Transform Methods & Signal Processing

lecture 08

Matthew Roughan
<matthew.roughan@adelaide.edu.au>

Discipline of Applied Mathematics
School of Mathematical Sciences
University of Adelaide

July 27, 2009

Windows

No we don't mean the common operating system. Windows are a way of minimizing leakage when performing Fourier transforms, but they lead into a more sophisticated time-sensitive versions of the Fourier Transform called the Short-Time Fourier Transform.

This lecture considers window functions, and the Short-Time Fourier Transform (STFT), which uses window functions to localize frequency analysis by the FT.

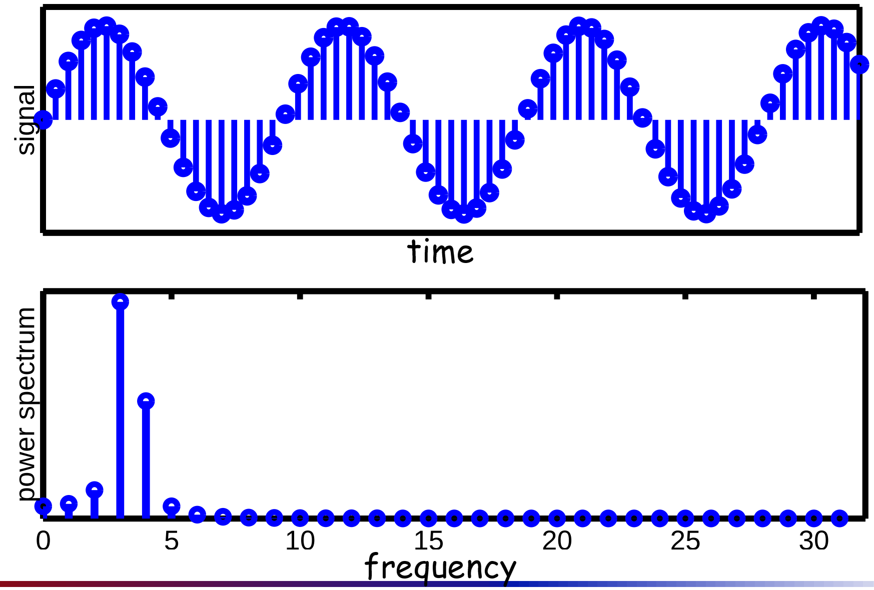
Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.1/79

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.2/79

Leakage

- always have finite signals
- implicit assumption in DFT is periodicity
  - we look at correlation of signal to sin's and cosines with periods that match the length of the data
- What if a signal is not periodic?
- What if the period is not the same as the length of the data?
- We get leakage

Leakage example



Note that the power spectrum doesn't have a single distinct peak. Rather, energy is "leaked" into neighboring frequencies.

What causes leakage

- The DFT uses a finite number of frequencies.
- Not all signals fit this mold exactly: what happens to sinusoids with non-integral frequencies?
- Their power is spread over a few frequencies.
- Note we are representing the signal by a series of numbers $X(k)$ which represent the correlation of the signal to a particular sinusoid with freq. kf_s/N,
- another way to understand, is to think of each element $X(k)$ of the DFT as a narrow bandpass filter, centered on frequency kf_s/N, but which have side lobes.

Alternative view

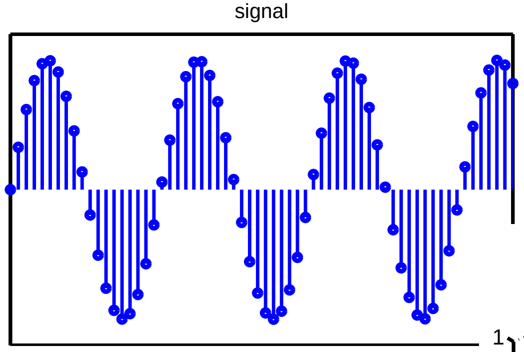
- alternative view: DFT truncated signal implicitly assumes signal is **periodic**, but it isn't, so what happens at the edges?
- Edges induce transients
- transients introduce extra frequency components

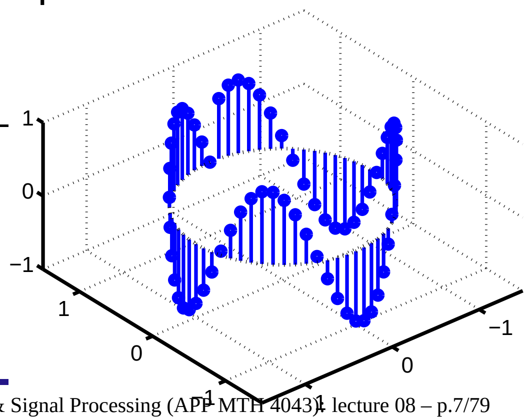
Why do we care?

- side lobes reduce **sensitivity**
- determine the smallest signal we can detect against a background of another signal

All filters have side-lobes. Why don't these impact the DFT for integral frequencies? Well, the side-lobes of a rectangular pulse look like a sinc function which has nulls in between the peaks. As we will see, the default DFT of a finite signal looks like the signal multiplied by a rectangular pulse that truncates a signal (which is otherwise assumed to be infinite). For integral frequencies, the nulls exactly line up with the other frequencies, and so there is no leakage. For non-integral frequencies, the nulls don't line up, and so we see leakage.

Periodic signal view





Yet another view

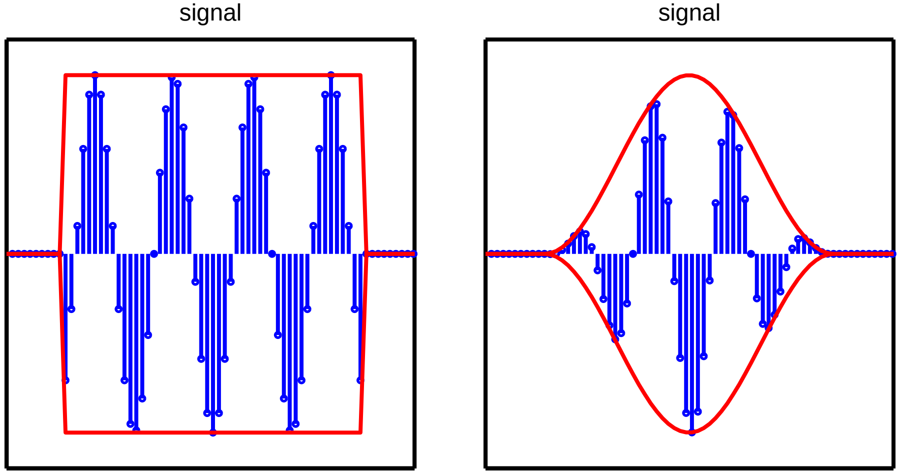
The signal can be thought of as an infinite duration signal, that has been truncated by rectangular window.

- input signal is the result of the product of an (infinite) signal with a rectangular window
  - convolution property
  - resulting FT is the convolution of the FT of the rectangle (a sinc), with the FT of the signal
  - FT of a rectangle is a sinc function
- what happens if we use a smoother window?


```
% file:       complex_view_4.m, (c) Matthew Roughan, Sun Aug 22 2004
N = 64;
x = (0:N-1)/N;
f1 = 3.4;
y1 = sin(2*pi*f1*x);
figure(11)
hold off
plot([x; x], [zeros(size(y1)); y1],'b', 'linewidth', 3);
hold on
plot(x, y1, 'bo', 'linewidth', 4);
set(gca, 'linewidth', 3, 'fontsize', 18, 'ylim', [-1.2 1.2], 'xlim', [0 max(x)], 'xtick', [], 'ytick',
title('signal');
set(gcf, 'PaperUnits', 'centimeters', 'PaperOrientation', 'portrait', 'PaperPosition', [0 0 20 12])
print('-depsc', 'Plots/circular_view_of_period_1.eps');

%% now do circular view
figure(21)
theta = 0:2*pi/N:2*pi-2*pi/N;
circ_x = cos(theta);
circ_y = sin(theta);
hold off
plot3(circ_x, circ_y, y1, 'o', 'linewidth', 5);
hold on
plot3([circ_x; circ_x], [circ_y; circ_y], [zeros(size(y1)); y1], 'b-', 'linewidth', 3);
plot3(circ_x, circ_y, zeros(size(y1)), 'k:', 'linewidth', 3);
grid on
set(gca,'xlim',[-1.5 1.5],'ylim',[-1.5 1.5], 'linewidth', 3, 'fontsize', 18);
view(-130,50);
set(gcf, 'PaperUnits', 'centimeters', 'PaperOrientation', 'portrait', 'PaperPosition', [0 0 18 14])
print('-depsc', 'Plots/circular_view_of_period_2.eps');
```

Windows

Windows reduce the transient at the edges, but giving edge points less weight, e.g.



Impact of Windows

Product of the signal with a window function

product in time domain = convolution in frequency domain

- just have to look at transfer functions of windows.
- want to reduce size of side-lobes
- we can choose our own window function!

Note that windows may drop the overall power of the signal so (by Rayleigh-Parseval) the power in the output signal drops. However, relative magnitudes are more important here than absolutes!

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.9/79

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.10/79

Windows

All defined for $n = 0, 1, \ldots, N-1$

- Rectangular (default) $w_N(n) = 1$
- Bartlett (triangular) $w_N(n) = 1 - \left|\frac{n-N/2}{N/2}\right|$
- Welch (Riesz) $w_N(n) = 1 - \left(\frac{n-N/2}{N/2}\right)^2$
- Hanning $w_N(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right)$
- Hamming $w_N(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$
- Blackman $w_N(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{N}\right) + 0.08\cos\left(\frac{4\pi n}{N}\right)$
- Blackman-Harris (3 term)
  $w_N(n) = 0.42323 - 0.49755\cos\left(\frac{2\pi n}{N}\right) + 0.07922\cos\left(\frac{4\pi n}{N}\right)$
- Gaussian $w_N(n) = \exp\left[-4.5\left(\frac{n-N/2}{N/2}\right)^2\right]$

Basis for Comparison

- measure drop to largest side-lobe
  - measure of sensitivity
- also measure "width" of the windows frequency response by looking at where the power drops off by a factor of a half, i.e. we find $\Delta\omega$ such that

$$\frac{|F(\Delta\omega/2)|^2}{|F(0)|^2} = \frac{1}{2}$$

  - minimum resolution bandwidth
  - two peaks of same magnitude have to be at least this far apart to resolve them as separate
- we will also look at some other properties in a minute

See

- Lyons, pages 77, 179, and 486
- Bracewell, pages 164-169, and 171
- fred harris, "On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform", Proceedings of the IEEE, Vol.66, No.1, January 1978, pp 51-83.

Also see
http://www.cis.rit.edu/resources/software/sig_manual/windows.html
http://astronomy.swin.edu.au/~pbourke/other/windows/
http://en.wikipedia.org/wiki/Window_function

Other window functions: Blackman-Nuttall, Bartlett-Hann, Bessel, von Hann, Tukey, Cauchy, Bohman, Dolph-Tschebyscheff, Taylor, Extended Cosine Bell Window, Riemann, ...

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.11/79

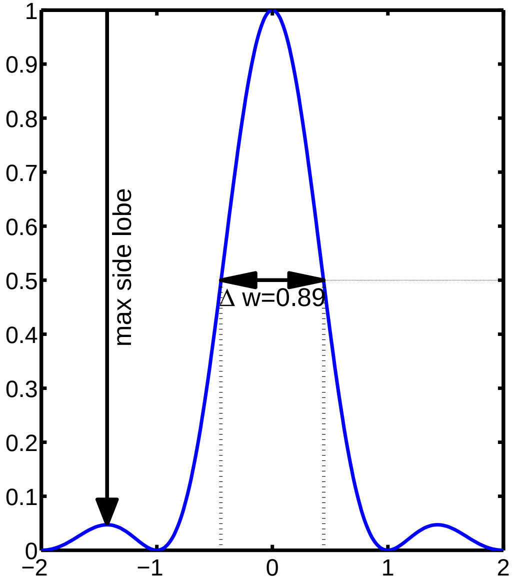
Other metrics that aren't considered here

- equivalent noise bandwidth
- scalloping loss

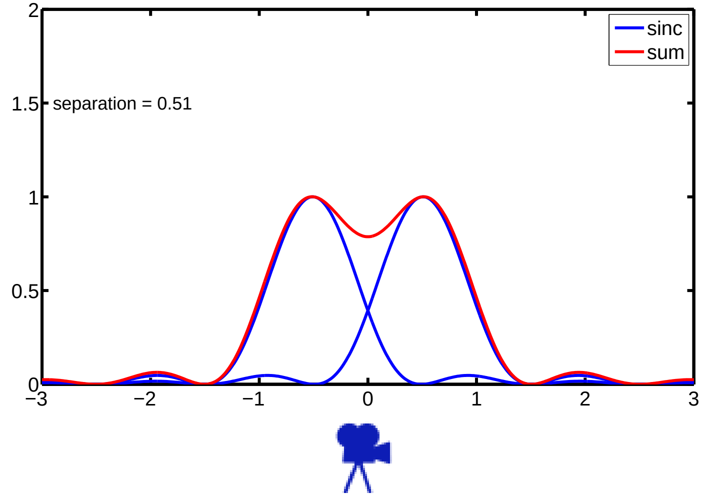
Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.12/79

Rectangular Window

- calculate results for rectangular window
- $\mathcal{F}\{r(t)\} = \mathrm{sinc}(s)$
- figure shows $\mathrm{sinc}(s)^2$
- $\Delta w = 0.89$ is the width
- drop to the max side lobe is -13 dB



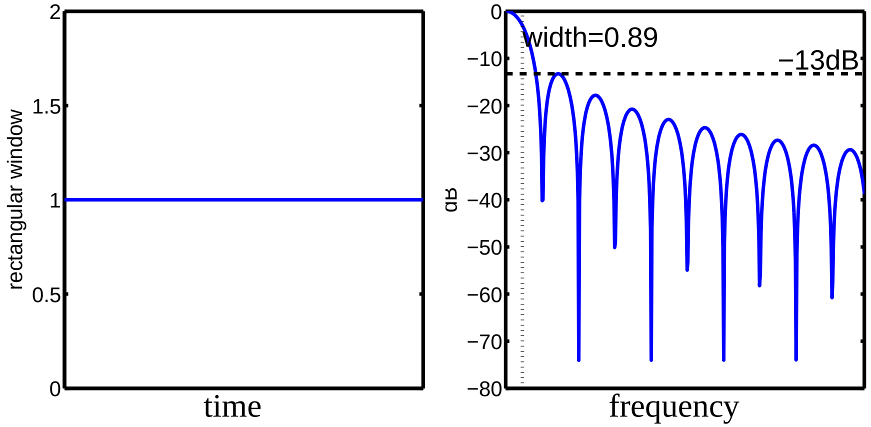
Rectangular Window



Note the "nulls", or places where the spectrum drops to zero. These are at $\pm 1, 2, \ldots$, and so exactly line up with other frequencies for integral frequencies – so no leakage.

Rectangular Window

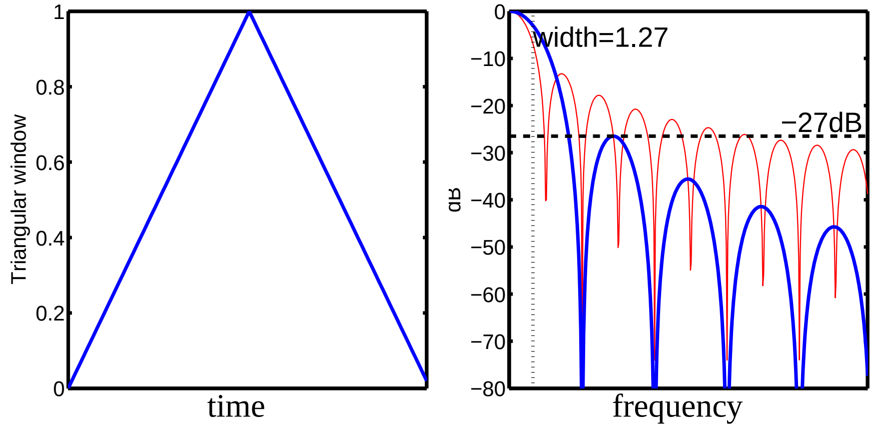
The window you have when you don't have a window



$$w_N(n) = 1$$

Triangular Window

Reduce the size of the discontinuity



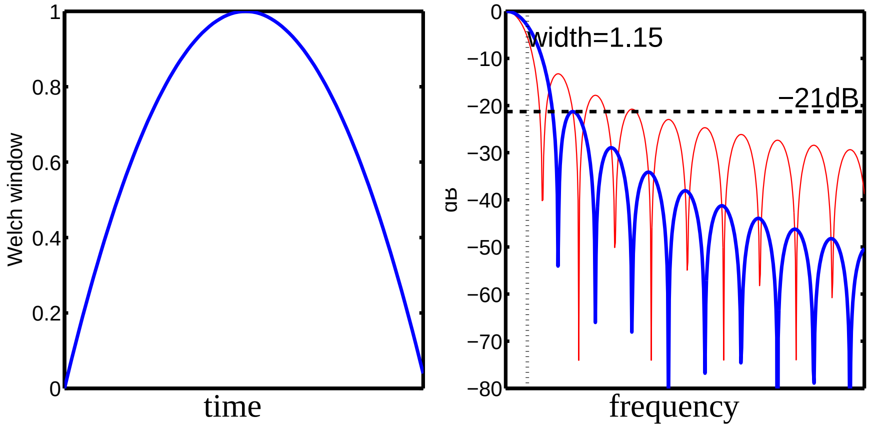
$$w_N(n) = 1 - \left| \frac{n - N/2}{N/2} \right|$$

The red curve provides a reference to the frequency response of the rectangular window (for comparison).

We can see immediately that the first side lobe of the triangular window is much lower (-27 dB instead of -13 dB). However, the width is significantly larger, 1.27 as opposed to 0.89.

Welch Window

Reduce the size of the discontinuity, but keep power.

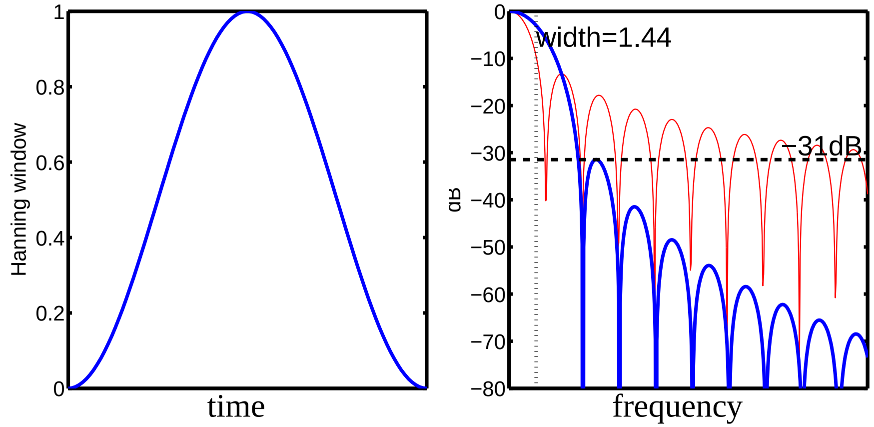


$$w_N(n) = 1 - \left(\frac{n - N/2}{N/2}\right)^2$$

The first side lobe is lower than the rectangular window, but not as low as for the triangular window, but the main lobe is narrower.

Hanning Window

Make the discontinuity smooth, as well as small.

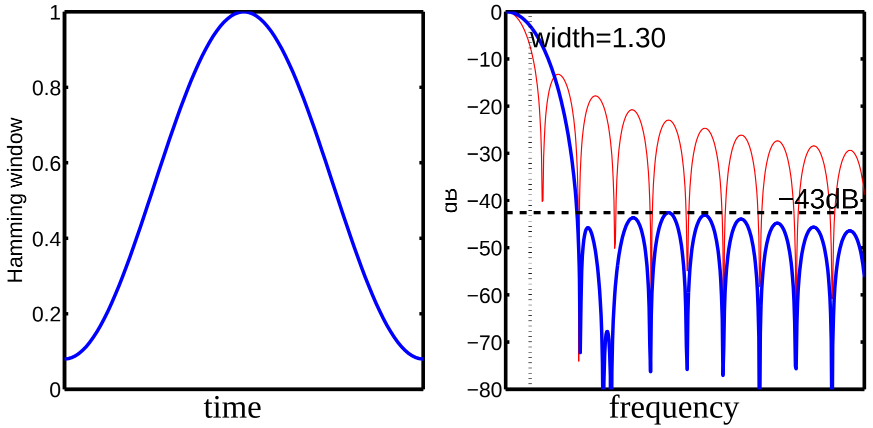


$$w_N(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right)$$

The aim here is to reduce the discontinuity in the first derivative (as well at the step discontinuity). The result (as we will see later) is not just a lower side lobe, but also a faster roll off in the side lobes.

Hamming Window

Side-lobe reduction, from $w_N(n) = \alpha + (1-\alpha)\cos\left(\frac{2\pi n}{N}\right)$



$$w_N(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$$

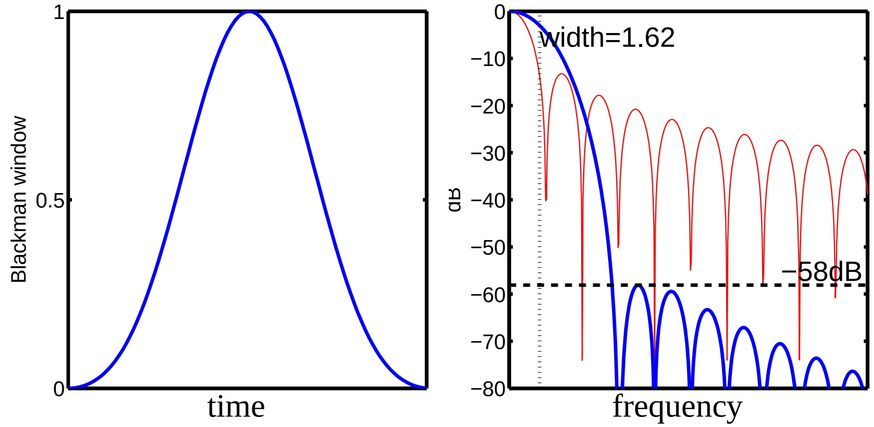
The Hamming window starts from a family of windows $w_N(n) = \alpha + (1-\alpha)\cos\left(\frac{2\pi n}{N}\right)$ and chooses the value of $\alpha = 0.54$ that minimizes the side lobe.

A noticeable feature of the Hamming window is that unlike the Hanning window its side-lobes don't roll off (but they do start lower). We will study why this is in a minute, but note that the window does have a small discontinuity at the edge.

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.19/79

Blackman Window

Side-lobe reduction through an extra term.

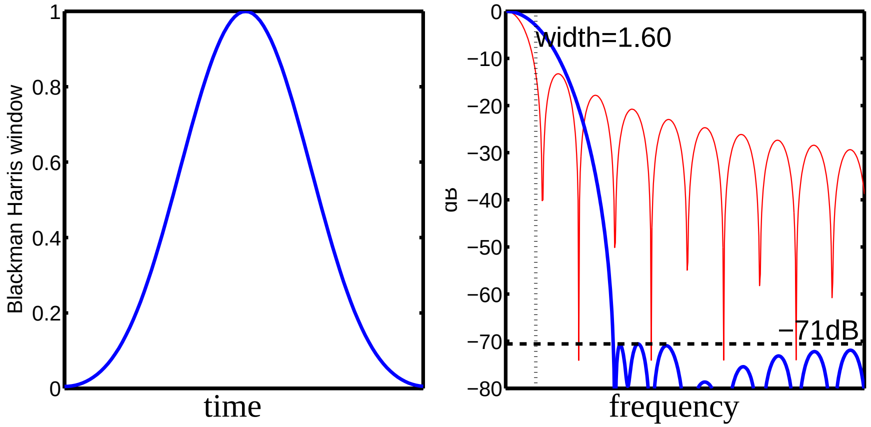


$$w_N(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{N}\right) + 0.08\cos\left(\frac{4\pi n}{N}\right)$$

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.20/79

Blackman-Harris Window (3 term)

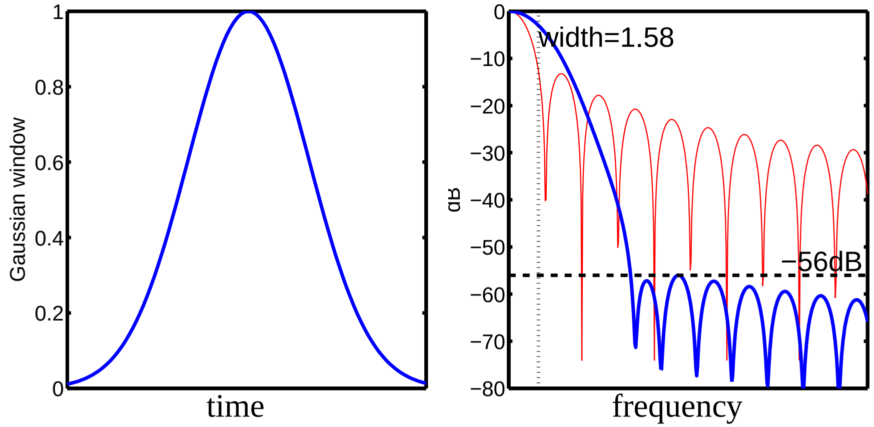
Optimize Blackman.



$$w_N(n) = 0.42323 - 0.49755\cos\left(\frac{2\pi n}{N}\right) + 0.07922\cos\left(\frac{4\pi n}{N}\right)$$

Gaussian Window

Minimum uncertainty (see later)



$$w_N(n) = \exp\left[-4.5\left(\frac{n - N/2}{N/2}\right)^2\right]$$

Note that the co-efficients of the Blackman-Harris Window are very close to the Blackman window, but that this still has a large impact. Clearly, if such small changes can have an impact, we even have to consider the impact of finite precision on side-lobes.

The Gaussian windows are actually a family, with a parameter (4.5 in the example) that can be tuned to tradeoff resolution for the side lobe reduction.

Generally speaking there are often such tradeoffs here.

Windowing

There is a tradeoff between resolution and sensitivity!

- better sensitivity (lower side-lobes)
  $\Rightarrow$ less resolution
- better resolution (of frequencies)
  $\Rightarrow$ worse sensitivity

Another tradeoff in the roll-off of side lobes.

- smoother function
  $\Rightarrow$ steeper roll-off
  but less drop off in first side-lobe

Some windows have a parameter that can tune the tradeoffs

Tunable tradeoffs

Windows with tunable tradeoffs

- Kaiser-Bessel

$$w_N(n) = \frac{I_0\left[B\sqrt{1-\left(\frac{n-p}{p}\right)^2}\right]}{I_0(B)}$$

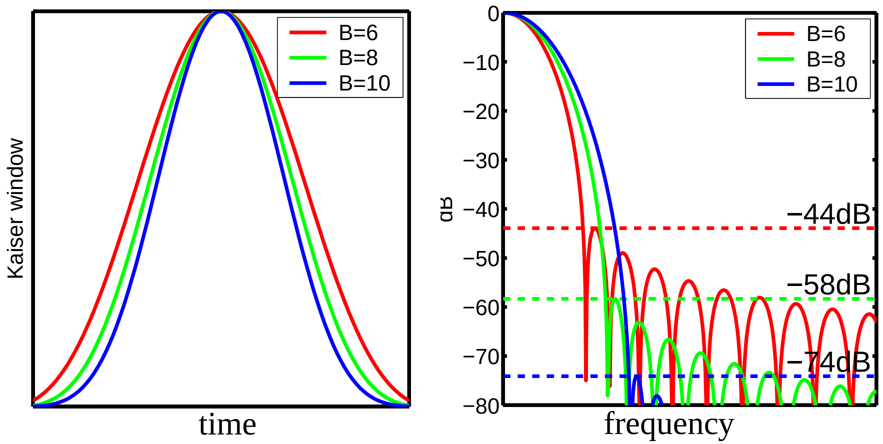
where $p = \frac{N-1}{2}$, and I_0 is the zero order modified Bessel function of the first kind, given by $I_0(x) = 1 + \sum_{k=0}^{\infty} \frac{(x/2)^{2k}}{(k!)^2}$

Choosing different values of B tunes the tradeoff

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.23/79

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.24/79

Kaiser Window



$$w_N(n) = I_0\left[B\sqrt{1-\left(\frac{n-p}{p}\right)^2}\right]/I_0(B)$$

Tunable tradeoffs

Windows with tunable tradeoffs

- Chebyshev (Dolph-Chebyshev)
  $w_N(n) =$ the N-point inverse DFT of
  $$\frac{\cos\left[N\cos^{-1}\left(\alpha\cos\left(\frac{\pi m}{N}\right)\right)\right]}{\cosh(N\cosh^{-1}(\alpha))}$$
  where
  $$\alpha = \cosh\left[\frac{1}{N}\cosh^{-1}(10^{\gamma})\right]$$

not shown here

The figure shows the window and its frequency response for three different values of B.

Tunable tradeoffs

Windows with tunable tradeoffs

- Gaussian (tune the standard deviation)

$$w_N(n) = \exp\left[-\alpha\left(\frac{n-N/2}{N/2}\right)^2\right]$$

We find size of the discontinuity at the edge of the window by taking $n=0$, e.g. it is $\exp(-\alpha)$. The side-lobes from such an edge will resemble the rectangular side-lobes, with their -13dB attenuation, and so the side-lobes of the Gaussian will be approximately

$$\text{side-lobe} = -13 + 20\log_{10} e^{-\alpha} = -13 - 20\alpha\log_{10} e$$

Actually they vary from this a little, but the relationship is useful, as we can also predict the width of the Gaussian window precisely as it is just a scaled version of itself.

The Uncertainty Principle

We have seen there are basic tradeoffs in window choice. The uncertaintly principle shows that these tradeoffs are fundamental and unavoidable.

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.27/79

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.28/79

Uncertainty principle

The tradeoff relates to a general principle: **uncertainty**

- We can't squeeze more information out of a sequence
- we can only change the way we see the information
- here we tradeoff sensitivity for resolution

Scaling property of FTs tells us something

$$f(at) \rightarrow \frac{1}{|a|} F\left(\frac{s}{a}\right)$$

- if we make the window 'narrower' to exclude more of the transients (that cause leakage), then we make the FT 'wider'

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.29/79

Uncertainty principle

Another way to understand

- frequency resolution depends on the number of data points in our dataset
- Windowing reduces the power from some data points
- a little like reducing the number of data points
- so we need a longer data sequence for a finer resolution

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.30/79

Regularity and decay

We can extend the intuition from the above by looking at relationship between regularity of the function $f(t)$ and the decay rate of $|F(s)|$, e.g.

If there exists a constant K, and $\varepsilon > 0$ such that

$$|F(s)| \leq \frac{K}{1+|s|^{p+1+\varepsilon}}$$

Then f has at least p continuous derivatives.

Hence, if $F(s)$ has compact support then $f \in C^\infty$.

Regularity and decay

Proof: By definition of the IFT

$$f(t) = \int_{-\infty}^{\infty} F(s) e^{i2\pi st}\, ds$$

If $F \in L^1(\mathbb{R})$ then the above implies f is continuous and bounded, because

$$|f(t)| \leq \int_{-\infty}^{\infty} |F(s)e^{i2\pi st}|\, ds = \int_{-\infty}^{\infty} |F(s)|\, ds$$

Take the kth order derivative, WRT to t, and we get

$$|f^{(k)}(t)| \leq \int_{-\infty}^{\infty} |(i2\pi s)^k F(s) e^{i2\pi st}|\, ds \leq (2\pi)^k \int_{-\infty}^{\infty} |s|^k |F(s)|\, ds$$

We use the inequality

$$\left| \int_{-\infty}^{\infty} g(s)\, ds \right| \leq \int_{-\infty}^{\infty} |g(s)|\, ds$$

and the fact that $|e^{i2\pi st}| = 1$.

Then remember the differentiation formula for FTs (from Lecture 2)

$$\mathcal{F}\left\{ \frac{d^n}{dt^n} f(t) \right\} = (i2\pi s)^n F(s)$$

and $|i| = 1$.

Regularity and decay

Proof: Now, if

$$|F(s)| \leq \frac{K}{1+|s|^{p+1+\varepsilon}}$$

Then,

$$\int_{-\infty}^{\infty} |F(s)|\,(1+|s|^p)\,ds \leq \int_{-\infty}^{\infty} \frac{K(1+|s|^p)}{1+|s|^{p+1+\varepsilon}}\,ds < \infty$$

which also implies that

$$\int_{-\infty}^{\infty} |F(s)|\,|s|^k\,ds < \infty$$

for all $k \leq p$, so the derivative $f^{(k)}(t)$ exists and is bounded.

See Mallat, p.29-30.

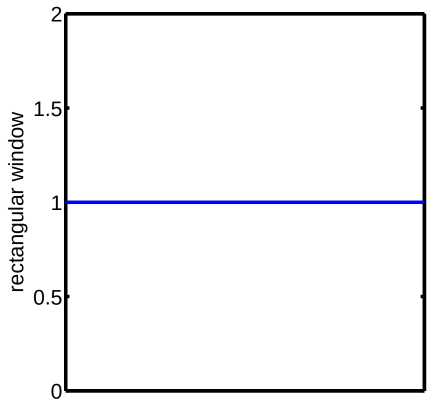
Regularity and decay

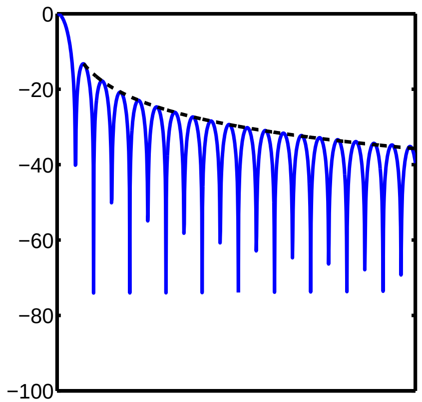
Windowing examples: consider 5 windowing functions

- **Rectangular:** $w_N(n) = 1$
  This has a discontinuity.
- **Triangular:** $w_N(n) = 1 - \left|\frac{n-N/2}{N/2}\right|$
  This has a discontinuity in the first derivative.
- **Welch:** $w_N(n) = 1 - \left(\frac{n-N/2}{N/2}\right)^2$
  This has a discontinuity in the first derivative.
- **Hanning:** $w_N(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right)$
  This has a discontinuity in the 2nd derivative.
- **Hamming:** $w_N(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$
  This has a discontinuity, but of smaller size than for the rectangular window.

Rectangular Window

Discontinuous

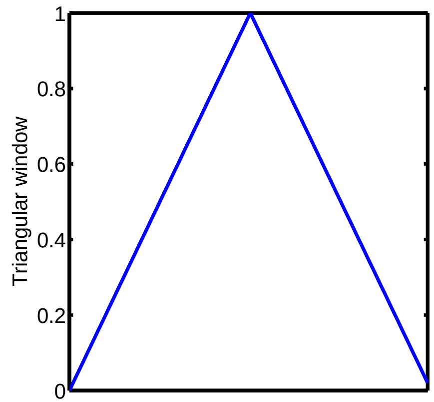


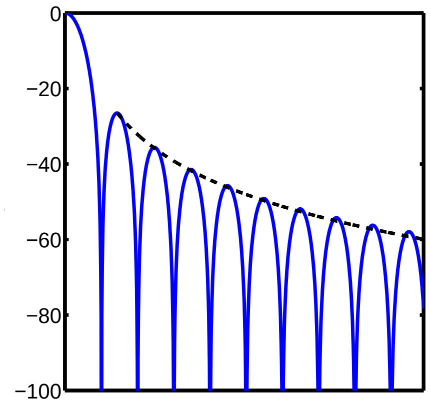


Decay $F(s) \sim \frac{K}{1+|s|}$

Triangular Window

Discontinuity in the first derivative





Decay $F(s) \sim \frac{K}{1+|s|^2}$

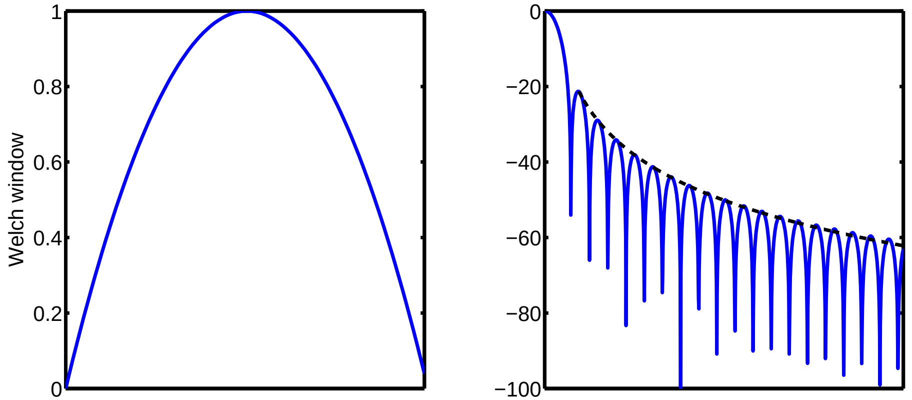
The black dashed line is $\propto 1/(1+|s|)$, the decay of the side-lobes.

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.35/79

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.36/79

Welch Window

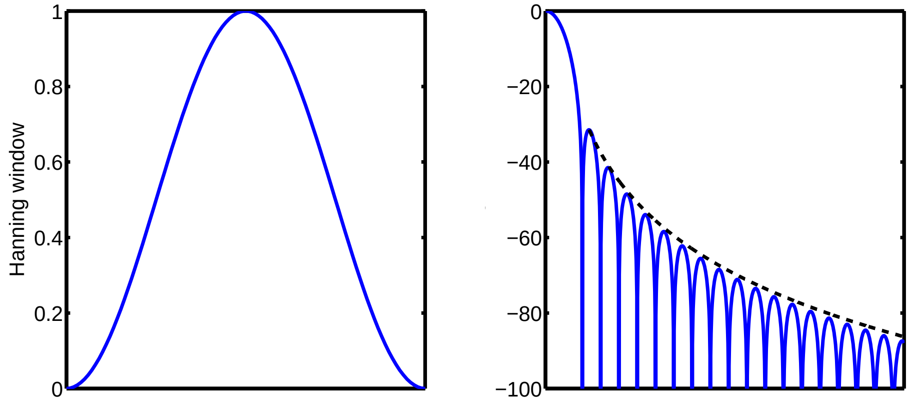
Discontinuity in the first derivative



Decay $F(s) \sim \dfrac{K}{1+|s|^2}$

Hanning Window

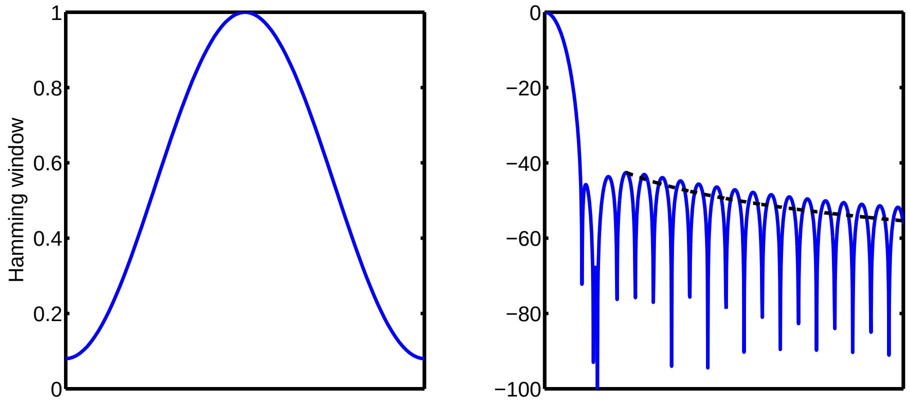
Discontinuity in the 2nd derivative



Decay $F(s) \sim \dfrac{K}{1+|s|^3}$

Hamming Window

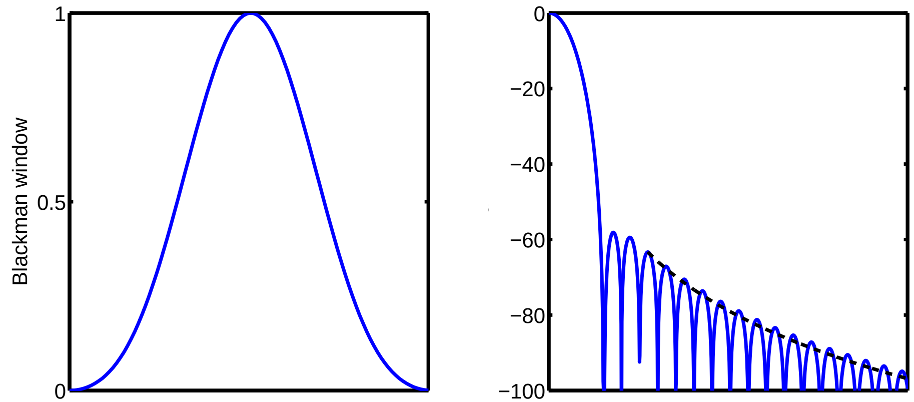
Discontinuous ($w_N(0) = 0.54 - 0.46 = 0.08$)



Decay $F(s) \sim \dfrac{K}{1+|s|}$

Blackman Window

Discontinuity in the 2nd derivative



Decay $F(s) \sim \dfrac{K}{1+|s|^3}$

Regularity and decay and duality

Duality implies that $\mathcal{F}\{F(t)\} = f(-s)$, so the above regularity properties work in reverse as well, e.g. if there exists a constant K, and $\varepsilon > 0$ such that

$$|f(t)| \leq \frac{K}{1+|t|^{p+1+\varepsilon}}$$

Then F has at least p continuous derivatives.

Hence, if $f(t)$ has compact support then $F \in C^\infty$.

Compactness

Theorem: If $f \neq 0$ has compact support then $F(s)$ can't be 0 on a whole interval. Similarly, if $F \neq 0$ has compact support then $f(t)$ can't be zero on a whole interval.

Proof: Assume $F(s) \neq 0$ has compact support in the interval $[-b,b]$. From the definition of the IFT

$$f(t) = \int_{-b}^{b} F(s) e^{i2\pi st}\, ds$$

If non-trivial function $f(t) = 0$ for $t \in [c,d]$, then $f^{(n)}(t_0) = 0$ inside the interval (c,d), and so by differentiating n times under the integral at t_0,

$$0 = f^{(n)}(t_0) = \int_{-b}^{b} F(s)(i2\pi s)^n e^{i2\pi st_0}\, ds$$

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.41/79

The **support** of a function is the set where the function takes non-zero values. We write the support as

$$\text{supp}\{f\} = \{x | f(x) \neq 0\}$$

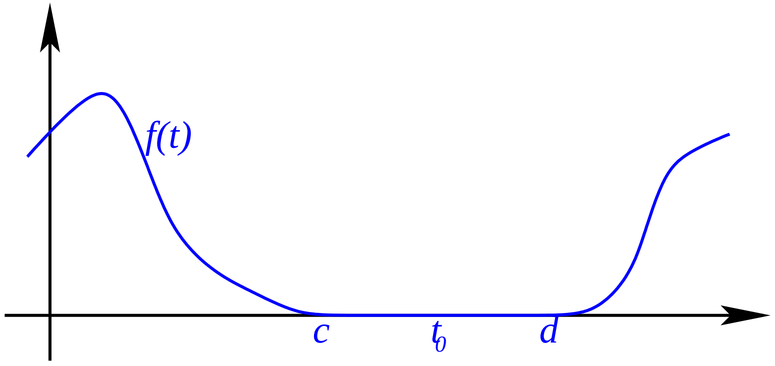
Functions with **compact support** are zero outside a compact set. For example, a function with compact support would only be non-zero in the set $[-a,b] \in \mathbb{R}$,

$$f(x) = 0 \quad \text{for } x \notin [-a,b]$$

NB: I have tried to keep this simple, omitting definitions of compact set (a closed, bounded set), and the fact that we define support to be the closure of a set.

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.42/79

Compactness



- The function is flat (constant) at t_0
- The derivatives of f at t_0 must all be 0

Compactness

Proof: We can write the IFT as

$$f(t) = \int_{-b}^{b} F(s) e^{i2\pi s(t-t_0)} e^{i2\pi s t_0} \, ds$$

And we can expand the exponential $\exp(i2\pi s(t-t_0))$ in a power series about t_0 to get

$$f(t) = \sum_{n=0}^{\infty} \frac{(t-t_0)^n}{n!} \int_{-b}^{b} F(s) (i2\pi s)^n e^{i2\pi s t_0} \, ds$$

However note that we have already shown that each of the integrals in the above sum are zero, and so the sum is zero leading to $f(t) = 0$, which contradicts the assumption that $f(t)$ is nontrivial (i.e. $f(t) \neq 0$ for some t).

See Mallat, pp.32-33

Remember that the Taylor series for an exponential is given by

$$e^{i2\pi s(t-t_0)} = \sum_{n=0}^{\infty} \frac{(t-t_0)^n}{n!} (i2\pi s)^n.$$

Meaning

Given a

- more compact
- irregular
- sharper

function in the time (frequency) domain we get a

- less compact
- smoother
- wider

function in the frequency (time) domain.

Windows summary

Name	max side lobe	width	roll off polynomial
rectangular	-13 dB	0.89	$K/(1+\lvert s\rvert)$
triangular (Bartlett)	-27 dB	1.28	$K/(1+\lvert s\rvert^2)$
Welch (Riesz)	-21 dB	1.16	$K/(1+\lvert s\rvert^2)$
Hanning	-32 dB	1.44	$K/(1+\lvert s\rvert^3)$
Hamming	-43 dB	1.30	$K/(1+\lvert s\rvert)$
Blackman	-58 dB	1.68	$K/(1+\lvert s\rvert^3)$
Blackman-Harris	-67 dB	1.66	$K/(1+\lvert s\rvert)$
Kaiser (B=6)	-44 dB	1.40	$K/(1+\lvert s\rvert)$
Kaiser (B=8)	-58 dB	1.58	$K/(1+\lvert s\rvert)$
Kaiser (B=10)	-74 dB	1.74	$K/(1+\lvert s\rvert)$
Gaussian (α=4.5)	-56 dB	1.55	$K/(1+\lvert s\rvert)$

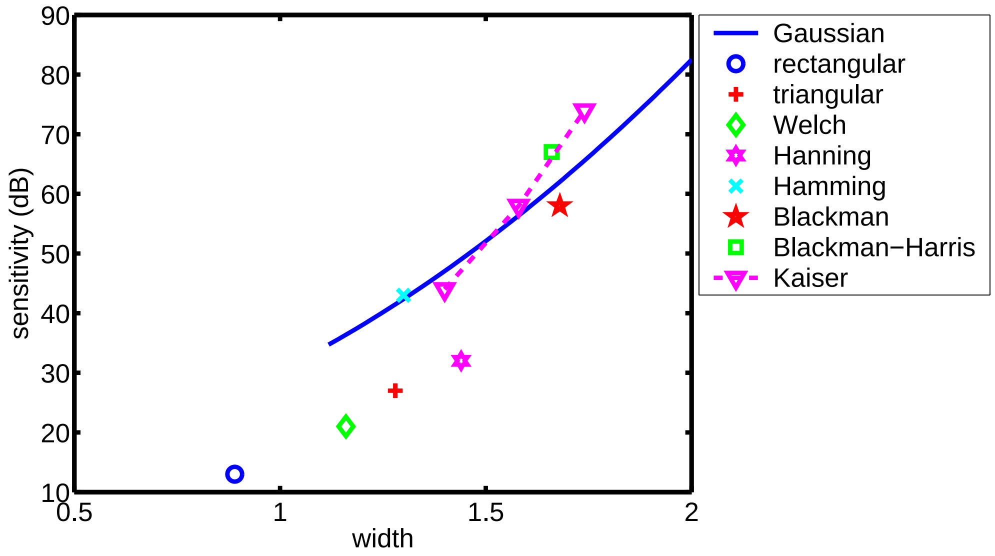
Where available results from "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform", F.J.Harris, Proc. of the IEEE, Vol.66, No.1, Jan. 1978, pp.51-83.

We will extend these concepts later when we consider the uncertainty principle.

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.45/79

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.46/79

Sensitivity vs resolution



Transient Signal Analysis

We have previously seen that the Fourier transform is not appropriate for analyzing transient signals, but what should we do then? The first step is to look at the Short-Time or Windowed Fourier Transform.

Figure shows the tradeoff between sensitivity and resolution. As noted, there are other tradeoffs not shown in the figure.

Transient signals

- all signals are transient
  - they have a start and stop at least
- sometimes this doesn't matter
- often it does
  - conversation is full of transients
  - music
  - images
- the Fourier transform
  - the Fourier transform is nice because it diagonalises **time-invariant** linear systems
  - doesn't localize in time at all
- we need something more for transient signals

EM Frequency Band Allocations

Frequency Band	Designation	Typical use
3-30 kHz	Very Low Frequency (VLF)	Long-range navigation
30-300 kHz	Low Frequency (LF)	Marine Communications
300-3000 kHz	Medium Frequency (MF)	AM radio
3-30 MHz	High Frequency (HF)	Jindalee, Amateur Radio
30-300 MHz	Very High Frequency (VHF)	FM radio, VHF TV
300-3000 MHz	Ultra High Frequency (UHF)	UHF TV, radar
3-30 GHz	SuperHigh Frequency (SHF)	Satellite Comms

From p. 308 of Philips, Parr and Riskin.

- Audible sound frequencies: $\sim 20 - 20{,}000 Hz$
- AM radio frequencies: $535 - 1615 kHz$
- FM radio frequencies: $88 - 108 MHz$
- how should we carry sound on radio?

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.49/79

Some references:
http://encyclopedia.thefreedictionary.com/Radio\%20spectrum
http://encyclopedia.thefreedictionary.com/Electromagnetic\%20spectrum
http://encyclopedia.thefreedictionary.com/AM\%20band
*– ignore \ sign before % in the URLs above

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.50/79

Amplitude Modulation (AM)

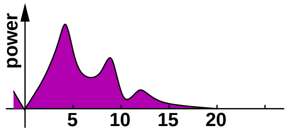
AM Radio

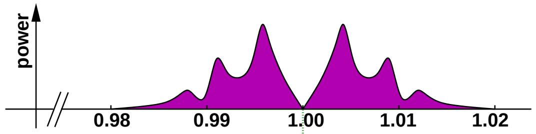
- modulation of signal $x(t)$ with a cosine function

$$y(t) = \cos(2\pi f_{\text{carrier}} t)\,[1 + x(t)]$$

- $f_{\text{carrier}} \in [535 - 1615] kHz$
- modulation property of FT

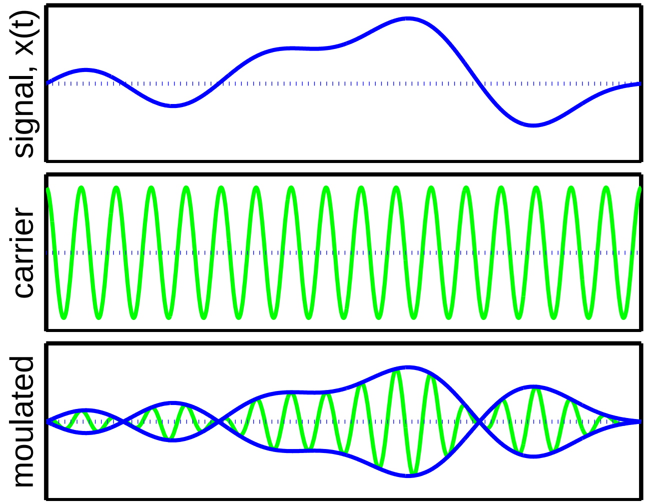
$$\mathcal{F}\{f(t)\cos(2\pi s_0 t)\} = \frac{1}{2}F(s - s_0) + \frac{1}{2}F(s + s_0)$$





Amplitude Modulation (AM)

$$y(t) = \cos(2\pi f_{\text{carrier}} t)\,|x(t)|$$



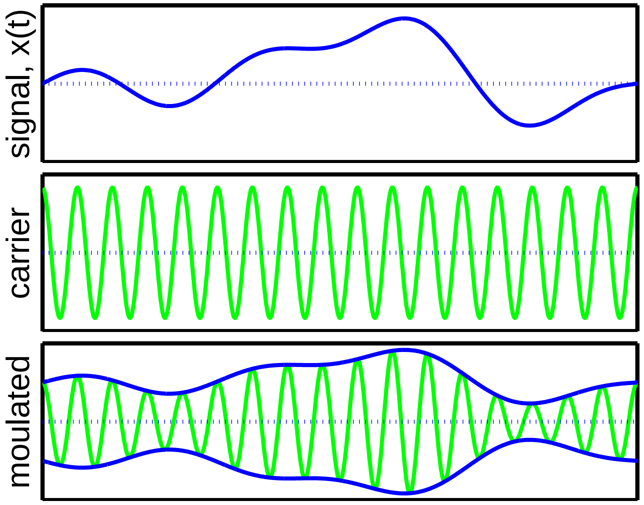
Amplitude modulation allows us to send a audio signal (with frequencies of up to a few kHz, in the frequency band of hundreds of kHz.

p. 309 of Philip, Parr and Riskin


```
% file:       am.m, (c) Matthew Roughan, Wed Oct  6 2004
%
N = 1000; x = (1:N)/N;
f = 0.3 * sin(pi*x) - 0.4 * sin(pi*3*x) + 0.2 * sin(pi*4*x) + 0.2 * sin(pi*7*x);
carrier_freq = 17;
carrier = cos(2*pi*carrier_freq*x);
am_signal = carrier .* abs(f);

figure(1)
hold off
plot(x, f, 'b', 'linewidth', 3);
hold on
plot([0 1], [0 0], ':', 'linewidth', 3);
set(gca, 'xtick', [], 'ytick', [], 'fontsize', 24, 'linewidth', 3);
ylabel('signal, x(t)');
set(gcf, 'PaperUnits', 'centimeters', 'PaperPosition', [0 0 20 5])
print('-depsc', 'Plots/am_1.eps');

figure(2)
hold off
plot(x, carrier, 'g', 'linewidth', 3);
hold on
plot([0 1], [0 0], ':', 'linewidth', 3);
set(gca, 'xtick', [], 'ytick', [], 'ylim', [-1.2 1.2], 'fontsize', 24, 'linewidth', 3);
ylabel('carrier');
set(gcf, 'PaperUnits', 'centimeters', 'PaperPosition', [0 0 20 5])
print('-depsc', 'Plots/am_2.eps');

figure(3)
hold off
plot(x, am_signal, 'g', 'linewidth', 3);
hold on
plot([0 1], [0 0], ':', 'linewidth', 3);
plot(x, [f; -f], 'b', 'linewidth', 3);
set(gca, 'xtick', [], 'ytick', [], 'ylim', [-1.2 1.2], 'fontsize', 24, 'linewidth', 3);
ylabel('moulated');
set(gcf, 'PaperUnits', 'centimeters', 'PaperPosition', [0 0 20 5])
print('-depsc', 'Plots/am_3.eps');
```

Amplitude Modulation (AM)

$$y(t) = \cos\left(2\pi f_{\text{carrier}} t\right)\left[1 + x(t)\right]$$



Frequency Modulation (FM)

FM Radio (88-108 MHz in the US)

- modulate the frequency of the signal
- given a signal $x(t)$ to be transmitted

$$y(t) = \cos(2\pi\phi(t))$$

  where $\phi(t)$ is now a (non-linear) function of time, depending on the signal $x(t)$ to be transmitted.

- ***instantaneous frequency*** is the rate of change of phase, e.g.

$$f(t) = \frac{d}{dt}\phi(t)$$

- so take

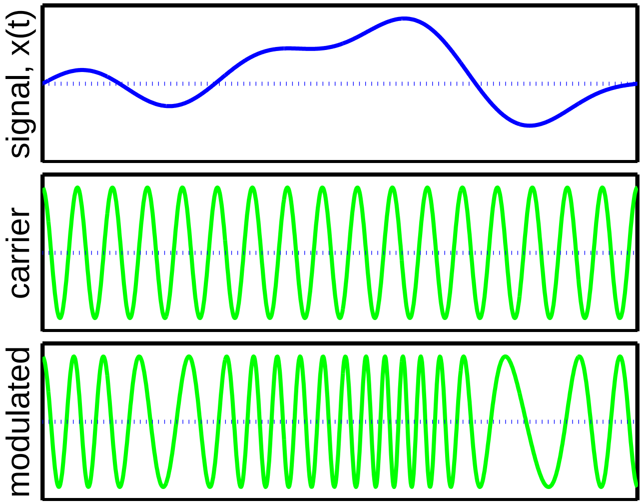
$$\phi(t) = \int_0^t f_{\text{carrier}} + x(t)\, dt$$

Note that the result is a transient signal! The (carrier) frequency is constant, but the amplitude changes over time. This is not that interest (yet) because the only non-stationarity in the signal is in the input signal.

Some references:
`http://encyclopedia.thefreedictionary.com/Frequency\%20modulation`
`http://encyclopedia.thefreedictionary.com/FM\%20radio`
*– ignore \ sign before % in the URLs above

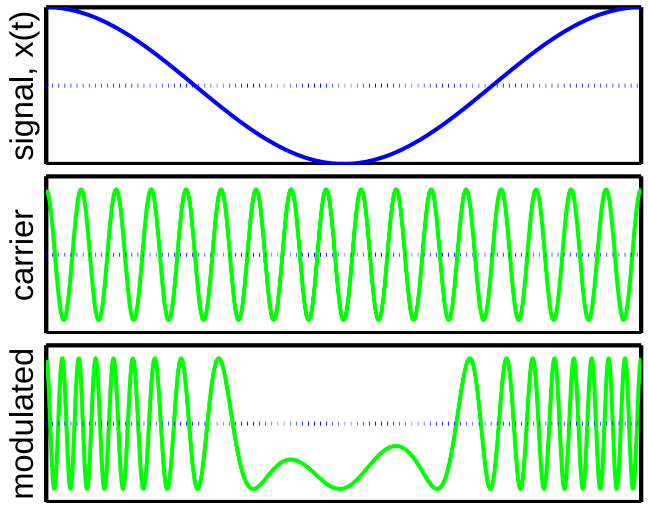
Frequency Modulation (FM)

$y(t) = \cos(2\pi\phi(t))$ where $\phi(t) = f_{\text{carrier}}t + \beta \int_0^t x(t)\,dt$



Frequency Modulation (FM)

The FM signal is transient (even if input isn't)




```
% file:       fm.m, (c) Matthew Roughan, Wed Oct  6 2004
%
N = 1000; x = (1:N)/N;
f = 0.3 * sin(pi*x) - 0.4 * sin(pi*3*x) + 0.2 * sin(pi*4*x) + 0.2 * sin(pi*7*x);
carrier_freq = 17;
carrier = cos(2*pi*carrier_freq*x);
phi = cumsum(carrier_freq+20*f)/N;
fm_signal = cos(2*pi*phi);

figure(1)
hold off
plot(x, f, 'b', 'linewidth', 3);
hold on
plot([0 1], [0 0], ':', 'linewidth', 3);
set(gca, 'xtick', [], 'ytick', []);
set(gca, 'fontsize', 24, 'linewidth', 3);
ylabel('signal, x(t)');
set(gcf, 'PaperUnits', 'centimeters', 'PaperPosition', [0 0 20 5])
print('-depsc', 'Plots/fm_1.eps');

figure(2)
hold off
plot(x, carrier, 'g', 'linewidth', 3);
hold on
plot([0 1], [0 0], ':', 'linewidth', 3);
set(gca, 'xtick', [], 'ytick', [], 'ylim', [-1.2 1.2], 'fontsize', 24, 'linewidth', 3);
ylabel('carrier');
set(gcf, 'PaperUnits', 'centimeters', 'PaperPosition', [0 0 20 5])
print('-depsc', 'Plots/fm_2.eps');

figure(3)
hold off
plot(x, fm_signal, 'g', 'linewidth', 3);
hold on
plot([0 1], [0 0], ':', 'linewidth', 3);
set(gca, 'xtick', [], 'ytick', [], 'ylim', [-1.2 1.2], 'fontsize', 24, 'linewidth', 3);
ylabel('modulated');
set(gcf, 'PaperUnits', 'centimeters', 'PaperPosition', [0 0 20 5])
print('-depsc', 'Plots/fm_3.eps');
```


The modulated signal is

$$y(t) = \cos(2\pi\phi(t)) \text{ where } \phi(t) = f_{\text{carrier}}t + \beta \int_0^t x(t)\,dt$$

Take input $x(t) = \cos(2\pi t)$, and then

$$\begin{aligned} \phi(t) &= f_{\text{carrier}}t + \beta \int_0^t x(t)\,dt \\ &= f_{\text{carrier}}t + \beta' \sin(2\pi t) \end{aligned}$$

Envelope and phase

We can change two things

- Envelope (amplitude modulation)
- Phase (frequency modulation)

Result is a signal

$$y(t) = A(t)\cos\left(2\pi\phi(t)\right)$$

In transient analysis of this signal we would like to be able to determine $A(t)$ and $\phi(t)$.

- note that a real signal (e.g. music) would consist of a superposition of a number of such terms, e.g.
  - plucked string has a number of harmonics
  - each decays at different rates

A Chirp

Both Amplitude and Frequency Modulation can happen at once, a simple example being a chirp. Examples:

- a linear chirp

$$y(t) = A(t)\cos\left[2\pi(bt^2 + ct)\right]$$

  Instantaneous frequency

$$f(t) = \frac{d}{dt}\left[bt^2 + ct\right] = 2bt + c$$

- a hyperbolic chirp

$$y(t) = \cos\left(\frac{2\pi\alpha}{\beta - t}\right) \quad \text{and} \quad f(t) = \frac{\alpha}{(\beta - t)^2}$$

Note that the representation is not necessarily unique, e.g.

$$2\sin x \sin y = \cos(x-y) - \cos(x+y)$$

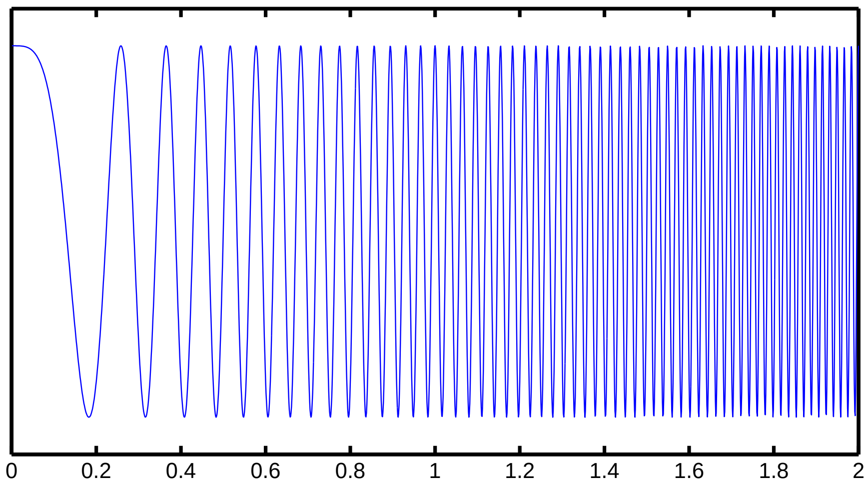
So we write

$$\begin{aligned} 2\sin(2\pi ft)\sin(2\pi\phi(t)) &= \cos\left[2\pi(ft - \phi(t))\right] - \cos\left[2\pi(ft + \phi(t))\right] \\ A(t)\sin(2\pi\phi(t)) &= \cos\left[2\pi(ft - \phi(t))\right] - \cos\left[2\pi(ft + \phi(t))\right] \end{aligned}$$

So we could represent our signal as above (with varying amplitude and frequency term), or in another representation with only varying frequency.

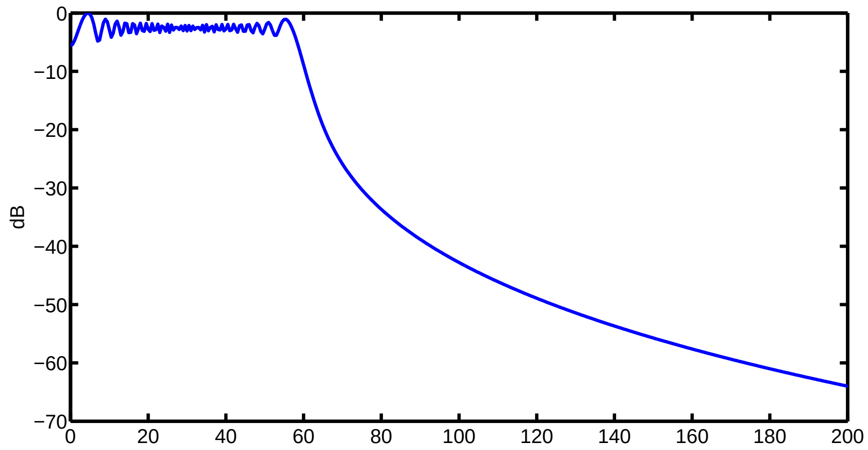
An Example Chirp

A Chirp $y(t) = \cos(2\pi 15 t^2)$, so instantaneous freq. $f(t) = 30t$



A Chirp

DFT of a chirp




```
% file:       chirp_spectrogram.m, (c) Matthew Roughan, Thu Oct  7 2004
%
N = 1000; K=2;
Fs = N;
t = (1:K*N)/N;
b = 15.0;
c = 0.0;
chirp = cos(2*pi*(b*t.^2 + c * t));
% b = -15.0;
% c = 60.0;
% chirp = cos(2*pi*(b*t.^2 + c * t));
z = fft(chirp);
w = abs(fftshift(z));
w = 20*log10(w/max(w));

figure(12)
hold off
plot(t, chirp, 'linewidth', 1);
set(gca, 'ytick', [], 'ylim', [-1.2 1.2], 'xlim', [0 2], 'fontsize', 18, 'linewidth', 3);
set(gcf, 'PaperUnits', 'centimeters', 'PaperPosition', [0 0 32 16])
print('-depsc', 'Plots/chirp_spect_1.eps');

figure(15)
plot(-Fs/2:Fs/(K*N):Fs/2-Fs/(K*N), w, 'linewidth', 3);
set(gca, 'fontsize', 18, 'linewidth', 3);
set(gca, 'xlim', [0 200]);
ylabel('dB');
set(gcf, 'PaperUnits', 'centimeters', 'PaperPosition', [0 0 32 16])
print('-depsc', 'Plots/chirp_spect_5.eps');

figure(16)
s = -Fs/2+Fs/(K*N):Fs/(K*N):Fs/2;
result = calc_sinc(s);
hold off
plot(s, w, 'linewidth', 3);
hold on
plot(s, 20*log10(abs(result)), 'g');
% plot(s, 20*log10(abs(cos(s.^2/(8*pi*b) + pi/4))), 'm');
plot(s, 20*log10(abs(conv2(cos(s.^2/(pi*b) + pi/4), result, 'same'))), 'r');
```


Note that the spectrum appears to be roughly flat over a range from 0 to 60 Hz. We know, however, that the frequency should have distinct peaks (at any point in time), and that the spread occurs because we are averaging a frequency that changes from 0 to 60 over the course of the 2 second observation.

Instantaneous frequency

Just to reiterate, given a signal

$$y(t) = A(t)\cos[2\pi\phi(t)]$$

The **Instantaneous frequency** is

$$f(t) = \frac{d\phi}{dt}$$

Windows, windows everywhere

We can use windowing functions in other ways

- analysis of transient signals
- use windows to select a chunk of data
- move the window along
- so perform FT of a series of functions

$$g(\tau;t) = f(\tau)w^*(\tau - t)$$

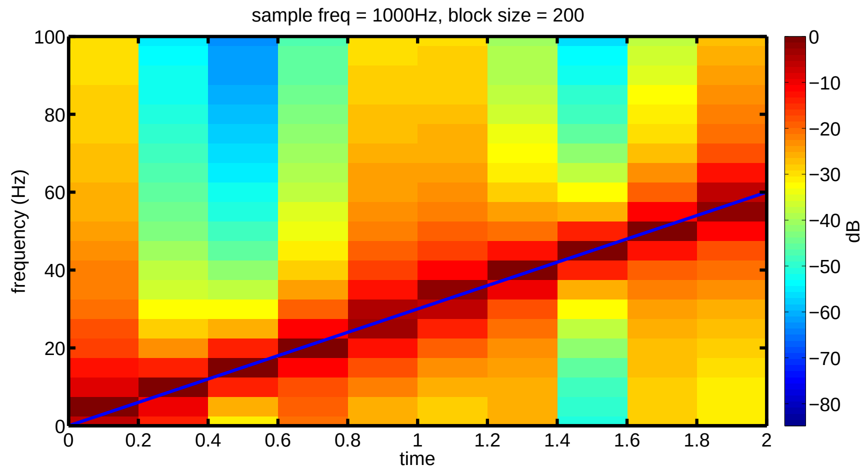
- we get the Short-Time Fourier Transform (STFT)

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.61/79

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.62/79

STFT of a Chirp

STFT of the chirp from slide 59.



Short Time Fourier Transform

- The Fourier transform goes from time- to frequency-domain
  - lose all time dependence
- but, e.g. music does not have same frequency over long time periods
- want to frequencies over shorter time periods
- get STFT by applying time-shifted window function $w(\tau - t)$

$$STFT\{f;t,s\} = \int_{-\infty}^{\infty} f(\tau)w^*(\tau - t)e^{-i2\pi s\tau}\,d\tau$$

Magnitude^2 of the STFT results in the **spectrogram**.

$$\text{spectrogram}(f;t,s) = |STFT\{f;t,s\}|^2$$

Spectrogram code:
`http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=1553&objectType=file`

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.63/79

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.64/79

Discrete time STFT

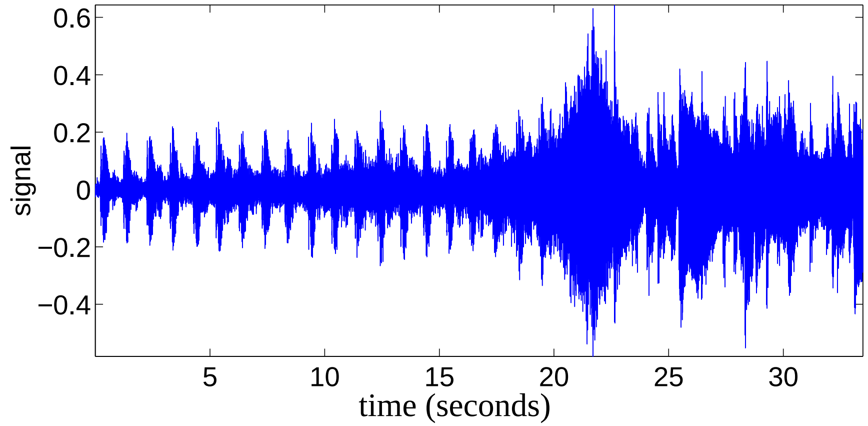
- apply a standard window (e.g. Hamming)
- DFT a block of data of size M from n to $n+M-1$.
- do so for all n.

$$DSTFT\{x;n,k\} = X(n;k) = \sum_{m=n}^{n+M-1} f(m)w^*(m-n)e^{-i2\pi ikm/N}$$

- In $X(n;k)$ the n indexes time (in the trans. domain)
- In $X(n;k)$ the k indexes frequency (as in the DFT)
- often it is only performed on non-overlapping blocks
  - only calculate $X(n;k)$ at time points $n = 0, M, 2M, \ldots$

Example spectrogram

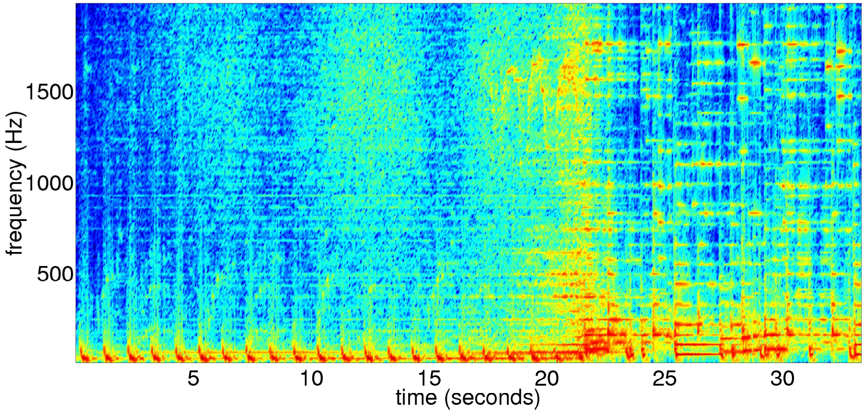
Most sounds aren't continuous, they are **transient**



Dark side of the moon: Breathe clip

NB: in the STFT, doesn't matter if the sequence is finite, or not, because we only analyze finite chunks.

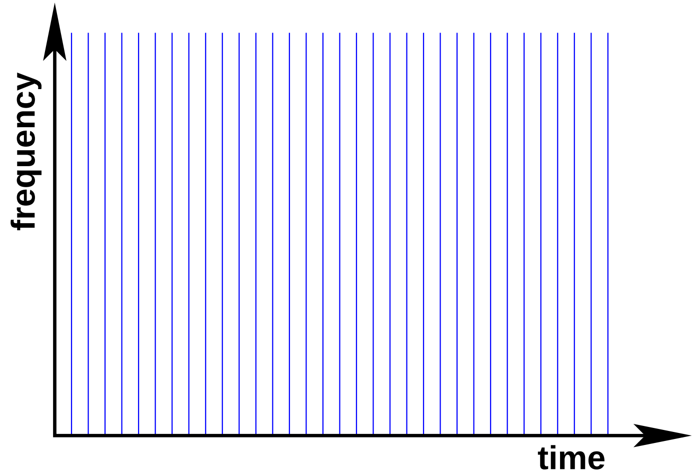
Example spectrogram



Dark side of the moon: Breathe clip

Cutting up the time-frequency space

Time domain

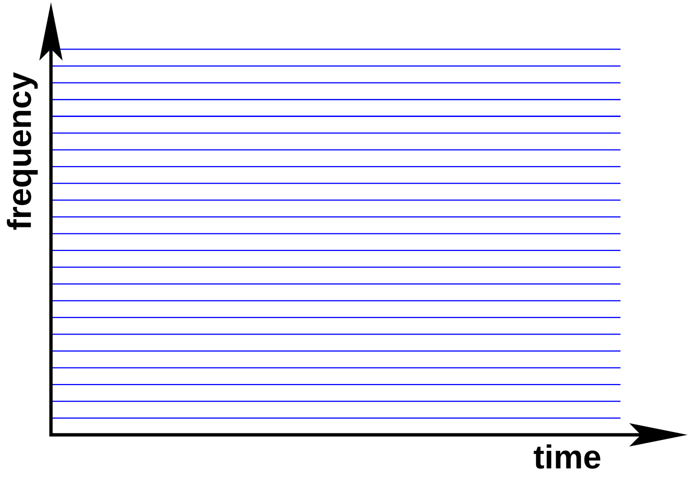


Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.67/79

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.68/79

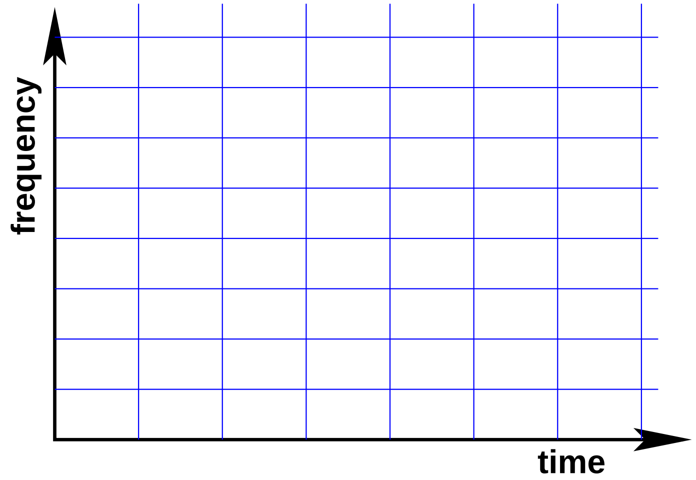
Cutting up the time-frequency space

Frequency domain (Fourier transform)



Cutting up the time-frequency space

STFT



Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.69/79

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.70/79

Uncertainty

Fundamental limitation on STFT is uncertainty

- if we make the window short, we get good time resolution, but poor frequency resolution
- if we make the window long, we get poor time resolution, but good frequency resolution
- we can't do better in both
- there is an uncertainty bound between time and frequency

Resolution

Given rectangular window of width M samples, and sampling intervals t_s

- time resolution is just Mt_s
  - signals have to be in different boxes to be resolved
- frequency resolution is $1/Mt_s$
  - standard frequency resolution for a series with sample rate $f_s = 1/t_s$ and M samples.

Notice that the product of the two resolutions is a constant!

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.71/79

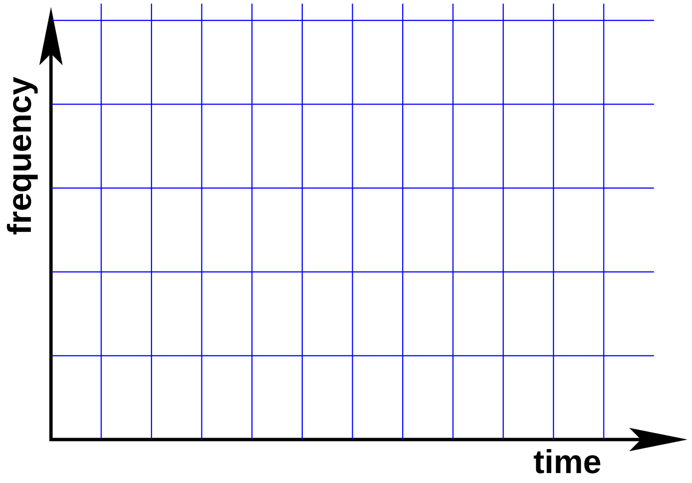
Overlapping windows doesn't really help. It may make the spectrogram look like it has finer time resolution, but it just has more points. We still couldn't **resolve** details any finer, but it can be useful to make the spectrogram look smoother.

Using different windows can change the effective frequency resolution, and therefore change the constant, but it is still constant for a given window function.

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.72/79

Cutting up the time-frequency space

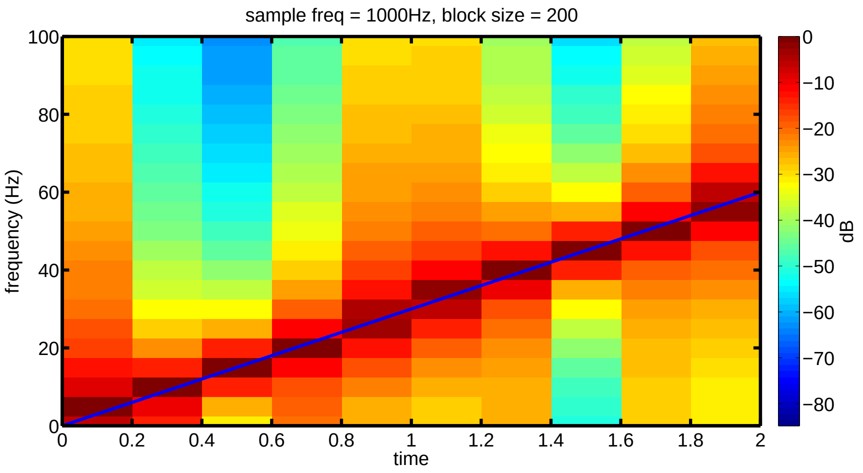
STFT (different width window)



Areas of boxes don't get smaller!

Spectrogram of a Chirp

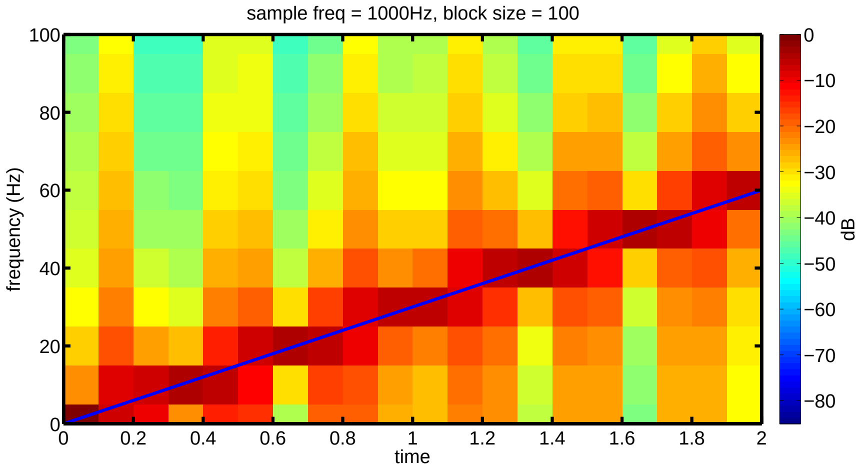
Spectrogram of a chirp



Area of boxes is the time resolution times the frequency resolution, and so from the argument above, the area is a constant.

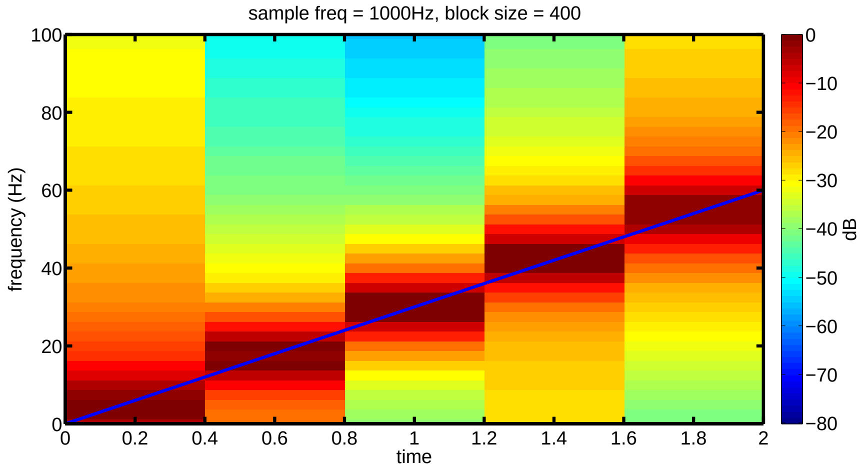
Spectrogram of a Chirp

Spectrogram of a chirp



Spectrogram of a Chirp

Spectrogram of a chirp



Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.75/79

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.76/79

Windows

- STFT with a window: sometimes called the Windowed FT (WFT)
- What window should we use?
- Choice involves tradeoffs
  - length of window (and hence computational cost) (e.g. does it have compact support)
  - size of uncertainty (Gabor function has minimal uncertainty region)
  - regularity of window, and roll off in Fourier domain
  - windows side-lobes vs its width in Fourier domain
  - can scale window and tradeoff frequency for time resolution

Limitations of the STFT

- computational cost $O(nm \log m)$
- time/frequency resolution tradeoff
  - small m better time, worse frequency resolution
  - large m better frequency, worse time resolution
- time/frequency resolution tradeoff is fixed
  - higher frequencies can change faster than lower frequencies
  - would be nice to have appropriate resolution for each frequency

The answer: wavelets

- next lecture

Transform Methods & Signal Processing (APP MTH 4043): lecture 08 – p.79/79