combination of signals is the linear combination of the respective Fourier transforms of the individual signals,

$$\mathcal{F}(ax+by) = a\mathcal{F}(x) + b\mathcal{F}(y).$$

Proof.

- Let $Z := \mathcal{F}(ax+by)$. From the Fourier transform definition

$$Z(f) = \int_{-\infty}^{\infty} \left[ax(t) + by(t)\right] e^{-j2\pi ft} dt$$

- Expand the product, reorder terms, identify transforms of x and y

$$Z(f) = a\int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt + b\int_{-\infty}^{\infty} y(t) e^{-j2\pi ft} dt = aX(f) + bY(f) \qquad \square$$

Energy conservation

Theorem (Parseval)
Let $X = \mathcal{F}(x)$ be the Fourier transform of signal x. The energies of x and X are the same, i.e.,

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \|x\|^2 = \|X\|^2 = \int_{-\infty}^{\infty} |X(f)|^2 df$$

- It follows that $X(f)$ is the energy density concentrated around f
- E.g., removing frequency component $\equiv$ remove corresponding energy

We omit proof as it is analogous to DFT case. Need to use finite integration region and take limit after exchanging order of integration. Not worth repeating.

Shift $\Leftrightarrow$ modulation

- Two more properties we didn't study for DFTs
  - $\Rightarrow$ They (sort of) hold for DFTs, but are difficult to explain
- Time shift $\Rightarrow$ multiplication by complex exponential in frequency
- Multiplication by complex exponential in time $\Rightarrow$ Shift in frequency
- Properties are dual of each other $\Rightarrow$ inverse transform symmetry
  - $\Rightarrow$ If one holds the other has to be true

Time shift

- Given signal x and shift τ define shifted signal $x_\tau \Rightarrow x_\tau = x(t-\tau)$
- Fourier transform of x is $X = \mathcal{F}(x)$. Transform of x_τ is $X_\tau = \mathcal{F}(x_\tau)$.

Theorem
A time shift of τ units in the time domain is equivalent to multiplication by a complex exponential of frequency $-\tau$ in the frequency domain

$$x_\tau = x(t-\tau) \iff X_\tau(f) = e^{-j2\pi f\tau} X(f)$$

- The phase of $X(f)$ changes, but the modulus remains the same

$$|X_\tau(f)| = \left|e^{-j2\pi f\tau} X(f)\right| = \left|e^{-j2\pi f\tau}\right| \times |X(f)| = |X(f)|$$

- Useful in signal detection $\Rightarrow$ Don't have to compare different shifts

Proof of time shift property

Proof.

- Shifted signal transform $\Rightarrow X_\tau(f) = \int_{-\infty}^{\infty} x(t-\tau) e^{-j2\pi ft} dt$
- Change of variables $u = t - \tau$. Separate exponent in two factors

$$X_\tau(f) = \int_{-\infty}^{\infty} x(u) e^{-j2\pi f(u+\tau)} du = \int_{-\infty}^{\infty} x(u) e^{-j2\pi f\tau} e^{-j2\pi fu} du$$

- Pull the term $e^{-j2\pi f\tau}$ out of the integral. Identify $X(f)$

$$X_\tau(f) = e^{-j2\pi f\tau} \int_{-\infty}^{\infty} x(u) e^{-j2\pi fu} du = e^{-j2\pi f\tau} X(f) \qquad \square$$

Modulation

- For signal x and freq. g define modulated signal $\Rightarrow x_g = e^{-j2\pi gt} x(t)$
- Fourier transform of x is $X = \mathcal{F}(x)$. Transform of x_g is $X_\tau = \mathcal{F}(x_g)$.

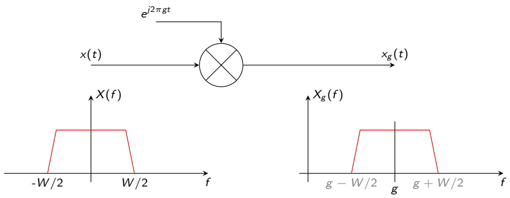
Theorem
A multiplication by a complex exponential of frequency g in the time domain is equivalent to a shift of g units in the frequency domain

$$x_g = e^{j2\pi gt} x(t) \iff X_g(f) = X(f-g)$$

- Dual of time shift result $\Rightarrow$ Proof not really necessary
- Principle behind transmission of signals on electromagnetic spectrum

Modulation of bandlimited signals

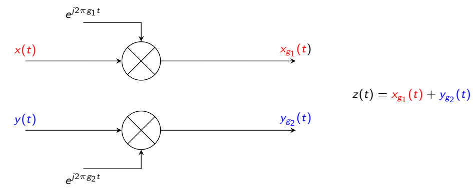
- Signal x has bandwidth W $\Rightarrow X(f) = 0$ for $f \notin [-W/2, W/2]$
- Multiplying by complex exponential shifts spectrum to the right
  $\Rightarrow$ Re-center spectrum at frequency g



- Can recover signal x by multiplying with conjugate frequency $e^{-j2\pi gt}$

Modulation of multiple bandlimited signals

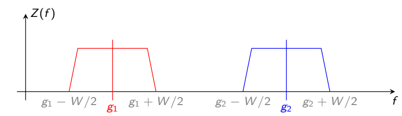
- Modulate two signals with bandwidth W using frequencies g_1 and g_2
  $\Rightarrow$ Spectrum of x recentered at g_1. Spectrum of y recentered at g_2



- Sum up to construct signal $z(t) = x_{g_1}(t) + y_{g_2}(t)$
  $\Rightarrow$ Can we recover x and y from mixed signal z? $\Rightarrow$ Yes

Spectrum of multiple modulated signals

- No spectral mixing if modulating frequencies satisfy $g_2 - g_1 > W$



- To recover x multiply by conjugate frequency $e^{-j2\pi g_1 t}$
- And eliminated all frequencies outside the interval $[-W/2, W/2]$
- To recover y multiply by conjugate frequency $e^{-j2\pi g_2 t}$
- And eliminated all frequencies outside the interval $[-W/2, W/2]$

Convolution

Convolution $\Leftrightarrow$ Product

- Both, Fourier transforms and DFTs are:
  $\Rightarrow$ Conjugate symmetric, linear, & conserve energy
- The Fourier transform also satisfies shift and modulation theorems
  $\Rightarrow$ They also (sort of) hold for DFTs (although we haven't shown)
  $\Rightarrow$ As they should, DFTs are close to Fourier transforms
- A sixth property of Fourier transforms, also sort of true for DFTs
  $\Rightarrow$ Convolution in time equivalent to multiplication in frequency

Convolution

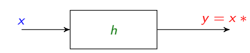
- Given signal x with values $x(t)$ and signal h with values $h(t)$
- Convolution of x with h is the signal $y = x * h$ with values

$$[x * h](t) = y(t) = \int_{-\infty}^{\infty} x(u)h(t-u)\,du$$

- Operation is commutative $\Rightarrow [x * h] \equiv [h * x]$

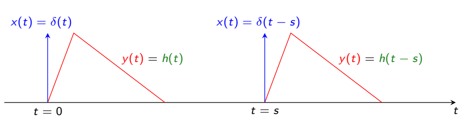
$$[h*x](t) = \int_{-\infty}^{\infty} h(u)x(t-u)\,du = \int_{-\infty}^{\infty} h(t-v)x(v)\,dv = [x*h](t)$$

- Still, prefer to interpret roles of x and h as asymmetric $\Rightarrow x$ hits h



Convolution with delta functions

- Convolution with $x(t) = \delta(t)$ $\Rightarrow y(t) = \int_{-\infty}^{\infty} \delta(u)h(t-u)\,du = h(t)$
- Hitting h with delta function produces convolution output $y \equiv h$



- Convolution with delayed delta $x(t) = \delta(t-s)$ ($u = s$ in integrand)

$$y(t) = \int_{-\infty}^{\infty} \delta(u-s)h(t-u)\,du = h(t-s)$$

- Hitting h with delayed delta produces delayed h as output

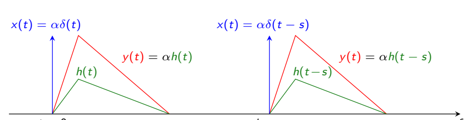
Convolution with scaled delta functions

- Convolution with scaled delta function $x(t) = \alpha\delta(t)$

$$y(t) = \int_{-\infty}^{\infty} \alpha\delta(u)h(t-u)\,du = \alpha\int_{-\infty}^{\infty} \delta(u)h(t-u)\,du = \alpha h(t)$$

- Convolution with scaled and delayed delta $x(t) = \alpha\delta(t-s)$

$$y(t) = \int_{-\infty}^{\infty} \alpha\delta(u-s)h(t-u)\,du = \alpha\int_{-\infty}^{\infty} \delta(u-s)h(t-u)\,du = \alpha h(t-s)$$



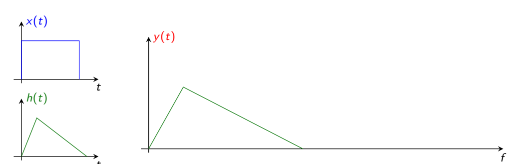
- Convolution with scaled and delayed delta is scaled and delayed h

Interpretation $\Rightarrow$ Scale, Shift, Sum (3S)

- Approximate convolution with Riemann sum (sampling at $u = u_n$)

$$y(t) = \int_{-\infty}^{\infty} x(u)h(t-u)\,du \approx T_s \sum_{n=-\infty}^{\infty} x(u_n)h(t-u_n)$$

- For each u_n $\Rightarrow$ Scale $h(t)$ by $x(u_n)$ to produce $x(u_n)h(t)$
  $\Rightarrow$ Shift to time u_n to produce $x(u_n)h(t-u_n)$
- Sum over all possible u_n $\Rightarrow$ integrate over all u, in the limit



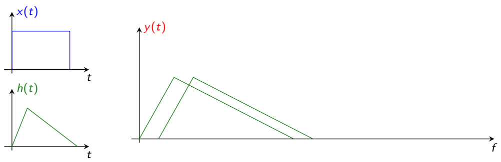
- Linear combination of shifted versions of h with coefficients $x(u)$

Interpretation $\Rightarrow$ Scale, Shift, Sum (3S)

- Approximate convolution with Riemann sum (sampling at $u = u_n$)

$$y(t) = \int_{-\infty}^{\infty} x(u)h(t-u)\,du \approx T_s \sum_{n=-\infty}^{\infty} x(u_n)h(t-u_n)$$

- For each u_n $\Rightarrow$ Scale $h(t)$ by $x(u_n)$ to produce $x(u_n)h(t)$
  $\Rightarrow$ Shift to time u_n to produce $x(u_n)h(t-u_n)$
- Sum over all possible u_n $\Rightarrow$ integrate over all u, in the limit



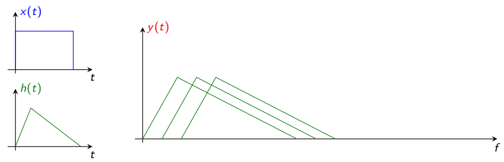
- Linear combination of shifted versions of h with coefficients $x(u)$

Interpretation $\Rightarrow$ Scale, Shift, Sum (3S)

- Approximate convolution with Riemann sum (sampling at $u = u_n$)

$$y(t) = \int_{-\infty}^{\infty} x(u)h(t-u)\,du \approx T_s \sum_{n=-\infty}^{\infty} x(u_n)h(t-u_n)$$

- For each u_n $\Rightarrow$ Scale $h(t)$ by $x(u_n)$ to produce $x(u_n)h(t)$
  $\Rightarrow$ Shift to time u_n to produce $x(u_n)h(t-u_n)$
- Sum over all possible u_n $\Rightarrow$ integrate over all u, in the limit



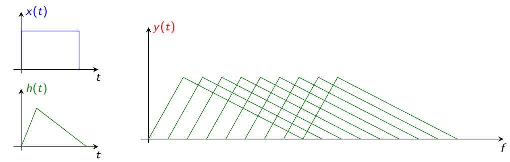
- Linear combination of shifted versions of h with coefficients $x(u)$

Interpretation $\Rightarrow$ Scale, Shift, Sum (3S)

- Approximate convolution with Riemann sum (sampling at $u = u_n$)

$$y(t) = \int_{-\infty}^{\infty} x(u)h(t-u)\,du \approx T_s \sum_{n=-\infty}^{\infty} x(u_n)h(t-u_n)$$

- For each u_n $\Rightarrow$ Scale $h(t)$ by $x(u_n)$ to produce $x(u_n)h(t)$
  $\Rightarrow$ Shift to time u_n to produce $x(u_n)h(t-u_n)$
- Sum over all possible u_n $\Rightarrow$ integrate over all u, in the limit



- Linear combination of shifted versions of h with coefficients $x(u)$

Time convolution $\equiv$ Frequency multiplication

Theorem (Convolution theorem)

Given signals x and y with transforms $X = \mathcal{F}(x)$ and $Y = \mathcal{F}(y)$. The Fourier transform $Z = \mathcal{F}(z)$ of the convolved signal $z = x * y$ is the product $Z = XY$

$$z = x * y \quad \Longleftrightarrow \quad Z = XY$$

- Convolution in time domain $\equiv$ to multiplication in frequency domain
- When we convolve signals x and y in the time domain
  $\Rightarrow$ Their transforms are multiplied in the frequency domain
- When we multiply two transforms in the frequency domain
  $\Rightarrow$ The signals get convolved in the time domain

Proof of convolution theorem

Proof.

- Use the definition of Fourier transform to write the transform of Z as

$$Z(f) = \int_{-\infty}^{\infty} z(t)e^{-j2\pi ft}\,dt$$

- Use the definition of convolution to write the signal z as

$$z(t) = \int_{-\infty}^{\infty} x(u)h(t-u)\,du$$

- Substitute the expression for $z(t)$ into expression for $Z(f)$

$$Y(f) = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x(u)h(t-u)\,du\right) e^{-j2\pi ft}\,dt$$

Proof of convolution theorem

Proof.

- Rewrite the nested integral as a double integral

$$Y(f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(u)h(t-u)e^{-j2\pi ft}\,du\,dt$$

- Make the change of variables $v = t - u$ and write

$$Y(f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(u)h(v)e^{-j2\pi f(u+v)}\,du\,dt$$

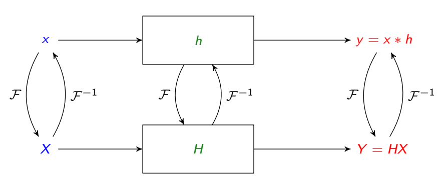
- Write $e^{-j2\pi f(u+v)} = e^{-j2\pi fu}e^{-j2\pi fv}$ and reorder terms to obtain

$$Y(f) = \left(\int_{-\infty}^{\infty} x(u)e^{-j2\pi fu}\,du\right)\left(\int_{-\infty}^{\infty} h(v)e^{-j2\pi fv}\,dv\right)$$

- Factors on the right are the Fourier transforms $X(f)$ and $Y(f)$ $\square$

System equivalence

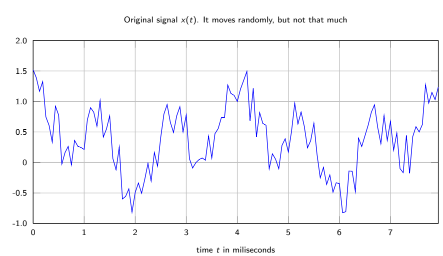
- Convolution in time equivalent to multiplication in frequency
  $\Rightarrow$ Is this useful in any way? $\Rightarrow$ Certainly, few facts are more useful
- Convolution theorem implies that these two systems are equivalent



- The lower path for design, the upper path for implementation

The signal and the noise

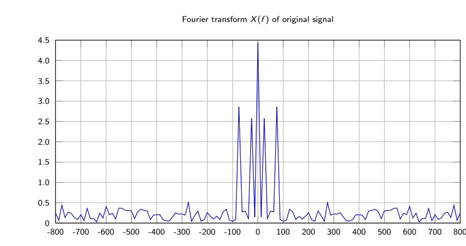
- There is signal and noise, but what is signal and what is noise?
- We already know answer $\Rightarrow$ Signal discernible in frequency domain



-

The signal and the noise

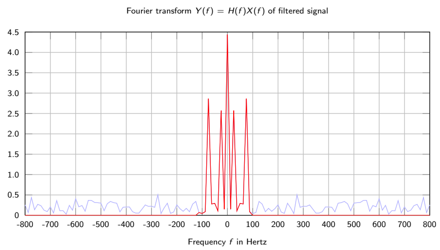
- There is signal and noise, but what is signal and what is noise?
- We already know answer $\Rightarrow$ Signal discernible in frequency domain



- Filter out all frequencies above 100Hz (and below -100Hz)

Noise removal – Low pass filter design

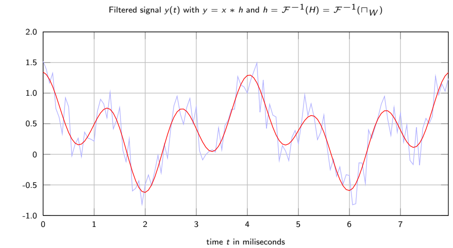
- Multiply spectrum with low pass filter $H(f) = \sqcap_W(f)$ with $W = 200$Hz
  - $\Rightarrow$ Only frequencies between $\pm W/2 = \pm 100$Hz are retained



- This spectral operation does separate signal from noise

Noise removal – Low pass filter design

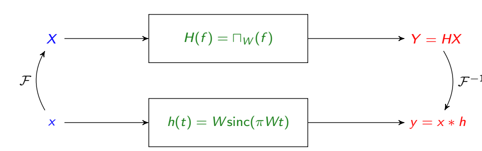
- Multiply spectrum with low pass filter $H(f) = \sqcap_W(f)$ with $W = 200$Hz
  - $\Rightarrow$ Only frequencies between $\pm W/2 = \pm 100$Hz are retained



- This spectral operation does separate signal from noise

Noise removal – Low pass filter implementation

- We can implement filtering in the frequency domain
  - $\Rightarrow$ Sample $\Rightarrow$ DFT $\Rightarrow$ Multiply by $H(f) = \sqcap_W(f)$ $\Rightarrow$ iDFT



- We can also implement filtering in the time domain
  - $\Rightarrow$ Inverse transform of $\sqcap_W(f)$ is $h(t) = W\text{sinc}(\pi W t)$
  - $\Rightarrow$ Sample (or not) $\Rightarrow$ Implement convolution with $h(t)$

Sampling

Alejandro Ribeiro
Dept. of Electrical and Systems Engineering
University of Pennsylvania
aribeiro@seas.upenn.edu
http://www.seas.upenn.edu/users/~aribeiro/

February 19, 2015

Discrete time signals

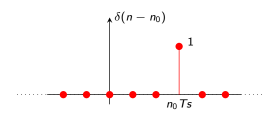
Discrete time signals

- To infinity, but no beyond $\Rightarrow$ Discrete but infinite time index $n \in \mathbb{Z}$.
- Discrete time signal x is a function mapping $\mathbb{Z}$ to complex value $x(n)$

$$x : \mathbb{Z} \to \mathbb{C} \qquad \text{(values } x(n) \text{ can be, often are, real)}$$

- Sampling time T_s is implicit. Time elapsed from sample n to $n+1$
- So is sampling frequency $f_s = 1/T_s$
- E.g., a shifted delta function $\delta(n - n_0)$ has a spike at time $n = n_0$

$$\delta(n - n_0) = \begin{cases} 1 & \text{if } n = n_0 \\ 0 & \text{else} \end{cases}$$



- Signal continuous to plus and minus infinity (unlike discrete signals)

Inner product and energy

- Given two signals x and y define the inner product of x and y as

$$\langle x, y \rangle := \sum_{n=-\infty}^{\infty} x(n) y^*(n)$$

- Projection of x on y. How much of x falls in y direction.
- How much x and y are like each other $\Rightarrow$ orthogonality $\equiv$ unrelated
- Define the energy of the signal as the inner product with itself

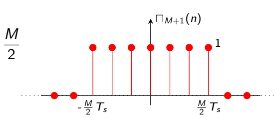
$$\|x\|^2 := \langle x, y \rangle = \sum_{n=-\infty}^{\infty} |x(n)|^2 = \sum_{n=-\infty}^{\infty} |x_R(n)|^2 + \sum_{n=-\infty}^{\infty} |x_I(n)|^2$$

- Sums extend to plus and minus infinity (they are series, not sums)
  - $\Rightarrow$ Inner product may not exist. Energy may be infinite

Energy and inner products of pulses

- Define square pulse of odd length $M+1$ as signal $\sqcap_{M+1}$ with values

$$\sqcap_{M+1}(n) = 1 \quad \text{if } -\frac{M}{2} \le n \le \frac{M}{2}$$
$$\sqcap_{M+1}(n) = 0 \quad \text{else } M \le n$$



- To compute energy of the pulse we just evaluate the definition

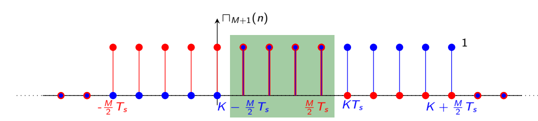
$$\|\sqcap_{M+1}\|^2 := \sum_{n=-\infty}^{\infty} |\sqcap_{M+1}(n)|^2 = \sum_{n=-M/2}^{M/2} (1)^2 = M+1$$

- Can normalize for unit energy as we did for discrete signal case
- But we rather not, as we did for continuous time (to let M grow)

Inner product of a pulse and a shifted pulse

- Inner product of pulse $\sqcap_{M+1}(n)$ and shifted pulse $\sqcap_{M+1}(n-K)$

$$\left\langle \sqcap_{M+1}(n), \sqcap_{M+1}(n-K) \right\rangle = \sum_{n=-\infty}^{\infty} \sqcap_{M+1}(n) \sqcap_{M+1}(n-K)$$



- For shifts $0 \le K \le M+1$, signals overlap for $K - M/2 \le n \le M/2$

$$\left\langle \sqcap_{M+1}(n), \sqcap_{M+1}(n-K) \right\rangle = \sum_{n=K-M/2}^{M/2} (1)(1) = (M+1) - K$$

- Proportional to overlap $\Rightarrow$ how much pulses "are like each other"

Discrete time Fourier transform

The discrete time Fourier transform (DTFT)

- The DTFT of discrete signal x is the function $X : \mathbb{R} \to \mathbb{C}$ with values

$$X(f) := T_s \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n T_s}$$

- Denote as $X = \mathcal{F}(x)$. Argument f is continuous and called frequency
- Sum need not exist $\Rightarrow$ Not all discrete time signals have a DTFT
- Definition depends on sampling time T_s. Facilitates connections later
- Fourier transform (FT) has continuous input and continuous output
- DFT is also well matched $\Rightarrow$ It has discrete input and discrete output
- DTFT is mismatched $\Rightarrow$ It has discrete input but continuous output
  - $\Rightarrow$ A little odd, but of little consequence

DTFT is also an inner product

- Define e_{fT_s} with values $e_{fT_s}(n) = T_s e^{j2\pi f n T_s}$. Write as inner product

$$X(f) = \langle x, e_{fT_s} \rangle = T_s \sum_{n=-\infty}^{\infty} x(n) e^{*}_{fT_s}(n)$$

- As in the case of the FT and the DFT, the DTFT value $X(f)$:
  - $\Rightarrow$ Is the projection of x onto discrete oscillation of freq. f
  - $\Rightarrow$ Measures how much $x(n)$ resembles discrete oscillation of freq. f
- Conceptually identical to FT & DFT $\Rightarrow$ Why a third definition?
  - $\Rightarrow$ All three, discrete time, discrete, and continuous signals exist
  - $\Rightarrow$ Deep connections between FT and DTFT and DTFT and DFT
- Analytical tool (as the FT). Not a computational tool (as the DFT)

Periodicity of DTFT

Theorem

The DFTF $X = \mathcal{F}(x)$ of discrete time signal x is periodic with period f_s

$$X(f + f_s) = X(f), \qquad \text{for all } f \in \mathbb{R}.$$

- Any frequency interval of length f_s contains all DTFT information
  - $\Rightarrow$ We will use the canonical set $\Rightarrow f \in [-f_s/2, f_s/2]$
- For sampling time T_s, freqs. larger than $f_s/2$ have no physical meaning
  - $\Rightarrow$ Frequency $-f$ is (more or less) the same as frequency f

Proof of periodicity property

Proof.

- Use the DTFT definition to write $X(f + f_s)$ as

$$X(f + f_s) = T_s \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi (f+f_s) n T_s}$$

- Separate the complex exponential in two factors

$$X(f + f_s) = T_s \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n T_s} e^{-j2\pi f_s n T_s}$$

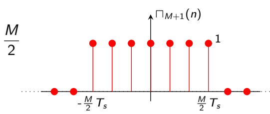
- Use $f_s T_s = 1$ in last factor $\Rightarrow e^{-j2\pi f_s n T_s} = e^{-j2\pi n} = \left(e^{j2\pi}\right)^{-n} = 1$
- Substitute in previous expression and observe definition of DTFT

$$X(f + f_s) = T_s \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n T_s} = X(f) \qquad \square$$

DTFT of a square pulse

- Consider square pulse of odd length $M + 1$

$$\sqcap_{M+1}(n) = 1 \qquad \text{if } -\frac{M}{2} \leq n \leq \frac{M}{2}$$
$$\sqcap_{M+1}(n) = 0 \qquad \text{else } M \leq n$$



- To compute the pulse DTFT $X = \mathcal{F}(\sqcap_{M+1})$ evaluate the definition

$$X(f) = T_s \sum_{n=-\infty}^{\infty} \sqcap_{M+1}(n) e^{-j2\pi f n T_s} = T_s \sum_{n=-M/2}^{M/2} e^{-j2\pi f n T_s}$$

- Write down the individual elements of the sum to express DTFT as

$$\frac{X(f)}{T_s} = e^{j2\pi f(-\frac{M}{2})T_s} + e^{j2\pi f(-\frac{M}{2}+1)T_s} + \ldots + e^{j2\pi f(\frac{M}{2}-1)T_s} + e^{j2\pi f(\frac{M}{2})T_s}$$

DTFT of a square pulse (computation, 1 of 2)

- Multiply by $e^{j2\pi f(\frac{1}{2})T_s}$ and $e^{j2\pi f(-\frac{1}{2})T_s}$ to write the equalities

$$e^{j2\pi f(\frac{1}{2})T_s}\frac{X(f)}{T_s} = e^{j2\pi f(-\frac{M}{2}+\frac{1}{2})T_s} + e^{j2\pi f(-\frac{M}{2}+\frac{3}{2})T_s} + \ldots + e^{j2\pi f(\frac{M}{2}-\frac{1}{2})T_s} + e^{j2\pi f(\frac{M}{2}+\frac{1}{2})T_s}$$
$$e^{-j2\pi f(\frac{1}{2})T_s}\frac{X(f)}{T_s} = e^{j2\pi f(-\frac{M}{2}-\frac{1}{2})T_s} + e^{j2\pi f(-\frac{M}{2}+\frac{1}{2})T_s} + \ldots + e^{j2\pi f(\frac{M}{2}-\frac{3}{2})T_s} + e^{j2\pi f(\frac{M}{2}-\frac{1}{2})T_s}$$

- First term in first row = second term in second row
- Second term in first row = third term in second row (unseen)

⋮

- Penultimate term in first row = last term in second row
- Subtracting second row from first row only two terms survive
  - $\Rightarrow$ The last term in the first row and the first term in the second row

DTFT of a square pulse (computation, 1 of 2)

- Multiply by $e^{j2\pi f(\frac{1}{2})T_s}$ and $e^{j2\pi f(-\frac{1}{2})T_s}$ to write the equalities

$$e^{j2\pi f(\frac{1}{2})T_s}\frac{X(f)}{T_s} = e^{j2\pi f(-\frac{M}{2}+\frac{1}{2})T_s} + e^{j2\pi f(-\frac{M}{2}+\frac{3}{2})T_s} + \ldots + e^{j2\pi f(\frac{M}{2}-\frac{1}{2})T_s} + e^{j2\pi f(\frac{M}{2}+\frac{1}{2})T_s}$$
$$e^{-j2\pi f(\frac{1}{2})T_s}\frac{X(f)}{T_s} = e^{j2\pi f(-\frac{M}{2}-\frac{1}{2})T_s} + e^{j2\pi f(-\frac{M}{2}+\frac{1}{2})T_s} + \ldots + e^{j2\pi f(\frac{M}{2}-\frac{3}{2})T_s} + e^{j2\pi f(\frac{M}{2}-\frac{1}{2})T_s}$$

- First term in first row = second term in second row
- Second term in first row = third term in second row (unseen)

⋮

- Penultimate term in first row = last term in second row
- Subtracting second row from first row only two terms survive
  - $\Rightarrow$ The last term in the first row and the first term in the second row

DTFT of a square pulse (computation, 2 of 2)

- Implementing the subtraction results in the equality

$$\frac{X(f)}{T_s}\left[e^{j2\pi f(\frac{1}{2})T_s} - e^{-j2\pi f(\frac{1}{2})}T_s\right] = e^{j2\pi f(\frac{M}{2}+\frac{1}{2})T_s} - e^{j2\pi f(-\frac{M}{2}-\frac{1}{2})T_s}$$

- Complex exponentials are conjugate. Subtraction cancels real parts
- We keep imaginary parts only, which are sines

$$\frac{X(f)}{T_s}\left[2j\sin\left(2\pi f\left(\frac{1}{2}\right)T_s\right)\right] = 2j\sin\left(2\pi f\left(\frac{M+1}{2}\right)T_s\right)$$

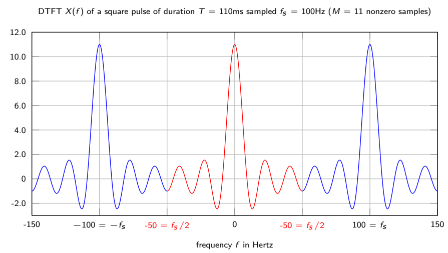
- Solve for $X(f)$ and simplify terms. Pulse length $T = (M+1)T_s$

$$X(f) = T_s\frac{\sin\left(\pi f\,(M+1)\,T_s\right)}{\sin\left(\pi f T_s\right)} = T_s\frac{\sin\left(\pi f T\right)}{\sin\left(\pi f T_s\right)}$$

- A slow sine over a fast sine $\Rightarrow$ not unlike a sinc pulse

Evaluation of the DTFT of a square pulse

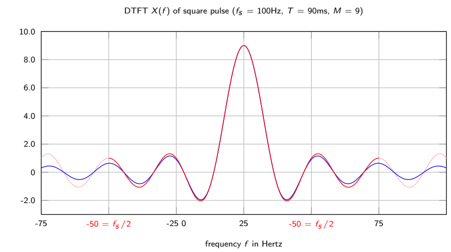
- Sampling freq. $f_s = 100$Hz. Pulse length in time $T = 110$ms pulse
  $\Rightarrow$ Resulting in $M+1 = 11$ nonzero samples



- DTFT is periodic, as we know it should. Focus on $f \in [-fs/2, f_s/2]$

The DTFT of a square pulse and the sinc pulse

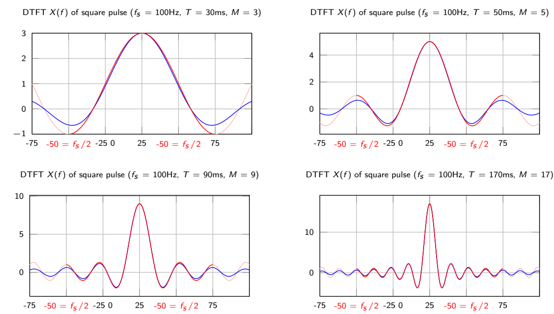
- Similar to the sinc pulse $\Rightarrow T\frac{\sin(\pi f T)}{\pi f T} = T\text{sinc}(\pi f T)$
- Fourier transform of unsampled pulse



- Some difference for f close to $\pm f_2/2$. Also, sinc is not periodic

Pulses of different length

- As the pulse widens, the DTFT concentrates. Same as FT and DFT
- As pulse widens difference with FT of continuous time pulse diminishes



The FT and the DTFT

- Interpret signal $x(n)$ as samples $x_C(nT_s)$ of continuous signal $x_C(t)$
- DTFT $X = \mathcal{F}(x)$ is Riemann sum approximation of FT $X_C = \mathcal{F}(x_C)$

$$X_C(f) = \int_{-\infty}^{\infty} x_C(t)e^{-j2\pi ft}dt \approx T_s \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi fnT_s} = X(f)$$

- Only frequencies between $\pm f_s/2$ have meaning in DTFT $\Rightarrow$ Chop
- FT $X_C(f)$ $\Rightarrow$ sample in time, chop in frequency $\Rightarrow$ DTFT $X(f)$

The DTFT and the DFT

- Chop x to $n \in [0, N-1]$ $\Rightarrow$ Discrete signal x_D with DFT $X_D = \mathcal{F}(x_D)$
- If elements discarded from x are small

$$X(f) = T_s \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi fnT_s} \approx T_s \sum_{n=0}^{N-1} x_D(n)e^{-j2\pi fnT_s}$$

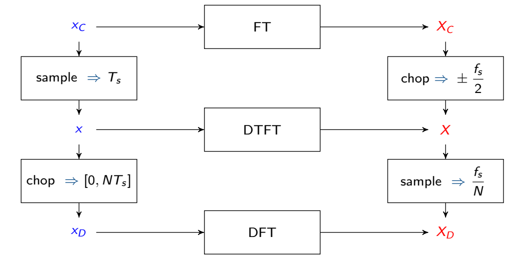
- True for all frequencies f. Sample in frequency at $f = (k/N)f_s$

$$X\left(\frac{k}{N}f_s\right) \approx T_s \sum_{n=0}^{N-1} x_D(n)e^{-j2\pi(k/N)f_snT_s} = T_s \sum_{n=0}^{N-1} x_D(n)e^{-j2\pi kn/N} = T_s\sqrt{N}X_D(k)$$

- DTFT $\Rightarrow$ Chop in time, sample in frequency $\Rightarrow$ DFT

The FT, the DTFT, and the DFT

- The DTFT bridges FT and DFT by dual sample and chopping



- The argument was careless though $\Rightarrow$ We will probe deeper

Inverse discrete time Fourier transform

The inverse (i)DTFT

- The iDTFT x of DTFT X, is the discrete time signal with elements

$$x(n) := \int_{-f_s/2}^{f_s/2} X(f)e^{j2\pi fnT_s}\,df$$

- We denote $x = \mathcal{F}^{-1}(X)$. Sampling time T_s (freq. f_s) implicit in X
- Sign in exponent changes with respect to DTFT.
- DTFT is an indefinite sum but iDTFT is a definite integral
  $\Rightarrow$ DTFT mismatch. Odd, but of little consequence
- Since DTFT X is periodic, any interval of width f_s does it. E.g.

$$x(n) = \int_{-f_s/2}^{f_s/2} X(f)e^{j2\pi fnT_s}\,df = \int_0^{f_s} X(f)e^{j2\pi fnT_s}\,df$$

Indeed, the iDTFT is the inverse of the DTFT

Theorem
The iDTFT $\tilde{x}$ of the DTFT X of the discrete time signal x is the signal x

$$\tilde{x} = \mathcal{F}^{-1}(X) = \mathcal{F}^{-1}[\mathcal{F}(x)] = x.$$

- What a surprise. It's getting tired. But this is the last one.
- As usual, discrete time signals can be written as sums of oscillations

$$x(n) = \int_{-f_s/2}^{f_s/2} X(f)e^{j2\pi fnT_s}\,df \approx (\Delta f) \sum_{n=-N/2}^{N/2} X(f_k)e^{j2\pi f_k nT_s}$$

- Conceptual; cf. continuous signals. Not literal; cf. discrete signals.

Proof of inverse Fourier transform

Proof.

- We want to show $\Rightarrow \tilde{x} = \mathcal{F}^{-1}(X) = \mathcal{F}^{-1}[\mathcal{F}(x)] = x$. Use definitions
- Definition of inverse transform of $X \Rightarrow \tilde{x}(\tilde{n}) := \int_{-f_s/2}^{f_s/2} X(f) e^{j2\pi f \tilde{n} T_s} df$
- From definition of transform of $x \Rightarrow X(f) := T_s \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n T_s}$
- Substituting expression for $X(f)$ into expression for $\tilde{x}(\tilde{n})$ yields

$$\tilde{x}(\tilde{n}) = \int_{-f_s/2}^{f_s/2} \left[T_s \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n T_s} \right] e^{j2\pi f \tilde{n} T_s} df$$

- Same as done for iDFT and iFT but with one integral and one sum

Proof of inverse Fourier transform

Proof.

- Exchange integration with sum $\Rightarrow$ Integrate first over f, then sum over n

$$\tilde{x}(\tilde{n}) = T_s \sum_{n=-\infty}^{\infty} x(n) \left[\int_{-f_s/2}^{f_s/2} e^{j2\pi f \tilde{n} T_s} e^{-j2\pi f n T_s} df \right]$$

- Pulled $x(n)$ out because it doesn't depend on f
- Up until now we repeated steps we already did for iDFT and iFT
  $\Rightarrow$ They worked for iDFT but didn't for iFT $\Rightarrow$ They work here.
- The innermost integral we have computed repeatedly $\Rightarrow$ It's a sinc

$$\int_{-f_s/2}^{f_s/2} e^{j2\pi f \tilde{n} T_s} e^{-j2\pi f n T_s} df = f_s \text{sinc}(\pi f_s (n - \tilde{n}) T_s) = f_s \text{sinc}(\pi (n - \tilde{n}))$$

- We used $f_s T_s = 1$ in second equality. Recall that n and $\tilde{n}$ are discrete

Proof of inverse Fourier transform

Proof.

- Evaluate sinc for $n = \tilde{n} \Rightarrow f_s \text{sinc}(\pi(n - \tilde{n})) = f_s$ because $\text{sinc}(0) = 1$
- Evaluate sinc for $n \neq \tilde{n} \Rightarrow f_s \text{sinc}(\pi(n - \tilde{n})) = 0$ because $\text{sinc}(k\pi) = 0$
- Lucky for us, the innermost integral was a delta function in disguise

$$\int_{-f_s/2}^{f_s/2} e^{j2\pi f \tilde{n} T_s} e^{-j2\pi f n T_s} df = f_s \delta(n - \tilde{n})$$

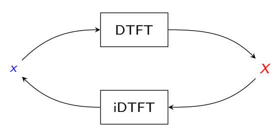
- Substituting in expression for $\tilde{x}(\tilde{n})$, only one term in sum is not null

$$\tilde{x}(\tilde{n}) = T_s f_s \sum_{n=-\infty}^{\infty} x(n) \delta(n - \tilde{n}) = x(\tilde{n})$$

- Also used $f_s T_s = 1$. Since we have $\tilde{x}(\tilde{n}) = x(\tilde{n})$ for all $\tilde{n} \Rightarrow \tilde{x} \equiv x$ □

From time to frequency and back

- If a discrete signal x has a DTFT X, its DTFT has an iDTFT
  $\Rightarrow$ The iDTFT of the DTFT X recovers original signal x
- The DTFT is a transformation without loss of information
  $\Rightarrow$ Can always come back from frequency domain to time domain



- True of DFT–iDFT and FT–iFT as well. Hadn't need to mention yet

DTFT of a constant

The DTFT of a constant

- Discrete time constant x has value $x(n) = 1$ for all n. The DTFT is

$$X(f) = T_s \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n T_s} = T_s \sum_{n=-\infty}^{\infty} e^{-j2\pi f n T_s}$$

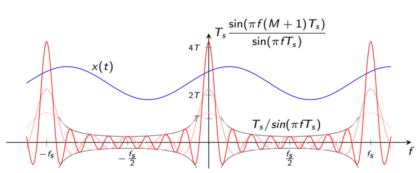
- It does not exist. For $n = 0$, $X(f) \to \infty$, for other n oscillates
- We know how to solve this problem $\Rightarrow$ Use delta function
- Write constant as pulse limit. DTFT of pulse we saw is ratio of sines
- Then, can think of writing DTFT of constant as the limit

$$X(f) = \lim_{M \to \infty} T_s \sum_{n=-M/2}^{M/2} e^{-j2\pi f n T_s} = \lim_{M \to \infty} T_s \frac{\sin(\pi f (M+1) T_s)}{\sin(\pi f T_s)}$$

- Except that it is this limit the one that does not exist

The limit of the DTFT of a square pulse

- As M grows, DTFT grows and narrows around $f = 0$. And $f = \pm k f_s$
  $\Rightarrow$ But it doesn't decrease for other frequencies



- But when multiplying by $Y(f)$ and integrating we recover $Y(0)$

$$\lim_{M \to \infty} \int_{-f_s/2}^{f_s/2} Y(f) T_s \frac{\sin(\pi f (M+1) T_s)}{\sin(\pi f T_s)} df = Y(0)$$

- Define (already did) delta function as the entity with this property

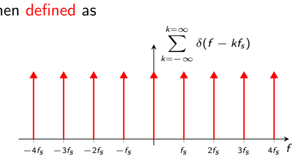
The Dirac train

- The delta function δ is a generalized function such that for all Y

$$\int_{-\infty}^{\infty} Y(f) \delta(f) df = Y(0)$$

- We can then *define* the DTFT of a constant as a delta function
- Almost correct, but observe that we also have peaks at $f = \pm k f_s$
- The DTFT of a constant is then defined as

$$X(f) = \sum_{k=-\infty}^{k=\infty} \delta(f - k f_s)$$



- We call this signal a train of deltas, a Dirac train, or a Dirac comb

What it means? Does it make sense?

- Informally $\Rightarrow \delta(f) = \infty$ for $f = 0$, $f = \pm f_s$, $f = \pm 2 f_s$, ...
  $\Rightarrow \delta(f) = 0$ for all other f
- Mathematically, only has sense after multiplication and integration

$$\int_{-\infty}^{\infty} Y(f) X(f) df = \int_{-\infty}^{\infty} Y(f) \sum_{k=-\infty}^{k=\infty} \delta(f - k f_s) df = \sum_{k=-\infty}^{k=\infty} Y(f - k f_s)$$

- Recovers the values of $Y(f)$ at the points where the train has spikes
- In particular, the iDTFT recovers the constant

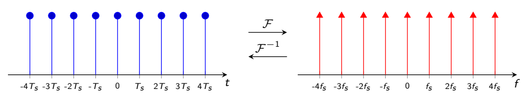
$$\int_{-f_s/2}^{f_s/2} X(f) e^{j2\pi f n T_s} df = \int_{-f_s/2}^{f_s/2} \sum_{k=-\infty}^{k=\infty} \delta(f - k f_s) e^{j2\pi f n T_s} df = e^{j2\pi 0 n T_s} = 1$$

- Definition makes sense $\Rightarrow$ Preserves consistency of DTFT analyses

Fourier transform of a Dirac train

The constant - Dirac train non-pair

- DTFT of a constant is a Dirac train $\Rightarrow$ suspiciously similar

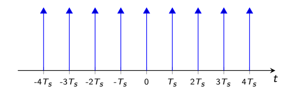


- Can we use duality to say the FT of a train is another train?
  $\Rightarrow$ Not quite. Left signal is discrete. Right signal is continuous
- Not a transform pair $\Rightarrow$ Can't define Dirac train in discrete time
  $\Rightarrow$ Definition of delta functions relies on integration
- But we are on to something

A Dirac train in the time domain

- For continuous time index t define continuous signal x as

$$x_C(t) = T_s \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$



- This signal is a Dirac train in time. Not a discrete time constant
- Being continuous, the Dirac train has a Fourier transform X_C

$$X_C(f) = \int_{-\infty}^{\infty} x_C(t) e^{-j2\pi ft}\, dt = \int_{-\infty}^{\infty} \left[T_s \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \right] e^{-j2\pi ft}\, dt$$

- Can be related to the DTFT of a discrete time constant

DTFT of a constant $\equiv$ FT of a Dirac train

- Exchange order of sum and integration, use delta function definition

$$X_C(f) = T_s \sum_{n=-\infty}^{\infty} \left[\int_{-\infty}^{\infty} \delta(t - nT_s) e^{-j2\pi ft}\, dt \right] = T_s \sum_{n=-\infty}^{\infty} e^{-j2\pi fnT_s}$$

- The sum on the right is the DTFT of a constant

$$X(f) = T_s \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi fnT_s} = T_s \sum_{n=-\infty}^{\infty} e^{-j2\pi fnT_s}$$

- The DTFT of a constant and the FT of a Dirac train coincide

$$X_C(f) = X(f) = \sum_{k=-\infty}^{\infty} \delta(t - kf_s)$$

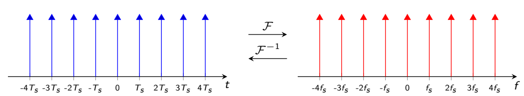
- Both are a Dirac trains in frequency with spacing f_s

The Dirac train - Dirac train FT pair

- FT of Dirac train with spacing T_s is a Dirac train with spacing f_s

$$x_C(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \quad \Longleftrightarrow \quad X_C(f) = \sum_{k=-\infty}^{\infty} \delta(t - kf_s)$$

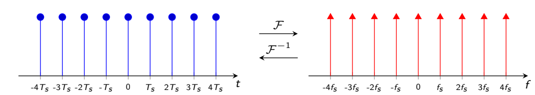
- The set of Dirac trains is an invariant class with respect to the FT



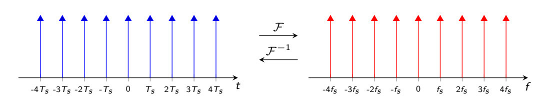
- This is a Fourier transform pair because both are continuous signals

Fundamentally different but equal

- Discrete time constant sampled at T_s $\Rightarrow$ DTFT $\Rightarrow$ Dirac train spaced f_s



- Dirac train spaced every T_s $\Rightarrow$ FT $\Rightarrow$ Dirac train spaced every f_s



- Discrete time constant fundamentally different from continuous time train
- Thus, DFTF of constant fundamentally different from FT of Dirac train
- But they coincide $\Rightarrow$ Something deeper is at play here

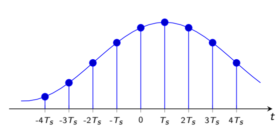
Sampling

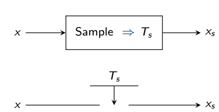
Sampling

- Consider continuous time signal x and sampling time T_s (freq. f_s)
- The sampled signal x_s is a discrete time signal with values

$$x_s(n) = x(nT_s)$$

- Creates discrete time signal x_s from continuous time signal x
- We've been doing this since first day. We want to understand it now
  $\Rightarrow$ Information lost from x when discarding all but samples $x(nT_s)$?





Sampling as multiplication by a Dirac train

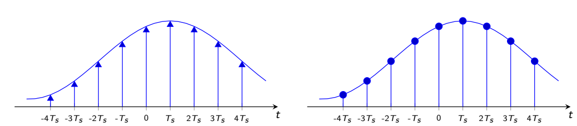
- Equivalently, we represent sampling as multiplication by a Dirac train

$$x_\delta(t) = x(t) \times T_s \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

- Indeed, since the only value that is relevant for $\delta(t - nT_s)$ is $x(nT_s)$

$$x_\delta(t) = T_s \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s)$$

- We can construct x_s if given x_δ and construct x_δ if given x_s

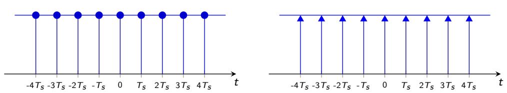


DTFT & FT of sampled signals coincide

Theorem
The DTFT $X_s = \mathcal{F}(x_s)$ of the sampled signal x_s and the FT $X_\delta = \mathcal{F}(x_\delta)$ of the Dirac sampled signal x_δ coincide

$$X_\delta(f) = X_s(f)$$

- True for all freqs., not just between $\pm f_s/2$. FT $X_\delta(f)$ is periodic
- We already saw this property for sampling continuous time constants
  $\Rightarrow$ Discrete time constant and Dirac train



DTFT & FT of sampled signals coincide (proof)

Proof.

- Write the definition of the FT $X_\delta = \mathcal{F}(x_\delta)$ of Dirac sampled signal

$$X_\delta(f) = \int_{-\infty}^{\infty} \left[T_s \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s) e^{-j2\pi ft} \right] df$$

- Exchange the order of summation and integration

$$X_\delta(f) = T_s \sum_{n=-\infty}^{\infty} \left[\int_{-\infty}^{\infty} x(nT_s)\delta(t - nT_s) e^{-j2\pi ft}\, df \right]$$

- Multiplying by delta and integrating recovers value at spike. Thus,

$$X_\delta(f) = T_s \sum_{n=-\infty}^{\infty} x(nT_s) e^{-j2\pi f nT_s} = T_s \sum_{n=-\infty}^{\infty} x_s(n) e^{-j2\pi f nT_s} = X_s(f)$$

- We use $x_s(n) = x(nT_s)$ and definition of DTFT in last two equalities □

Product in time convolution in frequency

- When we convolve signals in time we multiply their spectra
- Duality $\Rightarrow$ When we multiply them in time we convolve their spectra
  $\Rightarrow$ Don't need to prove. It has to be true because iFT is like an FT
- We obtain Dirac sampled signal x_δ by multiplying x with Dirac train

$$x_\delta(t) = x(t) \times T_s \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

- Spectrum X_δ is convolution of $X = \mathcal{F}(x)$ with the FT of Dirac train

$$X_\delta = X * \mathcal{F}\left[T_s \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \right]$$

- Fourier transform of the Dirac train (T_s) is another Dirac train (f_s)

The spectrum of the Dirac sampled signal

- Spectrum X_δ convolves X with a Dirac train with spacing f_s

$$X_\delta = X * \left[\sum_{k=-\infty}^{\infty} \delta(t - kf_s) \right]$$

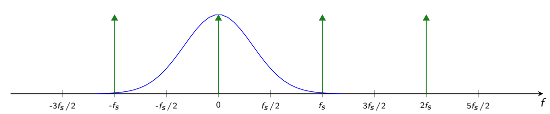
- But convolution is a linear operation $\Rightarrow X_\delta = \sum_{k=-\infty}^{\infty} X * \delta(f - kf_s)$
- Convolving with shifted delta is a shift $\Rightarrow X_\delta(f) = \sum_{k=-\infty}^{\infty} X(f - kf_s)$

Theorem
Spectrum of sampled signal is a sum of shifted versions of original spectrum

$$X_s(f) = X_\delta(f) = \sum_{k=-\infty}^{\infty} X(f - kf_s)$$

Spectrum periodization

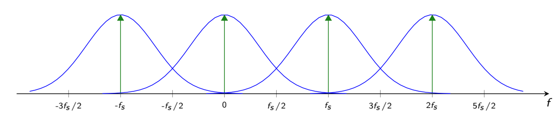
- We start with the spectrum X of x and the Dirac train in frequency
- Sampling to create x_s $\Rightarrow$ Multiplication with time Dirac train (T_s)
- Which in frequency domain entails convolution with Dirac train (f_s)
- Which is equivalent to summing shifted copies of the spectrum X



- FT X of continuous time signal x

Spectrum periodization

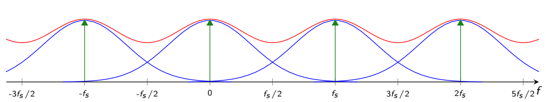
- We start with the spectrum X of x and the Dirac train in frequency
- Sampling to create x_s $\Rightarrow$ Multiplication with time Dirac train (T_s)
- Which in frequency domain entails convolution with Dirac train (f_s)
- Which is equivalent to summing shifted copies of the spectrum X



- First convolution step is to duplicate and shift spectrum to kf_s

Spectrum periodization

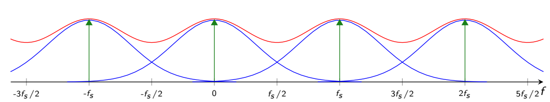
- We start with the spectrum X of x and the Dirac train in frequency
- Sampling to create x_s $\Rightarrow$ Multiplication with time Dirac train (T_s)
- Which in frequency domain entails convolution with Dirac train (f_s)
- Which is equivalent to summing shifted copies of the spectrum X



- Second convolution step is to sum all shifted copies

Information loss

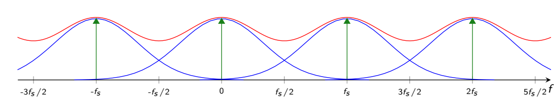
- When sampling x to x_s we lose information at high frequencies
  $\Rightarrow$ Everything that happens above $f_s/2$ is lost
  $\Rightarrow$ Freqs. close to $f_s/2$ distorted by superposition with freqs. above $f_s/2$



- We say that the sampling process results in spectral aliasing
  $\Rightarrow$ When f_s is small, severe aliasing destroys all information

Increasing sampling time

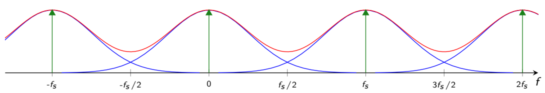
- As we increase the sampling time, aliasing becomes less severe



- Aliasing eventually disappears $\Rightarrow$ Approximately true in general
- But exactly true for bandlimited signals.
  $\Rightarrow$ Signals with $X(f) = 0$ for $f \notin [-W/2, W/2]$ (bandwidth W)

Increasing sampling time

- As we increase the sampling time, aliasing becomes less severe

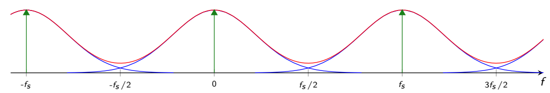


- Aliasing eventually disappears $\Rightarrow$ Approximately true in general
- But exactly true for bandlimited signals.

  $\Rightarrow$ Signals with $X(f) = 0$ for $f \notin [-W/2, W/2]$ (bandwidth W)

Increasing sampling time

- As we increase the sampling time, aliasing becomes less severe



- Aliasing eventually disappears $\Rightarrow$ Approximately true in general
- But exactly true for bandlimited signals.

  $\Rightarrow$ Signals with $X(f) = 0$ for $f \notin [-W/2, W/2]$ (bandwidth W)

Sampling of bandlimited signals

- We have therefore proved the following theorem

Theorem

Let x be a signal of bandwidth W. If the signal is sampled at a frequency $f_s \geq W$ we have that

$$X_\delta(f) = X_s(f) = X(f)$$

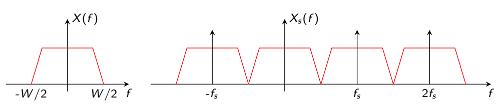
for all frequencies $f \in [-W/2, W/2]$

- There is no loss of information $\Rightarrow$ We can recover x from x_δ
- Use low pass filter to remove all frequencies outside of $[-W/2, W/2]$

Discussions

Sampling of bandlimited signals

- Signal with bandwidth W $\Rightarrow X(f) = 0$ for all $f \notin [-W/2, W/2]$
- Upon sampling, spectrum is periodized but not aliased

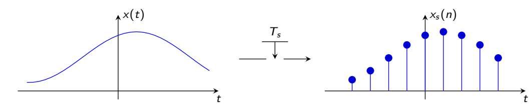


- This means that sampling entails no loss of information

  $\Rightarrow$ Can low pass x_s to recover x.

Non-vanishing sampling time

- That there is no loss of information is quite surprising
- We are discarding part of the signal, indeed, most of the signal



- It is reasonable to expect that we don't lose information as $T_s \to 0$

  $\Rightarrow$ But we don't have to let the sampling time vanish

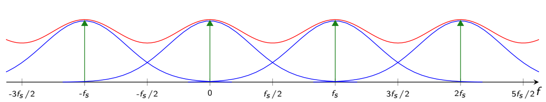
- Any sampling time $T_s \leq \dfrac{1}{W}$ yields $f_s \geq W$ and no information loss

Sampling of non-bandlimited signals

- Information in frequency components larger than $f_s/2$ is lost

  $\Rightarrow$ Nothing we can do about that other than increasing f_s

- Can't capture variability faster than $f_s/2$ with sampling time T_s

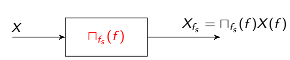


- But aliasing is also distorting information in components below $f_s/2$

Prefiltering

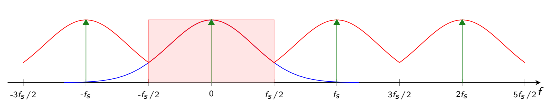
- To avoid aliasing distortion we preprocess x with a low pass filter
- I.e., we transform x into signal x_{f_s} with spectrum $X_{f_s} = \mathcal{F}(x_{f_s})$

$$X_{f_s}(f) = X(f)\sqcap_{f_s}(f)$$



- The signal x_{f_s} has bandwidth f_s and can be sampled without aliasing

  $\Rightarrow$ Frequency components below $f_s/2$ are retained with no distortion

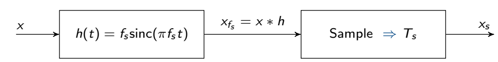


Prefiltering in time domain

- Prefiltering can be implemented as convolution in the time domain

$$x_{f_s} = x * h$$

- where h is iFT of low pass filter $X(f)\sqcap_{f_s}$ $\Rightarrow h(t) = f_s \text{sinc}(\pi f_s t)$

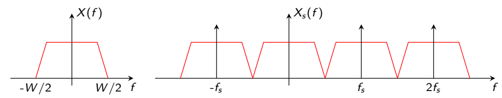


- Convolution has to be implemented in continuous time (circuits)

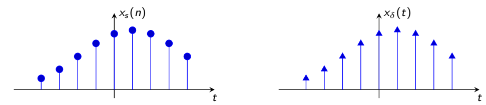
Signal reconstruction

Low pass filter recovery

- Bandwidth W ($X(f) = 0$ for all $f \notin [-W/2, W/2]$). Sample at $f_s \geq W$
- Can recover signal x from sampled signal x_s with low pass filter
  - $\Rightarrow$ What does exactly mean that "we use a low pass filter"?



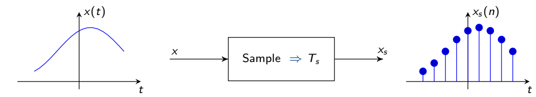
- Can't filter discrete time signal and have continuous time magically appear



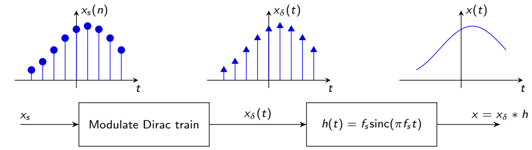
- But we can filter the continuous time Dirac sampled signal $x_\delta(t)$

Ideal sampling – reconstruction system

- We sample by keeping observations at nT_s $\Rightarrow x_s(n) = x(nT_s)$



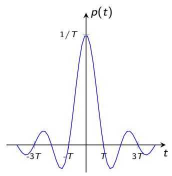
- To reconstruct we modulate Dirac train $\Rightarrow x_\delta(t) = T_s \sum_{n=-\infty}^{\infty} x_s(n)\delta(t - nT_s)$
- And low pass filter Dirac train x_δ $\Rightarrow x = x_\delta * \left[f_s \text{sinc}(\pi f_s t)\right]$



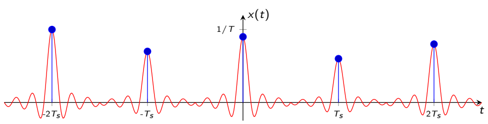
Reconstruction with a pulse train

- Dirac train is an abstract representation $\Rightarrow$ Can't be generated
- Modulate train of (narrow) pulses

$$x_p(t) = T_s \sum_{n=-\infty}^{\infty} x_s(n)p(t - nT_s)$$



- If pulse is sufficiently narrow $\Rightarrow x_p \approx x_\delta$
- E.g. $p(t) = \frac{1}{T}\text{sinc}\left(\pi\frac{t}{T}\right)$ with $T \ll T_s$
- Scale pulse by $x(n)$, shift to $t = nT_s$, sum all copies $\Rightarrow$ convolution?



Dirac train representation of pulse train

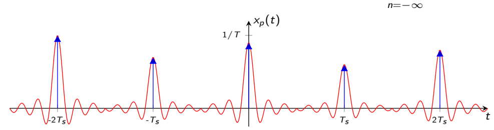
- Pulse train modulation can be represented as convolution with x_δ

$$x_p = p * x_\delta$$

- Indeed use definition of x_δ and convolution linearity to write $p * x_\delta$ as

$$x_p = p * \left[T_s \sum_{n=-\infty}^{\infty} x_s(n)\delta(t - nT_s)\right] = T_s \sum_{n=-\infty}^{\infty} x_s(n)\left[p * \delta(t - nT_s)\right]$$

- Convolving with shifted delta is a shift $\Rightarrow x_p(t) = T_s \sum_{n=-\infty}^{\infty} x_s(n)p(t - nT_s)$



Spectrum of modulated pulse train

- Convolution in time is equivalent to multiplication in frequency
- Then, the spectrum of $X_p = \mathcal{F}(x_p)$ is the product of $P = \mathcal{F}(p)$ and X_δ

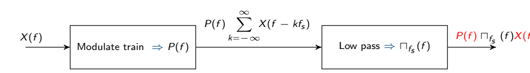
$$X_p(f) = P(f)X_\delta(f) = P(f) \sum_{k=-\infty}^{\infty} X(f - kf_s)$$

- Reconstructed signal x_r obtained by low pass filtering. FT $X_r = \mathcal{F}(x_r)$ is

$$X_r(f) = P(f)X_\delta(f)\sqcap_{f_s}(f) = P(f)\sqcap_{f_s}(f) \sum_{k=-\infty}^{\infty} X(f - kf_s)$$

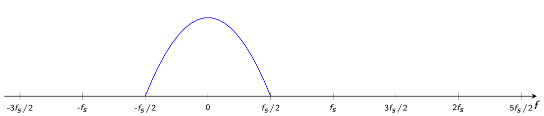
- Low pass filter eliminates all frequencies outside of $[-f_s/2, f_s/2]$

$$X_r(f) = P(f)\sqcap_{f_s}(f)X(f)$$



More on the spectrum of sampling and recovery

- We start with a bandlimited signal that we sample at $f_s = W$
- Spectrum is $\Rightarrow X(f)$

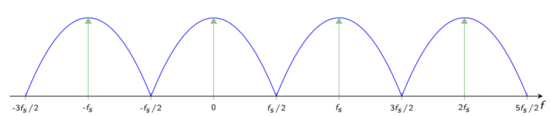


-

More on the spectrum of sampling and recovery

- The spectrum X_s of the sampled signal is periodization of X

$$\Rightarrow X_s(f) = \sum_{k=-\infty}^{\infty} X(f - kf_s)$$

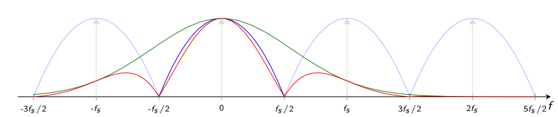


-

More on the spectrum of sampling and recovery

- To recover the signal we modulate a pulse train. Pulse FT is $P(f)$

$$\Rightarrow X_p(f) = P(f) \times \sum_{k=-\infty}^{\infty} X(f - kf_s)$$

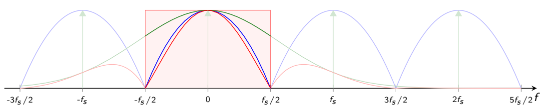


-

More on the spectrum of sampling and recovery

- We finalize recovery with a low pass filter of bandwidth f_s

  $\Rightarrow X_r(f) = \sqcap_{f_s}(f) P(f) X(f - kf_s)$



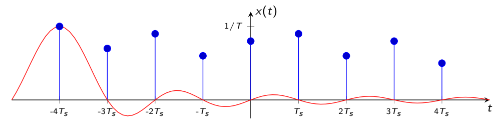
- Good pulse for recovery $\Rightarrow X(f) = 1$ for $f \in [-f_s/2, f_s/2]$

Modulation of a sinc train

- Do we know a pulse with $X(f) = 1$ for $f \in [-f_s/2, f_s/2]$?
  $\Rightarrow$ We do! $\Rightarrow$ The sinc pulse $f_s \text{sinc}(\pi f_s t)$
- Don't even need to use low pass filter $\Rightarrow$ sinc pulse already lowpass

Theorem
A signal of bandwidth $W \leq f_s$ can be recovered from samples $x(nT_s)$ as

$$x(t) = f_s T_s \sum_{n=-\infty}^{\infty} x(nT_s) \text{sinc}(\pi f_s (t - nT_s))$$



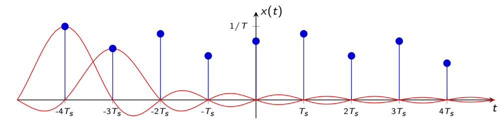
- Reconstruction without a Dirac train $\Rightarrow$ (mostly) implementable

Modulation of a sinc train

- Do we know a pulse with $X(f) = 1$ for $f \in [-f_s/2, f_s/2]$?
  $\Rightarrow$ We do! $\Rightarrow$ The sinc pulse $f_s \text{sinc}(\pi f_s t)$
- Don't even need to use low pass filter $\Rightarrow$ sinc pulse already lowpass

Theorem
A signal of bandwidth $W \leq f_s$ can be recovered from samples $x(nT_s)$ as

$$x(t) = f_s T_s \sum_{n=-\infty}^{\infty} x(nT_s) \text{sinc}(\pi f_s (t - nT_s))$$



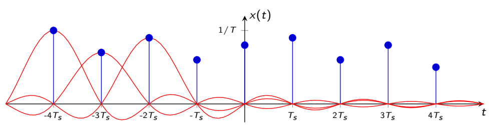
- Reconstruction without a Dirac train $\Rightarrow$ (mostly) implementable

Modulation of a sinc train

- Do we know a pulse with $X(f) = 1$ for $f \in [-f_s/2, f_s/2]$?
  $\Rightarrow$ We do! $\Rightarrow$ The sinc pulse $f_s \text{sinc}(\pi f_s t)$
- Don't even need to use low pass filter $\Rightarrow$ sinc pulse already lowpass

Theorem
A signal of bandwidth $W \leq f_s$ can be recovered from samples $x(nT_s)$ as

$$x(t) = f_s T_s \sum_{n=-\infty}^{\infty} x(nT_s) \text{sinc}(\pi f_s (t - nT_s))$$



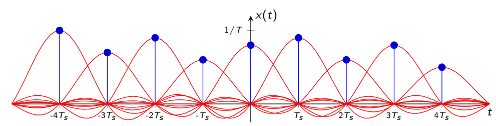
- Reconstruction without a Dirac train $\Rightarrow$ (mostly) implementable

Modulation of a sinc train

- Do we know a pulse with $X(f) = 1$ for $f \in [-f_s/2, f_s/2]$?
  $\Rightarrow$ We do! $\Rightarrow$ The sinc pulse $f_s \text{sinc}(\pi f_s t)$
- Don't even need to use low pass filter $\Rightarrow$ sinc pulse already lowpass

Theorem
A signal of bandwidth $W \leq f_s$ can be recovered from samples $x(nT_s)$ as

$$x(t) = f_s T_s \sum_{n=-\infty}^{\infty} x(nT_s) \text{sinc}(\pi f_s (t - nT_s))$$

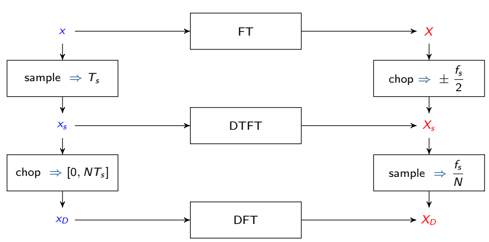


- Reconstruction without a Dirac train $\Rightarrow$ (mostly) implementable

From the FT to the DFT

The DFT as a proxy for the FT

- We use the DFT for frequency analysis of continuous time signals
- Justifiable $\Rightarrow$ They're approximately equal for small T_s and large N



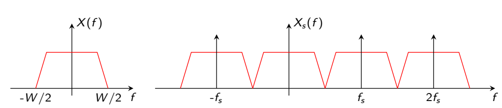
- Sampling $\Rightarrow$ Can understand what is lost in the approximation

Sampling $\Rightarrow$ From the FT to the DTFT

- Sampling in time $\equiv$ periodization (not "chop") in frequency

$$x_s(n) = x(nT_s) \qquad \Longleftrightarrow \qquad X_s(f) = \sum_{k=-\infty}^{\infty} X(f - kf_s)$$

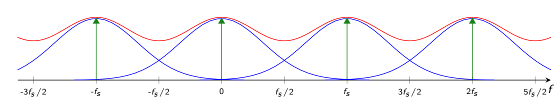
- Replicate. Shift to recenter at $f = kf_s$. Add all shifted copies
- If signal is bandlimited $\Rightarrow X_s(f) = X(f)$ for all $f \in [-f_s/2, f_s/2]$
  $\Rightarrow$ Spectra coincide perfectly $\Rightarrow$ No approximation



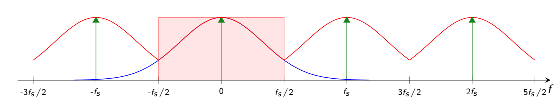
- In general, signals are not bandlimited and we expect some distortion

Lost in approximation

- Signal is not bandlimited $\Rightarrow$ freqs. above $f_s/2$ not seen in DTFT
- Without prefiltering $\Rightarrow$ aliasing distorts freqs. close to $f_s/2$



- With prefiltering $\Rightarrow$ all freqs. below $f_s/2$ approximated correctly



- Which means that we do use a low pass filter prior to sampling

The DTFT as proxy for the FT (1 of 3)

- Filter $\Rightarrow$ multiply in frequency by H $\Rightarrow$ convolve in time with h

$$X_f = HX \quad \Longleftrightarrow \quad x_f = x * h$$

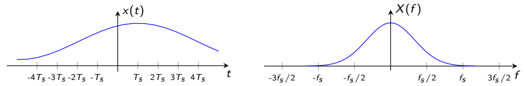
- Sample filtered signal X_f $\Rightarrow$ Periodize filtered spectrum X_f

$$x_s(n) = x_f(nT_s) \quad \Longleftrightarrow \quad X_s(f) = \sum_{k=-\infty}^{\infty} X_f(f - kf_s)$$

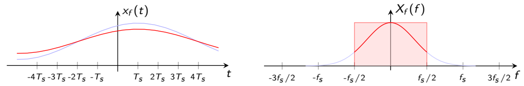
- Distortion (information loss) occurs during filtering step
  - $\Rightarrow$ Frequency $\Rightarrow$ Loss above $f_s/2$ + some distortion if H not perfect
  - $\Rightarrow$ Time $\Rightarrow$ Convolution with h

The DTFT as proxy for the FT (2 of 3)

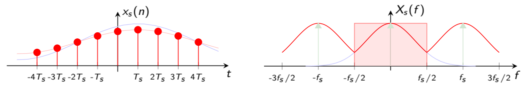
- Continuous time signal x with FT X $\Rightarrow$ Not necessarily bandlimited



- Continuous time filtered signal x_f $\Rightarrow$ filtering smoothes (distorts) x

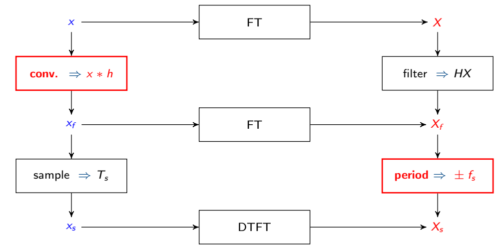


- Sampled signal x_s obtained from filtered x_f $\Rightarrow$ No further distortion



The DTFT as proxy for the FT (3 of 3)

- Filtering (chop) induces convolution. Sampling induces periodization



- Small distortion $\Rightarrow$ Make f_s so that $X(f) \approx 0$ for $f \notin [-f_s/2, f_s/2]$

Windowing $\Rightarrow$ From the DTFT to the DFT

- DTFT of sampled signal x_s is $\Rightarrow X_s(f) = T_s \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n T_s}$
- Windowed signal $\Rightarrow$ Nullify signal values outside of interval $[0, N-1]$

$$x_w(n) = x_s(n), \qquad \text{for all} \quad n \in [0, N-1]$$

- Windowed signal is $x_w(n) = 0$ outside of window (all $n \notin [0, N-1]$)
- DTFT of windowed signal x_w is $\Rightarrow X_s(f) = T_s \sum_{n=0}^{N-1} x(n) e^{-j2\pi f n T_s}$

Spectrum after windowing

- Windowing equivalent to multiplication with square pulse
- More generically $\Rightarrow$ define a window signal w_N as one for which

$$w_N(n) = 0 \qquad \text{for all} \quad n \notin [0, N-1]$$

- Rewrite discrete time windowed signal as $\Rightarrow x_w(n) = x(n) \times w_N(n)$
- Since multiplication in time is equivalent to convolution in frequency

$$X_w(f) = X_s(f) * W_N(f)$$

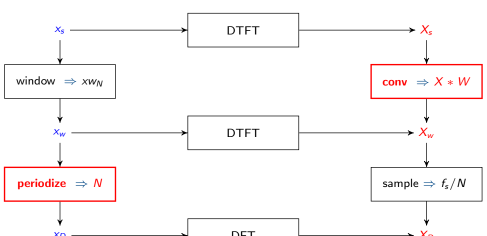
- Multiplicative distortion given by DTFT of window function
- If x_s is already finite $\Rightarrow$ No distortion (dual of bandlimited)

Frequency sampling

- DTFT of windowed signal x_w is $\Rightarrow X_s(f) = T_s \sum_{n=0}^{N-1} x(n) e^{-j2\pi f n T_s}$
- Reinterpret x_w as discrete signal x_D (null vs undefined outside $[0, N-1]$)
- Signal x_D has a DFT (finite) $\Rightarrow X_D(f) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_D(n) e^{-j2\pi kn/N}$
- Comparing expressions $\Rightarrow X_s\left(\frac{k}{N} f_s\right) = T_s \sqrt{N} X_D(k)$
- Sample in time $\equiv$ periodize in frequency $\Rightarrow$ Dual property holds?
  - $\Rightarrow$ Yes. The iDFT is a periodic operation
  - $\Rightarrow$ We have $x_D(n+N) = x_D(N)$ because $e^{j2\pi k(n+N)/N} = e^{j2\pi kn/N}$

The DFT as proxy for the DTFT (1 of 2)

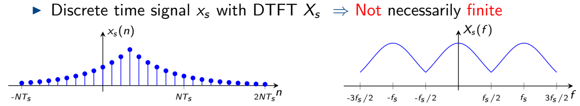
- Window (chop) induces convolution. Sampling induces periodization



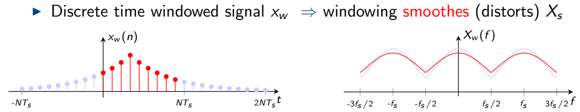
- Small distortion $\Rightarrow$ Make N so that $x(n) \approx 0$ for $n \notin [0, N-1]$

The DFT as proxy for the DTFT (2 of 2)

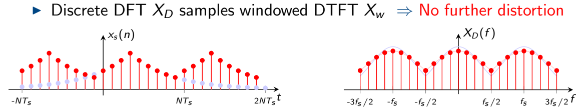
- Discrete time signal x_s with DTFT X_s $\Rightarrow$ Not necessarily finite



- Discrete time windowed signal x_w $\Rightarrow$ windowing smoothes (distorts) X_s



- Discrete DFT X_D samples windowed DTFT X_w $\Rightarrow$ No further distortion



Linear time invariant systems

Alejandro Ribeiro
Dept. of Electrical and Systems Engineering
University of Pennsylvania
aribeiro@seas.upenn.edu
http://www.seas.upenn.edu/users/~aribeiro/

February 27, 2015

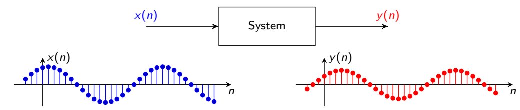
Linear time invariant systems

Fourier transform and convolution

- Fourier transform enables signal and information processing
  - ⇒ Patterns and properties easier to discern on frequency domain
- Also enables analysis and deign of linear time invariant (LTI) systems
  - ⇒ Not altogether unrelated to pattern discernibility
- Two properties of LTI systems
  - ⇒ Characterized by their (impulse) response to a delta input
  - ⇒ Responses to other inputs are convolutions with impulse response
- Equivalent properties in the frequency domain
  - ⇒ Characterized by frequency response = $\mathcal{F}$(impulse response)
  - ⇒ Output spectrum = input spectrum × frequency response

Systems

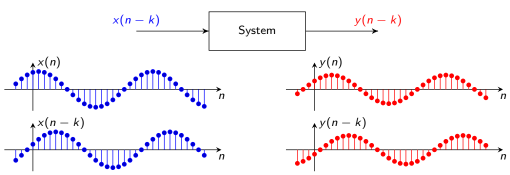
- A system is characterized by an input ($x(n)$) output ($y(n)$) relation
- This relation is between functions, not values
- Each output value $y(n)$ depends on all input values $x(n)$



- We can, alternatively, consider continuous time systems. The same.

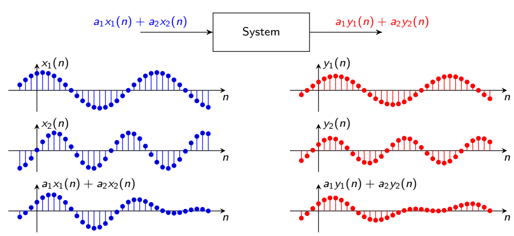
Time invariant systems

- A system is time invariant if a delayed input yields a delayed output
- If input $x(n)$ yields output $y(n)$ then input $x(n-k)$ yields $y(n-k)$
- Think of output when input is applied k time units later



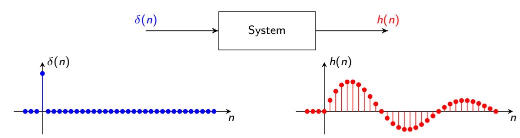
Linear systems

- In a linear system ⇒ input a linear combination of inputs
  - ⇒ Output the same linear combination of the respective outputs
- I.e., if input $x_1(n)$ yields output $y_1(n)$ and $x_2(n)$ yields $y_2(n)$
  - ⇒ Input $a_1x_1(n) + a_2x_2(n)$ yields output $a_1y_1(n) + a_2y_2(n)$



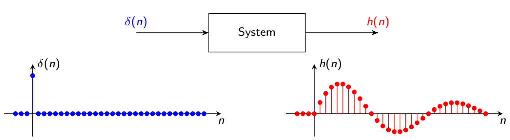
Linear time invariant systems

- Linear + time invariant system = linear time invariant system (LTI)
- Also called a LTI filter, or a linear filter, or simply a filter
- The impulse response is the output when input is a delta function
  - ⇒ Input is $x(n) = \delta(n)$ (discrete time, $\delta(0) = 1$)
  - ⇒ Output is $y(n) = h(n)$ = impulse response



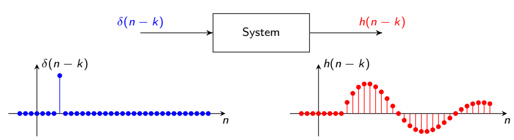
Scale and shifted impulse responses

- Since the system is time invariant (shift)
  - ⇒ Input $\delta(n-k)$ ⇒ Induces output response $h(n-k)$
- Since the system is linear (scale)
  - ⇒ input $x(k)\delta(n-k)$ ⇒ Output $x(k)h(n-k)$
- Since the system is linear (sum)
  - ⇒ $x(k_1)\delta(n-k_1) + x(k_2)\delta(n-k_2)$ ⇒ $x(k_1)h(n-k_1) + x(k_2)h(n-k_2)$



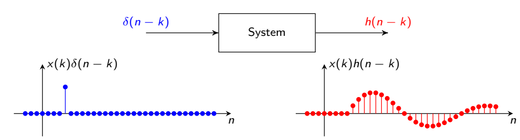
Scale and shifted impulse responses

- Since the system is time invariant (shift)
  - ⇒ Input $\delta(n-k)$ ⇒ Induces output response $h(n-k)$
- Since the system is linear (scale)
  - ⇒ input $x(k)\delta(n-k)$ ⇒ Output $x(k)h(n-k)$
- Since the system is linear (sum)
  - ⇒ $x(k_1)\delta(n-k_1) + x(k_2)\delta(n-k_2)$ ⇒ $x(k_1)h(n-k_1) + x(k_2)h(n-k_2)$



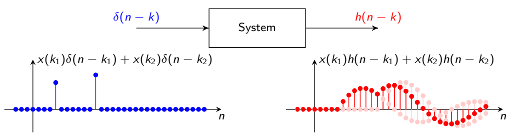
Scale and shifted impulse responses

- Since the system is time invariant (shift)
  - ⇒ Input $\delta(n-k)$ ⇒ Induces output response $h(n-k)$
- Since the system is linear (scale)
  - ⇒ input $x(k)\delta(n-k)$ ⇒ Output $x(k)h(n-k)$
- Since the system is linear (sum)
  - ⇒ $x(k_1)\delta(n-k_1) + x(k_2)\delta(n-k_2)$ ⇒ $x(k_1)h(n-k_1) + x(k_2)h(n-k_2)$



Scale and shifted impulse responses

- Since the system is time invariant (shift)
  - $\Rightarrow$ Input $\delta(n-k)$ $\Rightarrow$ Induces output response $h(n-k)$
- Since the system is linear (scale)
  - $\Rightarrow$ input $x(k)\delta(n-k)$ $\Rightarrow$ Output $x(k)h(n-k)$
- Since the system is linear (sum)
  - $\Rightarrow x(k_1)\delta(n-k_1) + x(k_2)\delta(n-k_2)$ $\Rightarrow x(k_1)h(n-k_1) + x(k_2)h(n-k_2)$



Output of a linear time invariant system

- Shift, Scale, and Sum $\Rightarrow$ Is this a Convolution? $\Rightarrow$ Of course
- Can write any signal x as $\Rightarrow x(n) = \sum_{k=-\infty}^{+\infty} x(k)\delta(n-k)$
- Thus, output of LTI with impulse response h to input x is given by

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n-k)$$

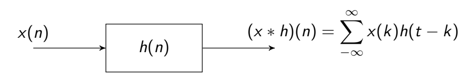
- The above sum is the convolution of x and h $\Rightarrow y = x * h$

Output of a linear time invariant system

Theorem

A linear time invariant system is completely determined by its impulse response h. In particular, the response to input x is the signal $y = x * h$.

- Innocent looking restrictions $\Rightarrow$ Linearity + time invariance
  - $\Rightarrow$ Induce very strong structure (anything but innocent)



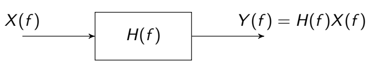
- Can derive exact same result for continuous time systems

Frequency response

- Frequency response = transform of impulse response $\Rightarrow H = \mathcal{F}(h)$

Corollary

A linear time invariant system is completely determined by its frequency response H. In particular, the response to input X is the signal $Y = HX$.



- Design in frequency $\Rightarrow$ Implement in time
  - $\Rightarrow$ Have done this already, but now we know its true for any LTI

Causality

- A causal filter is one with $h(n) = 0$ for all negative $n < 0$
  - $\Rightarrow$ Otherwise, we would respond to spike before seeing spike
- In general $\Rightarrow y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n-k) = \sum_{k=-\infty}^{n} x(k)h(n-k)$
- The value $y(n)$ is only affected by past inputs $x(k)$, with $k \leq n$
- If filter is not causal but $h(n) = 0$ for all $n < N$
  - $\Rightarrow$ Make it causal with a delay $\Rightarrow \tilde{h}(n) = h(n-N)$
- Frequency response of delayed filter $\Rightarrow \tilde{H}(f) = H(f)e^{j2\pi fN}$
  - $\Rightarrow$ Qualitatively the same filter

Finite impulse response

- A causal finite impulse response filter (FIR) is one for which

$$h(n) = 0 \qquad \text{for all } n \geq N$$

- We say the filter is of length N; only N values in $h(n)$ are not null
- Can write output at time n as

$$y(n) = h(0)x(n) + h(1)x(n-1) + \ldots h(N-1)x(n-N+1)$$

- Running input vector $\mathbf{x}_N(n) = [x(n); x(n-1); \ldots; x(n-N+1)]$
- FIR filter vector response $\mathbf{h} = [h(0), h(1), \ldots, h(N-1)]$
- Can then write output at time n as $\Rightarrow y(n) = \mathbf{h}^T \mathbf{x}_N$

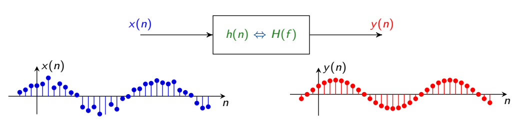
Finite impulse response filter design

Linear time invariant systems

Finite impulse response filter design

Filter design and implementation

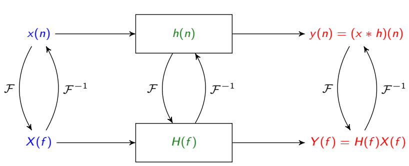
- We want to utilize a LTI system to process discrete time signal $x(n)$
  - $\Rightarrow$ E.g., to smooth out the signal $x(n)$ shown below



- All LTIs are completely determined by their impulse responses h
  - $\Rightarrow$ Design h and implement filter as time convolution $\Rightarrow y = x * h$
- All LTIs are completely determined by their frequency responses h
  - $\Rightarrow$ Design H and implement filter as spectral product $\Rightarrow Y = HX$

Frequency design and time implementation

- Time and frequency representations are equivalent

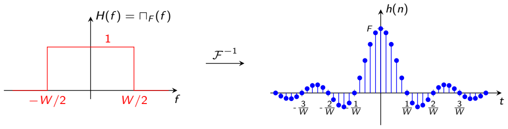


- Identify pattern transformation in frequency domain $\Rightarrow$ Design H
- Use inverse DTFT to compute impulse response $\Rightarrow h = \mathcal{F}^{-1}(H)$
- Implement convolution in time $\Rightarrow y(n) = (x * h)(n)$

Causality and infinite response

- Impulse response $h = \mathcal{F}^{-1}(H)$ is typically not causal and infinite
  $\Rightarrow$ E.g., Low pass filter with cutoff freq. $W/2 \;\Rightarrow\; H(f) = \sqcap_W(f)$

$$h(n) = \int_{-f_s/2}^{f_s/2} H(f) e^{j2\pi f n T_s}\, df = W \text{sinc}(\pi W n T_s)$$



- Multiply by window (chop) for finite response with N nonzero coeffs.
- Delay $h(n)$ to obtain a causal filter with $h(n) = 0$ for $n \leq 0$

FIR filter design

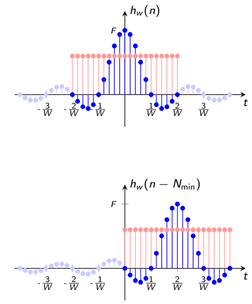
- Transform $h(n)$ into finite impulse response

$$h_w(n) = h(n)w(n)$$

- Window $w(n) = 0$ for $n \notin [N_{\min}, N_{\max}]$
- Filter length $N = N_{\max} - N_{\min} + 1$
- Transform $h_w(n)$ into causal response

$$h_w(n) \quad \Longrightarrow \quad h_w(n - N_{\min})$$

- Choose borders $N_{\min}$ and $N_{\max}$ to retain highest values of $h(n)$
- Often, around $n = 0$. But not always



Spectral effects of windowing and delaying

- Multiplication in time domain $\Rightarrow$ Convolution in frequency domain
- As a result, instead of filtering with $H(f)$, we filter with

$$H_w = H * W$$

- Choose windows with spectrum $W = \mathcal{F}(w)$ close to delta function
- Time delay $\Rightarrow$ Multiplication with complex exponential in frequency

$$H_w(f) \quad \Longrightarrow \quad H_w(f) e^{j2\pi f N_{\min} T_s}$$

- Irrelevant, as it should, we just shifted the response

FIR filter design methodology

- Procedure to design time coefficients of a FIR filter

(1) Spectral analysis to determine filter frequency response $H(f)$

(2) Inverse DFT (not DTFT) to determine impulse response $h(n)$

(3) Determine nr. of coefficients N and coefficient range $[N_{\min}, N_{\max}]$

(4) Select window $w(n) \;\Rightarrow$ Alters spectrum to $H_w = H * W$

(5) Shift impulse response by $N_{\min}$ time steps to make filter causal

- How to we use FIR filter coefficients $h(n)$ to implement the filter?

FIR implementation

- The output $y(n)$ of the FIR filter is given by the convolution value

$$y(n) = \sum_{k=-\infty}^{\infty} x(k) h(n-k)$$

- Since h is finite and causal, only N nonzero terms. Make $k = n - l$

$$y(n) = \sum_{k=n-(N-1)}^{n} x(k) h(n-k) = \sum_{l=0}^{N-1} h(l) x(n-l)$$

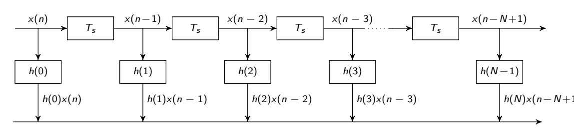
- Easier to visualize when written in expanded form

$$y(n) = h(0)\,x(n) + h(1)\,x(n-1) + \ldots + h(N-1)\,x(n-N+1)$$

- The expression above can be implemented with a shift register

Shift registers

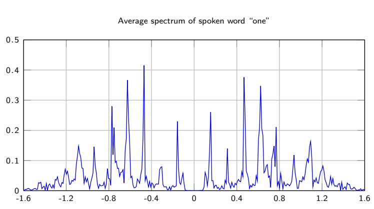
- Upon arrival of signal value $x(n)$ we compute output value $y(n)$ by
  $\Rightarrow$ Delay (shift) units to shift elements of signal x
  $\Rightarrow$ Product (scale) units to multiply with filter coefficients $x(n)$
  $\Rightarrow$ Sum units to aggregate the products $h(k)x(n-k)$



- Shift register can be implemented in hardware (or software)

Voice recognition $\Rightarrow$ Spectral design

- For a given word to be recognized we compare the spectra $\bar{X}$ and X
  $\Rightarrow \bar{X} \;\Rightarrow$ Average spectrum magnitude of word to be recognized
  $\Rightarrow X \;\Rightarrow$ Recorded spectrum during execution time

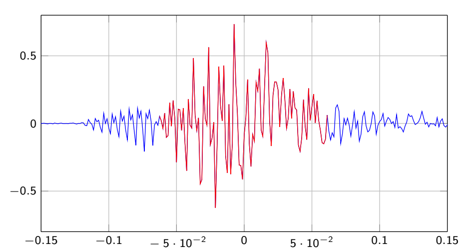


- Made coparison with inner product $\Rightarrow X^T \bar{X}$
- Equivalent to using $\bar{X}$ to filter $X \;\Rightarrow Y(f) = H(f)X(f)$ with $H(f) = \bar{X}$

Voice recognition $\Rightarrow$ Filter design

(2) Impulse response $h(n) \;\Rightarrow$ Inverse DFT of $\bar{X}$

(4) Window to keep $N = 1{,}000$ largest consecutive taps



Signal and information processing in time

Alejandro Ribeiro
Dept. of Electrical and Systems Engineering
University of Pennsylvania
aribeiro@seas.upenn.edu
http://www.seas.upenn.edu/users/~aribeiro/

March 10, 2015

Signals and information

Continuous time, discrete time, discrete signals

- We have studied continuous time, discrete time, and discrete signals
- Complex exponentials (CE), discrete time CE, and discrete CE
- And also the Fourier transform (FT), the DTFT, and the DFT
- For which we respectively studied the iFT, iDTFT and the iDFT
- Different versions of related concepts
  - $\Rightarrow$ Let's take time to summarize
  - $\Rightarrow$ And to emphasize analogies and differences

Signals

- Continuous time (CT) $t \in \mathbb{R}$ $\Rightarrow$ Continuous time signals

$$x : \mathbb{R} \rightarrow \mathbb{C}$$

- Discrete time (DT) $n \in \mathbb{Z}$ $\Rightarrow$ Discrete time signals

$$x : \mathbb{Z} \rightarrow \mathbb{C}$$

- Discrete and finite $n \in [0, N-1]$ $\Rightarrow$ Discrete signals

$$x : [0, N-1] \rightarrow \mathbb{C}$$

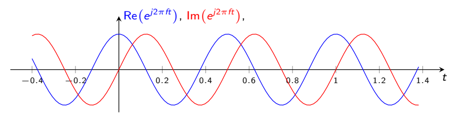
- From discrete signals we go to ...
  - ... infinity $\Rightarrow$ discrete time signals (extend borders)
  - ... and beyond $\Rightarrow$ continuos time signal (fill in spaces, dense)

Inner products and energy

- Inner product in continuous time $\Rightarrow \langle x, y \rangle := \int_{-\infty}^{\infty} x(t)y^*(t)dt$
- Inner product in discrete time $\Rightarrow \langle x, y \rangle := \sum_{n=-\infty}^{\infty} x(n)y^*(n)$
- Inner product of discrete signals $\Rightarrow \langle x, y \rangle := \sum_{n=0}^{N-1} x(n)y^*(n)$
- How much signals x and y are like each other
- Unrelated signals = orthogonality $\Rightarrow \langle x, y \rangle = 0$
- Energy, same definition works for all $\Rightarrow \|x\|^2 = \langle x, x \rangle$
- Inner product may not exist and energy may be infinite (CT and DT)

Continuous time complex exponentials

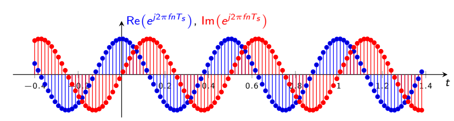
- Continuous time complex exponential e_f $\Rightarrow e_f(t) = e^{j2\pi ft}$
  - $\Rightarrow$ Signal is dense and extend to plus and minus infinity



- Frequency $f = 2$Hz shown. Time t in seconds

Discrete time complex exponentials

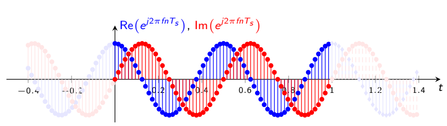
- Discrete time complex exponential e_{fT_s} $\Rightarrow e_{fT_s}(n) = e^{j2\pi fnT_s}$
  - $\Rightarrow$ Sample continuous time CE with sampling frequency $f_s = 1/T_s$
  - $\Rightarrow$ Signal extend to plus and minus infinity but is not dense



- Frequency $f = 2$Hz. Sampling freq. $f_s = 64$Hz. Time t in seconds.

Discrete complex exponentials

- Discrete complex exponential $\Rightarrow \sqrt{N} e_{kN}(n) = e^{j2\pi kn/N} = e^{j2\pi fnT_s}$
  - $\Rightarrow$ Discrete time CE observed during N samples $= NT_s$ time units
  - $\Rightarrow$ Defined for frequencies of the form $f = (k/N)f_s$ only
  - $\Rightarrow$ Exactly k oscillations during observation period $N \Leftrightarrow T$



- Frequency $f = 2$Hz. Sampling freq. $f_s = 64$Hz. Time t in seconds
- Observation time $T = 1$s $\Rightarrow$ number samples $N = Tf_s = 64$.
- Discrete frequency $k = N(f/f_s) = 2$

Orthogonality of complex exponentials

- Discrete complex exponentials are a set of N orthonormal signals

$$\langle e_{kN}, e_{lN} \rangle = \delta(k - l)$$

- We restrict k and l to interval of length N. E.g., $[-N/2+1, N/2]$
- CE with freqs. N apart are equivalent. Opposites are conjugates
- Discrete time complex exponentials are (sort of) orthogonal

$$\langle e_{fT_s}, e_{gT_s} \rangle = \delta(f - g)$$

- Continuous time delta $\Rightarrow$ Involves a limit. Generalized function
- Same is true in continuous time $\Rightarrow \langle e_f, e_g \rangle = \delta(f - g)$

Fourier transforms

Fourier transforms

- Fourier transform (FT) of continuous time signal x is the function

$$X(f) := \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t}\, dt$$

- The discrete time (DT)FT of discrete time signal x is the function

$$X(f) := T_s \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n T_s}$$

- The discrete (D)FT of discrete signal x is the function

$$X(k) := \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi f n T_s}$$

- Discrete frequency k equivalent to real $f = k/NT_s = kf_s/N$
- DFT is undefined for frequencies that are not $f = kf_s/N$ for some k

Fourier transforms as inner products

- Recall definitions of inner products and complex exponentials
- Write the FT of x as $\Rightarrow X(f) = \langle x, e_f \rangle = \int_{-\infty}^{\infty} x(t) e_f^*(t)\, dt$
- Write DTFT of x as $\Rightarrow X(f) = \langle x, e_{fT_s} \rangle = T_s \sum_{n=-\infty}^{\infty} x(n) e_{fT_s}^*(n)$
- Write the DFT of x as $\Rightarrow X(k) = \langle x, e_{kN} \rangle = \sum_{n=-\infty}^{\infty} x(n) e_{kN}^*(n)$
- All three transforms written as inner products in respective spaces

Different formalizations of the same concept

- Inner products with frequency f ($f = kf_s/N$) complex exponentials
- It follows that they are different formalizations of the same concept
  - $\Rightarrow$ They are projections of x onto oscillations of frequency f
  - $\Rightarrow$ They measure how much x resembles oscillation of frequency f
- Integrals, indefinite sums, sums $\Rightarrow$ Inherent differences in signals
- FT and DTFT are analysis tools. DFT is a computational tool

Input and output spaces

- Input and output spaces for FTs are continuous
- For DTFTs, discrete inputs, continuous and periodic outputs (odd)
- For DFTs, input and outputs are discrete and periodic or finite

	Input space	Output space
Fourier transform	Continuous	Continuous
DTFT	Discrete	Periodic Continuous
DFT	Discrete Periodic	Periodic Discrete

- Observe the duality between sampling and periodicity or finiteness

The DTFT as proxy for the FT (1 of 3)

- Filter $\Rightarrow$ multiply in frequency by H $\Rightarrow$ convolve in time with h

$$X_f = HX \quad \Longleftrightarrow \quad x_f = x * h$$

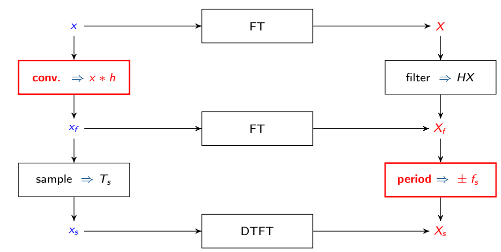
- Sample filtered signal X_f $\Rightarrow$ Periodize filtered spectrum X_f

$$x_s(n) = x_f(nT_s) \quad \Longleftrightarrow \quad X_s(f) = \sum_{k=-\infty}^{\infty} X_f(f - kf_s)$$

- Distortion (information loss) occurs during filtering step
  - $\Rightarrow$ Frequency $\Rightarrow$ Loss above $f_s/2$ + some distortion if H not perfect
  - $\Rightarrow$ Time $\Rightarrow$ Convolution with h

The DTFT as proxy for the FT (2 of 3)

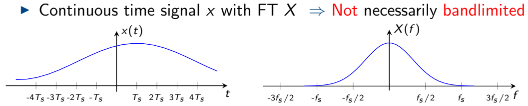
- Filtering (chop) induces convolution. Sampling induces periodization



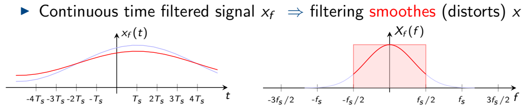
- Small distortion $\Rightarrow$ Make f_s so that $X(f) \approx 0$ for $f \notin [-f_s/2, f_s/2]$

The DTFT as proxy for the FT (3 of 3)

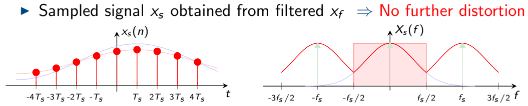
- Continuous time signal x with FT X $\Rightarrow$ Not necessarily bandlimited



- Continuous time filtered signal x_f $\Rightarrow$ filtering smoothes (distorts) x



- Sampled signal x_s obtained from filtered x_f $\Rightarrow$ No further distortion



The DFT as proxy for the DTFT (1 of 3)

- Filter $\Rightarrow$ multiply by window w_N $\Rightarrow$ convolve in frequency with W_N

$$x_w(n) = x(n) \times w_N(n) \quad \Longleftrightarrow \quad X_w(f) = X_s(f) * W_N(f)$$

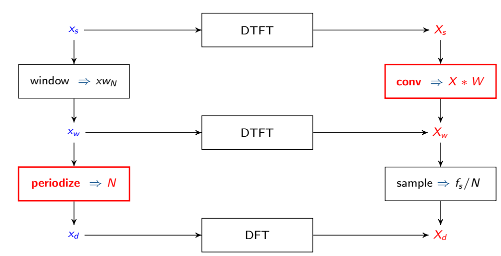
- Sample windowed spectrum X_w $\Rightarrow$ Periodize windowed signal x_w

$$x_d(n) = \sum_{k=-\infty}^{\infty} x_w(n - kN) \quad \Longleftrightarrow \quad X_d\left(\frac{kf_s}{N}\right) = T_s\sqrt{N}\, X_w(k)$$

- Distortion (information loss) occurs during windowing step
  - $\Rightarrow$ Frequency sampling is with no loss of information

The DFT as proxy for the DTFT (2 of 3)

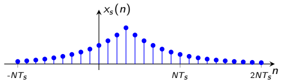
- Window (chop) induces convolution. Sampling induces periodization

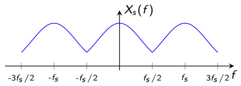


- Small distortion $\Rightarrow$ Make N so that $x(n) \approx 0$ for $n \notin [0, N-1]$

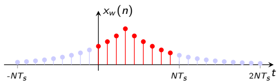
The DFT as proxy for the DTFT (3 of 3)

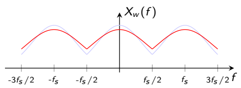
- Discrete time signal x_s with DTFT X_s $\Rightarrow$ Not necessarily finite



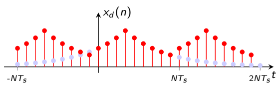


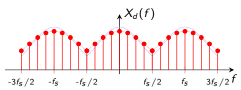
- Discrete time windowed signal x_w $\Rightarrow$ windowing smoothes (distorts) X_s





- Discrete DFT X_D samples windowed DTFT X_w $\Rightarrow$ No further distortion





Bandlimited and finite (periodic) signals

- If signal is bandlimited and sampled at frequency $f_s \geq W$
  $\Rightarrow$ The DTFT and the FT coincide in the interval $[-f_s/2, f_s/2]$
- If signal is finite, and windowed with N larger than its length
  $\Rightarrow$ DFT and DTFT coincide at the sampled frequencies $f = kf_s/N$
- What happens when signal is bandlimited and finite?
  $\Rightarrow$ Doesn't matter. These signals don't exist. Uncertainty principle

Inverse Fourier transforms

Inverse Fourier transforms

- Given a transform X, the inverse Fourier transform is defined as

$$x(t) := \int_{-\infty}^{\infty} X(f) e^{j2\pi f t}\, df$$

- The iDTFT x of DTFT X, is the discrete time signal with elements

$$x(n) = \int_{-f_s/2}^{f_s/2} X(f) e^{j2\pi f n T_s}\, df = \int_{0}^{f_s} X(f) e^{j2\pi f n T_s}\, df$$

- Given a Fourier transform X, the inverse (i)DFT is defined as

$$x(n) := \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} = \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} X(k) e^{j2\pi kn/N}$$

- Same as direct transform but for sign in the exponent $\Rightarrow$ duality

The inverses are inverses indeed

Theorem
The inverse FT (or inverse DTFT or inverse DFT) $\tilde{x}$ of the FT (respectively, DTFT or DFT) X of a given signal x is the given signal x

$$\tilde{x} = \mathcal{F}^{-1}(X) = \mathcal{F}^{-1}[\mathcal{F}(x)] = x$$

- We can recover signal from transform $\Rightarrow$ equivalent representation
  $\Rightarrow$ Neither less, nor more information. Just different interpretability
- Implies that we can write signal as a sum of complex exponentials
  $\Rightarrow$ Literally for iDFT, conceptually for iDTFT and iFT

Inverse DFT as sum of complex exponentials

- Signal as sum of exponentials $\Rightarrow x(n) = \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} X(k) e^{j2\pi kn/N}$
- Expand the sum inside out from $k = 0$ to $k = \pm 1$, to $k = \pm 2, \ldots$

$$\begin{array}{llllll}
x(n) = & X(0) & e^{j2\pi 0n/N} & & & \text{constant} \\
& +X(1) & e^{j2\pi 1n/N} & +X(-1) & e^{-j2\pi 1n/N} & \text{single oscillation} \\
& +X(2) & e^{j2\pi 2n/N} & +X(-2) & e^{-j2\pi 2n/N} & \text{double oscillation} \\
& \vdots & \vdots & \vdots & \vdots & \vdots \\
& +X\left(\frac{N}{2}-1\right) & e^{j2\pi\left(\frac{N}{2}-1\right)n/N} & +X\left(-\frac{N}{2}+1\right) & e^{-j2\pi\left(\frac{N}{2}-1\right)n/N} & \left(\frac{N}{2}-1\right) - \text{oscillation} \\
& +X\left(\frac{N}{2}\right) & e^{j2\pi\left(\frac{N}{2}\right)n/N} & & & \frac{N}{2} - \text{oscillation}
\end{array}$$

- Start with slow variations and progress on to add faster variations

Properties of Fourier transforms

Linearity and conjugate symmetry

Theorem
The FT, DTFT, and DFT of linear combinations of signals are linear combinations of the respective transforms of the individual signals,

$$\mathcal{F}(ax + by) = a\mathcal{F}(x) + b\mathcal{F}(y).$$

- Useful to compute transforms when considering sums of signals

Theorem
The FT, DTFT, and DFT $X = \mathcal{F}(x)$ of a real signal x (one with $Im(x) \equiv 0$) are conjugate symmetric

$$X(-f) = X^*(f)$$

- Only the positive half of the spectrum carries information

Energy conservation (Parseval's Theorem)

Theorem (Parseval)
The energy of a signal x and its FT, DTFT, or DFT $X = \mathcal{F}(x)$ are the same, i.e.,

$$\|x\|^2 = \|X\|^2$$

- Energy definitions are different for different signal spaces
- For the FT $\Rightarrow \int_{-\infty}^{\infty} |x(t)|^2 dt = \|x\|^2 = \|X\|^2 = \int_{-\infty}^{\infty} |X(f)|^2 df$
- For the DTFT $\Rightarrow \sum_{n=-\infty}^{\infty} |x(n)|^2 = \|x\|^2 = \|X\|^2 = \int_{-f_s/2}^{f_s/2} |X(f)|^2 df$
- For the DFT $\Rightarrow \sum_{n=0}^{N-1} |x(n)|^2 = \|x\|^2 = \|X\|^2 = \sum_{k=-N/2+1}^{N/2} |X(k)|^2$

Shift and modulation

Theorem

A time shift of τ units in the time domain is equivalent to multiplication by a complex exponential of frequency $-\tau$ in the frequency domain

$$x_\tau = x(t-\tau) \quad \Longleftrightarrow \quad X_\tau(f) = e^{-j2\pi f\tau}X(f)$$

Theorem

A multiplication by a complex exponential of frequency g in the time domain is equivalent to a shift of g units in the frequency domain

$$x_g = e^{j2\pi gt}x(t) \quad \Longleftrightarrow \quad X_g(f) = X(f-g)$$

- Theorems are duals of each other. True for FT and DTFT
- For DFT we need to define circular shifts. Not covered in this course

Convolutions in continuous and discrete time

- Let x and h be continuous time signals
- Convolution of x with h is the signal $y = x * h$ with values

$$[x*h](t) = y(t) = \int_{-\infty}^{\infty} x(u)h(t-u)\,du$$

- Let x and h be discrete time signals
- Convolution of x with h is the signal $y = x * h$ with values

$$[x*h](n) = y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

Multiplication and convolution

- Convolution in time domain $\equiv$ to multiplication in frequency domain

Theorem (Convolution theorem)

Given signals x and y with transforms $X = \mathcal{F}(x)$ and $Y = \mathcal{F}(y)$. The FT $Z = \mathcal{F}(z)$ of the convolved signal $z = x * y$ is the product $Z = XY$

$$z = x * y \quad \Longleftrightarrow \quad Z = XY$$

- True for FT and DTFT. For DFT need to define circular convolution
- The dual is also true
- Convolution in frequency domain $\equiv$ to multiplication in time domain

Sampling and reconstruction

Sampling

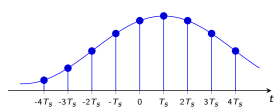
- The sampled signal x_s is a discrete time signal with values

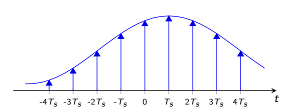
$$x_s(n) = x(nT_s)$$

- Creates discrete time signal x_s from continuous time signal x
- Equivalently, we represent sampling as multiplication by a Dirac train

$$x_\delta(t) = x(t) \times T_s \sum_{n=-\infty}^{\infty} \delta(t-nT_s)$$

- Dirac train lives in continuous time. Compare FT of x_δ to FT of x





Spectral effect of sampling

- Multiplication $\Leftrightarrow$ Convolution . Thus spectrum $X_\delta = \mathcal{F}(x_\delta)$ is

$$X_\delta = X * \mathcal{F}\left[T_s \sum_{n=-\infty}^{\infty} \delta(t-nT_s)\right]$$

- Fourier transform of the Dirac train (T_s) is another Dirac train (f_s)

$$X_\delta = X * T_s \sum_{n=-\infty}^{\infty} \delta(f-kf_s) = \sum_{n=-\infty}^{\infty} X * \delta(f-kf_s)$$

Theorem

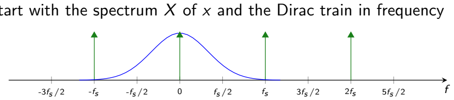
Sampled signal spectrum is a sum of shifted versions of original spectrum

$$X_s(f) = X_\delta(f) = \sum_{k=-\infty}^{\infty} X(f-kf_s)$$

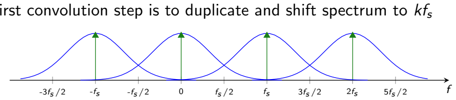
- We say the spectrum of X is periodized when the signal is sampled

Spectrum periodization

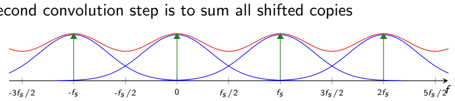
- Start with the spectrum X of x and the Dirac train in frequency



- First convolution step is to duplicate and shift spectrum to kf_s



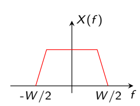
- Second convolution step is to sum all shifted copies

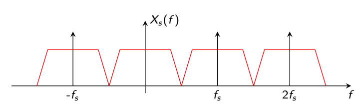


- Loose all info. above $f_s/2$. And some below to aliasing distortion

Sampling of bandlimited signals

- Signal with bandwidth W $\Rightarrow X(f) = 0$ for all $f \notin [-W/2, W/2]$
- Upon sampling, spectrum is periodized but not aliased





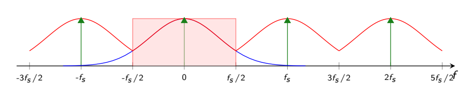
- This means that sampling entails no loss of information

Prefiltering

- To avoid aliasing preprocess x into x_{f_s} with a low pass filter

$$X_{f_s}(f) = X(f)\sqcap_{f_s}(f)$$

- The signal x_{f_s} has bandwidth f_s and can be sampled without aliasing
  - $\Rightarrow$ Frequency components below $f_s/2$ retained with no distortion



- Prefiltering can be implemented as convolution in the time domain

$$x_{f_s} = x * h, \qquad h(t) = f_s \text{sinc}(\pi f_s t)$$

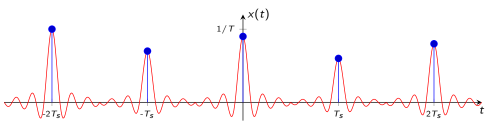
- iFT of low pass filter with cutoff $f_s/2$ is the sinc pulse with freq. f_s

Reconstruction

- In principle, we can recover x from x_δ with a low pass filter
- Since Dirac train can't be generated, we modulate train of pulses

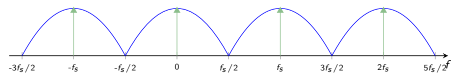
$$x_p(t) = T_s \sum_{n=-\infty}^{\infty} x_s(n)p(t - nT_s)$$

- For narrow pulses, pulse and Dirac modulation are close, i.e, $x_p \approx x_\delta$

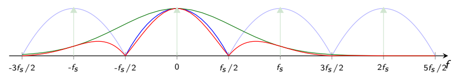


The spectrum of the reconstructed signal

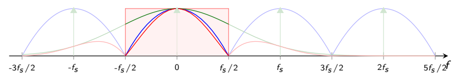
- Spectrum X_s of sampled signal $\Rightarrow X_s(f) = \sum_{k=-\infty}^{\infty} X(f - kf_s)$



- Spectrum X_p of pulse train $\Rightarrow X_p(f) = P(f) \times \sum_{k=-\infty}^{\infty} X(f - kf_s)$



- Reconstructed spectrum $X_r \Rightarrow X_r(f) = \sqcap_{f_s}(f)P(f)X(f - kf_s)$



- Good pulse for recovery $\Rightarrow X(f) = 1$ for $f \in [-f_s/2, f_s/2]$

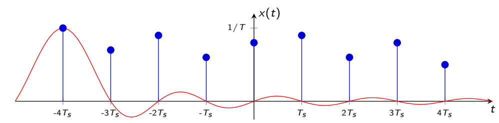
Modulation of a sinc train

- The sinc pulse $f_s\text{sinc}(\pi f_s t)$ has a flat spectrum for $f \in [-f_s/2, f_s/2]$
- Don't even need to use low pass filter $\Rightarrow$ sinc pulse already lowpass

Theorem

A signal of bandwidth $W \leq f_s$ can be recovered from samples $x(nT_s)$ as

$$x(t) = f_s T_s \sum_{n=-\infty}^{\infty} x(nT_s)\text{sinc}(\pi f_s(t - nT_s))$$



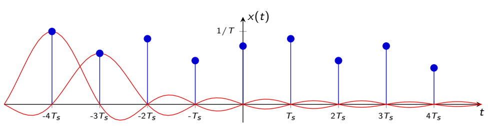
Modulation of a sinc train

- The sinc pulse $f_s\text{sinc}(\pi f_s t)$ has a flat spectrum for $f \in [-f_s/2, f_s/2]$
- Don't even need to use low pass filter $\Rightarrow$ sinc pulse already lowpass

Theorem

A signal of bandwidth $W \leq f_s$ can be recovered from samples $x(nT_s)$ as

$$x(t) = f_s T_s \sum_{n=-\infty}^{\infty} x(nT_s)\text{sinc}(\pi f_s(t - nT_s))$$



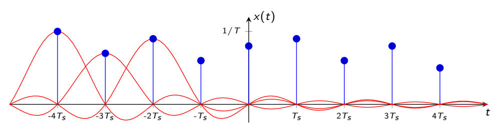
Modulation of a sinc train

- The sinc pulse $f_s\text{sinc}(\pi f_s t)$ has a flat spectrum for $f \in [-f_s/2, f_s/2]$
- Don't even need to use low pass filter $\Rightarrow$ sinc pulse already lowpass

Theorem

A signal of bandwidth $W \leq f_s$ can be recovered from samples $x(nT_s)$ as

$$x(t) = f_s T_s \sum_{n=-\infty}^{\infty} x(nT_s)\text{sinc}(\pi f_s(t - nT_s))$$



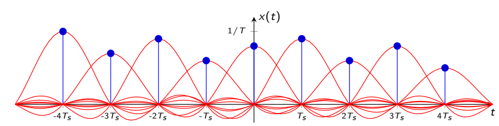
Modulation of a sinc train

- The sinc pulse $f_s\text{sinc}(\pi f_s t)$ has a flat spectrum for $f \in [-f_s/2, f_s/2]$
- Don't even need to use low pass filter $\Rightarrow$ sinc pulse already lowpass

Theorem

A signal of bandwidth $W \leq f_s$ can be recovered from samples $x(nT_s)$ as

$$x(t) = f_s T_s \sum_{n=-\infty}^{\infty} x(nT_s)\text{sinc}(\pi f_s(t - nT_s))$$



Philosophical digression

- Sampling is a straightforward operation, but its effects are obscure
  - $\Rightarrow$ Or not. If we look at the signal in frequency effects are also clear
- Loss of information contained at frequencies $f > f_s/2$
- Aliasing distortion for frequencies $f \leq f_s/2$
- Perfect recovery of bandlimited signals
- Avoid aliasing with profiteering
- Reconstruction distortion when modulating a train of pulses
- If we had a sixth sense for frequencies, all of this would be obvious
  - $\Rightarrow$ But we do have that sense, or rather have grown that sense

Linear time invariant systems

Signals and information

Fourier transforms

Inverse Fourier transforms

Properties of Fourier transforms

Sampling and reconstruction

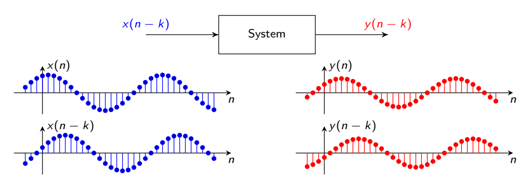
Linear time invariant systems

Applications

Signal representation

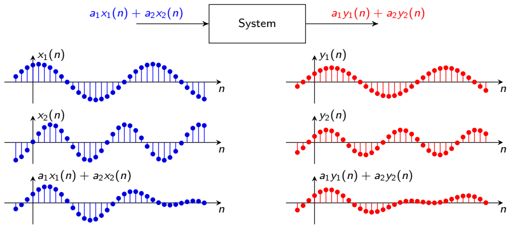
Time invariant systems

- Systems are characterized by input-output ($x \to y$) relationships
- A system is time invariant if a delayed input yields a delayed output
- If input $x(n)$ yields output $y(n)$ then input $x(n-k)$ yields $y(n-k)$



Linear systems

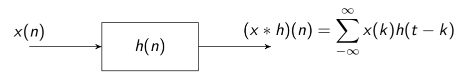
- In a linear system $\Rightarrow$ input a linear combination of inputs
  $\Rightarrow$ Output the same linear combination of the respective outputs
- I.e., if input $x_1(n)$ yields output $y_1(n)$ and $x_2(n)$ yields $y_2(n)$
  $\Rightarrow$ Input $a_1x_1(n) + a_2x_2(n)$ yields output $a_1y_1(n) + a_2y_2(n)$



Output of a linear time invariant system

- linear time invariant system (LTI) $\Rightarrow$ Linear + time invariant

Theorem
A linear time invariant system is completely determined by its impulse response h. In particular, the response to input x is the signal $y = x * h$.

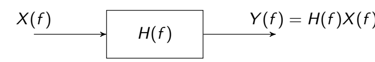


- Theorem true for discrete time and continuous time signals
  $\Rightarrow$ Convolutions are defined differently
- For discrete signals we need to use circular convolutions

Linear time invariant system frequency response

- Frequency response $\Rightarrow$ impulse response transform $\Rightarrow H = \mathcal{F}(h)$

Corollary
A linear time invariant system is completely determined by its frequency response H. In particular, the response to input X is the signal $Y = HX$.



- What a LTI system does to a signal is obscure
  $\Rightarrow$ Or not. If we look at the signal in frequency the effects are clear
- If we had a sixth sense for frequencies. Oh wait, we do
- It is obvious what LTI filters do $\Rightarrow$ They alter frequency components
- But they don't mix frequency components. Each of them is separate

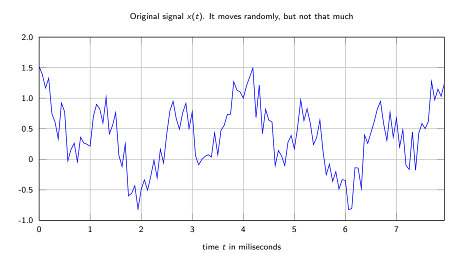
Applications

Applications

- Practical applications of frequency analysis are very common
- Here are a few applications that we have covered
  $\Rightarrow$ Noise removal,
  $\Rightarrow$ Music synthesis,
  $\Rightarrow$ Compression,
  $\Rightarrow$ Modulation,
  $\Rightarrow$ Signal detection (voice recognition)
- There are many more we have not covered
  $\Rightarrow$ E.g., equalization, high-pass filtering, band-pass filtering
- In all of these applications understanding time is complicated
  $\Rightarrow$ But understanding frequency is straightforward

Noise removal

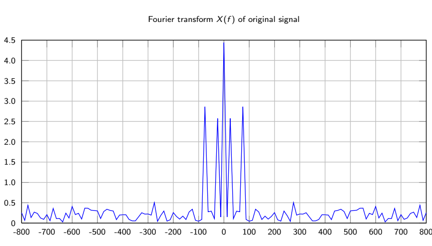
- There is signal and noise, but what is signal and what is noise?
- We already know answer $\Rightarrow$ Signal discernible in frequency domain



-

Noise removal

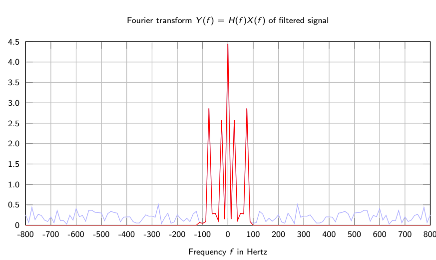
- There is signal and noise, but what is signal and what is noise?
- We already know answer $\Rightarrow$ Signal discernible in frequency domain



- Filter out all frequencies above 100Hz (and below -100Hz)

Low pass filter design

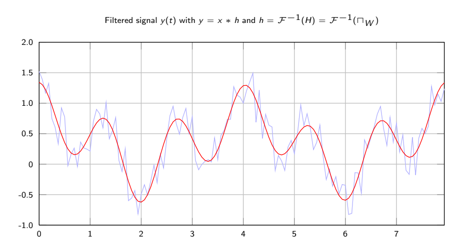
- Multiply spectrum with low pass filter $H(f) = \sqcap_W(f)$ with $W = 200$Hz
  $\Rightarrow$ Only frequencies between $\pm W/2 = \pm 100$Hz are retained



- This spectral operation does separate signal from noise

Low pass filter design

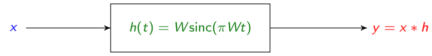
- Multiply spectrum with low pass filter $H(f) = \sqcap_W(f)$ with $W = 200$Hz
  $\Rightarrow$ Only frequencies between $\pm W/2 = \pm 100$Hz are retained



- This spectral operation does separate signal from noise

Low pass filter implementation

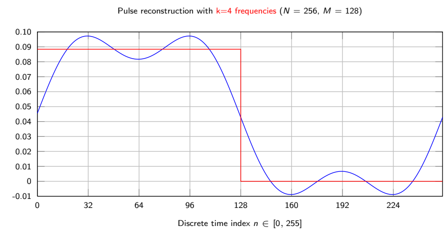
- We can implement filtering in the frequency domain
  $\Rightarrow$ Sample $\Rightarrow$ DFT $\Rightarrow$ Multiply by $H(f) = \sqcap_W(f)$ $\Rightarrow$ iDFT



- We can also implement filtering in the time domain
  $\Rightarrow$ Inverse transform of $\sqcap_W(f)$ is $h(t) = W\text{sinc}(\pi W t)$
- How is it that convolving with a sinc removes noise? $\Rightarrow$ obscure
- But is is very clear if we use our frequency sense
- Signal occupies some frequencies but noise occupies all frequencies

Signal compression

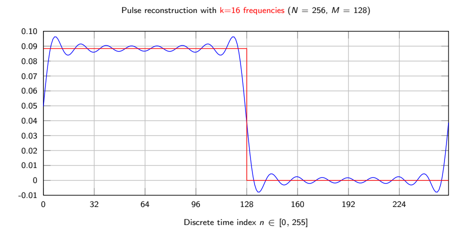
- Consider square pulse of duration $N = 256$ and length $M = 128$
- Reconstruct with 9 frequency components ($k \in [-4, 4]$)



- Compression $\Rightarrow$ Store 9 DFT values instead of $N = 128$ samples

Signal compression

- Consider square pulse of duration $N = 256$ and length $M = 128$
- Reconstruct with $k = 16$ frequency components



- Can tradeoff less compression for better signal accuracy

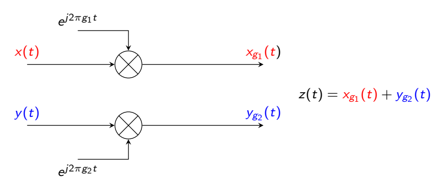
Signal compression

- Generic compression $\Rightarrow$ Keep largest DFT coefficients
  $\Rightarrow$ Not necessarily the lowest frequencies
- The approximation error energy is that of the coefficients dropped
- What's the advantage of comprising in frequency domain?
- Well, how would you compress in time domain
- Keep largest coefficients?
  $\Rightarrow$ No. Close values are redundant. Small values also important
- Keep values at certain spacing?
  $\Rightarrow$ Maybe. Actually that's sampling. Better think in freq. domain
- Compression is obscure but becomes clear if we use frequency sense

Modulation of multiple bandlimited signals

- Transmit multiple bandlimited signals (W) in a common support
  $\Rightarrow$ Wireless, optical fiber, coaxial cable, twisted pair
- Modulate (multiply by complex exponentials) with freqs. g_1 and g_2

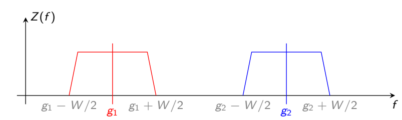
$$z(t) = e^{j2\pi g_1 t} x(t) + e^{j2\pi g_2 t} y(t)$$



- Spectrum of x recentered at g_1. Spectrum of y recentered at g_2

Spectrum of multiple modulated signals

- No spectral mixing if modulating frequencies satisfy $g_2 - g_1 > W$



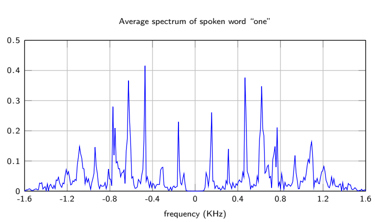
- To recover x multiply by conjugate frequency $e^{-j2\pi g_1 t}$
- And eliminated all frequencies outside the interval $[-W/2, W/2]$
- To recover y multiply by conjugate frequency $e^{-j2\pi g_2 t}$
- And eliminated all frequencies outside the interval $[-W/2, W/2]$

Modulation analysis and design

- Can we understand modulation in time?
  $\Rightarrow$ Actually, yes. Use orthogonality of complex exponentials
- But still, spectral analysis is clearer. Simplifies design
- Modulation is not entirely obscure
  $\Rightarrow$ But it becomes clearer if we use frequency sense

Signal detection (voice recognition)

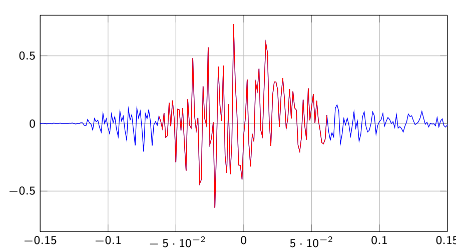
- For a given word to be recognized we compare the spectra $\bar{X}$ and X
  $\Rightarrow \bar{X}$ $\Rightarrow$ Average spectrum magnitude of word to be recognized
  $\Rightarrow X$ $\Rightarrow$ Recorded spectrum during execution time



- Energy $\sum_{k=-N/2+1}^{N/2} (X_i \bar{X}_i)^2 \Rightarrow$ Filter X with $\bar{X}$, i.e., $Y(f) = H(f)X(f)$ with $H(f) = \bar{X}$

Voice recognition $\Rightarrow$ Filter design

- Determine impulse response $h(n)$ as inverse DFT of spectrum $\bar{X}$
- Window $h(n)$ to keep, say, $N = 1{,}000$ largest consecutive taps



Signal detection analysis and design

- Can we understand signal detection in time?
  - ⇒ Actually, yes. It's called a matched filter
- But, as in modulation, spectral analysis is clearer. Simplifies design
- Signal detection is not entirely obscure
  - ⇒ But it becomes clearer if we use frequency sense

Signal representation

It's all oh so simple

- Once and again, things are invisible or obscure in time domain
  - ⇒ But they become, visible and clear in the frequency domain
- Even when clear in time, they are easier to understand in frequency
- Literally a new sense to view things that are otherwise invisible

"On ne voit bien qu'avec le coeur.
L'essentiel est invisible pour les yeux."

The Little Prince

- One sees clearly only with the frequency

Signal representation

- Why a new sense? ⇒ We can write signals as sums of shifted deltas

$$x(n) = \sum_{k=1}^{N} x(k)\delta(k-n)$$

- Conceptually, the same as writing signals as sums of oscillations

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

- Only difference is that we sense time but we don't sense frequency
- We say we change the signal representation or we change the basis
- It all hinges in our ability to represent the signal in a different domain

Moving forward

- If something is obscure in time but also obscure in frequency
  - ⇒ Change the representation ≡ Change the basis
- Images ⇒ multidimensional DFT, Discrete cosine transform (DCT)
- Stochastic processes ⇒ Principal component analysis (PCA)
  - ⇒ Eigenvectors of the correlation matrix
- Signals defined on graphs ⇒ Graph signal processing
  - ⇒ Eigenvalues of the graph Laplacian

Multidimensional Signal Processing

Mark Eisen, Alec Koppel, and Alejandro Ribeiro
Dept. of Electrical and Systems Engineering
University of Pennsylvania
aribeiro@seas.upenn.edu
http://www.seas.upenn.edu/users/~aribeiro/

March 27, 2015

Signal representation

It's all oh so simple

- Once and again, things are invisible or obscure in time domain
  - ⇒ But they become, visible and clear in the frequency domain
- Even when clear in time, they are easier to understand in frequency
- Literally a new sense to view things that are otherwise invisible

"On ne voit bien qu'avec le coeur.
L'essentiel est invisible pour les yeux."

The Little Prince

- One sees clearly only with the frequency
  The essential is invisible to the eyes

Signal representation

- Why a new sense? ⇒ We can write signals as sums of shifted deltas

$$x(n) = \sum_{k=1}^{N} x(k)\delta(k-n)$$

- Conceptually, the same as writing signals as sums of oscillations

$$x(n) = \sum_{k=1}^{N} X(k)e^{j2\pi kn/N}$$

- Only difference is that we sense time but we don't sense frequency
- We say we change the signal representation or we change the basis
- It all hinges in our ability to represent the signal in a different domain

Moving forward

- If something is obscure in time but also obscure in frequency
  - ⇒ Change the representation ≡ Change the basis
- Images ⇒ multidimensional DFT, Discrete cosine transform (DCT)
- Stochastic processes ⇒ Principal component analysis (PCA)
  - ⇒ Eigenvectors of the correlation matrix
- Signals defined on graphs ⇒ Graph signal processing
  - ⇒ Eigenvalues of the graph Laplacian

Images

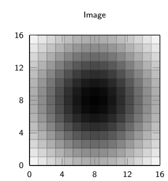
Images

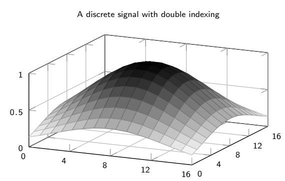
- A grid of pixels. Values define the luminescence of the point
  - ⇒ In a black an white image
- In a color image we record multiple channels for different colors
  - ⇒ E.g., red, green, and blue (RGB). Or Yellow Magenta Cyan blacK

- Not unlike signals we studied except that defined over two indices

Images as signals

- An image on the left and a signal on the right
  - ⇒ These are just different ways of visualizing the same information





- Can we perform DFT of image? ⇒ Yes, vectorize the matrix
- Vectorization records nearby pixels far away ⇒ 2D signal processing

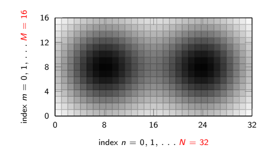
Two dimensional discrete signals

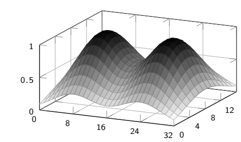
Two dimensional discrete signals

- Two dimensional (2D) discrete signal indexed by two indices (m, n)

$$m = 0, 1, \ldots, M-1 = [0, M-1]$$
$$n = 0, 1, \ldots, N-1 = [0, N-1]$$

- M rows and N columns. A total of MN different indices





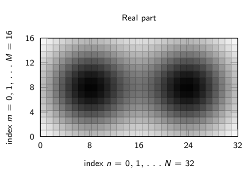
- 2D signal formally defined as map $x : [0, M-1] \times [0 : N-1] \to \mathbb{R}$
- The value that the signal takes at indices (m, n) is $x(m, n)$

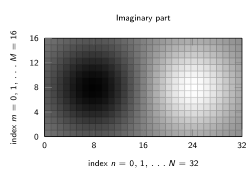
Complex 2D signals

- As in one dimensional case, may want to define complex signals

$$x : [0, M-1]x[0 : N-1] \to \mathbb{C}$$

- Space of $M\times$ 2D signals = space of $M \times N$ matrices $\mathbb{C}^{MxN}$ or $\mathbb{R}^{MxN}$



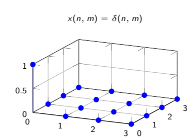


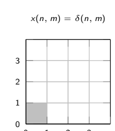
- Because, unsurprisingly, we are going to define two dimensional DFT

Deltas in two dimensions

- 2D delta function $\delta(m, n)$ is a spike at (initial) position $(m, n) = 0$

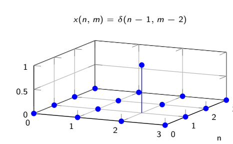
$$\delta(m, n) = \begin{cases} 1 & \text{if } m = n = 0 \\ 0 & \text{else} \end{cases}$$





- Shifted delta $\delta(m - m_0, n - n_0)$ has a spike at $(m, n) = (m_0, n_0)$

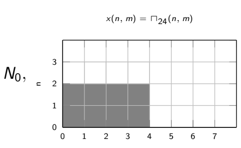
$$\delta(m - m_0, n - n_0) = \begin{cases} 1 & \text{if } (m, n) = (m_0, n_0) \\ 0 & \text{else} \end{cases}$$



Rectangular pulses

- Rectangular pulse of N rows and M columns $\sqcap_{M_0 N_0}$ is defined as

$$\sqcap_{M_0 N_0}(m, n) = \begin{cases} 1 & \text{if } m < M_0, n < N_0, \\ 0 & \text{else} \end{cases}$$

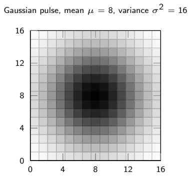


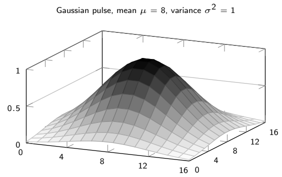
- If $M_0 = N_0$, rectangular pulse is said square. Denote $\sqcap_{N_0 N_0} = \sqcap_{N_0}$
- Can consider shifted pulses $\sqcap_{MN}(m - m_0, n - n_0)$
  - ⇒ Shifts must satisfy $m_0 < M - M_0$ and $n_0 < N - N_0$

Symmetric Gaussian pulses

- A 2D Gaussian pulse of mean μ and variance σ^2 is defined as

$$g_{\mu\sigma}(n,m) = \frac{1}{2\pi\sigma^2}\exp\left[-\frac{m-\mu}{2\sigma^2} - \frac{n-\mu}{2\sigma^2}\right]$$





- An actual bell shape. The pulse is symmetric centered at (μ, μ)
- Variance σ^2 controls how fast the pulse decays

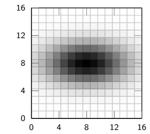
Generic Gaussian pulses

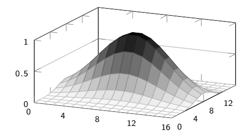
- Different centers in each coordinate and different variances
- Define coordinate vector $\mathbf{n} = [m, n]^T$. Just a variable
- Define center vector $\boldsymbol{\mu} = [\mu_1, \mu_2]$. Center coordinates
- Define covariance matrix $\mathbf{C} = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12} \\ \sigma_{12} & \sigma_{22}^2 \end{pmatrix}$
- Diagonal controls stretch in each direction. Off diagonals rotation
- The 2D Gaussian pulse of mean $\boldsymbol{\mu}$ and covariance $\mathbf{C}$ is

$$g_{\mu\sigma}(n,m) = \frac{1}{2\pi\sigma^2}\exp\left[-\frac{1}{2}(\mathbf{n}-\boldsymbol{\mu})^T\mathbf{C}^{-1}(\mathbf{n}-\boldsymbol{\mu})\right]$$

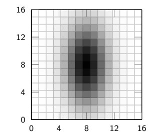
Generic Gaussian pulses

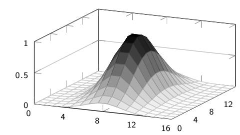
- A Gaussian pulse skewed in the m direction $\Rightarrow \mathbf{C} = \begin{pmatrix} 16 & 0 \\ 0 & 4 \end{pmatrix}$





- A Gaussian pulse skewed in the n direction $\Rightarrow \mathbf{C} = \begin{pmatrix} 4 & 0 \\ 0 & 16 \end{pmatrix}$





Inner product

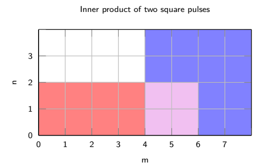
- Given 2D signals x and y define the inner product of x and y as

$$\langle x, y\rangle := \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x(m,n)y^*(m,n)$$

- It has the same properties of other inner products we encountered
  - $\Rightarrow$ Is a linear operator $\Rightarrow \langle x, y+z\rangle = \langle x, y\rangle + \langle x, z\rangle$
  - $\Rightarrow$ Reversing order entails conjugation $\Rightarrow \langle y, x\rangle = \langle x, y\rangle^*$
- It also has the same interpretation $\Rightarrow$ How much x looks like y
  - $\Rightarrow$ Positive = Positive correlation = same direction
  - $\Rightarrow$ Negative = Negative correlation = opposite directions
  - $\Rightarrow$ Null = Uncorrelated = Orthogonal = Perpendicular

Inner product of two rectangular pulses

- The inner product of two square pulses is the number of pixels in which both pulses are active (both are one)



- In the inner product sum $\langle x, y\rangle = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x(m,n)y^*(m,n)$ only the terms in which both pulses are not null count

Norm and energy

- The norm of the 2D signal x is $\Rightarrow \|x\| := \left[\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|x(m,n)|^2\right]^{1/2}$
- We define the energy of the 2D signal x as the norm squared

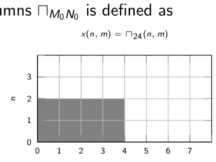
$$\|x\|^2 := \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|x(m,n)|^2 = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|x_R(m,n)|^2 + \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|x_I(m,n)|^2$$

- We can write the energy as self inner product $\Rightarrow \|x\|^2 = \langle x, x\rangle$

Energy of a square pulse

- Rectangular pulse of N rows and M columns $\sqcap_{M_0N_0}$ is defined as

$$\sqcap_{M_0N_0}(m,n) = \begin{cases} 1 & \text{if } m < M_0, n < N_0, \\ 0 & \text{else} \end{cases}$$



- To compute energy of the pulse we just evaluate the definition

$$\|\sqcap_N\|^2 := \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|\sqcap_{M_0N_0}(m,n)|^2 = \sum_{m=0}^{M_0-1}\sum_{n=0}^{N_0-1} 1^2 = M_0N_0$$

- The energy is the number of pixels (M_0N_0) in the square pulse
- Can normalize by $1/\sqrt{M_0N_0}$ to obtain pulse of unit energy

Two dimensional (2D) DFT

Signal representation

Images

Two dimensional discrete signals

Two dimensional (2D) discrete Fourier transform (DFT)

Two dimensional (2D) inverse (i) discrete Fourier transform (DFT)

Energy conservation (Parseval's theorem)

Convolution in 2 dimensions

Applications

Discrete Cosine Transform

2D Discrete Cosine Transform

JPEG image compression

Definition of 2D DFT

- 2D signal x With N rows and M columns. Elements $x(m,n)$
- We will focus on signals with $M = N$. To simplify notation
- Signal X is the 2D DFT of x if its elements $X(k,l)$ are

$$X(k,l) := \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)e^{-j2\pi(km+ln)/N}$$

- As in 1D we write $X = \mathcal{F}(x)$.
- X may be complex even for real 2D signals x. Focus on magnitude
- Argument k is horizontal frequency and l is the vertical frequency

The 2D DFT and the (regular, 1D) DFT

- Separate terms in the exponent and regroup factors to write

$$X(k,l) := \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \left[\frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(m,n) e^{-j2\pi ln/N} \right] e^{-j2\pi km/N}$$

- For fixed m, the term between parentheses is the DFT of $x(m,\cdot)$
- We then take the DFT of the resulting DFTs with respect to m
- The 2D DFT of x is the column-wise DFT of the row-wise DFTs
- Or the row-wise DFT of the column-wise DFTs. Just the same
- Useful to know. Not a new computation

Discrete Complex exponentials

- 2D Complex exponential of horizontal freq. k and vertical freq. l

$$e_{klN}(m,n) = \frac{1}{N} e^{-j2\pi(km+ln)/N} = \frac{1}{\sqrt{N}} e^{-j2\pi(km/N)} \frac{1}{\sqrt{N}} e^{-j2\pi(ln/N)}$$

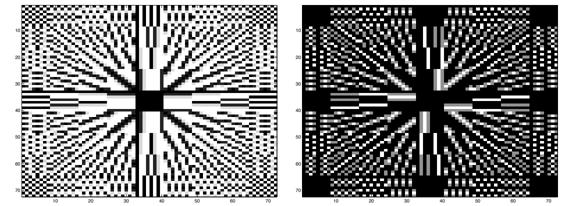
- Separate the exponential into two factors to write

$$e_{klN}(m,n) = \frac{1}{\sqrt{N}} e^{-j2\pi(km/N)} \frac{1}{\sqrt{N}} e^{-j2\pi(ln/N)} = e_{kN}(m) e_{lN}(n)$$

- 2D complex exponential is product of two 1D complex exponentials

How 2D complex exponentials look like

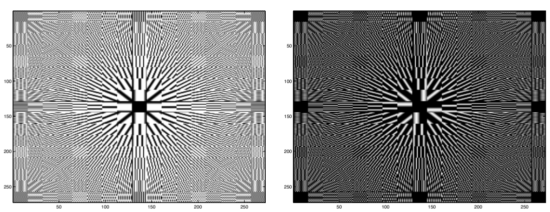
- Signal length $N = 8$. Total of $N^2 = 64$ different exponentials



- Horizontal / Vertical frequency $\Rightarrow$ Horizontal / Vertical variability
- Diagonals $\Rightarrow$ diagonal variabilty $\Rightarrow$ Directionality also important

How 2D complex exponentials look like

- Signal length $N = 16$. Total of $N^2 = 256$ different exponentials



- Horizontal / Vertical frequency $\Rightarrow$ Horizontal / Vertical variability
- Diagonals $\Rightarrow$ diagonal variabilty $\Rightarrow$ Directionality also important

DFT elements as inner products

- Rewrite 2D DFT using definition of 2D complex exponential

$$X(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n) e_{(-k)(-l)N}(m,n) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n) e^{*}_{klN}(m,n)$$

- From definition of inner product we have $\Rightarrow X(k,l) = \langle x, e_{klN} \rangle$
- DFT element $X(k,l)$ $\Rightarrow$ Inner product of $x(m,n)$ with $e_{kl,N}(m,n)$
- How much x is an oscillation of horizontal freq. k vertical freq. f
- 2D DFT contains information on rate of change as the 1D DFT
  $\Rightarrow$ But also in the direction of change

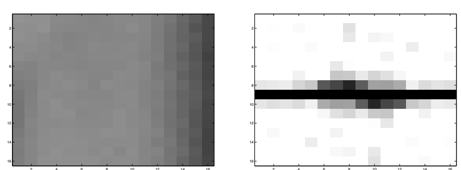
2D DFT of an image

- Lenna Sjööblom, playmate November 1972, Playboy magazine.
- And yet, we wonder why engineering is tough for women. Sorry.

- This is 256×256 image. We rarely do DFTs of full images
  $\Rightarrow$ Separate in 256 patches, each with 16×16 pixels

A patch and its 2D DFT

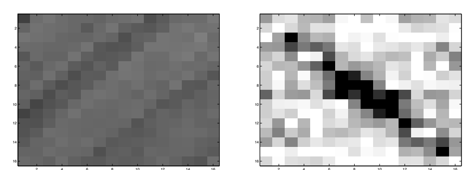
- Image patch on the left, 2D DFT coefficients on the right



- Signal mostly constant in vertical direction
  $\Rightarrow$ Large coefficients concentrated at low vertical frequencies

A patch and its 2D DFT

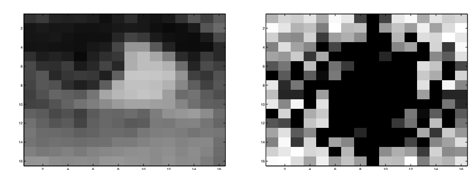
- Image patch on the left, 2D DFT coefficients on the right



- Signal changes diagonally from top left to bottom right
  $\Rightarrow$ Large coefficients on diagonal axis from top left to bottom right

A patch and its 2D DFT

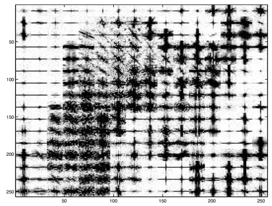
- Image patch on the left, 2D DFT coefficients on the right



- Signal shows variability in many different directions (piece of the eye)
  $\Rightarrow$ Large coefficients everywhere except when both freqs. are high

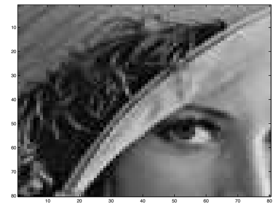
The DFT and variability

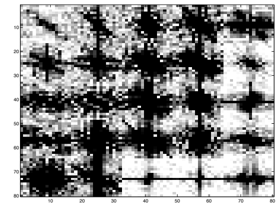
- The distribution of the 2D DFT coefficients captures variability
  - ⇒ Most coefficients are small on background patches
  - ⇒ Many coefficients are large on hat feathers patches



More on the the DFT and variability

- Many large coefficients on feather patches
- Large diagonal coefficients on hat ⇒ Direction of variability
- Face patches vary mostly in horizontal direction





Periodicity of complex exponentials

- We know that there are only N distinct complex exponentials
- Thus, there are only N^2 distinct 2D complex exponentials
  - ⇒ Horizontal frequencies k and $k+N$ are equivalent
  - ⇒ Vertical frequencies l and $l+N$ are equivalent
- Canonical sets $[0, N-1] \times [0, N-1]$ and $[-N/2, N/2] \times [-N/2, N/2]$
- 1D complex exponentials are conjugate symmetric. Thus

$$e_{(-k)(-l)N} \equiv e^*_{klN}$$

- Flipping sign of both freqs ≡ Conjugation of complex exponential

Periodicity of the 2D DFT

- Consider freqs (k, l) and $(k+N, l)$. DFT at $(k+N, l)$ is

$$X(k+N, l) = \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)e^*_{(k+N)lN}(m,n)$$

- Complex exponentials of freqs.(k, l) and $(k+N, l)$ are equivalent

$$X(k+N, l) = \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)e^*_{klN}(m,n) = X(k, l)$$

- 2D DFT has period N in horizontal direction.
- Likewise, 2D DFT has period N in vertical direction
- Suffices to look at $N \times N$ adjacent frequencies
- Canonical sets $[0, N-1] \times [0, N-1]$ and $[-N/2, N/2] \times [-N/2, N/2]$

Orthogonality of complex exponentials

Theorem
Complex exponentials with nonequivalent frequencies are orthogonal

$$\langle e_{klN}, e_{\tilde{k}\tilde{l}N}\rangle = \delta(k-\tilde{k})\delta(l-\tilde{l})$$

Proof.

- From definitions of inner product and discrete complex exponential

$$\langle e_{klN}, e_{pqN}\rangle = \frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} e^{-j2\pi(km+ln)/N}\left(e^{-j2\pi(\tilde{k}m+\tilde{l}n)/N}\right)^*$$

- Separate exponents and regroup factors

$$\langle e_{klN}, e_{pqN}\rangle = \frac{1}{N}\sum_{m=0}^{N-1} e^{-j2\pi km/N}\left(e^{-j2\pi\tilde{k}m/N}\right)^* \frac{1}{N}\sum_{n=0}^{N-1} e^{-j2\pi ln/N}\left(e^{-j2\pi\tilde{l}n/N}\right)^*$$

- Inner products of 1D exponentials. First is $\delta(k-\tilde{k})$, second is $\delta(l-\tilde{l})$ □

Two dimensional (2D) iDFT

Definition of 2D iDFT

- Given a Fourier transform X, the inverse (i)DFT $x = \mathcal{F}^{-1}(X)$ is

$$x(m,n) := \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} X(k,l)e^{j2\pi(km+ln)/N}$$

- Sum is over horizontal and vertical frequencies dimensions
- Recall that 2D DFT has period N in vertical and horizontal freqs.
- Any summation over $M \times N$ adjacent frequencies works as well. E.g.,

$$x(m,n) = \frac{1}{N}\sum_{k=-N/2+1}^{N/2}\sum_{l=-N/2+1}^{N/2} X(k,l)e^{j2\pi(km+ln)/N}$$

iDFT is, indeed, the inverse of the DFT

Theorem
The 2D inverse DFT $\tilde{x} = \mathcal{F}^{-1}(X)$ of the 2D DFT $X = \mathcal{F}(x)$ of any given signal x is the original signal x

$$\tilde{x} \equiv \mathcal{F}^{-1}(X) \equiv \mathcal{F}^{-1}(\mathcal{F}(x)) \equiv x$$

- Every 2D signal can be written as a sum of 2D complex exponentials

$$x(m,n) := \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} X(k,l)e^{j2\pi(km+ln)/N}$$

- The coefficient for horizontal frequency k and vertical frequency f is

$$X(k,l) := \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)e^{-j2\pi(km+ln)/N}$$

Proof of DFT inverse formula

Proof.

- To show $\tilde{x} \equiv x$ we prove $\tilde{x}(\tilde{m},\tilde{n}) = x(\tilde{m},\tilde{n})$ for all pairs of indices $(\tilde{m},\tilde{n})$
- From the definition of the 2D iDFT of X we write the value $\tilde{x}(\tilde{m},\tilde{n})$ as

$$\tilde{x}(\tilde{m},\tilde{n}) = \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} X(k,l)e^{j2\pi(k\tilde{m}+l\tilde{n})/N}$$

- From the definition of the 2D DFT of x we write the DFT value $X(k,l)$ as

$$X(k,l) := \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)e^{-j2\pi(km+ln)/N}$$

- Substituting expression for $X(k,l)$ into expression for $\tilde{x}(\tilde{n},\tilde{m})$ yields

$$\tilde{x}(\tilde{m},\tilde{n}) = \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\left[\frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)e^{-j2\pi(km+ln)/N}\right]e^{j2\pi(k\tilde{m}+l\tilde{n})/N}$$

Proof of DFT inverse formula

Proof.

- Exchange summation order, pull out $x(m,n)$, and distribute $1/N$ factors

$$\tilde{x}(\tilde{m},\tilde{n}) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)\left[\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} \frac{1}{N} e^{-j2\pi(km+ln)/N} \frac{1}{N} e^{j2\pi(k\tilde{m}+l\tilde{n})/N}\right]$$

- Can pull $x(m,n)$ out because it doesn't depend neither on k nor on l
- Innermost sum is inner product between $e_{\tilde{m}\tilde{n}N}$ and e_{mnN}. Orthonormality:

$$\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} \frac{1}{N} e^{-j2\pi(km+ln)/N} \frac{1}{N} e^{j2\pi(k\tilde{m}+l\tilde{n})/N} = \langle e_{\tilde{m}\tilde{n}N}, e_{mnN}\rangle = \delta(\tilde{m}-m)\delta(\tilde{n}-n)$$

- Reducing to $\Rightarrow \tilde{x}(\tilde{m},\tilde{n}) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x(m,n)\delta(\tilde{n}-n)\delta(\tilde{m}-m) = x(\tilde{m},\tilde{n})$
- Last equation true because only term $m=\tilde{m}, n=\tilde{n}$ is not null in the sum

□

The 2D DFT sense

- Can write image x as sum of deltas modulated by individual pixels

$$x(m,n) := \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} x(m,n)\delta(k-m, l-n)$$

- Also write as sum of oscillations modulated by 2D DFT coefficients

$$x(m,n) := \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} X(k,l)e^{j2\pi(km+ln)/N}$$

- These are mathematically analogous expressions.
- We can see (literally) pixels, but we can't see 2D DFT coefficients
- Easier to operate on the image, when written as sum of oscillations

Reconstruction of an image

- Separate in 16×16 patches (256 total). Compute 2D DFT of each patch
- Start with low frequencies and work up to larger frequencies

-

Reconstruction of an image

- Separate in 16×16 patches (256 total). Compute 2D DFT of each patch
- Start with low frequencies and work up to larger frequencies

- Reconstruction when using frequencies $-1 \leq k,l \leq 1$. Not too good

Reconstruction of an image

- Separate in 16×16 patches (256 total). Compute 2D DFT of each patch
- Start with low frequencies and work up to larger frequencies

- Reconstruction when using frequencies $-2 \leq k,l \leq 2$. Not bad

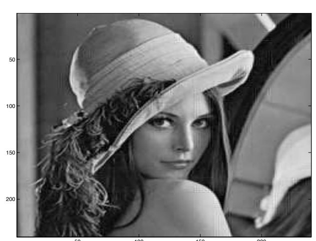
Reconstruction of an image

- Separate in 16×16 patches (256 total). Compute 2D DFT of each patch
- Start with low frequencies and work up to larger frequencies

- Using frequencies $-4 \leq k,l \leq 4$. Quite good, except for border effect

Reconstruction of an image

- Separate in 16×16 patches (256 total). Compute 2D DFT of each patch
- Start with low frequencies and work up to larger frequencies



- Freqs. $-7 \leq k,l \leq 7$. Border effect still present. Will solve later (DCT)

Energy conservation (Parseval's theorem)

Properties of the 2D DFT

- All properties of 1D DFTs have corresponding versions for 2D DFTs
  $\Rightarrow$ Linearity, conjugate symmetry, modulation $\Leftrightarrow$ shift
- We will cover energy conservation (to study compression)

$$\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} |x(m,n)|^2 = \|x\|^2 = \|X\|^2 = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} |X(k,l)|^2$$

- Will also cover the 2D convolution theorem (to study linear filtering)

$$y = x * h \quad \Longleftrightarrow \quad Y = HX$$

- Which will require defining the 2D convolution operation $x * h$

Energy conservation

Theorem (Parseval)

The energies of a signal x and its 2D DFT $X = \mathcal{F}(x)$ are the same, i.e.,

$$\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} |x(m,n)|^2 = \|x\|^2 = \|X\|^2 = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} |X(k,l)|^2$$

- Since 2D DFT is periodic, any set of adjacent freqs. would do. E.g.,

$$\|X\|^2 = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} |X(k,l)|^2 = \sum_{k=-M/2+1}^{M/2}\sum_{l=-N/2+1}^{N/2} |X(k,l)|^2$$

- From now on, we write $\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}(\cdot) = \sum_{m,n}(\cdot)$ and $\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}(\cdot) = \sum_{k,l}(\cdot)$
- To simplify notation. We would otherwise write up to six sums

Proof of Parseval's Theorem

Proof.

- The energy of the 2D DFT X is $\Rightarrow \|X\|^2 = \sum_{k,l} X(k,l)X^*(k,l)$
- The 2D DFT of x is $\Rightarrow X(k,l) := \frac{1}{N}\sum_{m,n} x(m,n)e^{-j2\pi(km+ln)/N}$
- Substitute expression for $X(k,l)$ into one for $\|X\|^2$ (observe conjugation)

$$\|X\|^2 = \sum_{k,l}\left[\left[\frac{1}{N}\sum_{m,n} x(m,n)e^{-j2\pi(km+ln)/N}\right]\left[\frac{1}{N}\sum_{\tilde{m},\tilde{n}} x^*(\tilde{m},\tilde{n})e^{+j2\pi(k\tilde{m}+l\tilde{n})/N}\right]\right]$$

- Distribute product, exchange sum order, pull $x(m,n)$ and $x^*(\tilde{m},\tilde{n})$ out

$$\|X\|^2 = \sum_{m,n}\sum_{\tilde{m},\tilde{n}} x(m,n)x^*(\tilde{m},\tilde{n})\left[\sum_{k,l}\frac{1}{N}e^{-j2\pi(km+ln)/N}\frac{1}{N}e^{+j2\pi(k\tilde{m}+l\tilde{n})/N}\right]$$

- Can pull out because $x(m,n)$ and $x^*(\tilde{m},\tilde{n})$ don't depend on (k,l)

Proof of Parseval's Theorem

Proof.

- Innermost sum is inner product between $e_{\tilde{m}\tilde{n}N}$ and e_{mnN}. Orthonormality:

$$\sum_{k,l}\frac{1}{N}e^{-j2\pi(km+ln)/N}\frac{1}{N}e^{j2\pi(k\tilde{m}+l\tilde{n})/N} = \langle e_{\tilde{m}\tilde{n}N}, e_{mnN}\rangle = \delta(\tilde{m}-m, \tilde{n}-n)$$

- Substitute $\delta(\tilde{m}-m, \tilde{n}-n)$ for innermost sum to simplify $\|X\|^2$ to

$$= \sum_{m,n}\sum_{\tilde{m},\tilde{n}} x(m,n)x^*(\tilde{m},\tilde{n})\delta(\tilde{m}-m, \tilde{n}-n) = \sum_{m,n} x(m,n)x^*(m,n)$$

- True because only terms with $m = \tilde{m}$ and $n = \tilde{n}$) are not null in the sum
- Conclude by noting that from definition of the energy of x, we have

$$\|X\|^2 = \sum_{m,n} x(m,n)x^*(m,n) = \|x\|^2$$ □

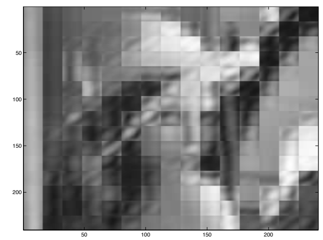
Reconstruction of an image

- Separate in 16×16 patches (256 total). Compute 2D DFT of each patch
- Start with low frequencies and work up to larger frequencies

- Energy of approximation error $\equiv$ Energy of 2D DFT coefficients dropped

Reconstruction of an image

- Separate in 16×16 patches (256 total). Compute 2D DFT of each patch
- Start with low frequencies and work up to larger frequencies



- Energy of reconstruction error $\Rightarrow$ 32% of image's energy (4 coefficients)

Reconstruction of an image

- Separate in 16×16 patches (256 total). Compute 2D DFT of each patch
- Start with low frequencies and work up to larger frequencies

- Energy of reconstruction error $\Rightarrow$ 9% of image's energy (16 coefficients)

Reconstruction of an image

- Separate in 16×16 patches (256 total). Compute 2D DFT of each patch
- Start with low frequencies and work up to larger frequencies

- Energy of reconstruction error $\Rightarrow$ 2% of image's energy (64 coefficients)

Convolution in 2 dimensions

2D Convolution

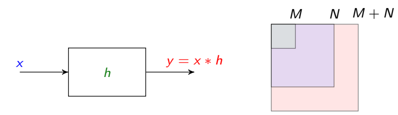
- Given 2D signal x of length $N \times N$ and filter h of length $M \times M$
- Reinterpret filter h as being null for all integers outside its range

$$h(m,n) = 0, \quad \text{for all } (m,n) \notin [0, M-1] \times [0, M-1]$$

- Convolution of x and h is the $(N+M) \times (N+M)$ signal $y = x * h$

$$y(m,n) = \sum_{p=0}^{N}\sum_{q=0}^{N} x(p,q)h(m-p, n-q)$$

- Hit filter h with input x to generate output y



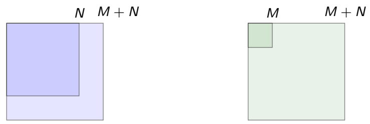
Padded signals

- The padded signal $\bar{x}$ is an $(N+M)\times(N+M)$ signal with

$$\bar{x}(m,n) = x(m,n), \quad \text{for } (m,n)\in[0,N-1]\times[0,N-1]$$
$$\bar{x}(m,n) = 0, \quad \text{else}$$

- The padded filter $\bar{h}$ is an $(N+M)\times(N+M)$ signal with

$$\bar{h}(m,n) = h(m,n), \quad \text{for } (m,n)\in[0,M-1]\times[0,M-1]$$
$$\bar{h}(m,n) = 0, \quad \text{else}$$



2D convolution theorem

- 2D DFTs of padded signal $\bar{X} = \mathcal{F}(\bar{x})$ and padded filter $\bar{H} = \mathcal{F}(\bar{h})$
- Regular DFT of output signal, $Y = \mathcal{F}(y)$

Theorem (2D Convolution)

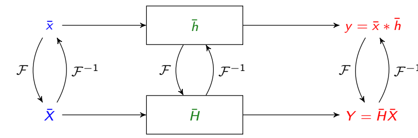
The convolution of padded signals in the space domain is equivalent to the multiplication of their 2D DFTs in the frequency domain

$$y = \bar{x} * \bar{h} \quad \Longleftrightarrow \quad Y = \bar{X}\bar{H}$$

- Transformation is obscure in space but crystal clear in frequency

Design in frequency, implement in space

- As we did in 1D, we design in frequency but implement in space



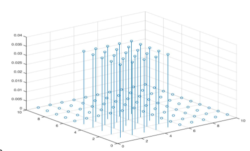
- Convolution doesn't change with padding $\Rightarrow y = \bar{x} * \bar{h} = x * h$
- 2D DFTs do change, but not by much when $M \ll N$
- Instead of padding x and h we crop y to make it $N\times N \Rightarrow \tilde{y}$
- Convolution theorem becomes approximate $\Rightarrow \tilde{Y} \approx HX$
  - $\Rightarrow$ There are differences close to the borders of the image

Applications

Averaging Filter

- An averaging filter is one with a square frequency response

$$h(m,n) = \frac{1}{M^2}\sqcap_M(m,n)$$



- The convolution $y = h * x$ is an average of adjacent pixels

$$y(m,n) = \frac{1}{M^2}\sum_{p=0}^{M-1}\sum_{q=0}^{M-1} x(m+p, n+q)$$

- What effect does an averaging filter has when applied to an image?

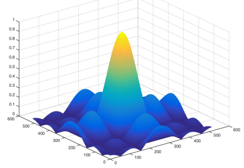
Image Blurring

- Averaging neighboring pixels has the effect of blurring the image


- What is the counterpart of blurring in the frequency domain?

Average Filter in frequency domain

- The 2D DFT of a 2D square pulse is a 2D sinc $\Rightarrow$ low pass filter



- Blurring entail removal of high frequencies (in all directions)
  - $\Rightarrow$ Smoothes edges, which makes image appear out of focus

Image Denoising

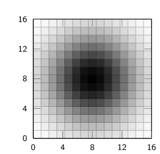
- Image is corrupted by white noise $\Rightarrow$ equal power at all frequencies

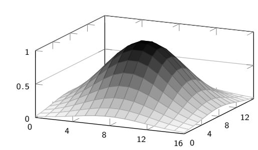

- Can remove noise with averaging filter $\Rightarrow$ Only low frequencies pass
  - $\Rightarrow$ Image has low frequencies only. Noise has all frequencies

Gaussian filter

- Or, apply 2D Gaussian filter $\Rightarrow$ 2D Gaussian pulse impulse response

$$h(n,m) = g_{\mu\sigma}(n,m) = \frac{1}{2\pi\sigma^2}\exp\left[-\frac{(m-\mu)^2}{2\sigma^2} - \frac{(n-\mu)^2}{2\sigma^2}\right]$$





- 2D Gaussian pulse also performs averaging with nearby pixels
- Also low pass $\Rightarrow$ 2D DFT is Gaussian pulse with inverse variance
  - $\Rightarrow$ Decrease σ^2 to let more frequencies pass

Gaussian filtering of a noisy image

- Remove noise with a Gaussian filter with variance $\sigma^2 = 1$

Noisy image | Filtered image

- Some noise is removed. Can remove more by increasing variance σ^2

Low pass filter of noisy image

- Remove noise with a Gaussian filter with variance $\sigma^2 = 4$

Noisy image | Filtered image

- More noise removed (good), but also more blurring (not good)

Edge detection

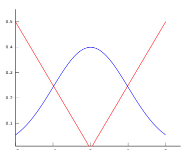
- Detect the edges of an image $\Rightarrow$ Rapid transitions
  $\Rightarrow$ A rapid transition is a high frequency $\Rightarrow$ Use a high pass filter

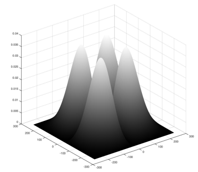
Gaussian Derivative Filter

- Multiply Gaussian filter frequency response by inverted pyramid

$$H(k,l) = G_{\mu\sigma}(k,l)|k+l|$$

- Derivative filter because freq. multiplication is derivation in space





- Very rapid variations are filtered out. They are regarded as noise
- Rapid, but now rapid variations are considered edges

Edge Detection

- Now applying this filter to our test image:

- After filter, only high frequencies (edges) remain in image

Image Sharpening

- We want to sharpen an image, e.g., because it's blurry, out of focus
  $\Rightarrow$ We can do that by heightening the edges
- Low frequencies are still important
  $\Rightarrow$ Want to boost high frequencies, as opposed to detecting them
- Add a constant α in frequency to let all frequencies pass

$$H(k,l) = (1-\alpha)\, G_{\mu\sigma}(k,l)|k+l| + \alpha$$

- In time, the constant is a delta $\Rightarrow$ we add the signal and the edges

Image Sharpening

- Increasing sharpening makes borders more defined

Discrete Cosine Transform

Border effects in image compression

- Patches are well approximated by a subset of 2D DFT coefficients
- Except for borders. And still a problem if we retain most coefficients

- Although didn't mention, also a problem with (1D) DFTs $\Rightarrow$ Why?

The DFT and the iDFT

- Start with real signal $x : [0, N-1] \to \mathbb{R}$. The DFT of signal x is

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

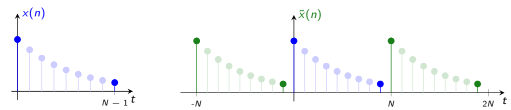
- We can recover x with the iDFT transformation defined by

$$\tilde{x}(n) := \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N}$$

- We know that $\tilde{x}(n) = x(n)$ for $n \in [0, N-1]$ (inverse transform)
- But the iDFT is defined for all n
- Signal $\tilde{x}$ is periodic with period N because exponentials $e^{j2\pi kn/N}$ are
  - $\Rightarrow$ We say that iDFT signal $\tilde{x}$ is a periodic extension of original x

The periodic extension of the iDFT

- First sample $x(0)$ and last sample $x(N-1)$ can be very different
  - $\Rightarrow$ Most likely are. Unless signal has some structure, e.g., symmetry
- This is a problem for the periodic extension
  - $\Rightarrow$ The value $x(0) = \tilde{x}(N)$ appears next to $x(N-1) = \tilde{x}(N-1)$



- It's tough to approximate a jump/discontinuity $\Rightarrow$ High frequency
- Never mind. We're more than Fourier people. We're fearless transformers

Inverse discrete cosine transform

- Say that we have a transform X so that we can write signal $\tilde{x}$ as

$$\tilde{x}(n) := \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(n+1/2)}{N}\right]$$

- Inverse discrete cosine transform (iDCT) of X $\Rightarrow \tilde{x} = \mathcal{C}^{-1}(X)$
- No complex numbers involved. Signals and transforms assumed real
- Haven't said how to find X so that $\tilde{x}(n) = x(n)$ for $n \in [0, N-1]$
- This is done with discrete cosine transform (DCT). We'll see later
- Details are different but this is still x written as a sum of oscillations
  - $\Rightarrow$ Still expect low frequency components to be most significant
  - $\Rightarrow$ But have written cosine in a way that avoids border discontinuities

The iDCT is an even function

- Put a mirror at $N + 1/2$ and compare samples in each direction
- The sample at $n = N-1$ can be written in terms of iDCT as

$$\tilde{x}(N-1) := \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(N-1+1/2)}{N}\right]$$
$$= \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(-1/2)}{N}\right]$$

- The sample at $n = N$ can be written in terms of iDCT as

$$\tilde{x}(N) := \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(N+1/2)}{N}\right]$$
$$= \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(+1/2)}{N}\right]$$

- Since cosines are even, sign is irrelevant. Thus $\Rightarrow \tilde{x}(N-1) = \tilde{x}(N)$

The iDCT is an even function

- Put a mirror at $N + 1/2$ and compare samples in each direction
- The sample at $n = N-2$ can be written in terms of iDCT as

$$\tilde{x}(N-1) := \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(N-2+1/2)}{N}\right]$$
$$= \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(-3/2)}{N}\right]$$

- The sample at $n = N+1$ can be written in terms of iDCT as

$$\tilde{x}(N+1) := \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(N+1+1/2)}{N}\right]$$
$$= \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(+3/2)}{N}\right]$$

- Since cosines are even, sign is irrelevant. Thus $\Rightarrow \tilde{x}(N-2) = \tilde{x}(N+1)$

The iDCT is an even function

- Put a mirror at $N + 1/2$ and compare samples in each direction
- The sample at $n = N-3$ can be written in terms of iDCT as

$$\tilde{x}(N-3) := \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(N-3+1/2)}{N}\right]$$
$$= \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(-5/2)}{N}\right]$$

- The sample at $n = N+2$ can be written in terms of iDCT as

$$\tilde{x}(N+2) := \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(N+2+1/2)}{N}\right]$$
$$= \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(+5/2)}{N}\right]$$

- Since cosines are even, sign is irrelevant. Thus $\Rightarrow \tilde{x}(N-3) = \tilde{x}(N+2)$

The iDCT is an even function

- Put a mirror at $N + 1/2$ and compare samples in each direction
- The sample at $n = N-4$ can be written in terms of iDCT as

$$\tilde{x}(N-4) := \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(N-4+1/2)}{N}\right]$$
$$= \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(-7/2)}{N}\right]$$

- The sample at $n = N+2$ can be written in terms of iDCT as

$$\tilde{x}(N+3) := \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(N+3+1/2)}{N}\right]$$
$$= \frac{1}{\sqrt{N}} X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} X(k) \cos\left[\frac{2\pi k(+7/2)}{N}\right]$$

- Since cosines are even, sign is irrelevant. Thus $\Rightarrow \tilde{x}(N-4) = \tilde{x}(N+3)$

The even extension of the iDCT

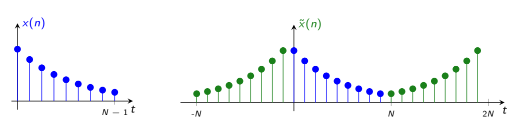
- Formalize argument to prove that the iDCT yields an even extension

$$\tilde{x}\Big[N + (n-1)\Big] = x\Big[N - n\Big]$$

- Or, to better visualize the symmetry

$$\tilde{x}\Big[(N-1/2) + (n-1/2)\Big] = x\Big[(N-1/2) - (n-1/2)\Big]$$

- Signal x written as sum of oscillations without border effects



The discrete cosine transform (DCT)

- Still have to find out a way of computing the coefficients $X(k)$
- Given a real signal x, the DCT $X = \mathcal{C}(x)$ is the real signal with

$$X(0) := \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) \cos\left[\frac{2\pi 0(n+1/2)}{N}\right]$$
$$X(k) := \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos\left[\frac{2\pi k(n+1/2)}{N}\right]$$

- Normalization constants are different for $k = 0$ and $k \in [1, N-1]$
- No complex numbers involved. Signals and transforms are real

DCT basis

- Define the elements of the DCT basis as the signals e_{kN} with

$$c_{0N}(n) := \frac{1}{\sqrt{N}} \qquad c_{kN}(n) := \sqrt{\frac{2}{N}}\cos\left[\frac{2\pi k(n+1/2)}{N}\right]$$

- Akin to the DFT basis defined by the N complex exponentials e_{kN}
- With basis defined can write DCT of x as $\Rightarrow X(k) = \langle x, c_{kN}\rangle$
- Inner product implies the usual interpretation
  $\Rightarrow$ X(k) is how much $x(n)$ resembles oscillation of frequency k

iDCT is the inverse of the DCT

Theorem
The iDCT $\tilde{x} = \mathcal{C}^{-1}(X)$ of the DCT $X = \mathcal{C}(x)$ of any given signal x is the original signal x, i.e.,

$$\tilde{x} \equiv \mathcal{C}^{-1}(X) \equiv \mathcal{C}^{-1}(\mathcal{C}(x)) \equiv x$$

- Equivalence means $\tilde{x}(n) = x(n)$ for $n \in [0, N-1]$.
  $\Rightarrow$ Otherwise, inverse transform $\tilde{x}$ is an even extension of original x
- To prove theorem, use DCT definition, iDCT definition, reverse summation order, and invoke orthogonality of the DCT basis.
- Conservation of energy (Parseval's) also holds $\Rightarrow$ orthogonality

2D Discrete Cosine Transform

Rewriting the 1D DCT

- For 1D signal x we defined the 1D DCT $X = \mathcal{C}(x)$ as

$$X(0) := \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} x(n)\cos\left[\frac{2\pi 0(n+1/2)}{N}\right]$$

$$X(k) := \sqrt{\frac{2}{N}}\sum_{n=0}^{N-1} x(n)\cos\left[\frac{2\pi k(n+1/2)}{N}\right]$$

- Define normalization constants $\nu_0 = 1$ and $\nu_k = \sqrt{2}$ for $k \neq 0$

$$X(k) := \frac{\nu_k}{\sqrt{N}}\sum_{n=0}^{N-1} x(n)\cos\left[\frac{2\pi k(n+1/2)}{N}\right]$$

- Just a definition to make notation more compact

Two dimensional discrete cosine transform

- Given a two dimensional signal x we define the 2D DCT X as

$$X(k,l) := \frac{\nu_k\nu_l}{N}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} x(m,n)\cos\left[\frac{2\pi k(m+1/2)}{N}\right]\cos\left[\frac{2\pi l(n+1/2)}{N}\right]$$

- 2D analogous of the 1D DCT. Or DCT analogous of the 2D DFT
- Can write the double sum as a pair of nested sums

$$X(k,l) := \frac{\nu_k\nu_l}{N}\sum_{n=0}^{N-1}\left[\sum_{m=0}^{N-1} x(m,n)\cos\left[\frac{2\pi k(m+1/2)}{N}\right]\right]\cos\left[\frac{2\pi l(n+1/2)}{N}\right]$$

- The 2D DCT is the vertical DCT of the horizontals DCTs
- Equivalently, it is also the horizontal DCT of the vertical DCTs

2D DCT as an inner product

- The 2D discrete cosine of horizontal freq. k and vertical freq. l is

$$c_{klN}(n,m) := \frac{c_k}{\sqrt{N}}\cos\left[\frac{2\pi k(m+1/2)}{N}\right]\frac{c_l}{\sqrt{N}}\cos\left[\frac{2\pi l(n+1/2)}{N}\right]$$

- Use to rewrite 2D DCT as inner product $\Rightarrow X(k,l) = \langle x, c_{kl,N}\rangle$
- The 2D DCT element $X(k,l)$ is the inner product of x with $c_{kl,N}$
- Observe that, similar to the 2D complex exponentials, we can write

$$c_{klN}(n,m) = c_{lN}c_{lN}$$

- Which implies orthonormality of the c_{klN}. Because the c_{kN} are

Rewrite the 1D iDCT

- For given DCT X we defined the iDCT as the signal $\tilde{x}$ with values

$$\tilde{x}(n) := \frac{1}{\sqrt{N}}X(0) + \sqrt{\frac{2}{N}}\sum_{k=1}^{N-1} X(k)\cos\left[\frac{2\pi k(n+1/2)}{N}\right]$$

- Use the same constants, $\nu_0 = 1$ and $\nu_k = \sqrt{2}$ for $k \neq 0$, to write

$$\tilde{x}(n) := \sum_{k=1}^{N-1}\frac{\nu_k}{\sqrt{N}}X(k)\cos\left[\frac{2\pi k(n+1/2)}{N}\right]$$

- Just a definition. To avoid writing four separate sums for 2D iDCT

Two dimensional inverse discrete cosine transform

- Given a 2D DCT X we define the 2D iDCT $\tilde{x}$ as

$$\tilde{x}(m,n) := \sum_{n=0}^{N-1}\sum_{m=0}^{N-1}\frac{\nu_k\nu_l}{N}X(k,l)\cos\left[\frac{2\pi k(m+1/2)}{N}\right]\cos\left[\frac{2\pi l(n+1/2)}{N}\right]$$

- 2D analogous of the 1D DCT. Or DCT analogous of the 2D DFT
- The 2D iDCT is even symmetric (not periodic). In both dimensions

$$\tilde{x}\Big[(N-1/2)+(m-1/2), n\Big] = x\Big[(N-1/2)-(m-1/2), n\Big]$$

$$\tilde{x}\Big[m, (N-1/2)+(n-1/2)\Big] = x\Big[m, (N-1/2)-(n-1/2)\Big]$$

- Thus, we don't have border effects in the reconstruction. Later

iDCT is the inverse of the DCT

Theorem
The iDCT $\tilde{x} = \mathcal{C}^{-1}(X)$ of the DCT $X = \mathcal{C}(x)$ of any given signal x is the original signal x, i.e.,

$$\tilde{x} \equiv \mathcal{C}^{-1}(X) \equiv \mathcal{C}^{-1}(\mathcal{C}(x)) \equiv x$$

- Equivalence means $\tilde{x}(n) = x(n)$ for $n \in [0, N-1]$.
  $\Rightarrow$ Otherwise, inverse transform $\tilde{x}$ is an even extension of original x
- To prove theorem, use DCT definition, iDCT definition, reverse summation order, and invoke orthogonality of the DCT basis.
- Conservation of energy (Parseval's) also holds $\Rightarrow$ orthogonality

Compression with the 2D DCT and 2D iDCT

- Compute 2D DCT of 16×16 patches. Reconstruct with low frequencies
- The signal is reconstructed with small error and no border effects

-

Compression with the 2D DCT and 2D iDCT

- Compute 2D DCT of 16×16 patches. Reconstruct with low frequencies
- The signal is reconstructed with small error and no border effects

- Reconstruction when using coefficients $0 \leq k, l \leq 4$. Not too good
- Compression factor 16 and error energy 1.59%

Compression with the 2D DCT and 2D iDCT

- Compute 2D DCT of 16×16 patches. Reconstruct with low frequencies
- The signal is reconstructed with small error and no border effects

- Reconstruction when using coefficients $0 \leq k, l \leq 6$. Quite good
- Compression factor 7.1 and error energy 0.81%

Compression with the 2D DCT and 2D iDCT

- Compute 2D DCT of 16×16 patches. Reconstruct with low frequencies
- The signal is reconstructed with small error and no border effects

- Reconstruction when using coefficients $0 \leq k, l \leq 8$. Excellent
- Compression factor 4 and error energy 0.46%

Compression with the 2D DCT and 2D iDCT

- Compute 2D DCT of 16×16 patches. Reconstruct with low frequencies
- The signal is reconstructed with small error and no border effects

- Reconstruction when using coefficients $0 \leq k, l \leq 10$. Flawless
- Compression factor 2.56 and error energy 0.26%

JPEG image compression

JPEG image compression

- Start with a color image $\Rightarrow$ three color channels x_R, x_B, x_G
  - $\Rightarrow$ Each pixel is represented by 8 bits
  - $\Rightarrow$ Values are integers in [0,255], or, equivalently [-127,128]
- Transform into luminance y and chrominance y_R and y_B
- Eye more sensitive to luminance. Sample chrominances every 2 pixels
- Work with luminance and chrominance separately.
- Separate each channel in 8×8 patches $\Rightarrow$ 64 pixels per patch
- For each patch x, compute the corresponding DCT X
  - $\Rightarrow$ Keep coefficients associated with largest frequency components
- Low frequencies more important but high frequencies not irrelevant
  - $\Rightarrow$ Introduce importance quantization

Importance quantization

- For each frequency pair k, l, define the importance coefficient $Q(k, l)$
- Encode each DCT frequent component as

$$\hat{X}(k, l) = \text{round}\left(\frac{X(k, l)}{Q(k, l)}\right)$$

- If $Q(k, l) \approx 1$ there is little change $\Rightarrow \hat{X}(k, l) \approx X(k, l)$
- If $Q(k, l)$ is large we reduce the range of $\hat{X}(k, l)$
- Numbers with smaller range can be encoded with less bits
  - $\Rightarrow$ Assign relatively small $Q(k, l)$ to low frequencies
  - $\Rightarrow$ Assign relatively large $Q(k, l)$ to high frequencies

Importance matrix

- The importance coefficients $Q(k, l)$ form the importance matrix $\mathbf{Q}$
  - $\Rightarrow$ Up to 20. Up to 50. Up to 90. More than 90.

$$\mathbf{Q} = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 36 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

- Instead of top left square, we assign importance to top left triangle
- Slight asymmetry $\Rightarrow$ More importance to horizontal frequencies
- All frequency components encoded to some extent
  - $\Rightarrow$ High frequency components encoded only when they are large

Principal Component Analysis

Aryan Mokhtari, Santiago Paternain, and Alejandro Ribeiro
Dept. of Electrical and Systems Engineering
University of Pennsylvania
aribeiro@seas.upenn.edu
http://www.seas.upenn.edu/users/~aribeiro/

April 10, 2015

The DFT and iDFT with Hermitian matrices

The discrete Fourier transform, again

- It is time to write and understand the DFT in a more abstract way
- Write signal x and complex exponential e_{kN} as vectors $\mathbf{x}$ and $\mathbf{e}_{kN}$

$$\mathbf{x} = \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{pmatrix} \qquad \mathbf{e}_{kN} = \frac{1}{\sqrt{N}} \begin{pmatrix} e^{j2\pi k0/N} \\ e^{j2\pi k1/N} \\ \vdots \\ e^{j2\pi k(N-1)/N} \end{pmatrix}$$

- Use vectors to write the kth DFT component as $(\mathbf{e}_{kN}^H = (\mathbf{e}_{kN}^*)^T)$

$$X(k) = \mathbf{e}_{kN}^H \mathbf{x} = \langle \mathbf{x}, \mathbf{e}_{kN} \rangle = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$$

- kth DFT component $X(k)$ is the product of $\mathbf{x}$ with exponential $\mathbf{e}_{kN}^H$

DFT in matrix notation

- Write DFT $\mathbf{X}$ as a stacked vector and stack individual definitions

$$\mathbf{X} = \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{0N}^H \mathbf{x} \\ \mathbf{e}_{1N}^H \mathbf{x} \\ \vdots \\ \mathbf{e}_{(N-1)N}^H \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{0N}^H \\ \mathbf{e}_{1N}^H \\ \vdots \\ \mathbf{e}_{(N-1)N}^H \end{bmatrix} \mathbf{x}$$

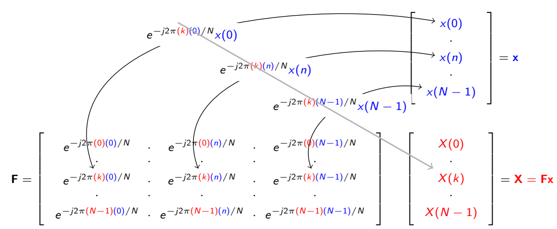
- Define the DFT matrix $\mathbf{F}^H$ so that we can write $\mathbf{X} = \mathbf{F}\mathbf{x}$

$$\mathbf{F} = \begin{bmatrix} \mathbf{e}_{0N}^H \\ \mathbf{e}_{1N}^H \\ \vdots \\ \mathbf{e}_{(N-1)N}^H \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-j2\pi(1)(1)/N} & \cdots & e^{-j2\pi(1)(N-1)/N} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi(N-1)(1)/N} & \cdots & e^{-j2\pi(N-1)(N-1)/N} \end{bmatrix}$$

- The DFT of signal x is a matrix multiplication $\Rightarrow \mathbf{X} = \mathbf{F}\mathbf{x}$

The DFT as a matrix product

- Indeed, in case you are having trouble visualizing the matrix product



- The kth DFT component $X(k)$ is the kth row of matrix product $\mathbf{F}\mathbf{x}$

Some properties of the DFT matrix F

- The (k, n)th element of the matrix $\mathbf{F}$ is the complex exponential

$$\big((\mathbf{F})\big)_{kn} = e^{-j2\pi(k)(n)/N} = \left(e^{-j2\pi(k)/N}\right)^{(n)}$$

- Since elements of rows are indexed powers we say $\mathbf{F}$ is Vandermonde
- Also observe that since $e^{-j2\pi(k)(n)/N} = e^{-j2\pi(n)(k)/N}$ we have

$$\big((\mathbf{F})\big)_{kn} = e^{-j2\pi(k)(n)/N} = e^{-j2\pi(n)(k)/N} \big((\mathbf{F})\big)_{nk}$$

- The DFT matrix $\mathbf{F}$ is symmetric $\Rightarrow \mathbf{F}^T = \mathbf{F}$
- Can then write $\mathbf{F}$ as $\Rightarrow \mathbf{F} = \mathbf{F}^T = \begin{bmatrix} \mathbf{e}_{0N}^* & \mathbf{e}_{1N}^* & \cdots & \mathbf{e}_{(N-1)N}^* \end{bmatrix}$

The Hermitian of the DFT matrix F

- Let $\mathbf{F}^H = (\mathbf{F}^*)^T$ be conjugate transpose of $\mathbf{F}$. We can write $\mathbf{F}^H$ as

$$\mathbf{F}^H = \begin{bmatrix} \mathbf{e}_{0N}^T \\ \mathbf{e}_{1N}^T \\ \vdots \\ \mathbf{e}_{(N-1)N}^T \end{bmatrix} \quad \Leftarrow \quad \mathbf{F} = \begin{bmatrix} \mathbf{e}_{0N}^* & \mathbf{e}_{1N}^* & \cdots & \mathbf{e}_{(N-1)N}^* \end{bmatrix}$$

- We say that $\mathbf{F}^H$ and $\mathbf{F}$ are Hermitians of each other (that's why $\mathbf{F}^H$)
- The nth row of $\mathbf{F}^H$ is the nth complex exponential $\mathbf{e}_{nN}^T$
- The kth column of $\mathbf{F}$ is the kth conjugate complex exponential $\mathbf{e}_{kN}^*$

The product of F and its Hermitian $\mathbf{F}^H$

- The product between the DFT matrix $\mathbf{F}$ and its Hermitian $\mathbf{F}^H$ is

$$\begin{bmatrix} \mathbf{e}_{0N}^T \\ \vdots \\ \mathbf{e}_{kN}^T \\ \vdots \\ \mathbf{e}_{(N-1)N}^T \end{bmatrix} \overset{\begin{bmatrix} \mathbf{e}_{0N}^* & \cdots & \mathbf{e}_{kN}^* & \cdots & \mathbf{e}_{(N-1)N}^* \end{bmatrix}}{\begin{bmatrix} \mathbf{e}_{0N}^T\mathbf{e}_{0N}^* & \cdots & \mathbf{e}_{0N}^T\mathbf{e}_{kN}^* & \cdots & \mathbf{e}_{0N}^T\mathbf{e}_{(N-1)N}^* \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{e}_{kN}^T\mathbf{e}_{0N}^* & \cdots & \mathbf{e}_{kN}^T\mathbf{e}_{kN}^* & \cdots & \mathbf{e}_{kN}^T\mathbf{e}_{(N-1)N}^* \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{e}_{(N-1)N}^T\mathbf{e}_{0N}^* & \cdots & \mathbf{e}_{(N-1)N}^T\mathbf{e}_{kN}^* & \cdots & \mathbf{e}_{(N-1)N}^T\mathbf{e}_{(N-1)N}^* \end{bmatrix}} = \mathbf{F}^H\mathbf{F}$$

- The (n, k) element of product matrix is the inner product $\mathbf{e}_{nN}^T \mathbf{e}_{kN}^*$
- Orthonormality of complex exponentials $\Rightarrow \mathbf{e}_{nN}^T \mathbf{e}_{kN}^* = \delta(n-k)$
  $\Rightarrow$ Only the diagonal elements survive in the matrix product

The matrix F and its inverse

- The DFT matrix $\mathbf{F}$ and its Hermitian are inverses of each other

$$\mathbf{F}^H\mathbf{F} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} = \mathbf{I}$$

- Matrices whose inverse is its Hermitian, are said Hermitian matrices
- Have proved the following fundamental theorem. Orthonormality

Theorem
The DFT matrix $\mathbf{F}$ is Hermitian $\Rightarrow \mathbf{F}^H\mathbf{F} = \mathbf{I}$

The iDFT in matrix form

- We can retrace methodology to also write the iDFT in matrix form
- No new definitions are needed. Use vectors $\mathbf{e}_{nN}$ and $\mathbf{X}$ to write

$$\tilde{x}(n) = \mathbf{e}_{nN}^T\mathbf{X} = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}$$

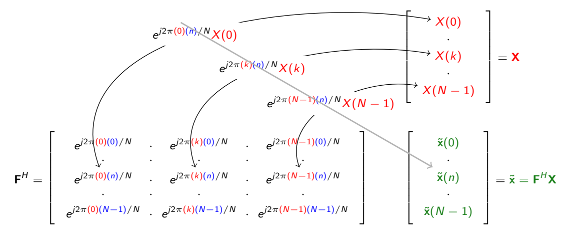
- Define stacked vector $\tilde{\mathbf{x}}$ and stack definitions. Use expression for $\mathbf{F}^H$

$$\tilde{\mathbf{x}} = \begin{bmatrix}\tilde{x}(0)\\ \tilde{x}(1)\\ \vdots \\ \tilde{x}(N-1)\end{bmatrix} = \begin{bmatrix}\mathbf{e}_{0N}^T\mathbf{X}\\ \mathbf{e}_{1N}^T\mathbf{X}\\ \vdots \\ \mathbf{e}_{(N-1)N}^T\mathbf{X}\end{bmatrix} = \begin{bmatrix}\mathbf{e}_{0N}^T\\ \mathbf{e}_{1N}^T\\ \vdots \\ \mathbf{e}_{(N-1)N}^T\end{bmatrix}\mathbf{X} = \mathbf{F}^H\mathbf{X}$$

- The iDFT is, as the DFT, just a matrix product $\Rightarrow \tilde{\mathbf{x}} = \mathbf{F}^H\mathbf{X}$

The iDFT in matrix form

- Again, in case you are having trouble visualizing the matrix product



- Can write the iDFT of $\mathbf{X}$ as the matrix product $\Rightarrow \tilde{\mathbf{x}} = \mathbf{F}^H\mathbf{X}$

Inverse theorem, like a pro

- When we proved theorems we had monkey steps and one smart step
  - $\Rightarrow$ That was orthonormality $\Rightarrow$ matrix $\mathbf{F}$ is Hermitian $\Rightarrow \mathbf{F}^H\mathbf{F} = \mathbf{I}$

Theorem

The iDFT is, indeed, the inverse of the DFT

Proof.

- Write $\tilde{\mathbf{x}} = \mathbf{F}^H\mathbf{X}$ and $\mathbf{X} = \mathbf{F}\mathbf{x}$ and exploit fact that $\mathbf{F}$ is Hermitian

$$\tilde{\mathbf{x}} = \mathbf{F}^H\mathbf{X} = \mathbf{F}^H\mathbf{F}\mathbf{x} = \mathbf{I}\mathbf{x} = \mathbf{x} \qquad \square$$

- Actually, this theorem would be true for any transform pair

$$X = \mathbf{T}\mathbf{x} \iff \tilde{\mathbf{x}} = \mathbf{T}^H\mathbf{X}$$

- As long as the transform matrix $\mathbf{T}$ is Hermitian $\Rightarrow \mathbf{T}^H\mathbf{T} = \mathbf{I}$

Energy conservation (Parseval) theorem, like a pro

Theorem

The DFT preserves energy $\Rightarrow \|\mathbf{x}\|^2 = \mathbf{x}^H\mathbf{x} = \mathbf{X}^H\mathbf{X} = \|\mathbf{X}\|^2$

Proof.

- Use iDFT to write $\mathbf{x} = \mathbf{F}^H\mathbf{X}$ and exploit fact that $\mathbf{F}$ is Hermitian

$$\|\mathbf{X}\|^2 = \mathbf{X}^H\mathbf{X} = (\mathbf{F}\mathbf{x})^H\mathbf{F}\mathbf{x} = \mathbf{x}^H\mathbf{F}^H\mathbf{F}\mathbf{x} = \mathbf{x}^H\mathbf{x} = \|\mathbf{x}\|^2 \qquad \square$$

- This theorem would also be true for any transform pair

$$X = \mathbf{T}\mathbf{x} \iff \tilde{\mathbf{x}} = \mathbf{T}^H\mathbf{X}$$

- As long as the transform matrix $\mathbf{T}$ is Hermitian $\Rightarrow \mathbf{T}^H\mathbf{T} = \mathbf{I}$

The discrete cosine transform

- Are there other useful transforms defined by Hermitian matrices $\mathbf{T}$?
  - $\Rightarrow$ Many. One we have already found is the DCT
- Define the inverse DCT matrix $\mathbf{C}^H$ to write the iDCT as $\tilde{\mathbf{x}} = \mathbf{C}^H\mathbf{X}$

$$\mathbf{C}^H = \frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \sqrt{2}\cos\left[\frac{2\pi(1)((1)+1/2)}{N}\right] & \cdots & \sqrt{2}\cos\left[\frac{2\pi(N-1)((1)+1/2)}{N}\right] \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \sqrt{2}\cos\left[\frac{2\pi(1)((N-1)+1/2)}{N}\right] & \cdots & \sqrt{2}\cos\left[\frac{2\pi(N-1)((N-1)+1/2)}{N}\right]\end{bmatrix}$$

- It is ready to verify that $\mathbf{C}$ is Hermitian (the cosines are orthonormal)
- From where the inverse and energy conservation theorems follow
  - $\Rightarrow$ Proofs hold for all Hermitians, $\mathbf{C}$ in particular

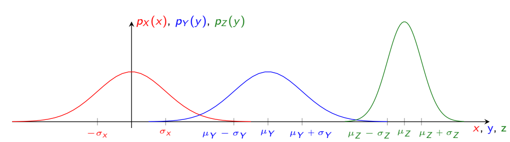
Designing transformations adapted to signals

- A basic information processing theory can be built for any $\mathbf{T}$
- Then, why do we specifically choose the DFT? Or the DCT?
  - $\Rightarrow$ Oscillations represent different rates of change
  - $\Rightarrow$ Different rates of change represent different aspects of a signal
- Not a panacea, though. E.g., $\mathbf{F}^H$ is independent of the signal
- If we know something about signal, should use it to build better $\mathbf{T}$
- A way of "knowing something" is a stochastic model of the signal
- PCA: Principal component analysis
  - $\Rightarrow$ Use the eigenvectors of the covariance matrix to build $\mathbf{T}$

Stochastic signals

Random Variables

- A random variable X models a random phenomena
  - $\Rightarrow$ One in which many different outcomes are possible
  - $\Rightarrow$ And one in which some outcomes may be more likely than others
- Thus, a random variable represents two things
  - $\Rightarrow$ All possible outcomes and their respective likelihoods



- Random variable X takes values around 0 and Y values around μ_Y
- Z takes values around μ_Z and the values are more concentrated

Probabilities

- Probabilities measure the likelihood of observing different outcomes
  - $\Rightarrow$ Larger probability means an outcome that is more likely
  - $\Rightarrow$ Or, observed more often when seeing many realizations
- Random variables represented by uppercase $\Rightarrow$ E.g., X
- Values that it can take represented by lowercase $\Rightarrow$ E.g., x
- The probability that X takes values between x and x' is written as

$$\mathrm{P}(x < X \le x')$$

- Here, we describe probabilities with density functions (pdf)
  - $\Rightarrow p_X(x)$

$$\mathrm{P}(x < X \le x') = \int_x^{x'} p_X(u)\,du$$

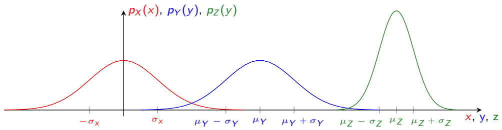
- $p_X(x) \approx$ How likely random variable X is to take a value around x

Gaussian random variables

- A random variable X is Gaussian (or Normal) if its pdf os of the form

$$p_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/\sigma^2}$$

- The mean μ determines center. The variance σ^2 determines width



- Means satisfy $0 = \mu_X < \mu_Y < \mu_z$. Variances are $\sigma_X^2 = \sigma^2 \mu_Y > \sigma_z^2$

Expectation

- Expectation of random variable is an average weighted by likelihoods

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x p_X(x)\, dx$$

- Regular average $\Rightarrow$ Sum all values and divide by number of values
- Expectation $\Rightarrow$ Weight values x by their relative likelihoods $p_X(x)$
- For a Gaussian random variable X the expectation is the mean μ

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/\sigma^2}\, dx = \mu$$

- Not difficult to evaluate integral, but besides the point to do so here

Variance

- Measure of variability around the mean weighted by likelihoods

$$\text{var}[X] = \mathbb{E}\left[\left(X - \mathbb{E}[X]\right)^2\right] = \int_{-\infty}^{\infty} \left(x - \mathbb{E}[X]\right)^2 p_X(x)\, dx$$

- Large variance $\equiv$ likely values are spread out around the mean
- Small variance $\equiv$ likely values are concentrated around the mean
- For a Gaussian random variable X the variance is the variance σ^2

$$\text{var}[X] = \int_{-\infty}^{\infty} \left(x - \mathbb{E}[X]\right)^2 \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/\sigma^2}\, dx = \sigma^2$$

- Not difficult to evaluate either. But also besides the point here

Random signals

- A random signal $\mathbf{X}$ is a collection of random variables (length N)

$$\mathbf{X} = [X(0),\ X(1),\ \ldots,\ X(N-1)]^T$$

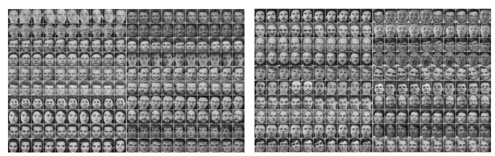
- Each of the random variables has its own pdf $\Rightarrow p_{X(n)}(x)$
- This pdf describes the likelihood of $X(n)$ taking a value around x
- This is not a sufficient description. Joint outcomes also important
- Joint pdf $p_{\mathbf{X}}(\mathbf{x})$ says how likely signal $\mathbf{X}$ is to be found around $\mathbf{x}$

$$\text{P}(\mathbf{x} \in \mathcal{X}) = \iint_{\mathcal{X}} p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}$$

- The individual pdfs $p_{X(n)}(x)$ are said to be marginal pdfs

Face images

- Random signal $\mathbf{X}$ $\Rightarrow$ All possible images of human faces
- More manageable $\Rightarrow$ $\mathbf{X}$ is a collection of 400 face images
  - $\Rightarrow$ The random variable represents all the images
  - $\Rightarrow$ The likelihood of each of them being chosen. E.g., 1/400 each



- Random variable specified by all outcomes and respective probabilities

Vectorization

- Do observe that the dataset consists of images $\equiv$ matrices
- Each image is stored in a matrix of size 112×92

$$\mathbf{M}_i = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,92} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,92} \\ \vdots & \vdots & \ddots & \vdots \\ m_{112,1} & m_{112,2} & \cdots & m_{112,92} \end{bmatrix}$$

- Stack columns of image M_i into the vector $\mathbf{x}_i$ with length $10,304$

$$\mathbf{x}_i = \left[m_{1,1},\ m_{21},\ \ldots,\ m_{112,1},\ m_{1,2},\ m_{2,2},\ \ldots,\ m_{112,2},\ \vdots,\ m_{1,92},\ m_{2,92},\ \ldots,\ m_{112,92}\right]^T$$

- Images are matrices $\mathbf{M}_i \in \mathbb{R}^{112\times 92}$. Signals are vectors $\mathbf{x}_i \in \mathbb{R}^{10,304}$

Realizations

- Realization $\mathbf{x}$ is an individual face pulled from set of possible outcomes
- Three possible realizations shown

- Realizations are just regular signals. Nothing random about them

Expectation, variance and covariance

- Signal's expectation is the concatenation of individual expectations

$$\mathbb{E}[\mathbf{X}] = \left[\mathbb{E}[X(0)],\ \mathbb{E}[X(1)],\ \ldots\ \mathbb{E}[X(N-1)]\right]^T = \iint \mathbf{x} p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}$$

- Variance of nth element $\Rightarrow \Sigma_{nn} = \text{var}[X(n)] = \mathbb{E}\left[\left(X(n) - \mathbb{E}[X(n)]\right)^2\right]$
- Measures variability of nth component
- Covariance between the signal components $X(n)$ and $X(m)$

$$\Sigma_{nm} = \mathbb{E}\left[\left(X(n) - \mathbb{E}[X(n)]\right)\left(X(m) - \mathbb{E}[X(m)]\right)\right] = \Sigma_{mn}$$

- Measures how much $X(n)$ predicts $X(m)$. Love, hate, and indifference
  - $\Rightarrow \Sigma_{nm} = 0$, components are unrelated. They are orthogonal
  - $\Rightarrow \Sigma_{nm} > 0$ ($\Sigma_{nm} < 0$), move in same (opposite) direction

Covariance matrix

- Assume that $\mathbb{E}[\mathbf{X}] = \mathbf{0}$ so that covariances are $\Sigma_{nm} = \mathbb{E}[X(n)X(m)]$
- Consider the expectation $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ of the (outer) product $\mathbf{x}\mathbf{x}^T$
- We can write the outer product $\mathbf{x}\mathbf{x}^T$ as

$$\mathbf{x}\mathbf{x}^T = \begin{bmatrix} x(0)x(0) & \cdots & x(0)x(n) & \cdots & x(0)x(N-1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x(n)x(0) & \cdots & x(n)x(n) & \cdots & x(n)x(N-1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x(N-1)x(0) & \cdots & x(N-1)x(n) & \cdots & x(N-1)x(N-1) \end{bmatrix}$$

-

Covariance matrix

- Assume that $\mathbb{E}[\mathbf{X}] = \mathbf{0}$ so that covariances are $\Sigma_{nm} = \mathbb{E}[X(n)X(m)]$
- Consider the expectation $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ of the (outer) product $\mathbf{x}\mathbf{x}^T$
- Expectation $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ implies expectation of each individual element

$$\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] = \begin{bmatrix} \mathbb{E}[x(0)x(0)] & \cdots & \mathbb{E}[x(0)x(n)] & \cdots & \mathbb{E}[x(0)x(N-1)] \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbb{E}[x(n)x(0)] & \cdots & \mathbb{E}[x(n)x(n)] & \cdots & \mathbb{E}[x(n)x(N-1)] \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbb{E}[x(N-1)x(0)] & \cdots & \mathbb{E}[x(N-1)x(n)] & \cdots & \mathbb{E}[x(N-1)x(N-1)] \end{bmatrix}$$

-

Covariance matrix

- Assume that $\mathbb{E}[\mathbf{X}] = \mathbf{0}$ so that covariances are $\Sigma_{nm} = \mathbb{E}[X(n)X(m)]$
- Consider the expectation $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ of the (outer) product $\mathbf{x}\mathbf{x}^T$
- The (n, m) element of the matrix $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ is the covariance $\Sigma_{n,m}$

$$\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] = \begin{bmatrix} \Sigma_{00} & \cdots & \Sigma_{0n} & \cdots & \Sigma_{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{n0} & \cdots & \Sigma_{nn} & \cdots & \Sigma_{n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{(N-1)0} & \cdots & \Sigma_{(N-1)n} & \cdots & \Sigma_{(N-1)(N-1)} \end{bmatrix}$$

- Define the covariance matrix of random signal $\mathbf{X}$ as $\mathbf{\Sigma} := \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$

Definition of covariance matrix

- When the mean is not null define the covariance matrix of $\mathbf{X}$ as

$$\mathbf{\Sigma} := \mathbb{E}\left[\left(\mathbf{x} - \mathbb{E}[\mathbf{x}]\right)\left(\mathbf{x} - \mathbb{E}[\mathbf{x}]\right)^T\right]$$

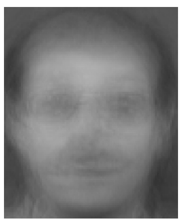
- As when null is mean, the (n, m) element of $\mathbf{\Sigma}$ is the covariance $\Sigma_{n,m}$

$$((\mathbf{\Sigma}))_{nm} = \mathbb{E}\left[\left(X(n) - \mathbb{E}[X(n)]\right)\left(X(m) - \mathbb{E}[X(m)]\right)\right] = \Sigma_{nm}$$

- The covariance matrix $\mathbf{\Sigma}$ is an arrangement of the covariances $\Sigma_{n,m}$
- The diagonal of $\mathbf{\Sigma}$ contains the (auto)variances $\Sigma_{nn} = \text{var}[X(n)]$
- Covariance matrix is symmetric $\Rightarrow ((\mathbf{\Sigma}))_{n,m} = \Sigma_{nm} = \Sigma_{mn} = ((\mathbf{\Sigma}))_{mn}$

Mean of face images

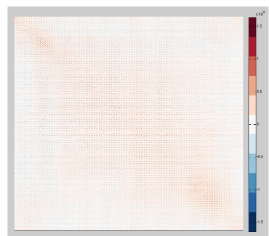
- All images are equally likely $\Rightarrow$ probability $1/400$ for each image
- The mean face is the regular average $\Rightarrow \mathbb{E}[\mathbf{x}] = \frac{1}{400}\sum_{i=1}^{400}\mathbf{x}_i$



- Average image looks something, sort of, like an average face

Covariance matrix of face images

- Covariance matrix $\Rightarrow \mathbf{\Sigma} = \frac{1}{400}\sum_{i=1}^{400}\left(\mathbf{x}_i - \mathbb{E}[\mathbf{x}]\right)\left(\mathbf{x}_i - \mathbb{E}[\mathbf{x}]\right)^T$



- Heat map of covariance matrix $\mathbf{\Sigma}$ shown on left
- Large correlation values around diagonal
- Large correlation values every 112 elements (jump a row on matrix)

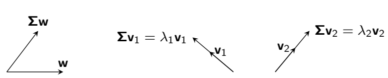
Principal component analisys (PCA) transform

Eigenvectors and eigenvalues of covariance matrix

- Consider a vector with N elements $\Rightarrow \mathbf{v} = [v(0), v(1), \ldots, v(N-1)]$
- We say that $\mathbf{v}$ is an eigenvector of $\mathbf{\Sigma}$ if for some scalar $\lambda \in \mathbb{R}$

$$\mathbf{\Sigma}\mathbf{v} = \lambda\mathbf{v}$$

- We say that λ is the eigenvalue associated to $\mathbf{v}$



- In general, $\mathbf{w}$ and $\mathbf{\Sigma}\mathbf{w}$ point in different directions
- But for eigenvalues $\mathbf{v}$, the product vector $\mathbf{\Sigma}\mathbf{v}$ is collinear with $\mathbf{v}$

Normalization

- If $\mathbf{v}$ is an eigenvector, $\alpha\mathbf{v}$ is also an eigenvector for any scalar $\alpha \in \mathbb{R}$,

$$\mathbf{\Sigma}(\alpha\mathbf{v}) = \alpha(\mathbf{\Sigma}\mathbf{v}) = \alpha\lambda\mathbf{v} = \lambda(\alpha\mathbf{v})$$

- Eigenvectors are defined up to a constant
- We use normalized eigenvectors with unit energy $\Rightarrow \|\mathbf{v}\|^2 = 1$
- If we compute $\mathbf{v}$ with $\|\mathbf{v}\|^2 \neq 1$ replace $\mathbf{v}$ with $\mathbf{v}/\|\mathbf{v}\|$
- There are N eigenvalues and distinct associated eigenvectors
  $\Rightarrow$ Some technical qualifications are needed in this statement

Ordering

Theorem
The eigenvalues of $\mathbf{\Sigma}$ are real and nonnegative $\Rightarrow \lambda \in \mathbb{R}$ *and* $\lambda \geq 0$

Proof.

- Begin by observing that we can write $\lambda = \mathbf{v}^H\mathbf{\Sigma}\mathbf{v}/\|\mathbf{v}\|^2$. Indeed

$$\mathbf{v}^H\mathbf{\Sigma}\mathbf{v} = \mathbf{v}^H(\mathbf{\Sigma}\mathbf{v}) = \mathbf{v}^H(\lambda\mathbf{v}) = \lambda\mathbf{v}^H\mathbf{v} = \lambda\|\mathbf{v}\|^2$$

- Complete by showing that $\mathbf{v}^T\mathbf{\Sigma}\mathbf{v}$ is nonnegative. Indeed (assume $\mathbb{E}[\mathbf{x}] = 0$)

$$\mathbf{v}^H\mathbf{\Sigma}\mathbf{v} = \mathbf{v}^H\mathbb{E}\left[\mathbf{x}\mathbf{x}^H\right]\mathbf{v} = \mathbb{E}\left[\mathbf{v}^H\mathbf{x}\mathbf{x}^H\mathbf{v}\right] = \mathbb{E}\left[\left(\mathbf{v}^H\mathbf{x}\right)\left(\mathbf{x}^H\mathbf{v}\right)\right] = \mathbb{E}\left[\left(\mathbf{v}^H\mathbf{x}\right)^2\right] \geq 0$$

□

- Order eigenvalues from largest to smallest $\Rightarrow \lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_{N-1}$
- Eigenvectors inherit order $\Rightarrow \mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{N-1}$
- The nth eigenvector of $\mathbf{\Sigma}$ is associated with its n largest eigenvalue

Eigenvectors are orthonormal

Theorem
Eigenvectors of $\mathbf{\Sigma}$ associated with different eigenvalues are orthogonal

Proof.

- Normalized eigenvectors $\mathbf{v}$ and $\mathbf{u}$ associated with eigenvalues $\lambda \neq \mu$

$$\mathbf{\Sigma v} = \lambda \mathbf{v}, \qquad \mathbf{\Sigma u} = \mu \mathbf{u}$$

- Since the matrix $\mathbf{\Sigma}$ is symmetric we have $\mathbf{\Sigma}^H = \mathbf{\Sigma}$, and it follows

$$\mathbf{u}^H \Sigma \mathbf{v} = \left(\mathbf{u}^H \Sigma \mathbf{v}\right)^H = \mathbf{v}^H \Sigma^H \mathbf{u} = \mathbf{v}^H \Sigma \mathbf{u}$$

- Make $\Sigma \mathbf{v} = \lambda \mathbf{v}$ on the leftmost side and $\Sigma \mathbf{u} = \mu \mathbf{u}$ on the rightmost

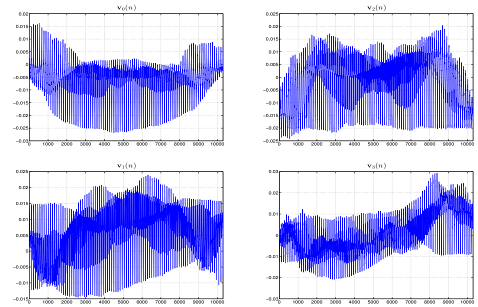
$$\mathbf{u}^H \lambda \mathbf{v} = \lambda \mathbf{u}^H \mathbf{v} = \mu \mathbf{v}^H \mathbf{u} = \mathbf{v}^H \mu \mathbf{u}$$

- Eigenvalues are different $\Rightarrow$ Relationship can only be true if $\mathbf{v}^H \mathbf{u} = 0$

□

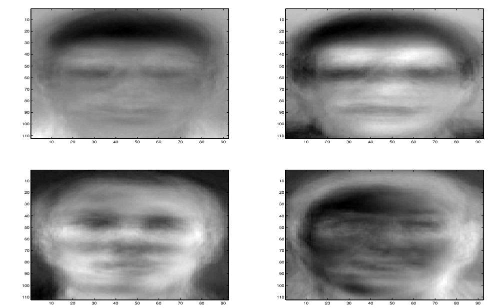
Eigenvectors of face images (1D)

- One dimensional representation of first four eigenvectors $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$



Eigenvectors of face images (2D)

- Two dimensional representation of first four eigenvectors $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$



Eigenvector matrix

- Define the matrix $\mathbf{T}$ whose kth column is the kth eigenvector of $\mathbf{\Sigma}$

$$\mathbf{T} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N]$$

- Since the eigenvectors $\mathbf{v}_k$ are orthonormal, the product $\mathbf{T}^H\mathbf{T}$ is

$$\mathbf{T}^H\mathbf{T} = \begin{bmatrix} \mathbf{v}_1^H \\ \vdots \\ \mathbf{v}_k^H \\ \vdots \\ \mathbf{v}_N^H \end{bmatrix} \left[\mathbf{v}_1 \; \cdots \; \mathbf{v}_k \; \cdots \; \mathbf{v}_N\right] = \begin{bmatrix} \mathbf{v}_1^H\mathbf{v}_1 & \cdots & \mathbf{v}_1^H\mathbf{v}_k & \cdots & \mathbf{v}_1^H\mathbf{v}_N \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{v}_k^H\mathbf{v}_1 & \cdots & \mathbf{v}_k^H\mathbf{v}_k & \cdots & \mathbf{v}_k^H\mathbf{v}_N \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{v}_N^H\mathbf{v}_N & \cdots & \mathbf{v}_N^H\mathbf{v}_k & \cdots & \mathbf{v}_N^H\mathbf{v}_N \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

- The eigenvector matrix $\mathbf{T}$ is Hermitian $\Rightarrow \mathbf{T}^H\mathbf{T} = \mathbf{I}$

Principal component analysis transform

- Any Hermitian $\mathbf{T}$ can be used to define an info processing transform
- Define principal component analysis (PCA) transform $\Rightarrow \mathbf{y} = \mathbf{Tx}$
- And the inverse (i)PCA transform $\Rightarrow \tilde{\mathbf{x}} = \mathbf{T}^H\mathbf{y}$
- Since $\mathbf{T}$ is Hermitian, iPCA is, indeed, the inverse of the PCA

$$\tilde{\mathbf{x}} = \mathbf{T}^H\mathbf{y} = \mathbf{T}^H(\mathbf{Tx}) = \mathbf{T}^H\mathbf{Tx} = \mathbf{Ix} = \mathbf{x}$$

- Thus $\mathbf{y}$ is an equivalent representation of $\mathbf{x}$ $\Rightarrow$ Back and forth
- And, also because $\mathbf{T}$ is Hermitian, Parseval's theorem holds

$$\|\mathbf{X}\|^2 = \mathbf{X}^H\mathbf{X} = (\mathbf{Tx})^H\mathbf{Tx} = \mathbf{x}^H\mathbf{T}^H\mathbf{Tx} = \mathbf{x}^H\mathbf{x} = \|\mathbf{x}\|^2$$

- Modifying elements y_k means altering energy composition of signal

Discussions

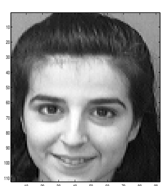
- The PCA transform is defined for any signal (vector) $\mathbf{x}$
  - $\Rightarrow$ But we expect to work well only when $\mathbf{x}$ is a realization of $\mathbf{X}$
- Write the iPCA in expanded form and compare with the iDFT

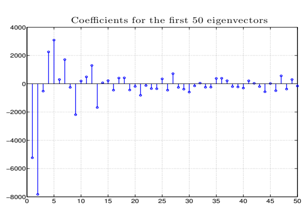
$$x(n) = \sum_{k=0}^{N-1} y(k)v_k(n) \quad \Leftrightarrow \quad x(n) = \sum_{k=0}^{N-1} X(k)e_{kN}(n)$$

- The same except that the use different bases for the expansion
- Still, like developing a new sense.
- But not one that is generic. Rather, adapted to the random signal $\mathbf{X}$

Coefficients of a projected face image

- PCA transform coefficients for given face image with 10,304 pixels
- Substantial energy in the first 15 PCA coefficients $y(k)$ with $k \leq 15$
- Almost all energy in the first 50 PCA coefficients $y(k)$ with $k \leq 50$
  - $\Rightarrow$ Thas is a compression factor of more than 200





Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - $\Rightarrow$ Increasing number of coefficients increases accuracy.
  - $\Rightarrow$ Using 50 coefficients suffices

Figure: image

Figure: No. P.C.s = 1

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - $\Rightarrow$ Increasing number of coefficients increases accuracy.
  - $\Rightarrow$ Using 50 coefficients suffices

Figure: image

Figure: No. P.C.s = 5

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - ⇒ Increasing number of coefficients increases accuracy.
  - ⇒ Using 50 coefficients suffices

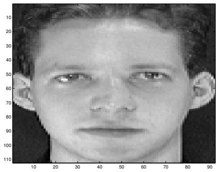


Figure: image

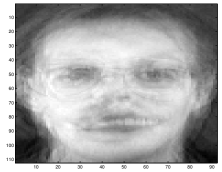


Figure: No. P.C.s = 10

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - ⇒ Increasing number of coefficients increases accuracy.
  - ⇒ Using 50 coefficients suffices

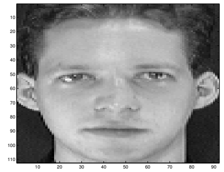


Figure: image

Figure: No. P.C.s = 20

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - ⇒ Increasing number of coefficients increases accuracy.
  - ⇒ Using 50 coefficients suffices

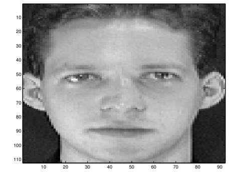


Figure: image

Figure: No. P.C.s = 30

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - ⇒ Increasing number of coefficients increases accuracy.
  - ⇒ Using 50 coefficients suffices

Figure: image

Figure: No. P.C.s = 40

Reconstructed face images

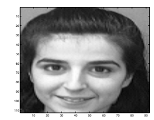
- Reconstructed image for increasing number of PCA coefficients
  - ⇒ Increasing number of coefficients increases accuracy.
  - ⇒ Using 50 coefficients suffices

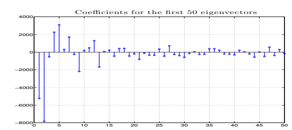
Figure: image

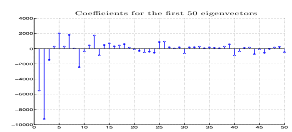
Figure: No. P.C.s = 50

Coefficients of the same person

- PCA transform y for two different pictures of the same person
- Coefficients are similar, even if pose and attitude are different
  - ⇒ E.g., first two coefficients almost identical

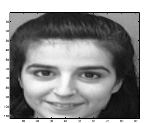


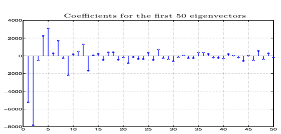


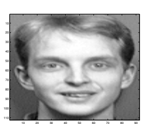


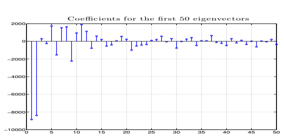
Coefficients of different persons

- PCA transform y for pictures of different persons
- Similar pose and attitude, but PCA coefficients are still different
  - ⇒ Can be used to perform face recognition. More later









Principal Components

The discrete Fourier transform with Hermitian matrices

Stochastic signals

Principal component analisys (PCA) transform

Principal Components

Principal Component analysis for Compression

Dimensionality reduction

Face recognition

Signals with uncorrelated components

- A random signal X with uncorrelated components is one with

$$\Sigma_{nm} = \mathbb{E}\left[\big(X(n) - \mathbb{E}[X(n)]\big)\big(X(m) - \mathbb{E}[X(m)]\big)\right] = 0$$

- Different components are unrelated to each other.
- They represent different (orthogonal) aspects of signal
- Components uncorrelated ⇒ The covariance matrix is diagonal

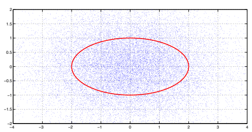
$$\mathbf{\Sigma} = \mathbb{E}\left[\big(\mathbf{x} - \mathbb{E}[\mathbf{x}]\big)\big(\mathbf{x} - \mathbb{E}[\mathbf{x}]\big)^T\right] = \begin{bmatrix} \Sigma_{00} & \cdots & \Sigma_{0n} & \cdots & \Sigma_{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{n0} & \cdots & \Sigma_{nn} & \cdots & \Sigma_{n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{(N-1)0} & \cdots & \Sigma_{(N-1)n} & \cdots & \Sigma_{(N-1)(N-1)} \end{bmatrix}$$

- How eigenvectors (principal components) of uncorrelated signals look?

Uncorrelated signal with 2 components

- Signal $\mathbf{X} = [X(0), X(1)]^T$ with 2 components and diagonal covariance

$$\mathbf{\Sigma} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$



- Covariance eigenvectors are

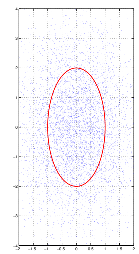
$$\mathbf{v}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \mathbf{v}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- The respective associated eigenvalues are $\lambda_0 = 2$ and $\lambda_1 = 1$
- Eigenvectors are orthogonal, as they should.
  - $\Rightarrow$ Represent directions of separate signal variability
  - $\Rightarrow$ Rate of variability given by associated eigenvalue

Another uncorrelated signal with 2 components

- Signal $\mathbf{X} = [X(0), X(1)]^T$ with 2 components and diagonal covariance

$$\mathbf{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$



- Covariance eigenvectors reverse order

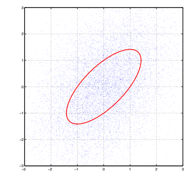
$$\mathbf{v}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- Associated eigenvalues are $\lambda_0 = 2$ and $\lambda_1 = 1$
- Eigenvectors still orthogonal, as they should.
  - $\Rightarrow$ Directions of separate signal variability
  - $\Rightarrow$ Rate given by associated eigenvalue

Signal with correlated components

- Signal $\mathbf{X} = [X(0), X(1)]^T$ with 2 components and diagonal covariance

$$\mathbf{\Sigma} = \begin{bmatrix} 3/2 & 1/2 \\ 1/2 & 3/2 \end{bmatrix}$$



- Covariance eigenvectors mix coordinates

$$\mathbf{v}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

- Eigenvalues are $\lambda_0 = 2$ and $\lambda_1 = 1$
- The eigenvalues are orthogonal. This is true for any covariance matrix
  - $\Rightarrow$ Mix coordinates but still represent directions of separate variability
  - $\Rightarrow$ Rate of change also given by associated eigenvalue

Eigenvectors in uncorrelated signals

- Uncorrelated components means diagonal covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} \Sigma_{00} & \cdots & \Sigma_{0n} & \cdots & \Sigma_{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{n0} & \cdots & \Sigma_{nn} & \cdots & \Sigma_{n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{(N-1)0} & \cdots & \Sigma_{(N-1)n} & \cdots & \Sigma_{(N-1)(N-1)} \end{bmatrix}$$

- If variances are ordered, kth eigenvector is k-shifted delta $\delta(n-k)$
- The corresponding variance Σ_{kk} is the associated eigenvalue
- Eigenvectors represent directions of orthogonal variability
- Rate of variability given by associated eigenvalue

Eigenvectors in correlated signals

- Correlated components means a full covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} \Sigma_{00} & \cdots & \Sigma_{0n} & \cdots & \Sigma_{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{n0} & \cdots & \Sigma_{nn} & \cdots & \Sigma_{n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{(N-1)0} & \cdots & \Sigma_{(N-1)n} & \cdots & \Sigma_{(N-1)(N-1)} \end{bmatrix}$$

- The eigenvectors $\mathbf{v}_k$ now mix different components
  - $\Rightarrow$ But they still represent directions of orthogonal variability
  - $\Rightarrow$ With the rate of variability given by associated eigenvalue
- PCA transform represents a signal as a sum of orthonormal vectors
  - $\Rightarrow$ Each of which represents independent variability
- Principal components (eigenvectors) with larger eigenvalues represent directions in which the signal has more variability

Principal Component analysis for Compression

Recap of DFT and iDFT in matrix notation

- Write DFT $\mathbf{X}$ as a stacked vector and stack individual definitions

$$\mathbf{X} = \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{0N}^H \mathbf{x} \\ \mathbf{e}_{1N}^H \mathbf{x} \\ \vdots \\ \mathbf{e}_{(N-1)N}^H \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{0N}^H \\ \mathbf{e}_{1N}^H \\ \vdots \\ \mathbf{e}_{(N-1)N}^H \end{bmatrix} \mathbf{x}$$

- Define the DFT matrix $\mathbf{F}^H$ so that we can write $\mathbf{X} = \mathbf{F}\mathbf{x}$

$$\mathbf{F} = \begin{bmatrix} \mathbf{e}_{0N}^H \\ \mathbf{e}_{1N}^H \\ \vdots \\ \mathbf{e}_{(N-1)N}^H \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-j2\pi(1)(1)/N} & \cdots & e^{-j2\pi(1)(N-1)/N} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi(N-1)(1)/N} & \cdots & e^{-j2\pi(N-1)(N-1)/N} \end{bmatrix}$$

- The DFT of signal $\mathbf{x}$ is a matrix multiplication $\Rightarrow \mathbf{X} = \mathbf{F}\mathbf{x}$
- The iDFT of signal $\mathbf{X}$ is a matrix multiplication $\Rightarrow \mathbf{x} = \mathbf{F}^H\mathbf{X}$

Recap of Compression by DFT

- We map signal $\mathbf{x}$ into the frequency domain $\mathbf{X}$ using DFT

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-j2\pi(1)(1)/N} & \cdots & e^{-j2\pi(1)(N-1)/N} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi(N-1)(1)/N} & \cdots & e^{-j2\pi(N-1)(N-1)/N} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

- Keep the k largest coefficients of $\mathbf{X}$ and make the rest $0 \rightarrow \tilde{\mathbf{X}}$

$$\mathbf{X}^T = [X(0), X(1), \dots, X(N-1)] \rightarrow \tilde{\mathbf{X}}^T = [X(0), 0, \dots, X(N-1)]$$

- Map the compressed signal $\tilde{\mathbf{X}}$ into the time domain $\tilde{\mathbf{x}}$ using iDFT

$$\begin{bmatrix} \tilde{x}(0) \\ \tilde{x}(1) \\ \vdots \\ \tilde{x}(N-1) \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-j2\pi(1)(1)/N} & \cdots & e^{-j2\pi(1)(N-1)/N} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi(N-1)(1)/N} & \cdots & e^{-j2\pi(N-1)(N-1)/N} \end{bmatrix}^H \begin{bmatrix} \tilde{X}(0) \\ \tilde{X}(1) \\ \vdots \\ \tilde{X}(N-1) \end{bmatrix}$$

Recap of Principal Component Transform

- Define the matrix $\mathbf{T}$ whose ith row is the ith eigenvector of $\mathbf{\Sigma}$

$$\mathbf{T} = \begin{bmatrix} \cdots & \mathbf{v}_0^T & \cdots \\ \cdots & \mathbf{v}_1^T & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \mathbf{v}_{N-1}^T & \cdots \end{bmatrix}$$

- The eigenvectors $\mathbf{v}_i$ are orthonormal
- Eigenvectors $\mathbf{v}_i$ are ordered based on their associated eigenvalues
  - $\Rightarrow \lambda_0 \geq \lambda_1 \geq \dots \lambda_{N-1}$
- principal component analysis (PCA) transform $\Rightarrow \mathbf{y} = \mathbf{T}\mathbf{x}$
- And the inverse (i)PCA transform $\Rightarrow \tilde{\mathbf{x}} = \mathbf{T}^H\mathbf{y}$
- For PCA compression we do not pick k largest coefficients of $\mathbf{y}$
- We pick the coefficients of the k largest eigenvectors

PCA Compression Idea

- We map signal $\mathbf{x}$ into $\mathbf{y}$ using PCA transform

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} v_0(0) & v_0(1) & \cdots & v_0(N-1) \\ v_1(0) & v_1(1) & \cdots & v_1(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ v_{N-1}(0) & v_{N-1}(1) & \cdots & v_{N-1}(N-1) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

- Keep k coefficients of $\mathbf{y}$ corresponding to the k largest eigenvectors

$$\mathbf{y}^T = [y(0), y(1), \ldots, y(N-1)] \rightarrow \tilde{\mathbf{y}}^T = [y(0), y(1), \ldots, y(k-1), 0, \ldots, 0]$$

- Reconstruct the compressed signal $\tilde{\mathbf{y}}$ using iPCA transform

$$\begin{bmatrix} \tilde{x}(0) \\ \tilde{x}(1) \\ \vdots \\ \tilde{x}(N-1) \end{bmatrix} = \begin{bmatrix} v_0(0) & v_0(1) & \cdots & v_0(N-1) \\ v_1(0) & v_1(1) & \cdots & v_1(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ v_{N-1}(0) & v_{N-1}(1) & \cdots & v_{N-1}(N-1) \end{bmatrix}^H \begin{bmatrix} \tilde{y}(0) \\ \tilde{y}(1) \\ \vdots \\ \tilde{y}(N-1) \end{bmatrix}$$

PCA Compression implementation

- PCA compression is equivalent to ignoring $\mathbf{v}_k, \ldots \mathbf{v}_{N-1}$
- Keep the eigenvectors that corresponds to the k largest eigenvalues

$$\mathbf{T} = \begin{bmatrix} \cdots & \mathbf{v}_0^T & \cdots \\ \cdots & \mathbf{v}_1^T & \cdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \cdots & \mathbf{v}_{N-1}^T & \cdots \end{bmatrix} \rightarrow \tilde{\mathbf{T}} = \begin{bmatrix} \cdots & \mathbf{v}_0^T & \cdots \\ \cdots & \mathbf{v}_1^T & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \mathbf{v}_{k-1}^T & \cdots \end{bmatrix}$$

- Transform the signal $\mathbf{x}$ to $\tilde{\mathbf{y}}$ using the transform matrix $\tilde{\mathbf{T}}$

$$\tilde{\mathbf{y}} = \tilde{\mathbf{T}}\mathbf{x}$$

- Reconstruct signal $\tilde{\mathbf{x}}$ by transforming back the $\tilde{\mathbf{y}}$ signal

$$\tilde{\mathbf{x}} = \tilde{\mathbf{T}}^H \tilde{\mathbf{y}}$$

Expected reconstruction error

- Define the expected reconstruction error as $\mathbb{E}\left[\|\mathbf{x} - \tilde{\mathbf{x}}\|^2\right]$
- Given the realizations $\{\mathbf{x}_1, \ldots, bbx_n\}$ for the random variable $\mathbf{x}$
  - $\Rightarrow$ The empirical expected reconstruction error is $\frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2$
- Consider the case that we only keep k eigenvectors for compression.
- How to choose k eigenvectors to minimize the expected reconstruction error?

$$\min \ \mathbb{E}\left[\|\mathbf{x} - \mathbf{x}'\|^2\right] \ \text{or} \ \min \ \frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i - \mathbf{x}_i'\|^2$$

Proof of optimality of PCA (1 of 3)

Theorem

The Expected reconstruction error is minimized by choosing the k largest principal components.

Proof:

- Consider $\hat{\mathcal{S}} := \{\hat{\mathbf{v}}_0, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_{k-1}\} \subset \mathcal{S} := \{\mathbf{v}_0, \ldots, \mathbf{v}_{N-1}\}$ as the set of eigenvectors for compression
- $\mathbf{y}_i$ is the mapped signal of $\mathbf{x}_i$ when we use all the eigenvectors
- The empirical expected reconstruction error can be simplified as

$$\frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i - \mathbf{x}_i'\|^2 = \frac{1}{n}\sum_{i=1}^n \Big\| \sum_{\mathbf{v}_j \in \mathcal{S}} \mathbf{y}_i(j)\mathbf{v}_j - \sum_{\mathbf{v}_j \in \hat{\mathcal{S}}} \mathbf{y}_i(j)\mathbf{v}_j \Big\|^2$$

$$= \frac{1}{n}\sum_{i=1}^n \Big\| \sum_{\mathbf{v}_j \in \mathcal{S} - \hat{\mathcal{S}}} \mathbf{y}_i(j)\mathbf{v}_j \Big\|^2$$

$$= \frac{1}{n}\sum_{i=1}^n \sum_{\mathbf{v}_j \in \mathcal{S} - \hat{\mathcal{S}}} \mathbf{y}_i(j)^2$$

Proof of optimality of PCA (2 of 3)

- Substituting $\mathbf{y}_i(j)$ by its definition $\mathbf{x}_i^T\mathbf{v}_j$

$$\frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i - \mathbf{x}_i'\|^2 = \frac{1}{n}\sum_{i=1}^n \sum_{\mathbf{v}_j \in \mathcal{S} - \hat{\mathcal{S}}} \mathbf{v}_j^T \mathbf{x}_i \mathbf{x}_i^T \mathbf{v}_j$$

$$= \sum_{\mathbf{v}_j \in \mathcal{S} - \hat{\mathcal{S}}} \mathbf{v}_j^T \left(\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T \right) \mathbf{v}_j$$

- The covariance matrix Σ of dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is defined as

$$\Sigma := \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T \in \mathbb{R}^{n \times n}$$

- Considering the definition of covariance matrix Σ, we obtain

$$\min \frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i - \mathbf{x}_i'\|^2 = \min \sum_{\mathbf{v}_j \in \mathcal{S} - \hat{\mathcal{S}}} \mathbf{v}_j^T \ \Sigma \ \mathbf{v}_j$$

Proof of optimality of PCA (3 of 3)

- The sum $\sum_{\mathbf{v}_j \in \mathcal{S}} \mathbf{v}_j^T \Sigma \mathbf{v}_j = trace(\Sigma)$ is constant. Therefore,

$$\min \sum_{\mathbf{v}_j \in \mathcal{S} - \hat{\mathcal{S}}} \mathbf{v}_j^T \ \Sigma \ \mathbf{v}_j = \max \sum_{\mathbf{v}_j \in \hat{\mathcal{S}}} \mathbf{v}_j^T \ \Sigma \ \mathbf{v}_j$$

- Therefore, minimizing the empirical expected reconstruction error is equivalent to

$$\min \frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i - \mathbf{x}_i'\|^2 = \max \sum_{\mathbf{v}_j \in \hat{\mathcal{S}}} \mathbf{v}_j^T \ \Sigma \ \mathbf{v}_j$$

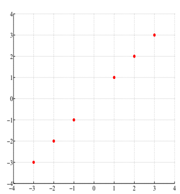
- The right hand side is maximized if we pick the k largest P.C.s.
  - $\Rightarrow \hat{\mathcal{S}} := \{\hat{\mathbf{v}}_0, \ldots, \hat{\mathbf{v}}_{k-1}\} = \{\mathbf{v}_0, \ldots, \mathbf{v}_{k-1}\}$

Dimension reduction

- Consider a set of realizations in $\mathbb{R}^2$ as

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 3 \\ 3 \end{bmatrix},$$

$$\mathbf{x}_4 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mathbf{x}_5 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \mathbf{x}_6 = \begin{bmatrix} -3 \\ -3 \end{bmatrix}.$$



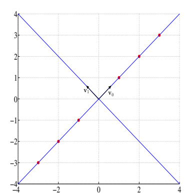
- The covariance matrix is $\mathbf{\Sigma} = \begin{bmatrix} 4.66 & 4.66 \\ 4.66 & 4.66 \end{bmatrix}$
- The eigenvalues are $\lambda_0 = 9.33$ and $\lambda_1 = 0$
- The eigenvectors are $\mathbf{v}_0 = [1/\sqrt{2}, 1/\sqrt{2}]^T$, $\mathbf{v}_1 = [-1/\sqrt{2}, 1/\sqrt{2}]^T$

Dimension reduction

- Consider a set of realizations in $\mathbb{R}^2$ as

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 3 \\ 3 \end{bmatrix},$$

$$\mathbf{x}_4 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mathbf{x}_5 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \mathbf{x}_6 = \begin{bmatrix} -3 \\ -3 \end{bmatrix}.$$



- The covariance matrix is $\mathbf{\Sigma} = \begin{bmatrix} 4.66 & 4.66 \\ 4.66 & 4.66 \end{bmatrix}$
- The eigenvalues are $\lambda_0 = 9.33$ and $\lambda_1 = 0$
- The eigenvectors are $\mathbf{v}_0 = [1/\sqrt{2}, 1/\sqrt{2}]^T$, $\mathbf{v}_1 = [-1/\sqrt{2}, 1/\sqrt{2}]^T$

Dimension reduction (continued)

- Axis $[1, 1]$ is very informative, while axis $[-1, 1]$ has no information
- Consider that we pick only one P.C. which is $\mathbf{v}_0$
- The mapped points are computed as $\mathbf{y}_i = \mathbf{v}_0^T\mathbf{x}_i$ for $i = 1, \ldots, 6$

$$\mathbf{y}_1 = \left[\frac{2}{\sqrt{2}}\right], \mathbf{y}_2 = \left[\frac{4}{\sqrt{2}}\right], \mathbf{y}_3 = \left[\frac{6}{\sqrt{2}}\right], \mathbf{y}_4 = \left[\frac{-2}{\sqrt{2}}\right], \mathbf{y}_5 = \left[\frac{-4}{\sqrt{2}}\right], \mathbf{y}_6 = \left[\frac{-6}{\sqrt{2}}\right]$$

- The reconstructed points are $\tilde{\mathbf{x}}_i = \mathbf{v}_0\mathbf{y}_i = \mathbf{x}_i$ for $i = 1, \ldots, 6$

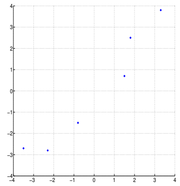
$$\tilde{\mathbf{x}}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tilde{\mathbf{x}}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \tilde{\mathbf{x}}_3 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \tilde{\mathbf{x}}_4 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \tilde{\mathbf{x}}_5 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \tilde{\mathbf{x}}_6 = \begin{bmatrix} -3 \\ -3 \end{bmatrix}$$

- The empirical reconstruction error is 0! (lossless compression).

Dimension reduction 2

- Consider a set of realizations in $\mathbb{R}^2$ as

$$\mathbf{x}_1 = \begin{bmatrix} 1.5 \\ 0.7 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 1.8 \\ 2.5 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 3.3 \\ 3.8 \end{bmatrix},$$
$$\mathbf{x}_4 = \begin{bmatrix} -0.8 \\ -1.5 \end{bmatrix}, \mathbf{x}_5 = \begin{bmatrix} -2.3 \\ -2.8 \end{bmatrix}, \mathbf{x}_6 = \begin{bmatrix} -3.5 \\ -2.7 \end{bmatrix}.$$

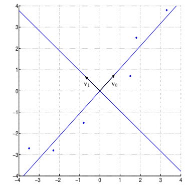


- The covariance matrix is $\mathbf{\Sigma} = \begin{bmatrix} 4.66 & 4.66 \\ 4.66 & 4.66 \end{bmatrix}$
- The eigenvalues are $\lambda_0 = 11.97$ and $\lambda_1 = 0.22$
- The eigenvectors are $\mathbf{v}_0 = [0.687, 0.727]^T$, $\mathbf{v}_1 = [-0.727, 0.687]^T$

Dimension reduction 2

- Consider a set of realizations in $\mathbb{R}^2$ as

$$\mathbf{x}_1 = \begin{bmatrix} 1.5 \\ 0.7 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 1.8 \\ 2.5 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 3.3 \\ 3.8 \end{bmatrix},$$
$$\mathbf{x}_4 = \begin{bmatrix} -0.8 \\ -1.5 \end{bmatrix}, \mathbf{x}_5 = \begin{bmatrix} -2.3 \\ -2.8 \end{bmatrix}, \mathbf{x}_6 = \begin{bmatrix} -3.5 \\ -2.7 \end{bmatrix}.$$



- The covariance matrix is $\mathbf{\Sigma} = \begin{bmatrix} 4.66 & 4.66 \\ 4.66 & 4.66 \end{bmatrix}$
- The eigenvalues are $\lambda_0 = 11.97$ and $\lambda_1 = 0.22$
- The eigenvectors are $\mathbf{v}_0 = [0.687, 0.727]^T$, $\mathbf{v}_1 = [-0.727, 0.687]^T$

Dimension reduction 2 (continued)

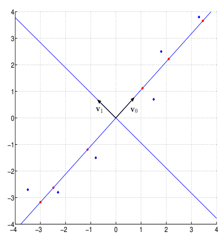
- $\mathbf{v}_0 = [0.687, 0.727]^T$ is more informative than $\mathbf{v}_1 = [-0.727, 0.687]^T$
- Consider that we pick only one P.C. which is $\mathbf{v}_0$
- The mapped points are computed as $\mathbf{y}_i = \mathbf{v}_0^T \mathbf{x}_i$ for $i = 1, \dots, 6$

$$\mathbf{y}_1 = [1.54] \quad \mathbf{y}_2 = [3.05] \quad \mathbf{y}_3 = [5.03]$$
$$\mathbf{y}_4 = [-1.64] \quad \mathbf{y}_5 = [-3.61] \quad \mathbf{y}_6 = [-4.37]$$

Dimension reduction 2 (continued)

- The reconstructed points are $\tilde{\mathbf{x}}_i = \mathbf{v}_0 \mathbf{y}_i = \mathbf{x}_i$ for $i = 1, \dots, 6$

$$\tilde{\mathbf{x}}_1 = \begin{bmatrix} 1.06 \\ 1.12 \end{bmatrix} \quad \tilde{\mathbf{x}}_2 = \begin{bmatrix} 2.10 \\ 2.22 \end{bmatrix}$$
$$\tilde{\mathbf{x}}_3 = \begin{bmatrix} 3.45 \\ 3.65 \end{bmatrix} \quad \tilde{\mathbf{x}}_4 = \begin{bmatrix} -1.11 \\ -1.19 \end{bmatrix}$$
$$\tilde{\mathbf{x}}_5 = \begin{bmatrix} -2.48 \\ -2.62 \end{bmatrix} \quad \tilde{\mathbf{x}}_6 = \begin{bmatrix} -2.99 \\ -3.17 \end{bmatrix}$$



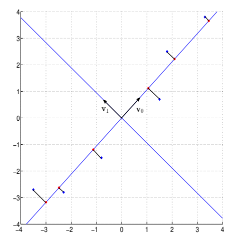
- The empirical reconstruction error is the average of the distances

$$\frac{1}{6} \sum_{i=1}^{6} \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 = 0.22$$

Dimension reduction 2 (continued)

- The reconstructed points are $\tilde{\mathbf{x}}_i = \mathbf{v}_0 \mathbf{y}_i = \mathbf{x}_i$ for $i = 1, \dots, 6$

$$\tilde{\mathbf{x}}_1 = \begin{bmatrix} 1.06 \\ 1.12 \end{bmatrix} \quad \tilde{\mathbf{x}}_2 = \begin{bmatrix} 2.10 \\ 2.22 \end{bmatrix}$$
$$\tilde{\mathbf{x}}_3 = \begin{bmatrix} 3.45 \\ 3.65 \end{bmatrix} \quad \tilde{\mathbf{x}}_4 = \begin{bmatrix} -1.11 \\ -1.19 \end{bmatrix}$$
$$\tilde{\mathbf{x}}_5 = \begin{bmatrix} -2.48 \\ -2.62 \end{bmatrix} \quad \tilde{\mathbf{x}}_6 = \begin{bmatrix} -2.99 \\ -3.17 \end{bmatrix}$$



- The empirical reconstruction error is the average of the distances

$$\frac{1}{6} \sum_{i=1}^{6} \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 = 0.22$$

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  $\Rightarrow$ Increasing number of coefficients reduces reconstruction error.
- Reconstruction error $= 1.25 \times 10^7$
- Sum of removed eigenvalues $= 5.02 \times 10^8$

Figure: image

Figure: No. P.C.s = 1

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  $\Rightarrow$ Increasing number of coefficients reduces reconstruction error.
- Reconstruction error $= 6.6 \times 10^6$
- Sum of removed eigenvalues $= 2.86 \times 10^8$

Figure: image

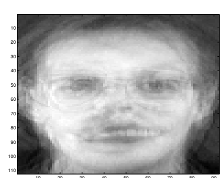


Figure: No. P.C.s = 5

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  $\Rightarrow$ Increasing number of coefficients reduces reconstruction error.
- Reconstruction error $= 3.9 \times 10^6$
- Sum of removed eigenvalues $= 1.9 \times 10^8$

Figure: image

Figure: No. P.C.s = 10

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  $\Rightarrow$ Increasing number of coefficients reduces reconstruction error.
- Reconstruction error $= 2.1 \times 10^6$
- Sum of removed eigenvalues $= 8.9 \times 10^7$

Figure: image

Figure: No. P.C.s = 20

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - $\Rightarrow$ Increasing number of coefficients reduces reconstruction error.
- Reconstruction error $= 1.3 \times 10^6$
- Sum of removed eigenvalues $= 3.11 \times 10^7$

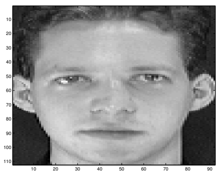


Figure: image

Figure: No. P.C.s = 30

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - $\Rightarrow$ Increasing number of coefficients reduces reconstruction error.
- Reconstruction error $= 5.8 \times 10^{-22}$
- Sum of removed eigenvalues $= 4.7 \times 10^{-7}$

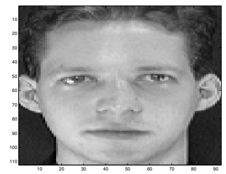


Figure: image

Figure: No. P.C.s = 40

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - $\Rightarrow$ Increasing number of coefficients reduces reconstruction error.
- Reconstruction error $= 6.24 \times 10^{-22}$
- Sum of removed eigenvalues $= 2.3 \times 10^{-7}$

Figure: image

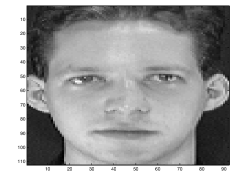
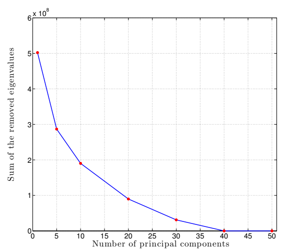
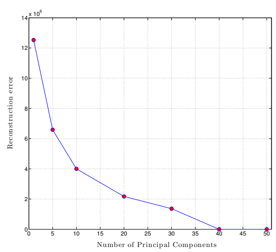


Figure: No. P.C.s = 50

Reconstruction error for one realization

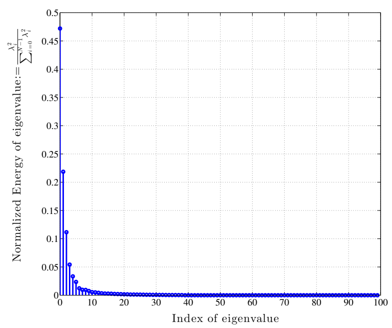
- Reconstruction error for one realization $\|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2$ decreases
- Sum of the removed eigenvalues decreases





???

- I don't know where to put this plot



Dimensionality reduction

Compression with the DFT

- Transform signal $\mathbf{x}$ into frequency domain with DFT $\mathbf{X} = \mathbf{F}\mathbf{x}$
- Recover $\mathbf{x}$ from $\mathbf{X}$ through iDFT matrix multiplication $\mathbf{x} = \mathbf{F}^H\mathbf{X}$
- We compress by retaining $K < N$ DFT coefficients to write

$$\tilde{x}(n) = \sum_{k=0}^{K-1} X(k) e^{j2\pi kn/N}$$

- Equivalently, we define the compressed DFT as

$$\tilde{X}(k) = X(k) \quad \text{for} \quad k < K, \qquad \tilde{X}(k) = 0 \text{ otherwise}$$

- Reconstructed signal is obtained with iDFT $\Rightarrow \tilde{\mathbf{x}} = \mathbf{F}^H\tilde{\mathbf{X}}$

Compression with the PCA

- Transform signal $\mathbf{x}$ into eigenvector domain with PCA $\mathbf{y} = \mathbf{T}\mathbf{x}$
- Recover $\mathbf{x}$ from $\mathbf{y}$ through iPCA matrix multiplication $\mathbf{x} = \mathbf{T}^H\mathbf{y}$
- We compress by retaining $K < N$ PCA coefficients to write

$$\tilde{x}(n) = \sum_{k=0}^{K-1} y(k) \mathbf{v}_k(n)$$

- Equivalently, we define the compressed PCA as

$$\tilde{y}(k) = y(k) \quad \text{for} \quad k < K, \qquad \tilde{y}(k) = 0 \text{ otherwise}$$

- Reconstructed signal is obtained with iDFT $\Rightarrow \tilde{\mathbf{x}} = \mathbf{T}^H\tilde{\mathbf{y}}$

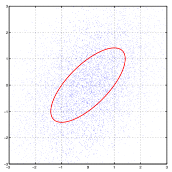
Why keeping the first K coefficients?

- Why do we keep the first K DFT coefficients?
  - $\Rightarrow$ Because faster oscillations tend to represent faster variation
  - $\Rightarrow$ Also, not always, sometimes we keep the largest coefficients
- Why do we keep the first K DFT coefficients?
  - $\Rightarrow$ Eigenvectors with lower ordinality have larger eigenvalues
  - $\Rightarrow$ Larger eigenvalues entail more variability
  - $\Rightarrow$ And more variability signifies more dominant features
- Eigenvectors with large ordinality represent finer signal features
  - $\Rightarrow$ And can often be omitted

Dimensionality reduction

- PCA compression is (more accurately) called dimensionality reduction
  - $\Rightarrow$ Do not compress signal. Reduce number of dimensions

$$\mathbf{\Sigma} = \begin{bmatrix} 3/2 & 1/2 \\ 1/2 & 3/2 \end{bmatrix}$$



- Covariance eigenvectors mix coordinates

$$\mathbf{v}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

- Eigenvalues are $\lambda_0 = 2$ and $\lambda_1 = 1$
- Signal varies more in $\mathbf{v}_0 = [1, 1]^T$ direction than in $\mathbf{v}_1 = [1, -1]^T$
  - $\Rightarrow$ Study one dimensional signal $\tilde{\mathbf{x}} = y(0)\mathbf{v}_0$
  - $\Rightarrow$ instead of the original two dimensional signal $\mathbf{x}$

Expected reconstruction error

- PCA dimensionality reduction is Minimizes the expected error energy
- To see that this is true, define the error signal as $\Rightarrow \mathbf{e} := \mathbf{x} - \tilde{\mathbf{x}}$
- The energy of the error signal is $\Rightarrow \|\mathbf{e}\|^2 = \|\mathbf{x} - \tilde{\mathbf{x}}\|^2$
- The expected value of the energy of the error signal is

$$\mathbb{E}\left[\|\mathbf{e}\|^2\right] = \mathbb{E}\left[\|\mathbf{x} - \tilde{\mathbf{x}}\|^2\right]$$

- Keeping the first K PCA coefficients minimizes $\mathbb{E}\left[\|\mathbf{e}\|^2\right]$
  - $\Rightarrow$ Among all reconstructions that use, at most K coefficients

Dimensionality reduction expected error

Theorem

The expectation of the reconstruction error is the sum of the eigenvalues corresponding to the eigenvectors of the coefficients that are discarded

$$\mathbb{E}\left[\|\mathbf{e}\|^2\right] = \sum_{k=K}^{N-1} \lambda_k$$

- It follows that keeping the first K PCA coefficients is optimal
  - $\Rightarrow$ In the sense that it minimizes the Expected error energy
- Good on average. Across realizations of the stochastic signal $\mathbf{X}$
- Need not be good for given realization(but we expect it to be good)

Proof of expected error expression

Proof.

- Error signal signal is $\mathbf{e} := \mathbf{x} - \tilde{\mathbf{x}}$. Define error PCA transform as $\mathbf{f} = \mathbf{T}^H\mathbf{x}$
- Using Parseval's (energy conservation) we can write the energy of $\mathbf{e}$ as

$$\|\mathbf{e}\|^2 = \|\mathbf{f}\|^2 = \sum_{k=K}^{N-1} y^2(k)$$

- In the last equality we used that $\mathbf{f} = \mathbf{y} - \tilde{\mathbf{y}} = [0, \ldots, 0, y(K), \ldots, y(N-1)]$
- Here, we are interested in the expected value of the error's energy
- Take expectation on both sides of equality $\Rightarrow \mathbb{E}\left[\|\mathbf{e}\|^2\right] = \sum_{k=K}^{N-1} \mathbb{E}\left[y^2(k)\right]$
- Used the fact that expectations are linear operators

Proof of expected error expression

Proof.

- Compute expected value $\mathbb{E}\left[y^2(k)\right]$ of the squared PCA coefficient $y(k)$
- As per PCA transform definition $y(k) = \mathbf{v}^H\mathbf{x}$, which implies

$$\mathbb{E}\left[y^2(k)\right] = \mathbb{E}\left[(\mathbf{v}_k^H\mathbf{x})^2\right] = \mathbb{E}\left[\mathbf{v}_k^H\mathbf{x}\mathbf{x}^T\mathbf{v}_k\right] = \mathbf{v}_k^H\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]\mathbf{v}_k$$

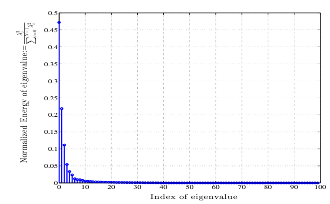
- Covariance matrix: $\mathbf{\Sigma} := \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$. Eigenvector definition $\mathbf{\Sigma}\mathbf{v}_k = \lambda_k$. Thus

$$\mathbb{E}\left[y^2(k)\right] = \mathbf{v}_k^H\mathbf{\Sigma}\mathbf{v}_k = \mathbf{v}_k^H\lambda_k\mathbf{v}_k = \lambda_k\|\mathbf{v}_k\|^2$$

- Substitute into expression for $\mathbb{E}\left[\|\mathbf{e}\|^2\right]$ to write $\Rightarrow \mathbb{E}\left[\|\mathbf{e}\|^2\right] = \sum_{k=K}^{N-1} \lambda_k$ □

Principal eigenvalues for face dataset

- Covariance matrix eigenvalues for faces dataset.
- Expected approximation error $\Rightarrow$ Tail sum of eigenvalue distribution
  - $\Rightarrow$ Average across all realizations. Not the same as actual error



- First 10 coefficients have 98% of energy.
- Eigenvectors with index $k > 50$ have 10^{-3}% of energy on average

Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")
- Keep 1 coefficient $\Rightarrow$ Reconstruction error $\Rightarrow$ 0.06
  - $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0.52

Reconstructed face images

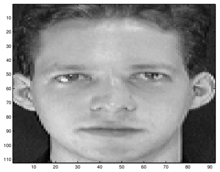
- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")
- Keep 5 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow$ 0.03
  - $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0.11

Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")
- Keep 10 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow$ 0.02
  - $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0.04

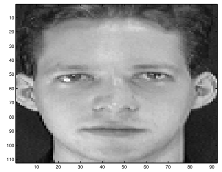
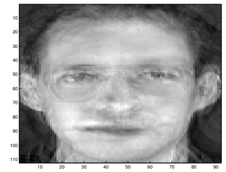
Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")
- Keep 20 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow$ 0.01
  $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0.01



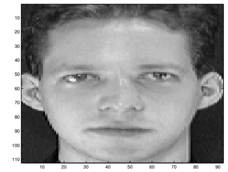

Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")
- Keep 30 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow$ 0.006
  $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0.003

Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")
- Keep 40 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow$ 0
  $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0

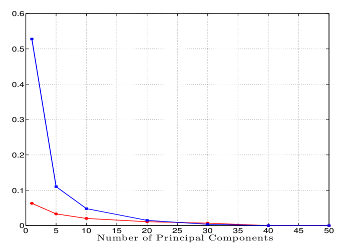



Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")
- Keep 50 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow$ 0
  $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0


Evolution of reconstruction error

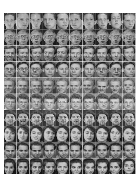
- Error for reconstruction process
- one realization (red), energy of removed eigenvalues (blue)

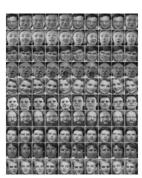


Dimension reduction

Face Recognition

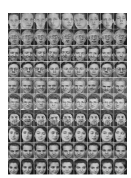
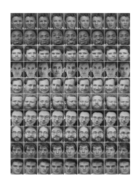
- Observe faces of known people $\Rightarrow$ Use them to train classifier
- Observe a face of unknown character $\Rightarrow$ Compare and classify
- The dataset we've used contains 10 different images of 40 people



Training set

- Separate the first 9 of each person to construct training set

- Interpret these images as know, and use them to train classifier

Test set

- Utilize the last image of each person to construct a test set




- Interpret these images as unknown, and use them to test classifier

Nearest neighbor classification

- Training set contains (signal, label) pairs $\Rightarrow \mathcal{T} = \{(\mathbf{x}_i, z_i)\}_{i=1}^{N}$
- Signal $\mathbf{x}$ is the face image. Label z is the person's "name"
- Given (unknown) signals $\mathbf{x}$, we want to assign a label
- Nearest neighbor classification rule
  - $\Rightarrow$ Find nearest neighbor signal in the training set

$$\mathbf{x}_{\text{NN}} := \underset{\mathbf{x}_i \in \mathcal{T}}{\operatorname{argmin}} \|\mathbf{x}_i - \mathbf{x}\|^2$$

  - $\Rightarrow$ Assign the label associated with the nearest neighbor

$$\mathbf{x}_{\text{NN}} \quad \Rightarrow \quad (\mathbf{x}_i, z_i) \quad \Rightarrow \quad z = z_i$$

- Reasonable enough. It should work. But it doesn't

The signal and the noise

- Image has a part that is inherent to the person $\Rightarrow$ The actual signal
- But it also contains variability $\Rightarrow$ Which we model as noise

$$\mathbf{x}_i = \tilde{\mathbf{x}}_i + \mathbf{w}$$

- Problem is, there is more variability (noise) than signal

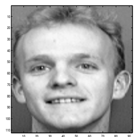


Figure: Test image

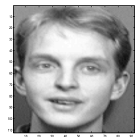


Figure: Nearest neighbor

PCA nearest neighbor classification

- Compute PCA for all elements of training set $\Rightarrow \mathbf{y}_i = \mathbf{T}^H\mathbf{x}_i$
- Redefine training set as one with PCA transforms $\Rightarrow \mathcal{T} = \{(\mathbf{y}_i, z_i)\}_{i=1}^{N}$
- Compute PCA transform of (unknown) signal $\mathbf{x}$ $\Rightarrow \mathbf{y} = \mathbf{T}^H\mathbf{x}$
- PCA nearest neighbor classification rule
  - $\Rightarrow$ Find nearest neighbor signal in training set with PCA transforms

$$\mathbf{y}_{\text{NN}} := \underset{\mathbf{y}_i \in \mathcal{T}}{\operatorname{argmin}} \|\mathbf{y}_i - \mathbf{y}\|^2$$

  - $\Rightarrow$ Assign the label associated with the nearest neighbor

$$\mathbf{y}_{\text{NN}} \quad \Rightarrow \quad (\mathbf{y}_i, z_i) \quad \Rightarrow \quad z = z_i$$

- Reasonable enough. It should work. And it does

Why does PCA work for face recognition?

- Recall: image = a part that belongs to the person + noise

$$\mathbf{x}_i = \tilde{\mathbf{x}}_i + \mathbf{w}$$

- PCA transformation $\mathbf{T} = [\mathbf{v}_0^T; \ldots; \mathbf{v}_{N-1}^T]$ leads to

$$\mathbf{y}_i = \mathbf{T}\mathbf{x}_i = \mathbf{T}\tilde{\mathbf{x}}_i + \mathbf{T}\mathbf{w}$$

- PCA concentrates energy of $\tilde{\mathbf{x}}_i$ on a few components
- But it keeps the energy of the noise on all components
- Keeping principal components improves the accuracy of classification
  - $\Rightarrow$ Because it increases the signal to noise ratio

PCA on the training set

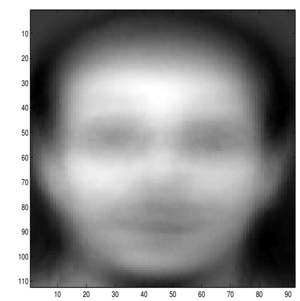
- The training set $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_{360}\}$ where $\mathbf{x}_i \in \mathbb{R}^{10304}$ is given
- Compute the mean vector and the covariance matrix as

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \quad \text{and} \quad \Sigma := \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}}_i)(\mathbf{x}_i - \bar{\mathbf{x}}_i)^T.$$

- Find the k largest eigenvalues of Σ
- Store their corresponding eigenvalues $\mathbf{v}_0, \ldots, \mathbf{v}_{k-1} \in \mathbb{R}^{10304}$ as P.C.
  - $\Rightarrow$ The Principal Components $\mathbf{v}_0, \ldots, \mathbf{v}_{k-1}$ are called eigenfaces
- Create the PCA transform matrix as $\mathbf{T} = [\mathbf{v}_0^T; \ldots; \mathbf{v}_{k-1}^T]$
- Project the training set into the space of P.C.s $\mathbf{y}_i = \mathbf{T}\mathbf{x}_i$
- $\boldsymbol{\Sigma}$ depends training set, but is also a good description of the test set

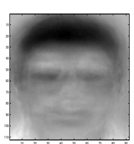
Average face of the training set

- The average face of the training set

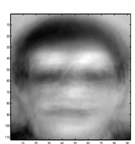


PCA on the training set

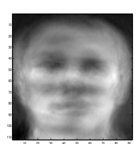
- The top 6 eigenfaces of the training set.



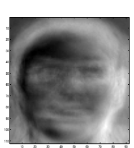
(1)



(2)

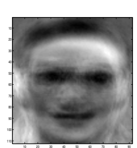


(3)



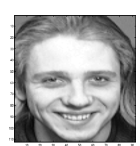
(4)

(5)

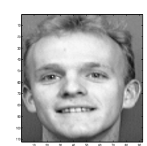
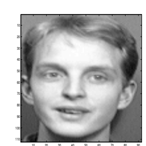
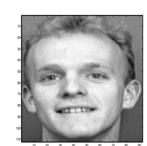
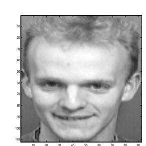


(6)

Finding the nearest neighbor

Num. of P.C.	test point	N.N. in the training set
$k = 1$		
$k = 5$		

PCA improves classification accuracy

Classification method	test point	result of classification
Naive N.N.		
PCA-ed($k = 5$) N.N.		

Signal Processing on Graphs

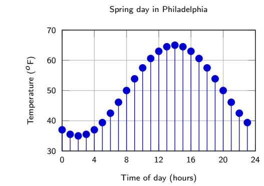
Santiago Segarra, Weiyu Huang, and Alejandro Ribeiro
Dept. of Electrical and Systems Engineering
University of Pennsylvania
aribeiro@seas.upenn.edu
http://www.seas.upenn.edu/~aribeiro/

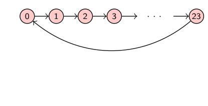
April 24, 2015

Graph Signals

The support of one dimensional signals

- We have studied one-dimensional signals, image processing, PCA
- It is time to understand them in a more unified way
- Consider the support of one-dimensional signals
- There is an underlying graph structure
  - ⇒ Each node represents discrete time instants (e.g. hours in a day)
  - ⇒ Edges are unweighted and directed

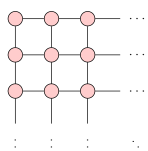




Underlying graph

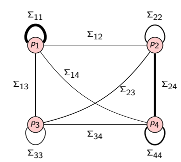
The support of images

- Similarly, images also have an underlying graph structure
- Each node represents a single pixel
- Edges denote neighborhoods of pixels
  - ⇒ Unweighted and undirected



PCA uses another underlying graph

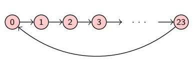
- The previous underlying graph assumes a structure between pixels (neighbors in lattice) a priori of seeing the images
- PCA considers images as defined on a different graph
- Each node represents a single pixel
- Edges denote covariance between pairs of pixels in the realizations
  - ⇒ A posteriori after seeing the images
  - ⇒ Undirected and weighted, including self loops



Graphs

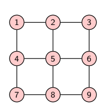
- Formally, a graph (or a network) is a triplet $(\mathcal{V}, \mathcal{E}, W)$
- $\mathcal{V} = \{1, 2, \dots, N\}$ is a finite set of N nodes or vertices
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges defined as order pairs (n, m)
  - ⇒ Write $\mathcal{N}(n) = \{m \in \mathcal{V} : (m, n) \in \mathcal{E}\}$ as the in-neighbors of n
- $W : \mathcal{E} \to \mathbb{R}$ is a map from the set of edges to scalar values, w_{nm}
  - ⇒ Represents the level of relationship from n to m
  - ⇒ Unweighted graphs ⇒ $w_{nm} \in \{0, 1\}$, for all $(n, m) \in \mathcal{E}$
  - ⇒ Undirected graphs ⇒ $(n, m) \in \mathcal{E}$ if and only if $(m, n) \in \mathcal{E}$ and $w_{nm} = w_{mn}$, for all $(n, m) \in \mathcal{E}$
    - ⇒ In-neighbors are neighbors
  - ⇒ More often weights are strictly positive, $W : \mathcal{E} \to \mathbb{R}_{++}$

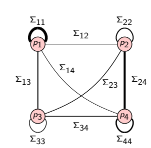
Graphs – examples



- Unweighted and directed graphs
  - ⇒ $\mathcal{V} = \{0, 1, \dots, 23\}$
  - ⇒ $\mathcal{E} = \{(0, 1), (1, 2), \dots, (22, 23), (23, 0)\}$
  - ⇒ $W : (n, m) \mapsto 1$, for all $(n, m) \in \mathcal{E}$

- Unweighted and undirected graphs
  - ⇒ $\mathcal{V} = \{1, 2, 3, \dots, 9\}$
  - ⇒ $\mathcal{E} = \{(1, 2), (2, 3), \dots, (8, 9), (1, 4), \dots, (6, 9)\}$
  - ⇒ $W : (n, m) \mapsto 1$, for all $(n, m) \in \mathcal{E}$





- Weighted and undirected graphs
  - ⇒ $\mathcal{V} = \{p_1, p_2, p_3, p_4\}$
  - ⇒ $\mathcal{E} = \{(p_1, p_1), (p_1, p_2), \dots, (p_4, p_4)\} = \mathcal{V} \times \mathcal{V}$
  - ⇒ $W : (n, m) \mapsto \Sigma_{nm} = \Sigma_{mn}$, for all (n, m)

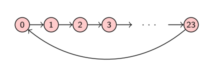
Adjacency matrices

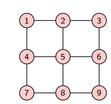
- Given a graph $G = (\mathcal{V}, \mathcal{E}, W)$ of N vertices,
- Its adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is defined as

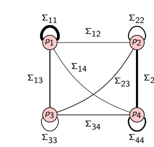
$$A_{nm} = \begin{cases} w_{nm}, & \text{if}(n, m) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}$$

- A matrix representation incorporating all information about G
  - ⇒ For unweighted graphs, positive entries represent connected pairs
  - ⇒ For weighted graphs, also denote proximities between pairs
- Inherently defines an ordering of vertices
  - ⇒ same ordering as in graph signals that we will see soon

Adjacency matrices – examples





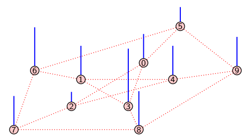


- Different ordering will yield different $\mathbf{A}$

$$\begin{bmatrix} & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ 1 & & & & \end{bmatrix}
\qquad
\begin{bmatrix} & 1 & & 1 & \\ 1 & & 1 & & \ddots \\ & 1 & & & \\ 1 & & & & \ddots \\ & \ddots & & \ddots & \end{bmatrix}
\qquad
\begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots \\ \Sigma_{12} & \Sigma_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} = \mathbf{\Sigma}$$

Graph signals

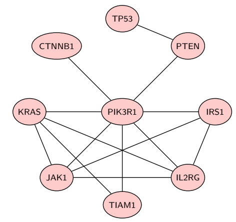
- Graph signals are mappings $x : \mathcal{V} \rightarrow \mathbb{R}$
- Defined on the vertices of the graph
- May be represented as a vector $\mathbf{x} \in \mathbb{R}^N$
- x_n represents the signal value at the nth vertex in $\mathcal{V}$
- Inherently utilizes an ordering of vertices
  - ⇒ same ordering as in adjacency matrices



$$\mathbf{x} = \begin{bmatrix} s_0 \\ s_1 \\ s_3 \\ \vdots \\ s_9 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}$$

Graphs – Gene networks

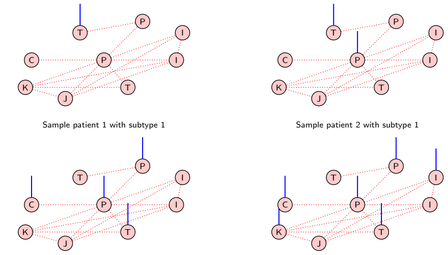
- Graphs representing gene-gene interactions
  - ⇒ Each node denotes a single gene (loosely speaking)
  - ⇒ Connected if their coded proteins participate in same metobolism



A sample network

Graph signals – Genetic profiles

- Genetic profiles for each patient can be considered as a graph signal
  - ⇒ Signal on each node is 1 if mutated and 0 otherwise



Sample patient 1 with subtype 1 — Sample patient 2 with subtype 1

Sample patient 1 with subtype 2 — Sample patient 2 with subtype 2

Plans

- We are going to derive following concepts for graph signal processing
  - ⇒ Total variations
  - ⇒ Frequency
  - ⇒ the notion of high or low frequency will be less obvious
  - ⇒ DFT and iDFT for graph signals
  - ⇒ Graph filtering
- And apply graph signal processing to gene mutation dataset

Graph Laplacian

Degree of a node

- The degree of a node is the sum of the weights of its incident edges
- Given a weighted and undirected graph $G = (\mathcal{V}, \mathcal{E}, W)$
- The degree of node i, $\deg(i)$ is defined as $\deg(i) = \sum_{j \in \mathcal{N}(i)} w_{ij}$
  - ⇒ where $\mathcal{N}(i)$ is the neighborhood of node i
- Equivalently, in terms of the adjacency matrix $\mathbf{A}$
  - ⇒ $\deg(i) = \sum_j A_{ij} = \sum_j A_{ji}$
- The degree matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a diagonal matrix s.t. $D_{ii} = deg(i)$
- In directed graphs, each node has an out-degree and an in-degree
  - ⇒ Weights in outgoing and incoming edges need not coincide

Laplacian of a graph

- Given a graph G with adjacency matrix $\mathbf{A}$ and degree matrix $\mathbf{D}$
- We define the Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

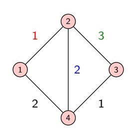
- Equivalently, $\mathbf{L}$ can be defined elementwise as

$$L_{ij} = \begin{cases} \deg(i) & \text{if } i = j \\ -w_{ij} & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

- We assume undirected G ⇒ $\deg(i)$ is well-defined
- The normalized Laplacian can be obtained as $\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$
  - ⇒ We will mainly focus on the unnormalized version

An example of a graph Laplacian

- Consider the weighted and undirected graph and its Laplacian



$$\mathbf{L} = \begin{bmatrix} 3 & -1 & 0 & -2 \\ -1 & 6 & -3 & -2 \\ 0 & -3 & 4 & -1 \\ -2 & -2 & -1 & 5 \end{bmatrix}$$

- Diagonal elements are strictly positive since no node is isolated
  - ⇒ Every node has a non-zero degree
- Off-diagonal elements are non-positive

Interpretation of the Laplacian

- Consider a graph G with Laplacian $\mathbf{L}$ and a signal $\mathbf{x}$ on G
  - ⇒ Define the new signal $\mathbf{y} = \mathbf{L}\mathbf{x}$

$$y_i = [\mathbf{L}\mathbf{x}]_i = \sum_{j \in \mathcal{N}(i)} w_{ij}(x_i - x_j)$$

- The summand j is large if one of two things happens
  - ⇒ The weight w_{ij} is large, i.e., edge $(i,j) \in \mathcal{E}$ is significant
  - ⇒ The value of x at node j is very different from the value at node i
- y_i measures the difference between $\mathbf{x}$ at a node and its neighborhood
- We can also define the Laplacian quadratic form of $\mathbf{x}$

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{(i,j) \in \mathcal{E}} w_{ij}(x_i - x_j)^2$$

- $\mathbf{x}^T\mathbf{L}\mathbf{x}$ quantifies the local variation of signal $\mathbf{x}$
  - ⇒ signals can be ordered depending on how wildly they vary
  - ⇒ will be important to order frequencies

Spectral properties of the Laplacian

- Denote by λ_i and $\mathbf{v}_i$ the eigenvalues and eigenvectors of $\mathbf{L}$
- Since $\mathbf{x}^T\mathbf{L}\mathbf{x} > 0$ for $\mathbf{x} \neq 0$, $\mathbf{L}$ is positive semi-definite
  - ⇒ All eigenvalues are nonnegative, i.e. $\lambda_i \geq 0$ for all i
- A constant vector $\mathbf{1}$ is an eigenvector of $\mathbf{L}$ with eigenvalue 0

$$[\mathbf{L1}]_i = \sum_{j \in \mathcal{N}(i)} w_{ij}(1-1) = 0$$

- Thus, $\lambda_1 = 0$ and $\mathbf{v}_1 = 1/\sqrt{N}\ \mathbf{1}$
  - ⇒ In connected graphs $\lambda_i > 0$ for $i = 2, \ldots, n$
  - ⇒ Multiplicity of $\lambda = 0$ equals the nr. of connected components

Graph Fourier Transform (GFT)

- Graph Signals
- Graph Laplacian
- Graph Fourier Transform (GFT)
- Ordering of frequencies
- Inverse graph Fourier transform (iGFT)
- Graph Filters
- Application: Gene Network
- Information sciences at ESE

Graph-shift operator

- Given an arbitrary graph $G = (\mathcal{V}, \mathcal{E}, W)$
- A graph-shift operator $\mathbf{S} \in \mathbb{R}^{N \times N}$ of graph G ia a matrix satisfying
  - ⇒ $S_{ij} = 0$ for $i \neq j$ and $(i, j) \notin \mathcal{E}$
- $\mathbf{S}$ can take nonzero values in the edges of G or in its diagonal
- We have already seen some possible graph-shift operators
  - ⇒ Adjacency $\mathbf{A}$, Degree $\mathbf{D}$ and Laplacian $\mathbf{L}$ matrices
- We restrict our attention to normal shifts $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^H$
  - ⇒ Columns of $\mathbf{V} = [\mathbf{v}_1 \mathbf{v}_2 \ldots \mathbf{v}_N]$ correspond to the eigenvectors of $\mathbf{S}$
  - ⇒ Λ is a diagonal matrix containing the eigenvalues of $\mathbf{S}$

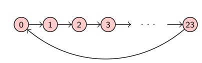
Graph Fourier Transform (GFT)

- Given a graph G and a graph signal $\mathbf{x} \in \mathbb{R}^N$ defined on G
  - ⇒ Consider a normal graph-shift $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^H$
- The Graph Fourier Transform (GFT) of $\mathbf{x}$ is defined as

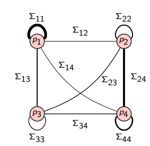
$$\tilde{\mathbf{x}}(k) = \langle \mathbf{x}, \mathbf{v}_k \rangle = \sum_{n=1}^{N} \mathbf{x}(n)\mathbf{v}_k^*(n)$$

- In matrix form, $\tilde{\mathbf{x}} = \mathbf{V}^H\mathbf{x}$
- Given that the columns of $\mathbf{V}$ are the eigenvectors $\mathbf{v}_i$ of $\mathbf{S}$
  - ⇒ $\tilde{\mathbf{x}}(k) = \mathbf{v}_k^H\mathbf{x}$ is the inner product between $\mathbf{v}_k$ and $\mathbf{x}$
  - ⇒ $\tilde{\mathbf{x}}(k)$ is how similar $\mathbf{x}$ is to $\mathbf{v}_k$
  - ⇒ In particular, GFT ≡ DFT when $\mathbf{V}^H = \mathbf{F}$, i.e. $\mathbf{v}_k = \mathbf{e}_{kN}$

DFT and PCA as particular cases of GFT



- For the directed cycle graph, GFT ≡ DFT
  - ⇒ if $\mathbf{S} = \mathbf{A}$ or
  - ⇒ if $\mathbf{S} = \mathbf{L}$ for symmetrized graph
  - ⇒ then $\mathbf{V}^H = \mathbf{F}$
- For the covariance graph, GFT ≡ PCA
  - ⇒ if $\mathbf{S} = \mathbf{A}$, then $\mathbf{V}^H = \mathbf{P}^H$



Ordering of frequencies

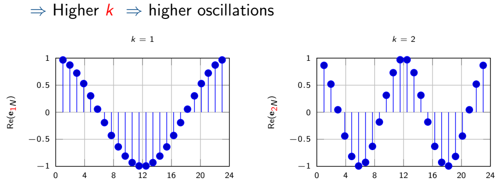
- Graph Signals
- Graph Laplacian
- Graph Fourier Transform (GFT)
- Ordering of frequencies
- Inverse graph Fourier transform (iGFT)
- Graph Filters
- Application: Gene Network
- Information sciences at ESE

Ordering of frequencies

- Recall in conventional DFT, the kth DFT component can be written

$$X(k) = \langle \mathbf{x}, \mathbf{e}_{kN} \rangle = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$$

- We say $X(k)$ the component for higher frequency given higher k
  - ⇒ There exists a natural ordering of frequencies
  - ⇒ Higher k ⇒ higher oscillations



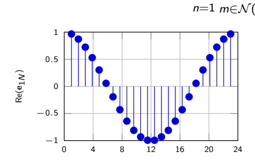
Quantifying oscillations – Zero crossings

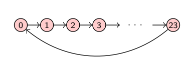
- We want to quantify the qualitative intuition of 'high oscillations'
- Classical zero crossings – # of places signals change signs

$$ZC(\mathbf{x}) = \sum_n \mathbf{1}\{x_n x_{n-1} < 0\}$$

- Graph zero crossings – # of edges signals on two ends differ in signs

$$ZC_G(\mathbf{x}) = \sum_{n=1}^{N} \sum_{m \in \mathcal{N}(n)} \mathbf{1}\{x_n x_m < 0\}$$





Underlying graph

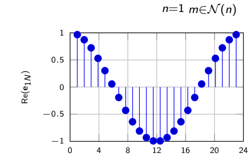
Quantifying oscillations – Total variations

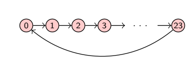
- Classical total variations – sum of squared differences in consecutive signal samples

$$TV(\mathbf{x}) = \sum_n (x_n - x_{n-1})^2$$

- Graph total variations – sum of squared differences between signals on two ends of edges multiplied by the corresponding edge weights
  - ⇒ Also known as Laplacian quadratic form

$$TV_G(\mathbf{x}) = \sum_{n=1}^{N} \sum_{m \in \mathcal{N}(n)} (x_n - x_m)^2 \, w_{mn} = \mathbf{x}^T L \mathbf{x}$$





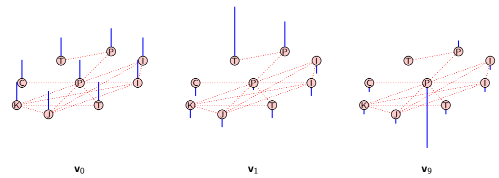
Underlying graph

Graph frequencies

- The Laplacian eigenvalues can be interpreted as frequencies
- Larger eigenvalues $\Rightarrow$ Higher frequencies
- The eigenvectors associated with large eigenvalues oscillate rapidly
  - $\Rightarrow$ Dissimilar values on vertices connected by edges with high weight
- The eigenvectors associated with small eigenvalues vary slowly
  - $\Rightarrow$ Similar values on vertices connected by edges with high weight
- Eigenvector associated with eigenvalue 0 is constant
  - $\Rightarrow$ for connected graph

Graph frequencies – Gene networks

- Three graph Laplacian eigenvectors for the gene networks



- $ZC_G(\mathbf{v}_0) = 0$
- $TV_G(\mathbf{v}_0) = 0$
- $ZC_G(\mathbf{v}_1) = 2$
- $TV_G(\mathbf{v}_1) = 0.4$
- $ZC_G(\mathbf{v}_1) = 20$
- $TV_G(\mathbf{v}_1) = 8.0$

Inverse graph Fourier transform (iGFT)

Inverse graph Fourier transform

- Recall the graph Fourier transform $\mathbf{x}$
  - $\Rightarrow$ of any signal $\mathbf{x} \in \mathbb{R}^N$ on the vertices of graph G
  - $\Rightarrow$ is the expansion of $\mathbf{x}$ of the eigenvectors of the Laplacian

$$\tilde{\mathbf{x}}(k) = \langle \mathbf{x}, \mathbf{v}_k \rangle = \sum_{n=1}^{N} x(n) v_k^*(n)$$

- In matrix form, $\tilde{\mathbf{x}} = \mathbf{V}^H \mathbf{x}$
- The inverse graph Fourier transform is

$$\mathbf{x}(n) = \sum_{k=0}^{N-1} \tilde{\mathbf{x}}(k) v_k(n)$$

- In matrix form, $\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$

Inverse theorem, like a pro

- Recap in proving theorems we have monkey steps and one smart step
  - $\Rightarrow$ That was orthonormality $\Rightarrow$ $\mathbf{V}^H$ is Hermitian $\Rightarrow \mathbf{V}\mathbf{V}^H = \mathbf{I}$

Theorem
The inverse graph Fourier transform (iGFT) is, indeed, the inverse of the GFT.

Proof.

- Write $\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}} = \mathbf{V}^H\mathbf{x}$ and exploit fact that $\mathbf{V}$ is Hermitian

$$\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}} = \mathbf{V}\mathbf{V}^H\mathbf{x} = \mathbf{I}\mathbf{x} = \mathbf{x} \qquad \square$$

- This is the last inverse theorem we will see...

Energy conservation (Parseval) theorem, like a pro

Theorem
The GFT preserves energy $\Rightarrow \|\mathbf{x}\|^2 = \mathbf{x}^H\mathbf{x} = \tilde{\mathbf{x}}^H\tilde{\mathbf{x}} = \|\tilde{\mathbf{x}}\|^2$

Proof.

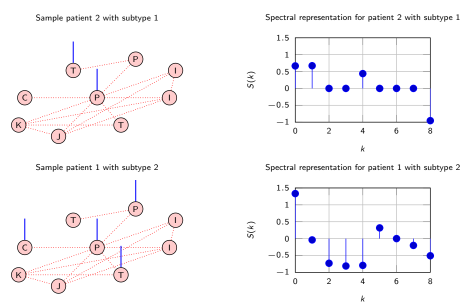
- Use GFT to write $\tilde{\mathbf{x}} = \mathbf{V}^H\mathbf{x}$ and the fact that $\mathbf{V}$ is Hermitian

$$\|\tilde{\mathbf{x}}\|^2 = \tilde{\mathbf{x}}^H\tilde{\mathbf{x}} = \left(\mathbf{V}^H\mathbf{x}\right)^H \mathbf{V}^H\mathbf{x} = \mathbf{x}^H\mathbf{V}\mathbf{V}^H\mathbf{x} = \mathbf{x}^H\mathbf{x} = \|\mathbf{x}\|^2 \qquad \square$$

- This is the last energy conservation theorem we will see...

Graph signal representations in two domains

- Graph signals can be equivalently represented in two domains
  - $\Rightarrow$ The vertex domain and the graph spectral domain



Graph Filters

Linearity and shift-invariance

- A graph filter $f : \mathbb{R}^N \to \mathbb{R}^N$ is a map between graph signals
  - $\Rightarrow$ Given a graph signal $\mathbf{x} \in \mathbb{R}^N$, its filtered version is $\mathbf{y} = f(\mathbf{x})$
- We will focus on filters f that are linear and shift-invariant
- A linear filter f is one that satisfies

$$\mathbf{y}_1 = f(\mathbf{x}_1), \quad \mathbf{y}_2 = f(\mathbf{x}_2) \quad \Longrightarrow \quad \alpha_1\mathbf{y}_1 + \alpha_2\mathbf{y}_2 = f(\alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2)$$

- A shift-invariant filter f satisfies

$$f(\mathbf{S}\mathbf{x}) = \mathbf{S}f(\mathbf{x})$$

where $\mathbf{S}$ is the graph-shift operator of the graph where $\mathbf{x}$ is defined

- Shift-invariance is the graph analog of time invariance in classical SP

Graph filters as matrix polynomials

- Given a graph G and a graph-shift operator $\mathbf{S} \in \mathbb{R}^{N\times N}$ on G
- We define the graph filter $\mathbf{H}$ as

$$\mathbf{H} := h_0\mathbf{S}^0 + h_1\mathbf{S}^1 + h_2\mathbf{S}^2 + \ldots = \sum_{\ell=0}^{L} h_\ell \mathbf{S}^\ell$$

- $\mathbf{H}$ is a polynomial on the graph-shift operator $\mathbf{S}$ with coefficients h_i
  - $\Rightarrow$ L is the degree of the filter
- Filter $\mathbf{H}$ acts on a graph signal $\mathbf{x} \in \mathbb{R}^N$ to generate $\mathbf{y} = \mathbf{H}\mathbf{x}$
  - $\Rightarrow$ If we define $\mathbf{x}^{(\ell)} := \mathbf{S}^\ell \mathbf{x} = \mathbf{S}\mathbf{x}^{(\ell-1)}$

$$\mathbf{y} = \sum_{\ell=0}^{L} h_\ell \mathbf{x}^{(\ell)}$$

- Why is $\mathbf{H}$ defined as a polynomial on $\mathbf{S}$?

Matrix polynomials are linear and shift-invariant

Proposition

The graph filter $\mathbf{H} = \sum_{\ell=0}^{L} h_\ell \mathbf{S}^\ell$ is linear and shift-invariant.

Proof.

- Since $\mathbf{H}$ is a matrix, linearity is trivial

$$\mathbf{y}_1 = \mathbf{H}\mathbf{x}_1, \quad \mathbf{y}_2 = \mathbf{H}\mathbf{x}_2 \quad \Longrightarrow \quad \alpha_1\mathbf{y}_1 + \alpha_2\mathbf{y}_2 = \mathbf{H}(\alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2)$$

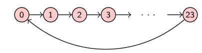
- For shift-invariance, note that $\mathbf{S}$ commutes with $\mathbf{S}^i$ for all i

$$\mathbf{H}(\mathbf{S}\mathbf{x}) = \left(\sum_{\ell=0}^{L} h_\ell \mathbf{S}^\ell\right)\mathbf{S}\mathbf{x} = \mathbf{S}\left(\sum_{\ell=0}^{L} h_\ell \mathbf{S}^\ell\right)\mathbf{x} = \mathbf{S}(\mathbf{H}\mathbf{x})$$

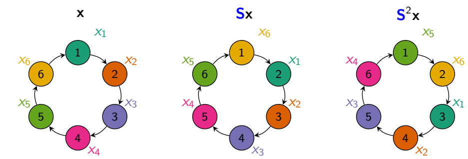
In fact, no other formulation of $\mathbf{H}$ is linear and shift-invariant

$\Rightarrow$ We will not show this

Connection with filters of time-varying signals



- Consider the particular case where $\mathbf{S} = \mathbf{A}_{dc}$
  - $\Rightarrow$ Adjacency matrix of a directed cycle
- Focus on a signal $\mathbf{x}$ defined on a cyclic graph with 6 nodes



- Consider the output signal $\mathbf{y} = \mathbf{H}\mathbf{x}$

$$\mathbf{y} = h_0\mathbf{x} + h_1\mathbf{S}^1\mathbf{x} + h_2\mathbf{S}^2\mathbf{x} + h_3\mathbf{S}^3\mathbf{x} + h_4\mathbf{S}^4\mathbf{x} + h_5\mathbf{S}^5\mathbf{x}$$

Connection with filters of time-varying signals

- Let's focus on the first component of signal $\mathbf{y}$

$$y_1 = h_0[\mathbf{S}^0\mathbf{x}]_1 + h_1[\mathbf{S}^1\mathbf{x}]_1 + h_2[\mathbf{S}^2\mathbf{x}]_1 + h_3[\mathbf{S}^3\mathbf{x}]_1 + h_4[\mathbf{S}^4\mathbf{x}]_1 + h_5[\mathbf{S}^5\mathbf{x}]_1$$
$$= h_0 x_1 + h_1 x_6 + h_2 x_5 + h_3 x_4 + h_4 x_3 + h_5 x_2$$

- In general, for element y_n of $\mathbf{y}$, exploiting the fact that $\mathbf{x}$ is cyclic

$$y_n = \sum_{l=0}^{N-1} h_l x_{n-l}$$

- Defining $\mathbf{h} := [h_0, h_1, \ldots, h_5]^T$ we may write

$$\mathbf{y} = \mathbf{h} * \mathbf{x}$$

- Thus, for the particular case where $\mathbf{S} = \mathbf{A}_{dc}$
  - $\Rightarrow$ $\mathbf{h}$ recovers the impulse response of the filter

Frequency response of a graph filter

- Recalling that $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^H$, we may write

$$\mathbf{H} = \sum_{\ell=0}^{L} h_\ell \mathbf{S}^\ell = \mathbf{V}\left(\sum_{\ell=0}^{L} h_\ell \Lambda^\ell\right)\mathbf{V}^H$$

- The application $\mathbf{H}\mathbf{x}$ of filter $\mathbf{H}$ to $\mathbf{x}$ can be split into three parts
  - $\Rightarrow$ $\mathbf{V}^H$ takes signal $\mathbf{x}$ to the graph frequency domain $\tilde{\mathbf{x}}$
  - $\Rightarrow$ $\widehat{\mathbf{H}} := \sum_{\ell=0}^{L} h_\ell \Lambda^\ell$ modifies the frequency coefficients to obtain $\tilde{\mathbf{y}}$
  - $\Rightarrow$ $\mathbf{V}$ brings the signal $\tilde{\mathbf{y}}$ back to the graph domain $\mathbf{y}$
- Since $\widehat{\mathbf{H}}$ is diagonal, define $\widehat{\mathbf{H}} =: diag(\widehat{\mathbf{h}})$
  - $\Rightarrow$ $\widehat{\mathbf{h}}$ is the frequency response of the filter $\mathbf{H}$
  - $\Rightarrow$ Output at frequency i depends only on input at frequency i

$$\tilde{y}_i = \widehat{h}_i \tilde{x}_i$$

Frequency response and filter coefficients

- In order to design a graph with a particular frequency response $\widehat{\mathbf{h}}$
  - $\Rightarrow$ Need to know the relation between $\widehat{\mathbf{h}}$ and the filter coefficients $\mathbf{h}$
- Define the matrix $\boldsymbol{\Psi} := \begin{pmatrix} 1 & \lambda_1 & \ldots & \lambda_1^{L-1} \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_N & \ldots & \lambda_N^{L-1} \end{pmatrix}$

Proposition

The frequency response $\widehat{\mathbf{h}}$ of a graph filter with coefficients $\mathbf{h}$ is given by

$$\widehat{\mathbf{h}} = \boldsymbol{\Psi}\mathbf{h}$$

Proof.

- Since $\widehat{\mathbf{h}} := \text{diag}(\sum_{\ell=0}^{L} h_\ell \Lambda^\ell)$ we have that $\widehat{h}_i = \sum_{\ell=0}^{L} h_\ell \lambda_i^\ell$
- Defining $\boldsymbol{\lambda}_i = [\lambda_i^0, \lambda_i^1, \ldots, \lambda_i^{L-1}]^T$ we have that $\widehat{h}_i = \boldsymbol{\lambda}_i^T \mathbf{h}$
- Stacking the values for all $\widehat{h}_i$, the result follows

Graph filter design

- Given the desired frequency response $\widehat{\mathbf{h}}$ of the graph filter
  - $\Rightarrow$ We can find the graph coefficients $\mathbf{h}$ as

$$\mathbf{h} = \boldsymbol{\Psi}^{-1}\widehat{\mathbf{h}}$$

- Since $\boldsymbol{\Psi}$ is Vandermonde
  - $\Rightarrow$ $\boldsymbol{\Psi}$ is invertible as long as $\lambda_i \neq \lambda_j$ for $i \neq j$
- For the particular case when $\mathbf{S} = \mathbf{A}_{dc}$, we have that $\lambda_i = e^{-j\frac{2\pi}{N}(i-1)}$

$$\boldsymbol{\Psi} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & e^{-j\frac{2\pi(1)(1)}{N}} & \ldots & e^{-j\frac{2\pi(1)(N-1)}{N}} \\ \vdots & \vdots & & \vdots \\ 1 & e^{-j\frac{2\pi(N-1)(1)}{N}} & \ldots & e^{-j\frac{2\pi(N-1)(N01)}{N}} \end{pmatrix} = \mathbf{F}$$

- $\Rightarrow$ The frequency response is the DFT of the impulse response

$$\widehat{\mathbf{h}} = \mathbf{F}\mathbf{h}$$

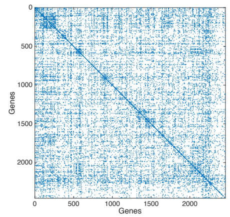
Application: Gene Network

Motivation

- Patients diagnosed with same disease exhibit different behaviors
- Each patient has a genetic profile describing gene mutations
- Would be beneficial to infer phenotypes from genotypes
  - $\Rightarrow$ Targeted treatments, more suitable suggestions, etc.
- Traditional approaches consider different genes to be independent
  - $\Rightarrow$ Not so ideal, as different genes may affect same metabolism
- Alternatively, consider genetic network
  - $\Rightarrow$ Genetic profiles becomes graph signals on genetic network
  - $\Rightarrow$ We will see how this consideration improves subtype classification

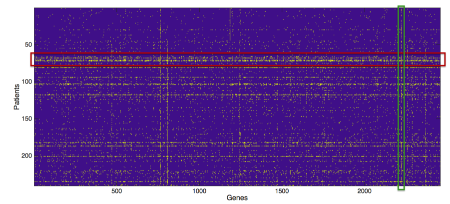
Genetic network

- Undirected and unweighted graph with 2458 nodes
  - ⇒ Describes gene-to-gene interactions
- Each node represents a gene in human DNA related to breast cancer
- An edge between two genes represents interaction
  - ⇒ Proteins encoded participate in the same metabolism process
- Adjacency matrix of the gene network



Genetic profiles

- Genetic profile of 240 women with breast cancer
  - ⇒ 44 with serous subtype and 196 with endometrioid subtype
  - ⇒ Patient i has an associated profile $\mathbf{x}_i \in \{0,1\}^{2458}$
- Mutations are very varied across patients
  - ⇒ Some patients present a lot of mutations
  - ⇒ Some genes are consistently mutated across patients



- Can we use the genetic profile to classify patients across subtypes?

k-nearest neighbor classification

- Quantify the distance between genetic profiles
  - ⇒ $d(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$
- Given a patient i to classify, all other patients' subtypes are known
- Find the k most similar profiles, i.e. j such that $d(i,j)$ is minimized
  - ⇒ Assign to i the most common subtype among these k neighbors
- Compare estimated with real subtype $\mathbf{y}$ for all patients
- We obtain the following error rates

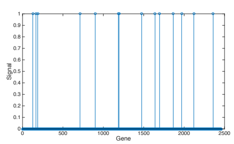
$$k = 3 \Rightarrow 13.3\%, \qquad k = 5 \Rightarrow 12.9\%, \qquad k = 7 \Rightarrow 14.6\%$$

- Can we do any better using graph signal processing?

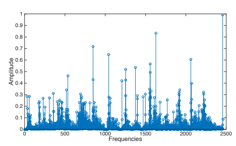
Genetic profile as a graph signal

- Each genetic profile $\mathbf{x}_i$ can be seen as a graph signal
  - ⇒ On the genetic network
- We can look at the frequency components $\tilde{\mathbf{x}}_i$ using the GFT
  - ⇒ Use as shift operator $\mathbf{S}$ the Laplacian of the genetic network

Example of signal $\mathbf{x}_i$



Frequency representation $\tilde{\mathbf{x}}_i$

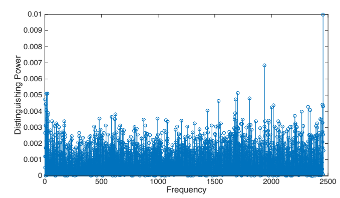


Distinguishing Power

- Define the distinguishing power of frequency $\mathbf{v}_k$ as

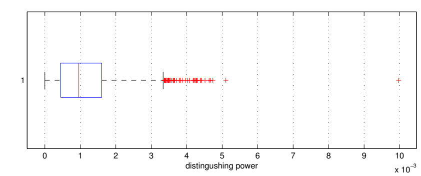
$$DP(\mathbf{v}_k) = \left| \frac{\sum_{i:y_i=1} \tilde{\mathbf{x}}_i(k)}{\sum_i \mathbf{1}\{y_i = 1\}} - \frac{\sum_{i:y_i=2} \tilde{\mathbf{x}}_i(k)}{\sum_i \mathbf{1}\{y_i = 2\}} \right| \Big/ \sum_i |\tilde{\mathbf{x}}_i(k)|,$$

- Normalized difference between the mean GFT coefficient for $\mathbf{v}_k$
  - ⇒ Among patients with serous and endometrioid subtypes
- Distinguishing power is not equal across frequencies



Distribution of distinguishing powers

- The distribution of distributing power



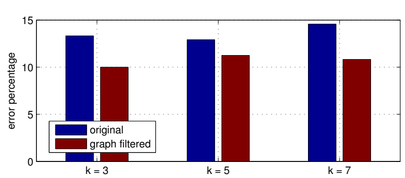
- Most frequencies have weak distinguishing power
  - ⇒ A few frequencies have strong differentiating power
  - ⇒ The most powerful frequency outperforms others sigificantly
- The distinguishing power defined is one of many proper heuristics

Increasing accuracy via graph filters

- Keeps only information in the most distinguishable frequency
- For the genetic profile $\mathbf{x}_i$ with its frequency representation $\tilde{\mathbf{x}}_i$
- Multiply $\tilde{\mathbf{x}}_i$ with graph filter H_1 having the frequency response

$$H_1(k) = \begin{cases} 1, & \text{if } k = \text{argmax}_k \, DP(\mathbf{v}_k); \\ 0, & \text{otherwise.} \end{cases}$$

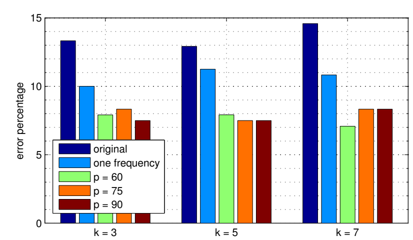
- Then perform inverse GFT to get the filtered graph signals $\hat{\mathbf{x}}_i$



Increasing accuracy via another graph filters

- Keeps information in frequencies with higher distinguishing power
- Multiply $\tilde{\mathbf{x}}_i$ with graph filter H_p having the frequency response

$$H_p(k) = \begin{cases} 1, & \text{if } DP(\mathbf{v}_k) \geq p\text{-th percentile of the distribution of } DP; \\ 0, & \text{otherwise,} \end{cases}$$



Information sciences at ESE

Courses to consider

- If you want to explore more about transforms and filters
  - ⇒ ESE210: Introduction to Dynamic Systems
  - ⇒ ESE303: Stochastic Systems Analysis and Simulation
  - ⇒ ESE325: Fourier Analysis and Applications ...
  - ⇒ ESE531: Digital Signal Processing

More courses to consider

- Once you have information you may want to something with it
- Controlling the state of a system
  - ⇒ ESE406: Control of Systems
  - ⇒ ESE500: Linear Systems Theory
- Making decisions that are good in some sense (optimal)
  - ⇒ ESE204: Decision Models
  - ⇒ ESE304: Optimization of Systems
  - ⇒ ESE504: Introduction to Optimization Theory
  - ⇒ ESE605: Modern Convex Optimization

Even more courses to consider

- At some point, you want to use what you've learned to do something
  - ⇒ ESE290: Introduction to ESE Research Methodology
  - ⇒ ESE350: Embedded Systems/Microcontroller Laboratory

Research

- Most professors use about 5% of their time on teaching
- The other 95% of their time they use on research
- It is a pity to come to Penn and not spend a summer doing research
- Most of us are happy to have help
- Even if we are not, our doctoral students are desperate for help

Thanks

- It has been my pleasure. I am very happy abut how things turned out
- If you need my help at some point in the next 30 years, let me know
- I will be retired after that