Transform Methods & Signal Processing

lecture 01

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

Discipline of Applied Mathematics
School of Mathematical Sciences
University of Adelaide

July 27, 2009

Introduction

OF bodies chang'd to various forms, I sing:
Ye Gods, from whom these miracles did spring,
Inspire my numbers with coelestial heat;
'Till I my long laborious work compleat:

Ovid, Metamorphoses

Note that not everything said in the lectures is included in the notes provided. You must listen, and take notes. These spaces in your handout notes are provided to allow you to take notes in lectures. You may be examined on material that is discussed in lectures, even if it does not appear explicitly in your notes!

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.1/61

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.2/61

Outline

- **Introduction**: (1 week)
- **Continuous Fourier transforms**: (1 week)
- **Discrete Fourier transforms**: (2 weeks)
- **Filters and Linear Systems**: (2 weeks)
- **The Radon Transform and tomography**: (1 week)
- **Random Processes and some theorems**: (1 week)
- **Wavelets**: (4 weeks)

More detailed outline available at

`http://internal.maths.adelaide.edu.au/people/mroughan/Lecture_notes/Transform_methods/`

Some reference books

- "Understanding Digital Signal Processing", R.G. Lyons, Prentice-Hall, 2nd edition, 2004.
- "Signals, Systems and Transforms", C.L.Phillips, J.M.Parr and E.A.Riskin, Prentice-Hall, 3rd edition, 2003.
- "The Fourier Transform and its Applications", R.N. Bracewell, McGraw-Hill, 2000.
- "A Wavelet Tour of Signal Processing", Stephan Mallat, Academic Press, 2001.
- "Digital Image Processing", R.C. Gonzalez and R.E. Woods, 3rd Ed., Prentice Hall, 2008.

On-line materials

All materials can also be found at

`http://internal.maths.adelaide.edu.au/people/mroughan/Lecture_notes/Transform_methods/`

MyUni is not used in this course.

Motivation

A man camped in a national park, and noticed Mr Snake and Mrs Snake slithering by. "Where are all the little snakes?" he asked.

Motivation for transformation

- operations on transformed data may be easier

$$\log(ab) = \log a + \log b$$

  insight behind slide rules
  - want to compute ab
  - take logs (with slide rule)
  - add the logs $\log a + \log b$
  - invert the log function $ab = \exp(\log a + \log b)$
- the same information is present
  but somehow more accessible
  - time domain vs frequency domain

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.6/61

The point is that sometimes transforms make it easier to do certain things, e.g. logarithms make it easier for us to multiply. The Fourier Transform is often used because it makes convolutions easier.

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.7/61

Application areas

- **signal and image processing**
- physics (e.g. astronomy)
- number theory
- probability theory and statistics
- cryptography
- acoustics
- oceanography and seismology
- optics and crystallography
- geometry
- everything else...

Applications: signal processing

- Internet traffic analysis
  - detect anomalies (DoS attacks and worms)
  - characterize traffic (as a fractal)
- Music generation and analysis
  - frequency, pitch and harmonics
  - music structure, and fractals
- Biomedical engineering
  - ECG processing
  - CAT and MRI scans
- Image processing
  - detecting objects in images
  - compression (JPEG, fingerprints)

More examples

- astronomy
  - enhancing blurry images
  - understanding repeated patterns: e.g. pulsars
- A/V
  - (cheap) digital music and TV
  - compression: music (e.g. MP3), images (JPEG), voice (cell phones, skype)
  - noise reduction
  - CGI (Computer Generated Images)
- telecommunications
  - echo suppression
  - equalization
- industry
  - process monitoring
  - finding problems (e.g. bad bearings on trains)
  - mining (finding ore bodies)
- military
  - sonar
  - radar

Integral transforms

- An **integral transform** is a transform defined in terms of an integral

$$f(t) \to \int f(t)g(t,s)dt$$

- Map a function (say of time) to a function of s
- $g(\cdot)$ is called the **kernel** of the transform
- notation (several alternatives)
  - $T\{f(t);s\} = \int f(t)g(t,s)\,dt$
  - $F(s) = \int f(t)g(t,s)\,dt$, $H(s) = \int h(t)g(t,s)\,dt$
  - $\mathcal{F}(s) = \int f(t)g(t,s)\,dt$, $\mathcal{H}(s) = \int h(t)g(t,s)\,dt$
  - $\tilde{f}(s) = \int f(t)g(t,s)\,dt$

Linear operators

- operators on functions (e.g. of time)
  could call it a functional
- linear operator $O\{f\}$ is defined by

$$O\{af+bh\} = aO\{f\}+bO\{h\}$$

  for $a,b,\in\mathbb{R}$
- integral transformations are linear operators

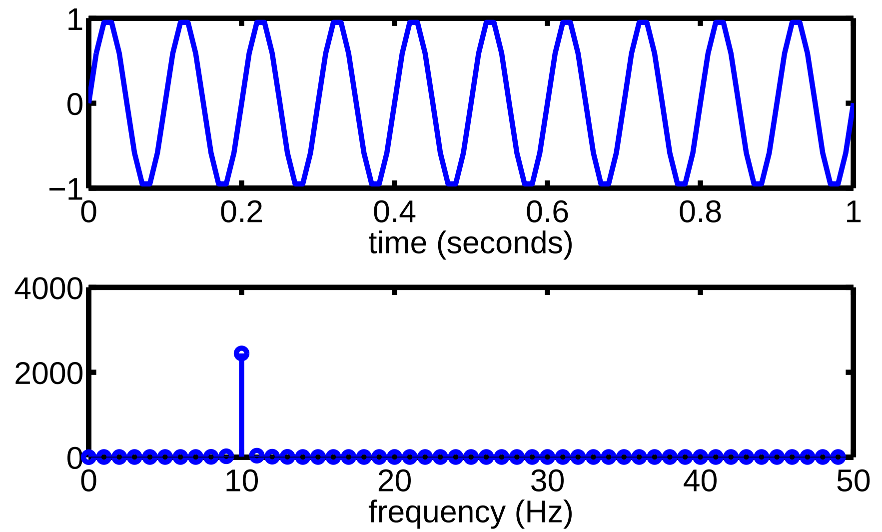
$$\int [af(t)+bh(t)]\,g(s,t)\,dt = a\int f(t)g(s,t)\,dt + b\int h(t)g(s,t)\,dt$$

Examples of integral transforms

Name	kernel $g(\cdot)$	transform of $f(t)$
Identity	$\delta(s-t)$	$F(s)=\int_{-\infty}^{\infty} f(t)\,\delta(s-t)\,dt$
Fourier	e^{-ist}	$F(s)=\int_{-\infty}^{\infty} f(t)\,e^{-ist}\,dt$
Laplace	e^{-st}, for $t \geq 0$	$F(s)=\int_{0}^{\infty} f(t)\,e^{-st}\,dt$
Hilbert	$\frac{1}{\pi(s-t)}$	$F(s)=\int_{-\infty}^{\infty} f(t)\,\frac{1}{\pi(s-t)}\,dt$
Mellin	t^{z-1}	$F(z)=\int_{0}^{\infty} f(t)\,t^{z-1}\,dt$
Fourier Cosine	$\cos(st)$	$F(s)=\int_{-\infty}^{\infty} f(t)\,\cos(st)\,dt$

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.12/61

An example: the Fourier transform



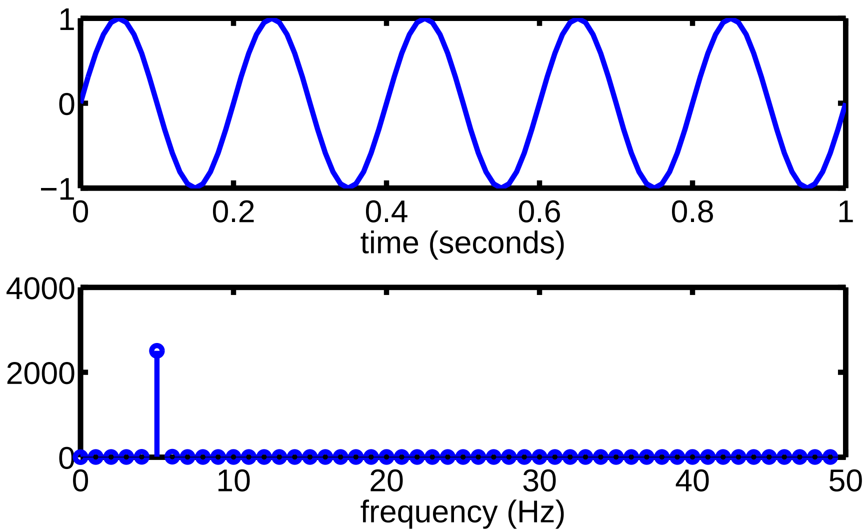
The figure shows a simple sine wave (top plot), and its power spectrum (lower plot), formed by taking the square of the (Discrete) Fourier transform.

Matlab code:

```
% file:       fft_sin_1.m, (c) Matthew Roughan, Sun Jun 27 2004
x = (0:0.01:1);
f = 10;              % frequency is ten cycles per second
y = sin(2* pi * f * x);

figure(1);
subplot(2,1,1)
hold off
plot(x, y, 'linewidth', 3);
set(gca,'xlim', [0 max(x)]);
set(gca,'linewidth', 3, 'fontsize', 18);
xlabel('time (seconds)');

z = abs(fft(y));
subplot(2,1,2)
hold off
stem(0:(length(z)-3)/2, z(1:end/2).^2, 'linewidth', 3);
set(gca,'linewidth', 3, 'fontsize', 18);
xlabel('frequency (Hz)');

set(gcf, 'PaperUnits', 'centimeters')
set(gcf, 'PaperPosition', [0 0 20 11.5])
print('-depsc', 'Plots/fft_sin_1.eps');
```

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.13/61

An example: the Fourier transform



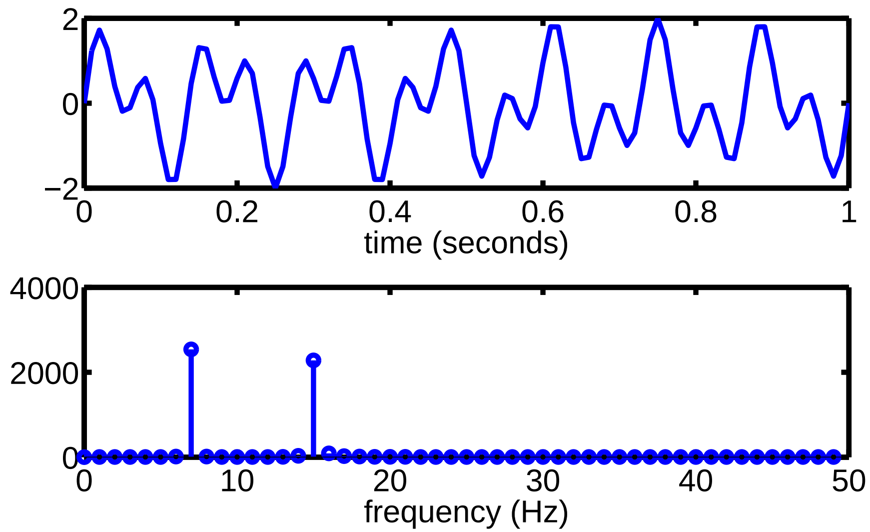
The figure shows a simple sine wave (top plot), and its power spectrum (lower plot), formed by taking the square of the (Discrete) Fourier transform.

Matlab code:

```
% file:       fft_sin_1a.m, (c) Matthew Roughan, Sun Jun 27 2004
x = (0:0.01:1);
f = 5;              % frequency is 5 cycles per second
y = sin(2* pi * f * x);

figure(1);
subplot(2,1,1)
hold off
plot(x, y, 'linewidth', 3);
set(gca,'xlim', [0 max(x)]);
set(gca,'linewidth', 3, 'fontsize', 18);
xlabel('time (seconds)');

z = abs(fft(y));
subplot(2,1,2)
hold off
stem(0:(length(z)-3)/2, z(1:end/2).^2, 'linewidth', 3);
set(gca,'linewidth', 3, 'fontsize', 18);
xlabel('frequency (Hz)');

set(gcf, 'PaperUnits', 'centimeters')
set(gcf, 'PaperPosition', [0 0 20 11.5])
print('-depsc', 'Plots/fft_sin_1a.eps');
```

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.14/61

An example: the Fourier transform

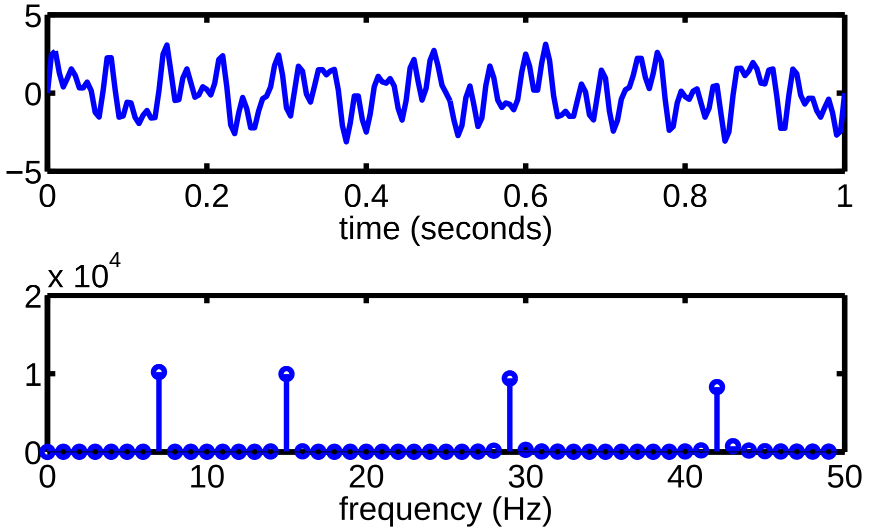


When we add two sine waves we can see them clearly separated in the frequency domain.

```
% file:       fft_sin_2.m, (c) Matthew Roughan, Sun Jun 27 2004
x = (0:0.01:1);
f1 = 7;             % frequency 1 = 7 Hz
f2 = 15;            % frequency 2 = 15 Hz
y = sin(2*pi*f1 * x) + sin(2*pi*f2 * x);

figure(2);
subplot(2,1,1)
hold off
plot(x, y, 'linewidth', 3);
set(gca,'xlim', [0 max(x)]);
set(gca,'linewidth', 3, 'fontsize', 18);
xlabel('time (seconds)');

z = abs(fft(y));
subplot(2,1,2)
hold off
stem(0:(length(z)-3)/2, z(1:end/2).^2, 'linewidth', 3);
set(gca,'linewidth', 3, 'fontsize', 18);
xlabel('frequency (Hz)');

set(gcf, 'PaperUnits', 'centimeters')
set(gcf, 'PaperPosition', [0 0 20 11.5])
print('-depsc', 'Plots/fft_sin_2.eps');
```

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.15/61

An example: the Fourier transform

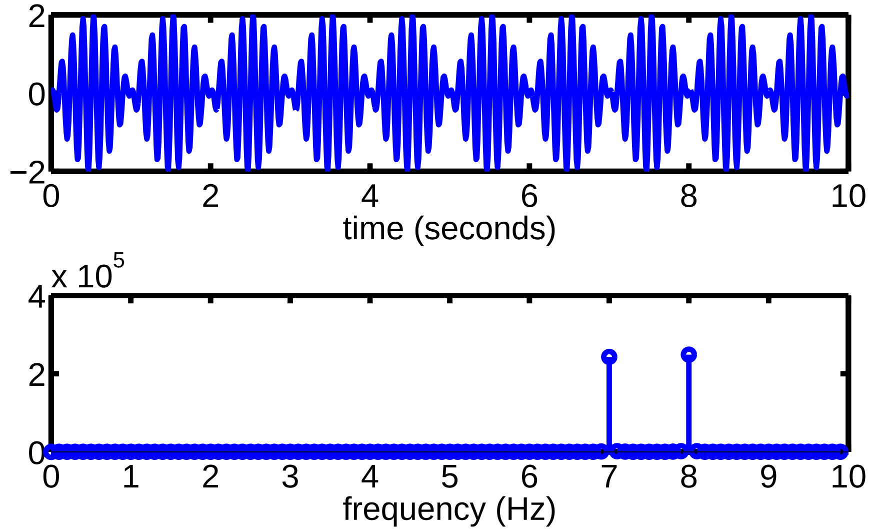


Even thought we know that the signal is constructed of four simple periodic functions, it is hard to see this when looking at the signal. However, it is obvious when we examine the Fourier transform.

```
% file:       fft_sin_3.m, (c) Matthew Roughan, Sun Jun 27 2004
x = (0:0.005:1);
f1 = 7;              % frequency 2 = 7 Hz
f2 = 15;             % frequency 2 = 15 Hz
f3 = 29;             % frequency 2 = 29 Hz
f4 = 42;             % frequency 2 = 42 Hz
y = sin(2*pi*f1 * x) + sin(2*pi*f2 * x) + sin(2*pi*f3 * x) + sin(2*pi*f4 * x);;

figure(3);
subplot(2,1,1)
hold off
plot(x, y, ’linewidth’, 3);
set(gca,’xlim’, [0 max(x)]);
set(gca,’linewidth’, 3, ’fontsize’, 18);
xlabel(’time (seconds)’);

z = abs(fft(y));
subplot(2,1,2)
hold off
stem(0:49, z(1:50).^2, ’linewidth’, 3);
set(gca,’linewidth’, 3, ’fontsize’, 18);
xlabel(’frequency (Hz)’);

set(gcf, ’PaperUnits’, ’centimeters’)
set(gcf, ’PaperPosition’, [0 0 20 11.5])
print(’-depsc’, ’Plots/fft_sin_3.eps’);
```

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.16/61

An example: Beats

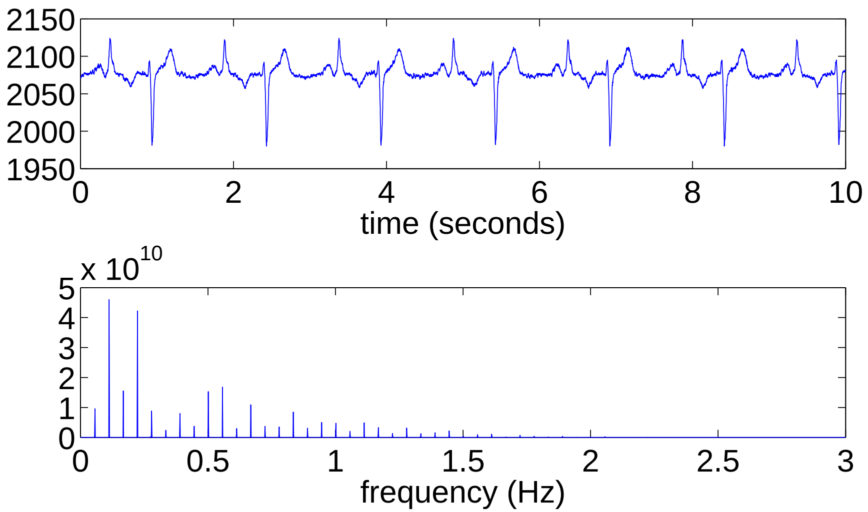


When two close notes are played together, we hear “beats”. The beat frequency is $|f_1 - f_2|$. This phenomena is often used in tuning instruments (e.g. guitars). In the above graph the beat frequency is obvious in the picture (and would be audible at one Hz). While the beats can be useful, it is also important to understand what frequencies are present in the signal, and in fact there is no signal at 1Hz, as we can see in the power spectrum.

```
% file:       beats.m, (c) Matthew Roughan, Wed Jul 23 2008
N = 10;
x = (0:0.01:N);
f1 = 7;              % frequency 1 = 7 Hz
f2 = 8;              % frequency 2 = 8 Hz
y = sin(2*pi*f1 * x + pi) + sin(2*pi*f2 * x);

figure(2);
subplot(2,1,1)
hold off
plot(x, y, ’linewidth’, 3);
set(gca,’xlim’, [0 max(x)]);
set(gca,’linewidth’, 3, ’fontsize’, 18);
xlabel(’time (seconds)’);

z = abs(fft(y));
subplot(2,1,2)
hold off
stem((0:99)/N, z(1:100).^2, ’linewidth’, 3);
set(gca,’linewidth’, 3, ’fontsize’, 18, ’xtick’, [0:10]);
xlabel(’frequency (Hz)’);

set(gcf, ’PaperUnits’, ’centimeters’)
set(gcf, ’PaperPosition’, [0 0 20 11.5])
print(’-depsc’, ’Plots/beats.eps’);
```

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.17/61

Example: ECG



Sound and Waves

Sound is formed from pressure waves in the air

- the disturbance propagates as the successive compression and rarefactions
- the number of compression-decompression sequences arriving at the detector during a chosen time interval is called the frequency
- The time interval between successive maximal compressions is called the period.
- The wavelength is the velocity divided by the frequency.
- At ground level and at 0^o C the speed of sound is approximately 331.5 metres per second

How does the machine that goes "beep" in a hospital work? Its looking for the period of your heartbeat, looking at a signal like this.

The wavelength of the note we call A=440Hz. proves to be about 753 mm (about 30 inches).

Pitch

Equal Temperament scale. Tuning Pitch: A=440Hz

Note	Frequency (Hz)						
A	27.50	55.00	110.00	220.00	440.00	880.00	1760.00
A#	29.13	58.27	116.54	233.08	466.16	932.32	1864.65
B	30.86	61.73	123.47	246.94	493.88	987.76	1975.53
C	32.70	65.40	130.81	261.62	523.25	1046.50	2093.00
C#	34.64	69.29	138.59	277.18	554.36	1108.73	2217.46
D	36.70	73.41	146.83	293.66	587.33	1174.65	2349.31
D#	38.89	77.78	155.56	311.12	622.25	1244.50	2489.01
E	41.20	82.40	164.81	329.62	659.25	1318.51	2637.02
F	43.65	87.30	174.61	349.22	698.45	1396.91	2793.82
F#	46.24	92.49	184.99	369.99	739.98	1479.97	2959.95
G	48.99	97.99	195.99	391.99	783.99	1567.98	3135.96
G#	51.91	103.82	207.65	415.30	830.60	1661.21	3322.43

Equal temperment scale

- Not the only possible scale
- convenient, because easy to change key
- each octave doubles the frequency
- 12 semitones per octave
- equal spacing on a log scale
- ratio between semi-tones is equal (hence the name of the scale), and therefore the ratio must be the 12th root of 2 $\simeq 1.0595$, e.g. ratio of D# to D

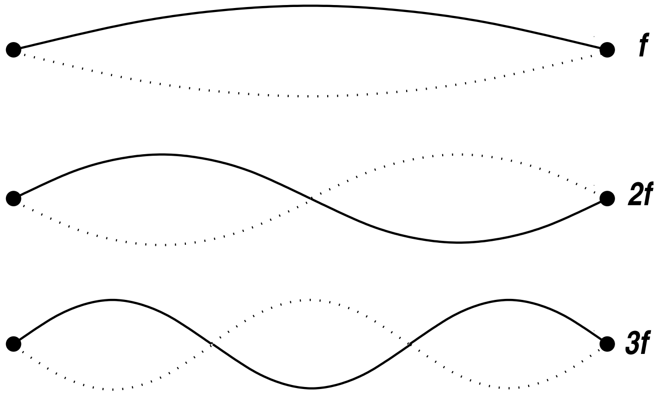
$$38.89/36.70 = 2^{1/12} = 1.0595$$

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.20/61

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.21/61

Harmonics

Real instruments don't generate pure sin waves



Vibrational resonances at fundamental frequency f and at $2f, 3f, \ldots$

We hear a mix of these **harmonics**

" Harmonie universelle, contenant la théorie et la pratique de la musique", published in Paris in 1636-7 by Franciscan friar Marin Mersenne, contains a bunch of good stuff:

- measured time of return of echos, and computed speed of sounds (accurate to within 10%)
- for a string under constant tension, frequency varies inversely as the length
- for a string of constant length, frequency is proportional to the square root of the tension
- for given length and constant tension, frequency varies inversely as the square root of the mass/unit length.
- determined frequencies of notes, by measuring slow cases, and using the relationships

Harmonics of vibrating string proposed by Daniel Bernoulli (1755).

http://www.dolmetsch.com/poshistory.htm

Harmonics of A (440 Hz)

Harmonic	Frequency	Normalized	Note name	how close
1 (fundamental)	440Hz	440Hz	A	100%
2	880Hz	440Hz	A	100%
3	1320Hz	660Hz	E	100%
4	1760Hz	440Hz	A	100%
5	2200Hz	550Hz	C#	99%
6	2640Hz	660Hz	E	100%
7	3080Hz	770Hz	G	98%
8	3520Hz	440Hz	A	100%
9	3960Hz	495Hz	B	100%
10	4400Hz	550Hz	C#	99%
11	4840Hz	605Hz	D	103%
12	5280Hz	660Hz	E	100%
13	5720Hz	715Hz	F#	97%
14	6160Hz	770Hz	G	98%
15	6600Hz	825Hz	G#	99%

Notice that the Harmonics are not all the same note. Also, the equal temperment scale doesn't exactly match the frequencies of a set of harmonics (the harmonics occur at powers of 2 of the fundamental frequency) – some are off by a few percent.

Musical Scale

A table showing the A scale

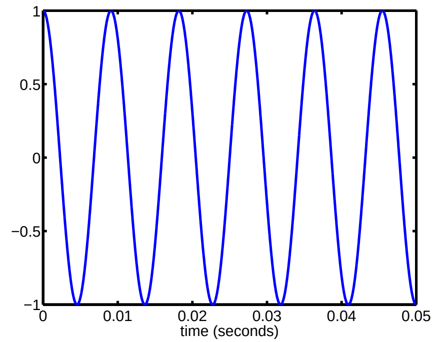
Interval name	Frequency Ratio	Example	Harmonic equivalent
Fundamental	1/1	A (440)	1st
Second	9/8	B (495)	9th
Third	5/4	C# (550)	5th and 10th
Fourth	4/3	D (586)	11th
Fifth	3/2	E (660)	3rd and 6th
Sixth	5/3	F# (733.3)	13th
Seventh	15/7	G# (825)	15th
Octave	2/1	A (880)	2nd, 4th, 8th

All of the notes in the scale fall (almost) on harmonics, and most of the harmonics are represented in the scale.

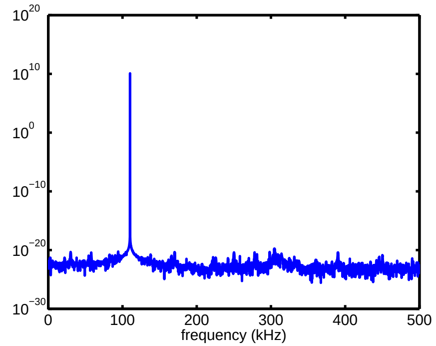
Tone/Timbre of instrument

Tone/Timbre of instrument is in part determined by proportion of different harmonics.

Cos wave



Fourier transform



Its all very confusing (to me at least), but the important point is that the harmonics sound good together. They are "consonant".

The only two harmonics omitted here are the 7th and 14th. I don't know why, but maybe its because these harmonics differ from their equal temperment pitch by the more than most (98%), though F# and D are worse.

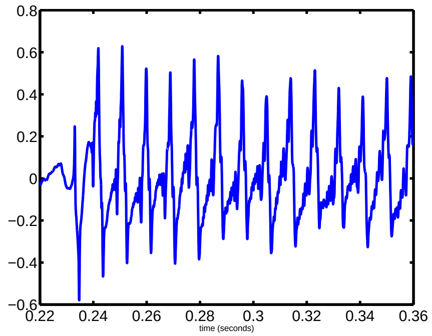
A pure cosine wave has a very simple spectral representation, but a strange artificial tone.

```
%
% file:       cos_wave.m, (c) Matthew Roughan, Sun Aug  1 2004
%
fs = 44100;
T = 5;
x = (1:fs*T)/fs;
f = 110;
y = 1*cos(2*pi*f*x);

figure(10)
plot(x,y, 'linewidth', 3);
set(gca, 'xlim', [0 0.05]);
set(gca, 'linewidth', 3, 'fontsize', 18);
xlabel('time (seconds)');
print('-depsc', 'Plots/cos_110.eps');

z = fft(y);
w = abs(z).^2;
w = fftshift(w);
q = (-length(w)/2+1:length(w)/2)/T;

figure(1)
semilogy(q, w, 'linewidth', 3);
set(gca, 'linewidth', 3, 'fontsize', 18);
set(gca, 'xlim', [0 500]);
xlabel('frequency (kHz)');
print('-depsc', 'Plots/cos_110_fft.eps');

set(gca, 'xlim', [-500 500]);
xlabel('frequency (kHz)');
print('-depsc', 'Plots/cos_110_fft_even.eps');
wavwrite(y, fs, 'Plots/cos_110.wav');
```

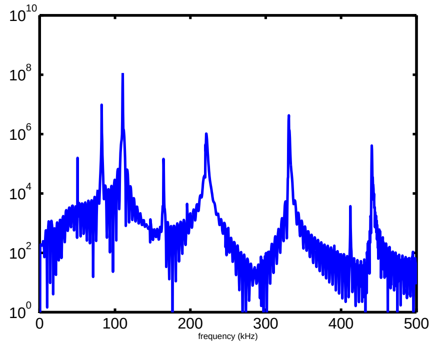
Tone/Timbre of instrument

Tone/Timbre of instrument is in part determined by proportion of different harmonics.

Note on a guitar



Fourier transform



A Guitar note has a much more complicated spectrum, including a range of harmonics. The selection of harmonics determine the tone/timbre of the instrument (along with other features such as the transient nature of the notes, i.e. the fact that different harmonics may die out at different rates).

Fourier analysis gives us a window into these phenomena.

```
%
% file:       guitar_pluck.m, (c) Matthew Roughan, Mon Jul 24 2006
%
file = 'My Song 2.wav';
[x, Fs, bits] = wavread(file);
dt = 1/Fs;

fprintf('file = %s\n', file);
fprintf(' sampled at %d bits at %d Hz\n', bits, Fs);
time = (1:length(x(:,1)))/Fs;

figure(2)
plot(time, x(:,1), 'linewidth', 3);
xlabel('time (seconds)');
set(gca, 'xlim', [0.22, 0.36]);
set(gca, 'linewidth', 3);
set(gca, 'fontsize', 18);
print('-depsc', 'Plots/guitar_A_110.eps');

%%%% choose a segment of the input data
temp = x(:,1);
temp = temp - mean(temp);

y = fft(temp);
K = length(temp);
freq = (1:K/2) / (K*dt);

figure(4)
hold off
```


Application: Pitch Estimation

- Autotuning guitar
  http://www.technologyreview.com/Infotech/19462/page1/
  - much of article is on mechanics
  - somewhere we must be estimating pitch of a string
- Simple approach is to use Fourier transform
  - refine using "harmonic comb"
    http://ccrma.stanford.edu/~jos/SimpleStrings/Plucked_Struck_String_Pitch_Estimation.html
  - look for the fundamental frequency $\hat{f}_0$ such that the sum of (log) power in the fundamental and harmonics is maximized.

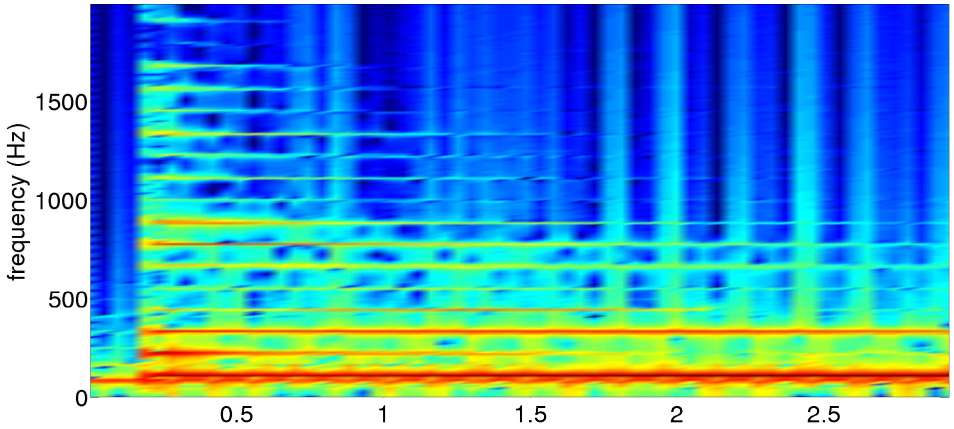
Simple version, choose f_0 that maximizes

$$\sum_{k=1}^{K} \log |X(kf_0)|$$

where K is the number of harmonics to include and X is the Fourier transform of the signal. Care must be taken to choose K, and skip any missing harmonics.

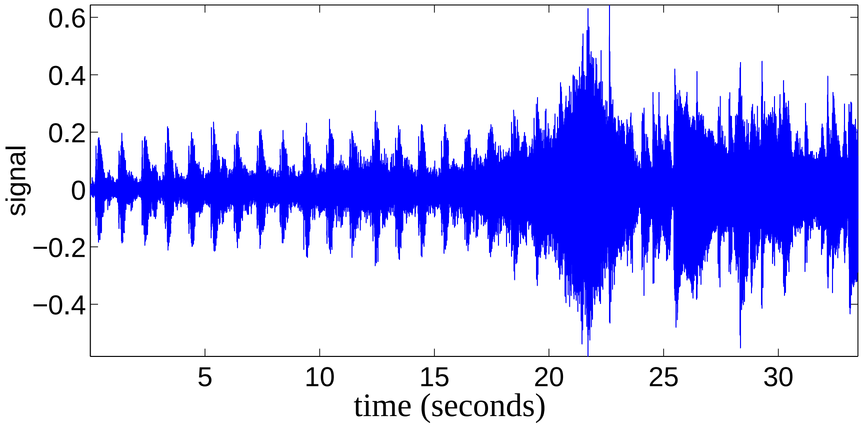
Example spectrogram

Spectrogram shows frequency content over time.
This example is the Guitar pluck we heard earlier.



Example spectrogram

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

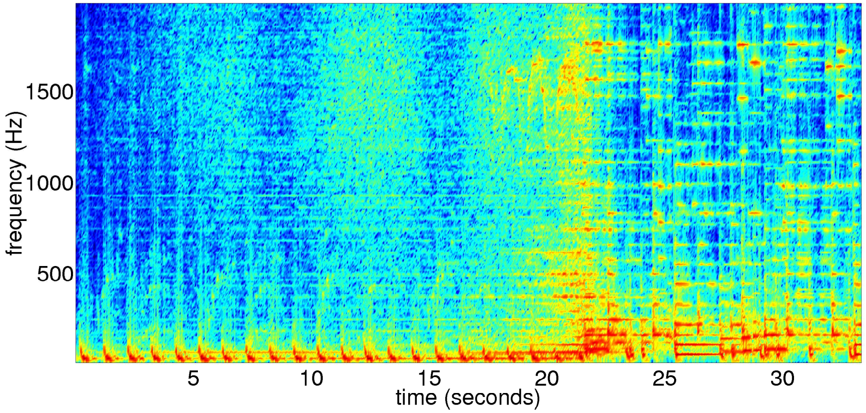


Dark side of the moon: Breathe clip

The clip of sound is from the first 30 seconds or so of "Breathe" from the album "The Dark Side of the Moon", by Pink Floyd.

It is a perfect clip, because it incorporates mid-term periodicities, and sounds ranging from mechanical to musical, and we can see the difference when we examine the spectrogram.

Example spectrogram



Dark side of the moon: Breathe clip

Application: Changing pitch

Sometimes we want to change pitch

- anonymizing an interview on TV (Vocoder)
- changing the speed of a recording to make writing a transcript easier
  - changing speed is easy
  - but when you double the speed, you double the frequency
  - need some way to correct for the change in pitch

Example:

- Clip from Bernard Fanning "Songbird",
- With the pitch increased by a factor of 2

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.30/61

Vocoders
http://www.ee.columbia.edu/ dpwe/resources/matlab/pvoc/
http://www.musicdsp.org/showone.php?id=126

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.31/61

Application: Compression

- raw audio, image and video files are large
- want to compress them
- best compression ratios allowed if we drop some data
- want to make sure that the data we drop is not perceptually important
- examples:
  - JPEG images
  - MP3 audio
- both do compression in the frequency domain

Application: Acoustic fingerprints

- How could we get a computer to identify a song?
  - last.fm can use fingerprints to ID songs, because ID3 tags in songs (put in by users) often have typos www.last.fm.
- Very large database of possible songs
  - fingerprint needs to be much smaller than song
  - even smaller than compressed song
- how do we deal with degradation
  - introduced noise
  - song might have been compressed
  - maybe not all of song is played/heard
- natural to do it in the frequency domain

An example patent: "Audio fingerprinting system and method", United States Patent 7,013,301, F. Holm, W. Hicken, March 14, 2006, http://patft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=/netahtml/PTO/search-bool.html&r=1&f=G&l=50&co1=AND&d=PTXT&s1=7,013,301.PN.&OS=PN/7,013,301&RS=PN/7,013,301

Another example: "Audio signal feature extraction", European Patent EP1403783, MORI YOSHIHIRO, OGAWA TOMOKI, MOCHINAGA KAZUHIRO, 2004-03-31, http://swpat.ffii.org/pikta/txt/ep/1403/783/

Both use frequencies present in a song to fingerprint it.

Application: Transcription

Allegri's "Miserere," was written in 1638, but by order of the Pope, it could only be sung in the Sistine Chapel during Easter week. About 140 years later a teenager heard the piece, and wrote the score from memory. There is some argument about whether he released it, or someone else did, but this is the perhaps the first example of teenagers vs the music industry.

- transcription is the process of taking audio, and converting it to written music (a score).
- it turns out to be jolly hard to get a computer to transcribe a general piece of music – we need to deal with transients.

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.34/61

Some 2D integral transforms

- Radon Transform (see also Hough transform)

$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$

- 2D Fourier transform (can go to N-dimensional)

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2\pi i(ux+vy)}\,dx\,dy$$

- Hankel transform (see also Fourier-Bessel)

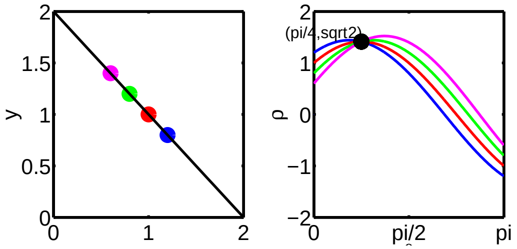
$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(r)e^{-2\pi i(ux+vy)}\,dx\,dy$$

Fourier trans. with a radially symmetric kernel

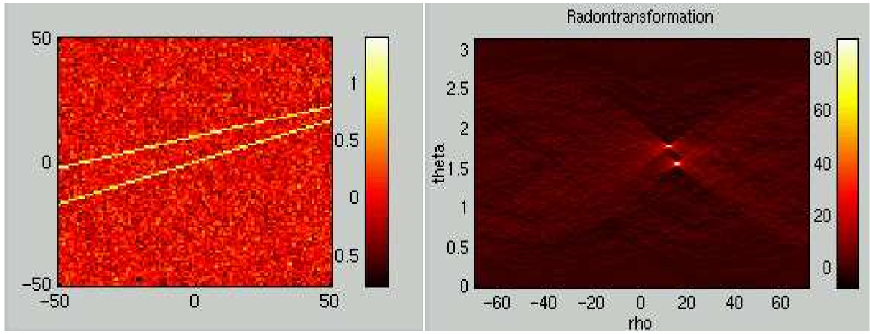
Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.35/61

An example: Radon transform

$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$



An example: Radon transform



http://eivind.imm.dtu.dk/staff/ptoft/Radon/Radon.html

Other integral transforms

- **Wavelet transform**, H-transform, Haar-transform
- **Z-transform**
- **Laplace-Stieltjes and Fourier-Stieltjes**
- bilateral-Laplace ($\int_{-\infty}^{\infty}$)
- Buschman and Mehler-Fock transforms, (power functions and Legendre polynomials)
- G- and Narain G-Transform (Meijer G-function)
- Hartley transform (cas = sin + cos)
- Hankel (Fourier-Bessel), Kontorovich-Lebedev and, Meijer transforms (Bessel functions)
- Stieltjes transform (gamma function and power)
- Abel transform (generalization of Hilbert transform)

Relationships between transforms

- Cosine transform = $\Re\{$Fourier transform$\}$
- Sin transform = $\Im\{$Fourier transform$\}$
- Laplace transform related to Fourier transform
- Fourier transform related to Fourier series (not the same)
- Wavelet transform related to Short Time Fourier transform
- Others ...

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.38/61

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.39/61

Basis functions

How do you represent a function?

- a function is a weighted sum (or integral) of basis functions

$$f(t) = \sum_k a_k g_k(t)$$

$$f(t) = \int a(s) g(t,s) ds$$

- simplest case: $a(s) = f(s)$, $g(t,s) = \delta(t-s)$
- a transformation is a change of basis

Linear algebra example

How do you represent a vector?

- a vector is a weighted sum of basis vectors

$$\mathbf{f} = \sum_k a_k \mathbf{g}_k$$

- simplest case: $a_k = f_k$, $\mathbf{g}_k = (0,\ldots,0,1,0\ldots,0)^t$
- a transformation is a change of basis

$$A\mathbf{f} = \sum_k b_k \mathbf{h}_k$$

- note that discrete-time (finite) case, is just the same

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.40/61

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.41/61

Examples of integral transforms

Name	basis functions
Identity	Delta functions $\delta(s-t)$
Fourier	Complex exponentials $e^{-ist} = \cos(st) - i\sin(st)$
Laplace	Real exponentials e^{-st}
Hilbert	Hyperbola $\frac{1}{\pi(s-t)}$
Mellin	Power functions t^{z-1}
Fourier Cosine	Cosines $\cos(st)$

Properties of basis functions

- orthogonal / bi-orthogonal / orthonormal
- redundancy, efficiency of representation
- finite/infinite support
- smoothness, regularity
- decay
- size of side lobes
- number of vanishing moments

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.42/61

Properties of basis functions can tell us something about the properties of various transforms.

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.43/61

Transform properties

- existence (when does integral converge)
- invertible (can we get back the original signal)
- complexity (how much work to compute)
- continuous vs discrete
- how does the transform behave when we change the original signal?
  - e.g. stretch the original signal
  - e.g. convolve two signals
- leakage (related to regularity and decay)

Inversion

Story of the frog prince
=> transformations can be invertible
Story of Pygmalion
=> not all transformations are invertible
How do we decide which is which?

- basis functions must not loose any information
- must be a practical way to extract the information back
- mapping must be one to one (preserves information in some way)
- orthogonal basis

The Frog Prince was turned into a frog by an evil witch. A princess restores him with a kiss.

Pygmalion is a mythological figure who fell in love with a statue he made. He prays to Venus (goddess of love), and she transforms the statue into a human (Galatea). http://en.wikipedia.org/wiki/Pygmalion_(mythology)

- In some more modern (19th century versions she rejects him)
- Many movies based on the same theme:
  - 80's: Weird Science
  - now: BuffyBot (BTVS, "I Was Made to Love You")

Inversion

Name	transform	inverse transform
Fourier	$F(s) = \int_{-\infty}^{\infty} f(t)\, e^{2\pi i s t}\, dt$	$f(t) = \int_{-\infty}^{\infty} F(s)\, e^{-2\pi i s t}\, ds$
Laplace	$F(s) = \int_{0}^{\infty} f(t)\, e^{-st}\, dt$	$f(t) = \frac{1}{2\pi i} \int_{\gamma - i\infty}^{\gamma + i\infty} F(s)\, e^{st}\, ds$
Hilbert	$F(s) = \int_{-\infty}^{\infty} \frac{f(t)}{\pi(s-t)}\, dt$	$f(t) = -\int_{-\infty}^{\infty} \frac{F(s)}{\pi(t-s)}\, ds$
Mellin	$F(z) = \int_{0}^{\infty} f(t)\, t^{z-1}\, dt$	$f(t) = \int_{c-i\infty}^{c+i\infty} F(s)\, s^{-z}\, ds$
Identity	$F(s) = \int_{\infty}^{\infty} f(t)\, \delta(s-t)\, dt$	$f(t) = \int_{\infty}^{\infty} F(s)\, \delta(s-t)\, ds$

Transform complexity

- mainly an issue for discrete transformations
- crude (numerical) integration not very efficient
- length N data, direct transformation $O(N^2)$
- Efficient algorithms exist
  - Fourier: Cooley-Tukey $O(N \log N)$
  - Wavelet: pyramidal filter bank $O(N)$

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.46/61

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.47/61

Key transform property

What do they do?

- Radon highlights **lines** in an image
- Fourier transformation highlights **frequencies**
- Short Time Fourier Transformation (spectrogram) **transient frequencies**
- Wavelet transformation highlights **transient fluctuations**

Property they highlight is related to basis functions.

What will we miss?

Too much!

- analogue devices (antenna, optical devices, analogue filters)
- Transform techniques for solving physical problems (e.g. DEs) where solution can be written in terms of basis functions, e.g. heat diffusion, vibration, ...
- other transforms: Laplace, Laplace-Stieltjes, Fourier-Stieltjes, wavelet packet, framelets, lifting schemes, ...
- too much else, ...

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.48/61

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.49/61

Basic terminology

This course relies on your knowledge of complex numbers, and basic calculus. We will briefly recap some of the assumed knowledge here, in part to ensure we are aware of the notation that will be used in this course..

Complex numbers

$$x = a + ib, \text{ where } i = \sqrt{-1}$$

- real part of x is $\Re(x) = a$
- imaginary part of x is $\Im(x) = b$
- complex conjugate $x^* = a - ib$
- Hermitian of a complex matrix $A = [a_{ij}]$ is $A^H = [a^*_{ji}]$.
- identities
  - $e^{ix} = \cos(x) + i\sin(x)$
  - $\cos(x) = \frac{1}{2}\left(e^{ix} + e^{-ix}\right)$
  - $\sin(x) = \frac{1}{2i}\left(e^{ix} - e^{-ix}\right)$

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.50/61

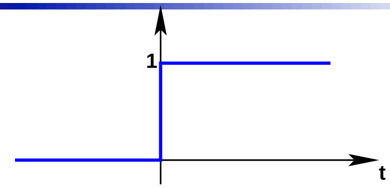
You should have gained a working knowledge of complex numbers before starting this course — if not, please see me as soon as possible.

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.51/61

Simple signals

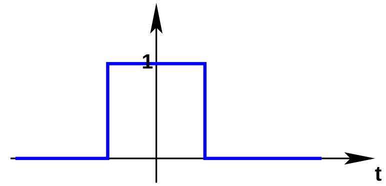
- **unit step:**

$$u(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$



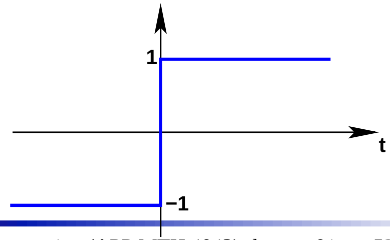
- **rectangular pulse:**

$r(t) = u(t+1/2) - u(t-1/2).$



- **sign (signum) function:**

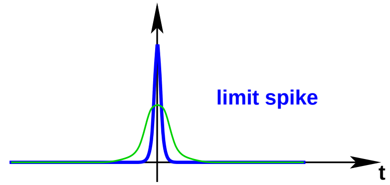
$$\text{sgn}(t) = \begin{cases} -1, & t < 0 \\ 1, & t > 0 \end{cases}$$



Delta "function" $\delta(t)$

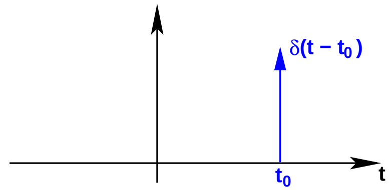
definition

$$\begin{aligned} \delta(-t) &= \delta(t) \\ \int_{-\infty}^{t} \delta(s)\, ds &= u(t) \\ \int_{-\infty}^{\infty} f(t)\delta(t-t_0)\, dt &= f(t_0) \end{aligned}$$



consequences

$$\begin{aligned} \delta(t) &= 0 \text{ for } t \neq 0 \\ \int_{-\infty}^{\infty} \delta(t)\, dt &= 1 \end{aligned}$$



The delta function(al) was introduced by Physicists well before it was accepted by mathematicians. Indeed it is often named the "Dirac delta function" after its creator, the famous physicist.

Operating with deltas is actually pretty simple, once one absorbs the definition (though the theory is a bit more complex).

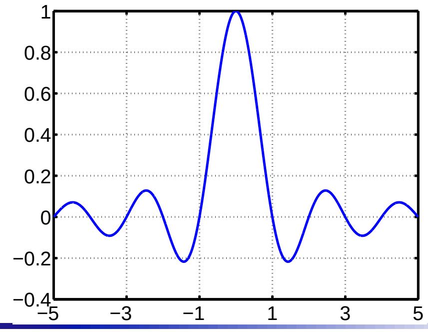
Some useful functions: sinc

The **sinc** function

$$\text{sinc}(x) = \begin{cases} 1, & \text{if } x = 0 \\ \frac{\sin \pi x}{\pi x}, & \text{otherwise,} \end{cases}$$

Properties:

- symmetric
- $\int_{-\infty}^{\infty} \text{sinc}(x)\, dx = 1$
- $\int_{-\infty}^{\infty} \text{sinc}(ax)\, dx = 1/|a|$
- $\lim_{a \to 0} \frac{1}{a} \text{sinc}\left(\frac{x}{a}\right) = \delta(x)$



The sinc function will be often used because it is the Fourier transform of a rectangular pulse.

There are multiple definitions of the sinc function: we use the normalized sinc function, but there's at least one other possibility:

- normalized sinc:

$$\text{sinc}(x) = \begin{cases} 1, & \text{if } x = 0 \\ \frac{\sin \pi x}{\pi x}, & \text{otherwise,} \end{cases}$$

- unnormalized sinc

$$\text{sinc}(x) = \begin{cases} 1, & \text{if } x = 0 \\ \frac{\sin x}{x}, & \text{otherwise,} \end{cases}$$

Note that sinc is an abbreviation of the full name "sine cardinal".

Signal characteristics

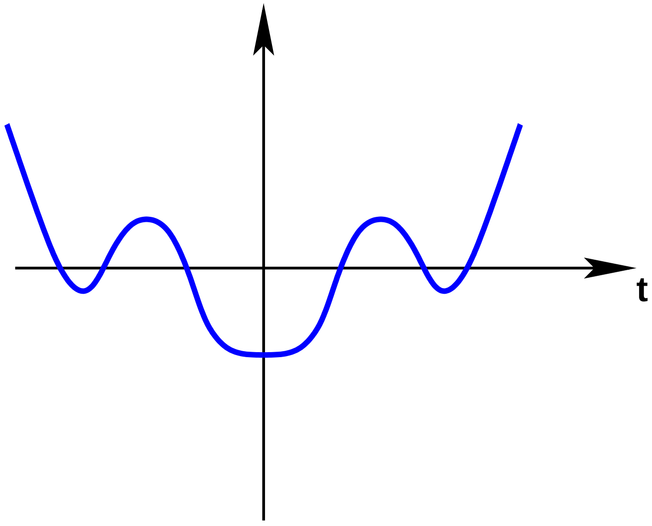
- **even:** $x(-t) = x(t)$
- **odd:** $x(-t) = -x(t)$
- any signal $x(t) = x_{\text{even}}(t) + x_{\text{odd}}(t)$ where

$$x_{\text{even}}(t) = \frac{1}{2}[x(t) + x(-t)] \text{ and } x_{\text{odd}}(t) = \frac{1}{2}[x(t) - x(-t)]$$

- **Hermitian:** $x(-t) = x^*(t)$
- **periodic:** $x(t + nT) = x(t)$ for any $n = 1, 2, \ldots$, and some $T > 0$.

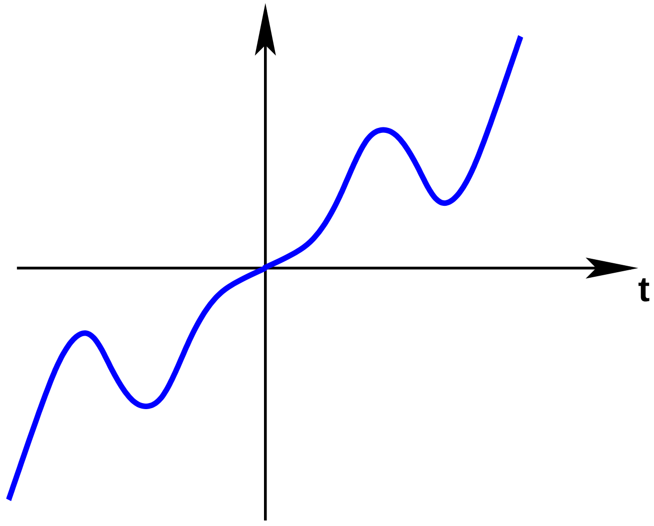
Even signals

$x(-t) = x(t)$



Odd signals

$x(-t) = -x(t)$

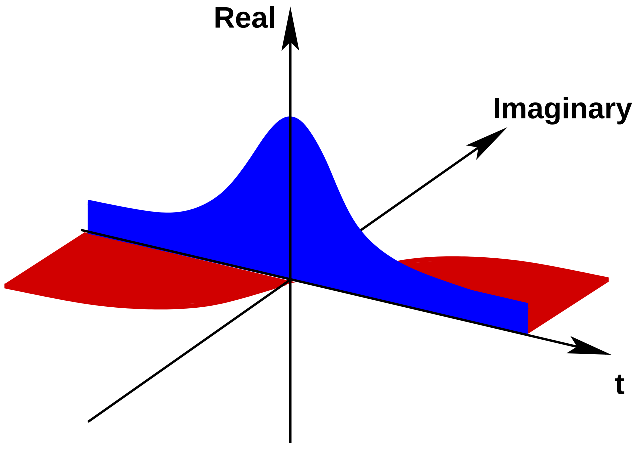


Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.56/61

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.57/61

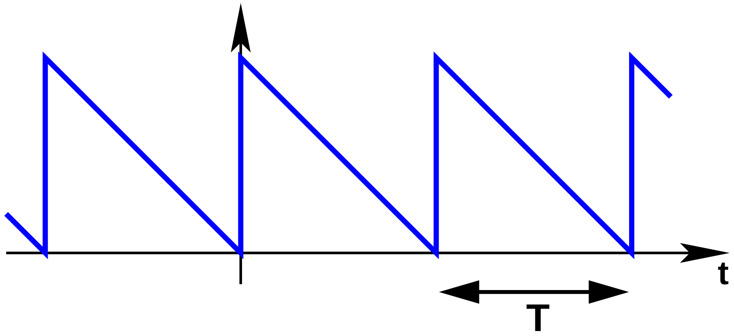
Hermitian signals

$x(-t) = x^*(t)$



Periodic signals

$x(t+nT) = x(t)$ for any $n = 1, 2, \ldots$



Real part is even
Imaginary part is odd

Transform Methods & Signal Processing (APP MTH 4043): lecture 01 – p.59/61

Frequency terminology

The minimal value $T = T_0 > 0$ for which periodic signal $x(t+nT) = x(t)$ for any $n = 1, 2, \ldots$, and some $T > 0$ is called the fundamental period, and has units of seconds.

$$\begin{aligned} T &= \text{period measured in seconds} \\ f &= 1/T = \text{frequency measured in Hz} \\ \omega &= 2\pi f \text{ measured in radians per second} \end{aligned}$$

Simple transformations

- time reversal $y(t) = x(-t)$
- time scaling $y(t) = x(at)$
- time shift $y(t) = x(t - t_0)$
- amplitude scaling $y(t) = Ax(t)$
- amplitude shift $y(t) = B + x(t)$.
- for complex signals $x(t) = a(t) + ib(t)$
  - real part $\Re(x(t)) = a(t)$
  - imaginary part $\Im(x(t)) = b(t)$
  - conjugate $x^*(t) = a(t) - ib(t)$
  - magnitude $|x(t)| = \sqrt{a(t)^2 + b(t)^2}$
  - phase angle $\theta(t) = \arctan(b(t)/a(t))$
  - $x(t) = |x(t)|e^{i\theta(t)}$

The unit “Hz” or Hertz refers to cycles per second.