

Statistical Signal Processing

Gustau Camps-Valls

Departament d'Enginyeria Electrònica
Image and Signal Processing (ISP) group
Universitat de València
gustau.camps@uv.es — <http://isp.uv.es>



UNIVERSITAT
DE VALÈNCIA

Theory:

- ① Probability and random variables
- ② Discrete time random processes
- ③ Spectral estimation
- ④ Signal decomposition and transforms
- ⑤ Introduction to information theory (bonus track)

Examples, demos and practices:

- Matlab source code, online material
- Examples and lab sessions

Chapters 1+2: Probability, random signals and variables

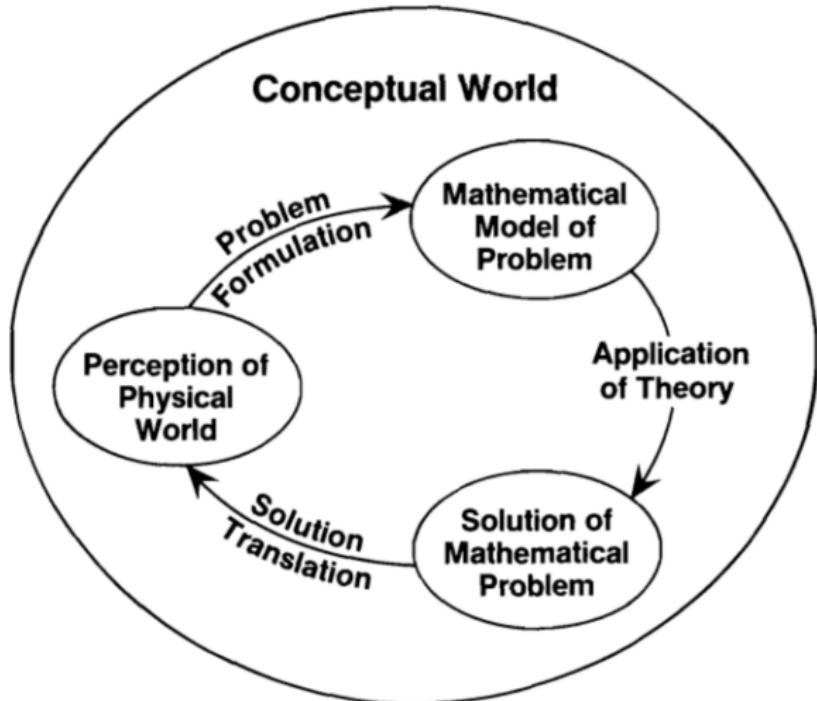
- "Introduction to random processes (with applications to signals and systems)", Gardner (1989)
- "Intuitive probability and random processes using MATLAB", Kay (2006)
- "Probability, random variables and random signal principles", Peebles (1987)
- "An introduction to statistical signal processing", Gray and Davisson (2004)
- "Probability and measure", Billingsley (1995)

Chapters 3+4: Spectral analysis and transforms

- "Spectral analysis of signals", Stoica and Moses (2005)
- Chapter 14 "Spectrum Estimation and Modeling" in Digital Signal Processing Handbook, Djuric and Kay (2005)
- Chapters 35-27 in Digital Signal Processing Handbook, Djuric and Kay (2005)
- Wikipedia, Vetterli and Gilles slides

Chapter 5 (bonus track): Introduction to information theory

- "Elements of Information Theory", Cover and Thomas (1991)
- "Information theory, inference and learning algorithms", D. MacKay (2004), <http://www.inference.phy.cam.ac.uk/mackay>



Part 1: Probability and random variables

'Probability is the chance that a given event will occur'

— Webster dictionary

'Probability is simply an average value'

'Probability theory is simply a calculus of averages'

'Probability theory is useful to design and analyze signal processing systems'

— Gardner, 1989

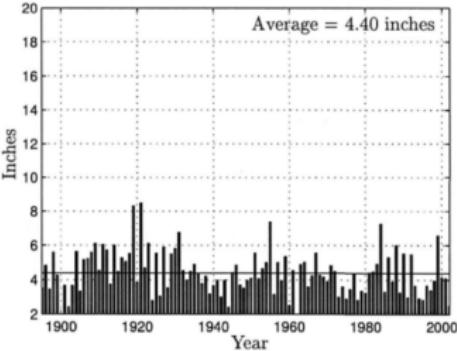
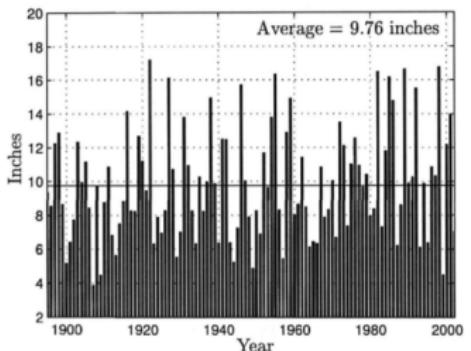
Ingredients of probability:

- We have a random experiment
- A set of outcomes
- The probabilities associated to these outcomes
- We cannot predict with certainty the outcome of the experiment
- We can predict “averages”!

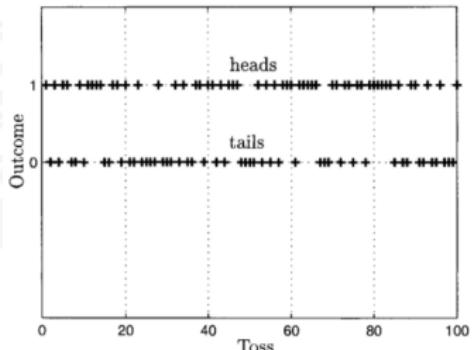
Philosophical aspects of probability:

- We want a probabilistic description of the physical problem
- We believe that there's a statistical regularity that describes the physical phenomenon

Example 1: cannot predict how much rain, but the average suggests not to plant in Arizona



Example 2: result of tossing a coin is not predictable, but the average 53% tells me it is fair coin



Types of probability: Probabilistic problems (and methods) can be discrete or continuous:

Q1 How many of the $N = 4$ people is chatting now via WhatsApp?

- Discrete answer: $0, \dots, N$
- Simple equiprobable decision: $1/(N + 1) = 1/5 = 20\%$

Q2 How long a particular guy is chatting between 15:00-15:10?

- Infinite answers: $T = [0, 10] \text{ min}$
- We need to decide if the outcome is discrete or continuous
- We need a probabilistic model!

[Q2] How long a particular guy is chatting between 15:00-15:10?

A1 Assume a simple probabilistic model with discrete answer:

- Assign a probability to each guy being on the phone in T , e.g.: $p = 0.75$
- Assume a Bernoulli distribution:

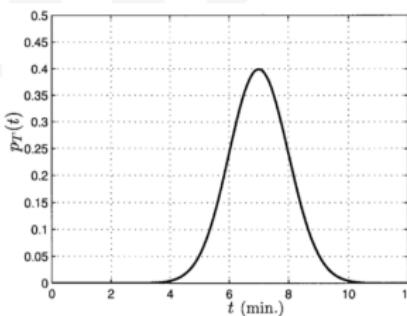
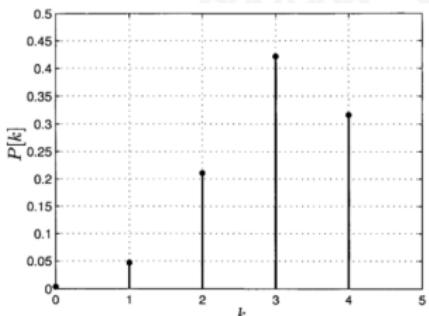
$$P[k] = \frac{N!}{k!(N-k)!} p^k (1-p)^{N-k}$$

- Too strong assumptions?: 1) each guy has a different p , 2) every p is affected by friends p ; and 3) p changes over time

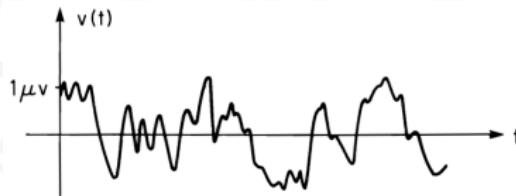
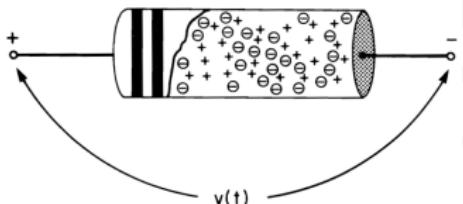
A2 Assume a more complex probabilistic model with continuous answer:

- Assume an average time of chat $\mu = 7\text{min}$
- Assume a Gaussian model for the time on the phone:

$$p_T(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(t-7)^2\right) \rightarrow P[5 \leq T \leq 6] = \int_5^6 p_T(t) dt = 0.1359$$



Notion of probability



- The thermal noise voltage: composition of +/- pulses
- What's the probability that $V(t) > 1\mu V$ at $t = t_o$? Need a probabilistic model for this physical situation!
- Event of interest A : $V(t_o) > 1\mu V$
- Event indicator: $I_A = 1$ if A occurs, $I_A = 0$ otherwise
- Imagine n resistors, and average the values of the indicator function:

$$P(A) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n I_a(i)$$

- If we have m resistors that fullfil A , then:

$$P(A) = \lim_{n \rightarrow \infty} \frac{m}{n}$$

- $P(A)$ is the *relative frequency* of occurrence of event A
- $P(A)$ is the *probability of occurrence* of event A

Sets

- A set S is a collection of entities (or elements):

$$S = \{s_1, \dots, s_n\}$$

- A set S belongs to a larger set S_o
- S is the set of all s contained in S_o that fulfill property Q_S :

$$S = \{s \in S_o : s \text{ satisfies } Q_S\}$$

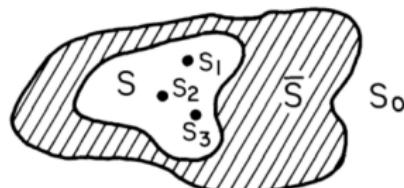
- S is a subset of S_o : $S \subset S_o$, $S \subseteq S_o$
- Complement of S is \bar{S}
- Union: $A \cup B = \{s \in S : s \in A \text{ or } s \in B\}$
- Intersection:

 - $A \cap B = \{s \in S : s \in A \text{ and } s \in B\}$

- DeMorgan's laws:

$$\overline{A \cup B} = \bar{A} \cap \bar{B}$$

$$\overline{A \cap B} = \bar{A} \cup \bar{B}$$



(a) Set containment and complement

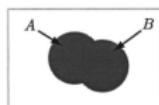
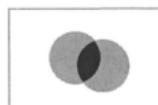


(b) Set union



(c) Set intersection

Sets operations

(a) Universal set \mathcal{S} (b) Set A (c) Set A^c (d) Set $A \cup B$ (e) Set $A \cap B$ (f) Set $A - B$

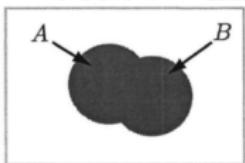
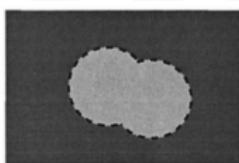
Sets relations

 \cap  $=$ 

Sets partition

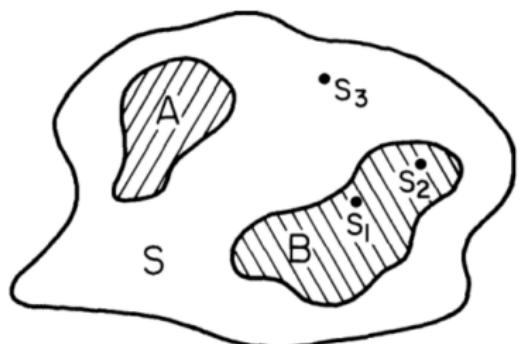
 $=$ 

DeMorgan's Law

(a) Set $A \cup B$ (b) Set $A^c \cap B^c$

Sample space

- **Experiment:** process of observing the state of the resistor at $t = t_o$
- **Sample point:** outcome of the experiment, sets of positions and velocities of all electrons/ions in the resistor
- **Sample space:** set S of all possible sample points (infinite possibilities!)
- **Event:** event $A \subset S$ that occurs (happens)



- s_1, s_2 and s_3 : sample points
- A, B : events
- S : sample space

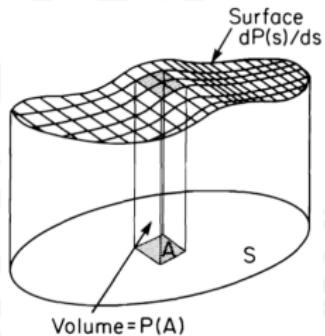
If enough net negatively charged regions reside near the $-$ terminal, and/or enough positively charged regions are near the $+$ terminal, then the event $A: V(t_o) > 1\mu V$ will occur.

Probability space

The sample space S is a probability space *iff* to every event A there is a number $P(A)$ that fulfills:

- $0 \leq P(A) \leq 1$
- $P(A \cup B) = P(A) + P(B)$, *iff* $A \cap B = \emptyset$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- $P(S) = 1$

Probability as a volume over a planar sample space



The infinitesimal probability of the event ds , centered at point s in set A is $dP(s)$ then the probability of the event A is the continuous integral (sum) of all the individual probabilities over the set:

$$P(A) = \int_{s \in A} dP(s) = \int_{s \in A} \frac{dP(s)}{ds} ds$$

This representation is not very useful because most of the problems are multidimensional, e.g. our problem is defined in a 6-dim space positions and velocities of a single electron.

Conditional probability

What if we have an extra condition on our problem? Given that $V(t_o) > 0$, what is the probability that $V(t_o) > 1\mu V$?

Conditional probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

In this example, A is a subset of B , $A \subset B$, so $A \cap B = A$:

$$P(A|B) = \frac{P(A)}{P(B)} = 2P(A), \text{ because } P(B) = 0.5$$

A conditional probability is a simple (unconditional) probability defined on a new (conditional) probability space, e.g. in our case $S_B = B$, and the new probability function is:

$$P_B(\cdot) = P(\cdot|B) = \frac{P(\cdot \cap B)}{P(B)}$$

Bayes' theorem

$$P(A|B) = P(B|A) \frac{P(A)}{P(B)}$$



Bayes was concerned about...

- *Forward problem:* Given a specified number of white and black balls in a box, what is the probability of drawing a black ball?
- *Reverse problem:* Given that one or more balls have been drawn, what can be said about the number of white and black balls in the box?

Bayes' Theorem: "Bayes' theorem relates the conditional and marginal probabilities of events A and B, where B has a non-zero probability"

$$\text{posterior} \equiv P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

Example "The department is formed by 60% men and 40% women. Men always wear trousers, women wear trousers or skirts in equal numbers".

- A: I see a girl
- B: A person is wearing trousers
- The probability of meeting a girl with trousers is:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{0.5 \times 0.4}{0.5 \times 0.4 + 1 \times 0.6} = 0.25$$

- Simple non-Bayesian probabilities would say: $0.4 \times 0.5 = 0.2$

Independent events

If the occurrence of event B has no effect on the occurrence of event A , we say that A is independent of B , $P(A|B) = P(A)$

Remember Bayes' theorem:

$$P(A|B) = P(B|A) \frac{P(A)}{P(B)}$$

then

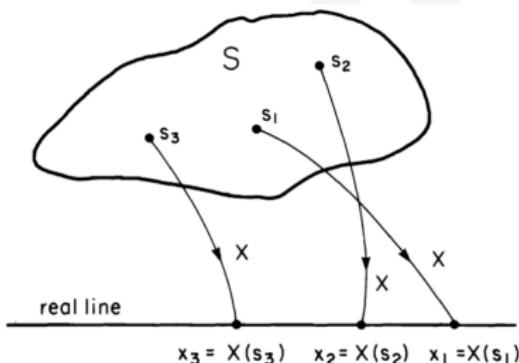
$$P(A \cap B) = P(A)P(B), \quad P(B|A) = P(B)$$

so if A is independent of B , then B is independent of A

Random variable

A random variable is a real-valued function $X(\cdot)$ of sample points in a sample space: a function that assigns a real number $x = X(s)$ to each sample point s . The real number x is called realization, or statistical sample of $X(\cdot)$

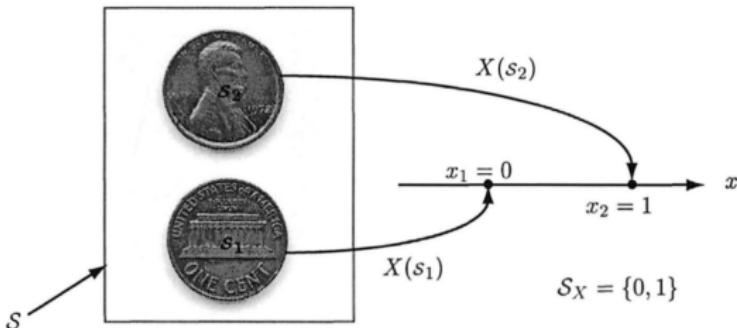
Representation of a random variable



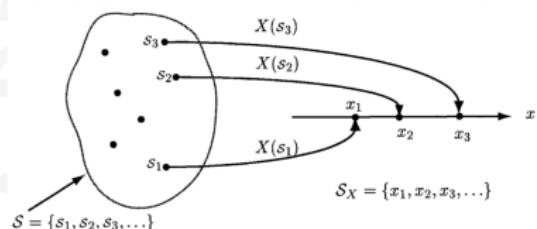
Our example:

$X = V(t_o)$ is a random variable, and after the experiment, the specific value $v(t_o)$ measured is a sample of the random variable $V(t_o)$:
 $x = v(t_o) = V(t_o, s) = X(s)$

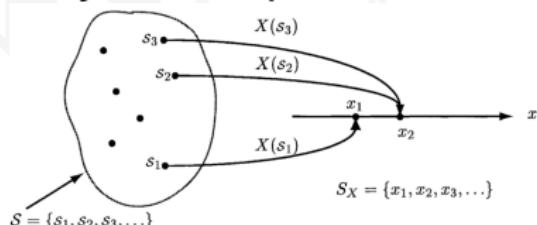
Discrete random variables



One-to-one map



Many-to-one map



Why “random” in random variable?

“Nothing in nature is random . . . A thing appears random only through the incompleteness of our knowledge.”

—Spinoza, Ethics I

“I do not believe that God rolls dice.”

—attributed to Einstein

Why “random” in random variable?

“A random or stochastic process is a mathematical model for a phenomenon that evolves in time in an unpredictable manner from the viewpoint of the observer.

The phenomenon may be a sequence of real-valued measurements of voltage or temperature, a binary data stream from a computer, a modulated binary data stream from a modem, a sequence of coin tosses, the daily Dow–Jones average, radiometer data or photographs from deep space probes, a sequence of images from a cable television, or any of an infinite number of possible sequences, waveforms, or signals of any imaginable type.

It may be unpredictable because of such effects as interference or noise in a communication link or storage medium, or it may be an information-bearing signal, deterministic from the viewpoint of an observer at the transmitter but random to an observer at the receiver.”

—Gray, 2004

Why “random” in random variable? (II)

“The theory of random processes quantifies the above notions so that one can construct mathematical models of real phenomena that are both tractable and meaningful in the sense of yielding useful predictions of future behavior.

Tractability is required in order for the engineer (or anyone else) to be able to perform analyses and syntheses of random processes, perhaps with the aid of computers. The meaningful requirement is that the models must provide a reasonably good approximation of the actual phenomena.

An oversimplified model may provide results and conclusions that do not apply to the real phenomenon being modeled. An overcomplicated one may constrain potential applications, render theory too difficult to be useful, and strain available computational resources. Perhaps the most distinguishing characteristic between an average engineer and an outstanding engineer is the ability to derive effective models providing a good balance between complexity and accuracy.”

—Gray, 2004

Why “random” in random variable? (III)

“Random processes usually occur in applications in the context of environments or systems which change the processes to produce other processes.

The intentional operation on a signal produced by one process, an input signal, to produce a new signal, an output signal, is generally referred to as signal processing, a topic easily illustrated by examples.”

—Gray, 2004

Why “random” in random variable? (IV)

- A time-varying voltage waveform is produced by a human speaking into a microphone or telephone. The signal can be modeled by a random process. This signal might be modulated for transmission, then it might be digitized and coded for transmission on a digital link. Noise in the digital link can cause errors in reconstructed bits, the bits can then be used to reconstruct the original signal within some fidelity. All of these operations on signals can be considered as signal processing, although the name is most commonly used for manmade operations such as modulation, digitization, and coding, rather than the natural possibly unavoidable changes such as the addition of thermal noise or other changes out of our control.
- For digital speech communications at very low bit rates, speech is sometimes converted into a model consisting of a simple linear filter (called an autoregressive filter) and an input process. The idea is that the parameters describing the model can be communicated with fewer bits than can the original signal, but the receiver can synthesize the human voice at the other end using the model so that it sounds very much like the original signal. A system of this type is called a *vocoder*.
- Signals including image data transmitted from remote spacecraft are virtually buried in noise added to them on route and in the front end amplifiers of the receivers used to retrieve the signals. By suitably preparing the signals prior to transmission, by suitable filtering of the received signal plus noise, and by suitable decision or estimation rules, high quality images are transmitted through this very poor channel.
- Signals produced by biomedical measuring devices can display specific behavior when a patient suddenly changes for the worse. Signal processing systems can look for these changes and warn medical personnel when suspicious behavior occurs.
- Images produced by laser cameras inside elderly North Atlantic pipelines can be automatically analyzed to locate possible anomalies indicating corrosion by looking for locally distinct random behavior.

Distribution function (DF) or Cumulative Density Function (CDF):

The probability distribution function for a random variable X is denoted by $F_X(\cdot)$, and is defined by

$$F_X(x) = \text{Prob}\{X < x\},$$

that is, $F_X(x)$ is the probability that the random variable X will take on a value less than the number x



Let $X = V(t_o)$, then:

- $A_X: V(t_o) < 1\mu V = 10^{-6}V$
- $\bar{A}_X: V(t_o) \geq 10^{-6}V$
- $\text{Prob}\{V(t_o) \geq 10^{-6}V\} = P(\bar{A}_X) = 1 - F_X(10^{-6}V)$

Therefore, we can answer the question if we determine the appropriate distribution function for the thermal noise voltage!

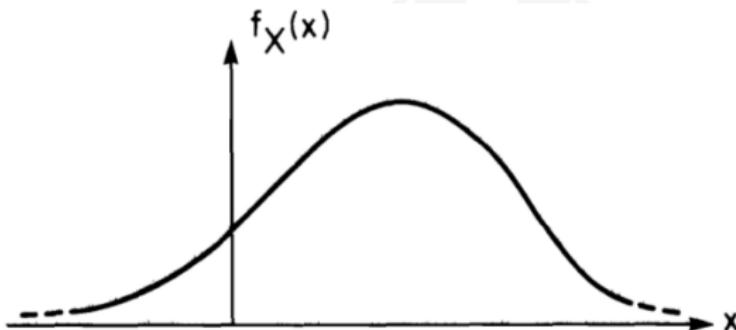
Probability Density function (PDF):

The probability that a random variable X takes a value in the interval $[x - \varepsilon, x + \varepsilon]$ is

$$\text{Prob}\{x - \varepsilon \leq X < x + \varepsilon\},$$

then the density of probability at the point x is

$$f_X(x) = \lim_{\varepsilon \rightarrow 0} \frac{1}{2\varepsilon} \text{Prob}\{x - \varepsilon \leq X < x + \varepsilon\}$$

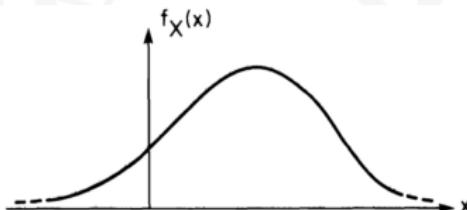
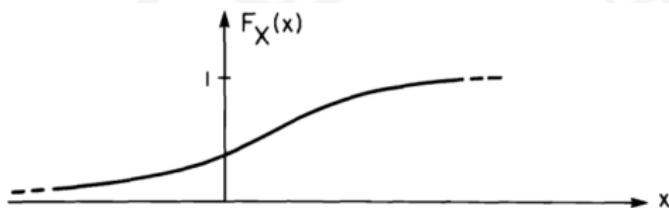


Properties:

- Non-negative function: $f_X(x) \geq 0$
- Unit area: $\int_{-\infty}^{\infty} f_X(x) dx = 1$

PDF and CDF are related:

$$f_X(x) = \frac{d}{dx} F_X(x) \quad F_X(x) = \int_{-\infty}^x f_X(y) dy$$

**Intuition:**

- Probability of the event $A_X : x \in [x_1, x_2]$ is

$$P(A_X) = F_X(x_2) - F_X(x_1) = \int_{x_1}^{x_2} \frac{dF_X(x)}{dx} dx$$

$$\text{Prob}\{x_1 \leq X < x_2\} = \int_{-\infty}^{x_2} f_X(x) dx$$

- “The probability that x is contained in some subset of real numbers A_X can be interpreted as the area under the probability density function $f_X(\cdot)$ above the subset”

The Gaussian (or normal) density function:

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right), \quad -\infty < x < \infty$$

where μ is the mean and σ is the standard deviation.

```
>> x = -10:0.1:10;
>> f = normpdf(x,1,1);
>> plot(x,f)
>> xlabel('Sample space x'); ylabel('Gaussian density f_X(x)');
>> x = randn(1000,1);
>> histfit(x);
>> x = rand(1000,1);
>> histfit(x);
```

The thermal noise voltage solution with the Gaussian model:

Assume $\mu = 0$ and $\sigma^2 = 4KTBR$ [V^2], where K is the Boltzmann's constant, T is the temperature [K], B is the bandwidth of the voltmeter [Hz], and R is the resistance [Ω]. For $T = 290K$ and a 100-MHz voltmeter, $\sigma^2 = 1.6 \cdot 10^{-10}$ [V^2]

- The probability of having $V(t_o) \geq 10^{-6}V$:

$$\text{Prob}\{V(t_o) \geq 10^{-6}V\} = 1 - \int_{-\infty}^{10^{-6}} f_X(x) dx,$$

and using the Gaussian density $f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)$, we obtain

$$\text{Prob}\{V(t_o) \geq 10^{-6}V\} = 1/2 - \text{erf}\left(10^{-6}/\sqrt{1.6 \cdot 10^{-10}}\right) \approx 0.48,$$

where $\text{erf}(\cdot)$ is the *error function*:

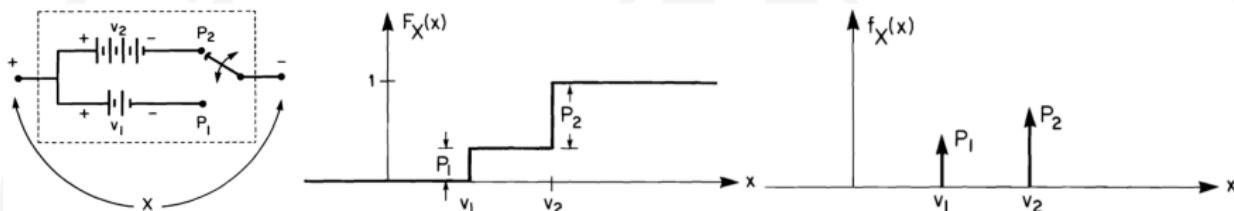
$$\text{erf}(y) = \frac{1}{\sqrt{2\pi}} \int_0^y \exp(-x^2/2) dx$$

```
>> help erf;
>> help erfinv;
```

- The conditional question: $P[V(t_o) \geq 10^{-6}V | V(t_o) > 0V] = 2 \times 0.48 = 0.96$

Probability density function of a discrete random variable:

Random variables with a Gaussian distribution are *continuous random variables*. The other interesting case are random variables that can take on only a countable number of values, the *discrete random variables* (e.g. quantized signals)

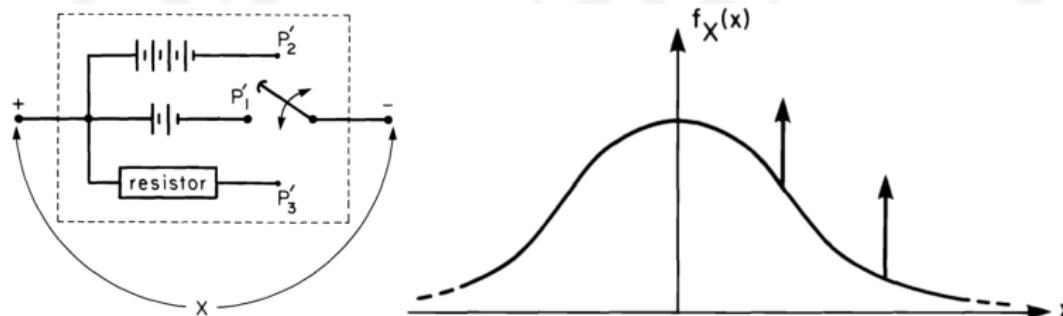


Intuition:

- The density function is just the differentiation of the (piecewise constant) distribution function
- Only two Dirac delta functions (impulses) are obtained (as expected)

Probability density function of mixed random variable:

Sometimes the systems provide a mixed (continuous+discrete) random variables.



Intuition:

-

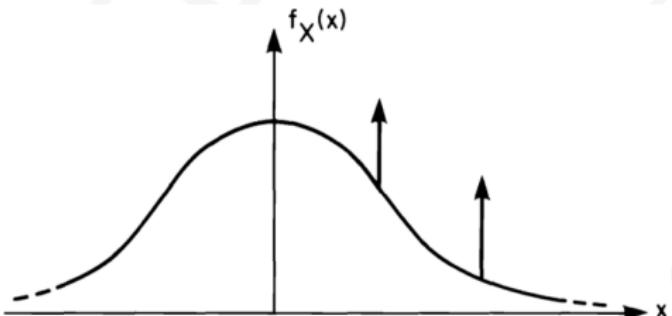
$$F_X(x) = (P'_1 + P'_2) F_Y(x) + P'_3 F_Z(x),$$

where Y denotes the random (discrete) battery and Z denotes the thermal-noise voltage (continuous)

- The density function is an additive function (yet not invertible)!

Probability mass function (PMF):

Useful to characterize the random variable instead of using the distribution function



$$P_X(x) = \begin{cases} P_1 & x = v_1 \\ P_2 & x = v_2 \\ 0 & \text{otherwise} \end{cases}$$

Joint distributions and densities:

The previous definitions extend from a single random variable to several random variables

For two RVs X and Y :

- Joint distribution function:

$$F_{XY}(x, y) = \text{Prob}\{X < x \text{ and } Y < y\}$$

- Joint density function:

$$f_{XY}(x, y) = \frac{\partial^2}{\partial x \partial y} F_{XY}(x, y)$$

- Marginal distributions obtained from the joint distribution and density:

$$F_X(x) = F_{XY}(x, \infty) \quad f_X(x) = \int_{-\infty}^{+\infty} f_{XY}(x, y) dy$$

- Conditional density:

$$f_{X|Y}(x|Y=y) = \frac{f_{XY}(x, y)}{f_Y(y)}$$

- Conditional distribution:

$$F_{X|Y}(x|Y < y) = \frac{F_{XY}(x, y)}{F_Y(y)}$$

Independent random variables:

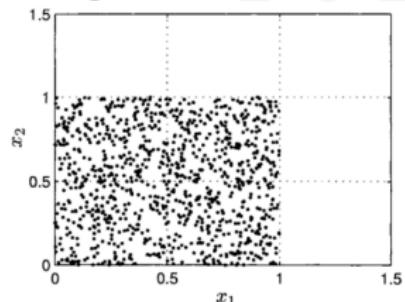
If X and Y are statistically independent then:

$$F_{XY}(x, y) = F_X(x) F_Y(y)$$

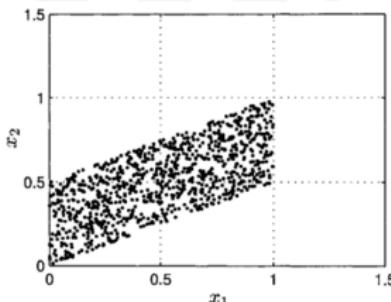
and therefore

$$f_{XY}(x, y) = f_X(x) f_Y(y)$$

Intuition: Independent variables only when you can describe X without the need of observing Y



(a) No dependency



(b) Dependency

Bivariate Gaussian density:

Two random variables X and Y are jointly Gaussian iff the random variable $Z = aX + bY$ is Gaussian for any real numbers a and b .

- If X and Y are not linearly dependent ($\nexists c, d : Y = cX + d$), then the joint Gaussian density is:

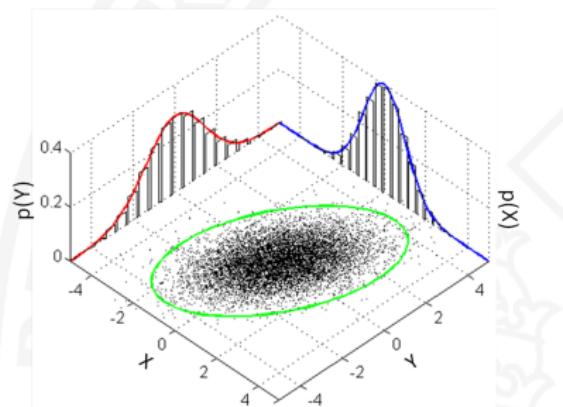
$$f_{XY}(x, y) = \frac{1}{2\pi\sigma\sigma'\sqrt{1-\rho^2}}$$

$$\times \exp\left(\frac{(x - \mu)^2/\sigma^2 - 2\rho(x - \mu)\sigma(y - \mu')\sigma' + (y - \mu')^2/\sigma'^2}{2(1 - \rho^2)}\right)$$

- If X and Y are linearly dependent, then $\rho = 1$, and this does not apply

Multivariate Gaussian density:

The multivariate normal distribution or multivariate Gaussian distribution, is a generalization of the one-dimensional (univariate) normal distribution to higher dimensions



The multivariate normal distribution of a k -dimensional random variable $X = [X_1, X_2, \dots, X_k]$ is written:

$$X \sim \mathcal{N}(\mu, \Sigma),$$

with k -dimensional mean vector

$$\mu = [\mu_1, \mu_2, \dots, \mu_k]$$

and $k \times k$ covariance matrix

$$\Sigma = [\sigma_{ij}], i, j = 1, 2, \dots, k$$

Multidimensional Probability density function (PDF)

- ① Joint PDF of a vector \mathbf{a} :

$$P_{\mathbf{a}}(\mathbf{a}) = P_{\mathbf{a}}(a_1, a_2, \dots, a_d), \quad \int P_{\mathbf{a}}(\mathbf{a}) d\mathbf{a} = 1$$

- ② Marginal PDF (of i th component of \mathbf{a}):

$$P_{a_i}(a_i) = \int P_{\mathbf{a}}(\mathbf{a}) da_1 da_{i-1} da_{i+1} da_d$$

is the integral of the joint PDF in all directions except i

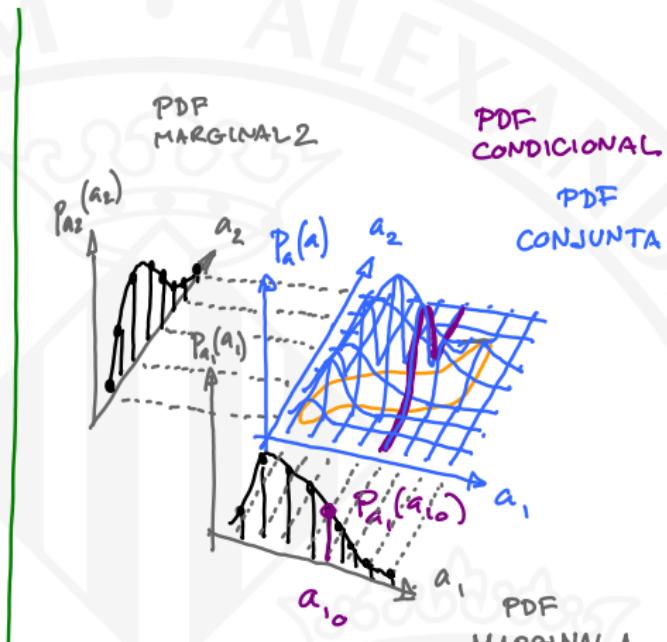
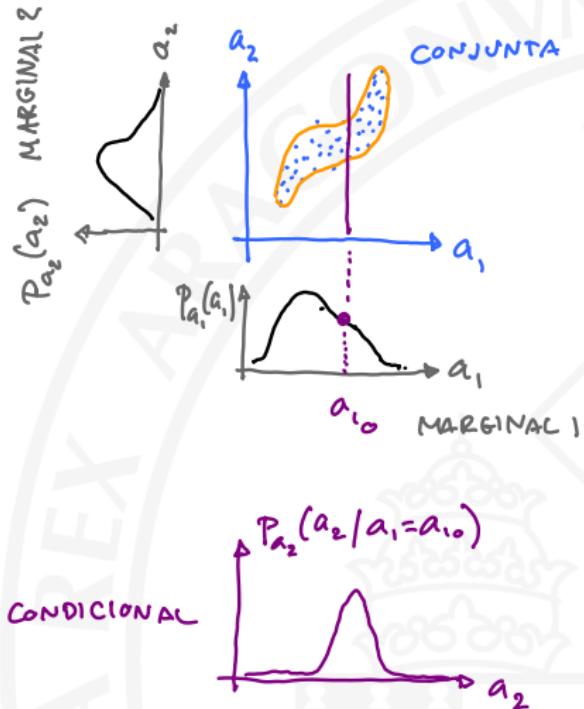
- ③ Conditional PDF (of a component i fixing the rest):

$$P_{a_i}(a_i | a_j) = \frac{P(a_i, a_j)}{P_{a_j}(a_j)} \quad \forall j \neq i$$

- ④ Bayes' rule says:

$$P(a_i | a_j) = \frac{p(a_i, a_j)}{p(a_j)}, \quad P(a_j | a_i) = \frac{p(a_i, a_j)}{p(a_i)}$$

$$P(a_i | a_j) = \frac{P(a_j | a_i)P(a_i)}{P(a_j)}$$



Probabilities and ensembles

An **ensemble** is a triple $(x, \mathcal{A}_x, \mathcal{P}_x)$:

- the **outcome** x is the value of a random variable,
- x takes values from set $\mathcal{A}_x = \{a_1, a_2, \dots, a_L\}$,
- with probabilities $\mathcal{P}_x = \{p_1, p_2, \dots, p_L\}$.
- $P(x = a_i) = p_i, \quad p_i \geq 0$
- $\sum_{a_i \in \mathcal{A}_x} P(x = a_i) = \sum_{i=1}^L p_i = 1.$

Simpler notation:

$$P(x = a_i) = P(x) = P(a_i)$$

i	a_i	p_i	
1	a	0.0575	a
2	b	0.0128	b
3	c	0.0263	c
4	d	0.0285	d
5	e	0.0913	e
6	f	0.0173	f
7	g	0.0133	g
8	h	0.0313	h
9	i	0.0599	i
10	j	0.0006	j
11	k	0.0084	k
12	l	0.0335	l
13	m	0.0235	m
14	n	0.0596	n
15	o	0.0689	o
16	p	0.0192	p
17	q	0.0008	q
18	r	0.0508	r
19	s	0.0567	s
20	t	0.0706	t
21	u	0.0334	u
22	v	0.0069	v
23	w	0.0119	w
24	x	0.0073	x
25	y	0.0164	y
26	z	0.0007	z
27	-	0.1928	-

(from David MacKay)

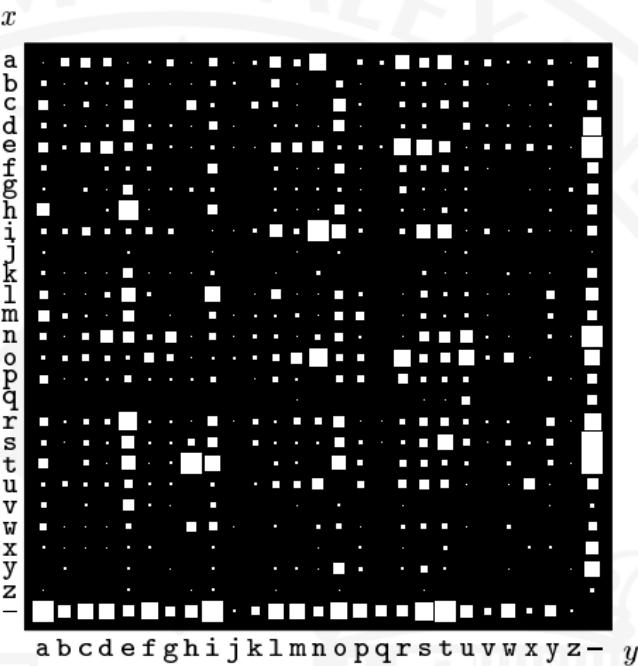
Example of joint probability

- Bigrams: probability of letter x followed by letter y
- Marginal probability from joint:

$$P(x = a_i) = \sum_{y \in \mathcal{A}_Y} P(x = a_i, y).$$

- Similarly

$$P(y) = \sum_{x \in \mathcal{A}_X} P(x, y).$$



(figure from David MacKay)

Functions of random variables:

Imagine that we have a function $g(\cdot)$ that transforms random variable X into Y , $Y = g(X)$.

Can we determine the probability of Y from the probability of X ? For some cases, yes!

If X is continuous and the inverse of $g(\cdot)$, denoted by $g^{-1}(\cdot)$ exists and is differentiable, the probability density for Y is:

$$f_Y(y) = f_X[g^{-1}(y)] \left| \frac{dg^{-1}(y)}{dy} \right| = \frac{f_X(x)}{|dg(x)/dx|}, \quad x = g^{-1}(y)$$

- This is very powerful! Avoid computing density $f_Y(y)$ directly (which is hard) and just derive the transformation
- Watch out with non-continuous functions (holes in the space) and bivalued (ambiguous) functions!

Density estimation under arbitrary transform

Now we are given $\mathbf{Y} = [Y_1, Y_2, \dots, Y_k]^\top$ obtained from $\mathbf{X} = [X_1, X_2, \dots, X_k]^\top$ using a deterministic function $\mathbf{g}(\cdot) : \mathbf{Y} = \mathbf{g}(\mathbf{X})$, and \mathbf{g}^{-1} exists and is differentiable, then the joint probability density of \mathbf{Y} is:

$$f_{\mathbf{Y}}(\mathbf{y}) = f_{\mathbf{X}}[\mathbf{g}^{-1}(\mathbf{y})] \left| \frac{\partial \mathbf{g}^{-1}(\mathbf{y})}{\partial \mathbf{y}} \right|,$$

where $|\partial \mathbf{g}^{-1}(\mathbf{y}) / \partial \mathbf{y}|$ is the absolute value of the determinant of the matrix of first-order partial derivatives $\partial g_i^{-1}(\mathbf{y}) / \partial y_j$ which is called the Jacobian of $\mathbf{g}^{-1}(\cdot)$.

Density estimation under arbitrary transform, intuition

Let $\mathbf{a} \in \mathbb{R}^k$ be a RV with PDF, $p_{\mathbf{a}}(\mathbf{a})$. Given some bijective, differentiable transform of \mathbf{a} into \mathbf{y} using $F : \mathbb{R}^k \rightarrow \mathbb{R}^k$, $\mathbf{y} = F(\mathbf{a})$, the PDFs are related:

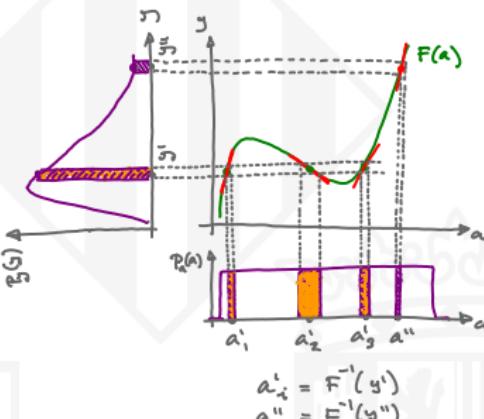
$$p_{\mathbf{a}}(\mathbf{a}) = p_{\mathbf{y}}(F(\mathbf{a})) \left| \frac{dF(\mathbf{a})}{d\mathbf{a}} \right|^{-1} = p_{\mathbf{y}}(F(\mathbf{a})) |\nabla_{\mathbf{a}} F(\mathbf{a})|^{-1}$$

where $|\nabla_{\mathbf{a}} F|$ is the determinant of the transform's Jacobian matrix.

Intuición (en caso 1D):

La población en un punto, y' , es la suma de las poblaciones en los puntos $a' = F^{-1}(y')$ pesada por la inversa de la pendiente en esos puntos $|\nabla F(a')|^{-1}$

Si: $|\nabla F(\mathbf{a})| \gg \Rightarrow p_{\mathbf{y}}(F(\mathbf{a})) \ll$



Example 1: A linear transformation

$$\mathbf{Y} = \mathbf{AX} + \mathbf{b}$$

Preliminaries:

$$\mathbf{X} = \mathbf{g}^{-1}(\mathbf{y}) = \mathbf{A}^{-1}[\mathbf{y} - \mathbf{b}]$$

$$\frac{\partial \mathbf{g}^{-1}(\mathbf{y})}{\partial \mathbf{y}} = \mathbf{A}^{-1}$$

$$|\mathbf{A}^{-1}| = 1/|\mathbf{A}|$$

Therefore:

$$f_Y(\mathbf{y}) = f_X[\mathbf{g}^{-1}(\mathbf{y})] \left| \frac{\partial \mathbf{g}^{-1}(\mathbf{y})}{\partial \mathbf{y}} \right| = \frac{f_X(\mathbf{A}^{-1}[\mathbf{y} - \mathbf{b}])}{|\mathbf{A}|} = \frac{f_X(\mathbf{X})}{|\mathbf{A}|},$$

where \mathbf{A}^{-1} is the inverse of \mathbf{A} , and $|\mathbf{A}|$ is the absolute value of the determinant of \mathbf{A}

Example 2: Sum of random variables

Let us define the sum of random variables X_1 and X_2 as $Y = Y_1 = X_1 + X_2$ and the second component $Y_2 = X_2$. Compute $f_Y(y_1, y_2)$.

Note that equivalently:

$$\mathbf{Y} = \mathbf{AX}, \quad \mathbf{Y} = [Y_1 \ Y_2]^\top, \quad \mathbf{X} = [X_1 \ X_2]^\top, \quad \mathbf{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

Therefore:

$$f_Y(\mathbf{y}) = \frac{f_{\mathbf{X}}(\mathbf{A}^{-1}\mathbf{y})}{|\mathbf{A}|} = f_{\mathbf{X}}\left(\begin{bmatrix} y_1 - y_2 \\ y_2 \end{bmatrix}\right) = f_{X_1 X_2}(y_1 - y_2, y_2) = f_{X_1 X_2}(y - x_2, x_2)$$

and

$$f_Y(y_1, y_2) = \int_{-\infty}^{\infty} f_{\mathbf{Y}}(\mathbf{y}) dy_2 = \int_{-\infty}^{\infty} f_{X_1 X_2}(y - x_2, x_2) dx_2$$

Example 3: Data rotation

Let X_1 and X_2 be independent Gaussian random variables with zero means and unity variances. Let us define the transform:

$$Y_1 = (X_1^2 + X_2^2)^{1/2}, \quad Y_2 = \tan^{-1}(X_2/X_1)$$

Compute the joint density $f_Y(y_1, y_2)$, and the marginals $f_{Y_1}(y_1)$ and $f_{Y_2}(y_2)$.

Note that: $\mathbf{Y} = \mathbf{AX}$, $\mathbf{Y} = [Y_1 \ Y_2]^\top$, $\mathbf{X} = [X_1 \ X_2]^\top$ and

$$\mathbf{Y} = \mathbf{g}(\mathbf{X}) = \begin{pmatrix} (X_1^2 + X_2^2)^{1/2} \\ \tan^{-1}(X_2/X_1) \end{pmatrix}, \quad \mathbf{X} = \mathbf{g}^{-1}(\mathbf{Y}) = \begin{pmatrix} Y_1 & \cos Y_2 \\ Y_1 & \sin Y_2 \end{pmatrix}$$

Joint density:

$$f_Y(\mathbf{y}) = f_X(\mathbf{g}^{-1}(\mathbf{y})) \left| \frac{\partial \mathbf{g}^{-1}(\mathbf{y})}{\partial \mathbf{y}} \right| = \dots = \frac{y_1}{2\pi} \exp \left(-\frac{y_1^2}{2} \right), \quad y_1 \geq 0, \quad 0 \leq y_2 < 2\pi$$

Marginal densities:

$$f_{Y_1}(y_1) = \int_{-\infty}^{\infty} f_Y(\mathbf{y}) dy_2 = \int_0^{2\pi} \frac{y_1}{2\pi} \exp \left(-\frac{y_1^2}{2} \right) dy_2 = y_1 \exp \left(-\frac{y_1^2}{2} \right), \quad y_1 \geq 0$$

$$f_{Y_2}(y_2) = \int_{-\infty}^{\infty} f_Y(\mathbf{y}) dy_1 = \int_0^{2\pi} \frac{y_1}{2\pi} \exp \left(-\frac{y_1^2}{2} \right) dy_1 = \frac{1}{2\pi}, \quad 0 \leq y_2 < 2\pi$$

Notion of expectation:

What do you expect the absolute value of the thermal-noise voltage to be?
 We need a mathematical definition of *expected value*!

Given a sample space S with only two points:

$$X(s) = \begin{cases} x_1 & s = s_1 \\ x_2 & s = s_2 \end{cases}$$

and perform n identical experiments that yield $\{s(1), s(2), \dots, s(n)\}$

Average value of the random variable $X(s)$ with $n \rightarrow \infty$?

$$\mathbb{E}\{X\} = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n X(s(i))$$

If m_1 and m_2 are the total number of times that equal s_1 and s_2 :

$$\mathbb{E}\{X\} = \lim_{n \rightarrow \infty} \left[\frac{m_1}{n} x_1 + \frac{m_2}{n} x_2 \right], \quad \text{and} \quad \lim_{n \rightarrow \infty} \frac{m_1}{n} = P(s_1), \quad \lim_{n \rightarrow \infty} \frac{m_2}{n} = P(s_2)$$

- Average of $X(s)$ is the *probability-weighted sum of all possible values*

$$\mathbb{E}\{X\} = X(s_1)P(s_1) + X(s_2)P(s_2)$$

- $\mathbb{E}\{X\}$ is the *expected value* of X

Expected value:

The expected value of a random variable X is denoted by $\mathbb{E}\{X\}$, and is a real (nonrandom) number defined by:

$$\mathbb{E}\{X\} = \sum_{s \in S} X(s)P(s)$$

Note:

- The expected value is a *probability-weighted average over the entire sample space of the sample values*
- For continuous random variables, replace \sum with \int :

$$\mathbb{E}\{X\} = \int_{s \in S} X(s)dP(s)$$

Example: $S = \{s_1, s_2, s_3, s_4, s_5\}$ and the probability function

$$P(s_1) = 1/8, \quad P(s_2) = 1/8, \quad P(s_3) = 1/8, \quad P(s_4) = 3/8, \quad P(s_5) = 1/8$$

and the random variable

$$X(s_1) = -1, \quad X(s_2) = +1, \quad X(s_3) = +1, \quad X(s_4) = +2, \quad X(s_5) = -1$$

Then: $\mathbb{E}\{X\} = \sum_{s \in S} X(s)P(s) = \dots = 7/8$

Properties of expectation:

- ① *Linearity*: for any two RVs X and Y , two real numbers a and b , and the RV $Z = aX + bY$:

$$\mathbb{E}\{Z\} = a\mathbb{E}\{X\} + b\mathbb{E}\{Y\}$$

- ② *Expected value of a function of a random variable*: for any function $g(\cdot)$ of a RV X , and $Y = g(x)$, we can show that

$$\mathbb{E}\{g(X)\} = \int_{-\infty}^{\infty} g(x)f_X(x)dx$$

Important note: this is the *fundamental theorem of expectation*, which is far much easier than estimating the PDF of Y and then using the definition:

$$\mathbb{E}\{g(X)\} = \mathbb{E}\{Y\} = \int_{-\infty}^{\infty} yf_Y(y)dy$$

Expected value of the thermal-noise voltage?

- Let define $X = V(t_o)$ and $g(\cdot) = |\cdot|$ the absolute value function. Then the expected value is:

$$\mathbb{E}\{g(X)\} = \int_{-\infty}^{\infty} g(x)f_X(x)dx$$

$$\mathbb{E}\{V(t_o)\} = \int_{-\infty}^{\infty} |x|f_X(x)dx$$

- For the Gaussian density:

$$\mathbb{E}\{V(t_o)\} = \alpha \sqrt{\frac{2}{\pi}}, \quad \alpha = 2\sqrt{KTBR}$$

For $T = 290K$, $R = 100\Omega$, $B = 100$ MHz, $\alpha \approx 1.3 \cdot 10^{-5}$ [V^2], and thus

$$\mathbb{E}\{V(t_o)\} \approx 10\mu V$$

Characteristic function

Example of a function of a random variable:

$$g(\cdot) = e^{i\omega(\cdot)}, \quad \text{with parameter } \omega$$

- The characteristic function is defined as:

$$\Phi_X(\omega) = \mathbb{E}\{e^{i\omega X}\}, \quad i = \sqrt{-1}$$

- Equivalent to:

$$\Phi_X(\omega) = \int_{-\infty}^{\infty} e^{i\omega x} f_X(x) dx,$$

which is the conjugate (sign reversed) Fourier transform of $f_X(\cdot)$

- A useful property of the characteristic function is that it yields the moments of the random variable: the n -th moment of X can be obtained by differentiation of Φ_X :

$$\mathbb{E}\{X^n\} = \left(\frac{1}{i^n} \right) \frac{d^n \Phi_X(\omega)}{d\omega^n} \Big|_{\omega=0}.$$

Characteristic functions:

- In probability theory and statistics, the characteristic function of any real-valued random variable completely defines its probability distribution
- If a random variable admits a probability density function, then the characteristic function is the inverse Fourier transform of the probability density function
- It provides the basis of an alternative route to analytical results compared with working directly with probability density functions or cumulative distribution functions

Distribution	Characteristic function $\varphi(t)$
Degenerate δ_a	e^{ita}
Bernoulli $Bern(p)$	$1 - p + pe^{it}$
Binomial $B(n, p)$	$(1 - p + pe^{it})^n$
Negative binomial $NB(r, p)$	$\left(\frac{1-p}{1-pe^{it}}\right)^r$
Poisson $Pois(\lambda)$	$e^{\lambda(e^{it}-1)}$
Uniform $U(a, b)$	$\frac{e^{itb} - e^{ita}}{it(b-a)}$
Laplace $L(\mu, b)$	$\frac{e^{it\mu}}{1+b^2t^2}$
Normal $N(\mu, \sigma^2)$	$e^{it\mu - \frac{1}{2}\sigma^2 t^2}$
Chi-squared χ^2	$(1 - 2it)^{-k/2}$
Cauchy $C(\mu, \theta)$	$e^{it\mu - \theta t }$
Gamma $\Gamma(k, \theta)$	$(1 - it\theta)^{-k}$
Exponential $Exp(\lambda)$	$(1 - it\lambda^{-1})^{-1}$
Multivariate normal $N(\mu, \Sigma)$	$e^{it^T \mu - \frac{1}{2} t^T \Sigma t}$
Multivariate Cauchy MultiCauchy(μ, Σ)	$e^{it^T \mu - \sqrt{t^T \Sigma t}}$

Sums of independent random variables: exploit the characteristic function

Let W be a random variable equal to the sum of two statistically independent random variables X and Y :

$$W = X + Y$$

- The characteristic function of W is defined as:

$$\Phi_W(\omega) = \mathbb{E}\{e^{i\omega W}\} = \mathbb{E}\{e^{i\omega(X+Y)}\} = \mathbb{E}\{e^{i\omega X}\}\mathbb{E}\{e^{i\omega Y}\} = \Phi_X(\omega)\Phi_Y(\omega)$$

which is the product of characteristic functions

- From the convolution property of the Fourier transform

$$f_w(\alpha) = f_x(\alpha) * f_y(\alpha) = \int_{-\infty}^{\infty} f(u)f(u - \alpha)du$$

The PDF of a sum of two statistically independent random variables is the convolution of the individual PDFs

First moments of a probability density function f_X :

Since the f_X is a non-negative function with unit area, the expected value

$$\mathbb{E}\{X\} = \int_{-\infty}^{\infty} xf_X(x)dx$$

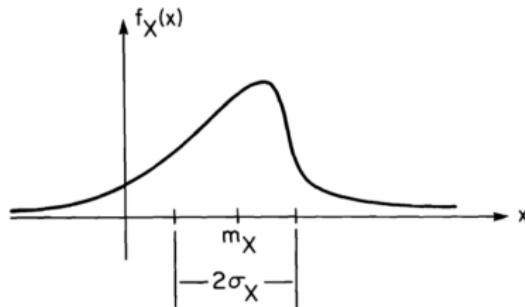
can be interpreted as the *first moment of the function $f_X(\cdot)$, which is a measure of the center of a function.*

- The center of the PDF $f_X(\cdot)$ is the *mean*:

$$m_X = \mathbb{E}\{X\} \quad >> \text{mean}(X)$$

- The square root of the second centralized moment measures the width of the function and is the *standard deviation*:

$$\sigma_X = \sqrt{\mathbb{E}\{(X - m_X)^2\}} \quad >> \text{std}(X)$$



Higher moments of a probability density function f_X :

- The center of the PDF $f_X(\cdot)$ is the *mean*:

$$m_X = \mathbb{E}\{X\} \quad >> \text{mean}(X)$$

→ “average value” of the distribution

- The square root of the second centralized moment measures the width of the function and is the *standard deviation*:

$$\sigma_X = \sqrt{\mathbb{E}\{(X - m_X)^2\}} \quad >> \text{std}(X)$$

→ “average dispersion” of the distribution

- The squared standard deviation is the *variance*:

$$\sigma_X^2 = \mathbb{E}\{(X - m_X)^2\} = \mathbb{E}\{X^2\} - m_X^2 \quad >> \text{var}(X)$$

→ “average dispersion” of the distribution

- The normalized 3rd central moment is the *skewness*:

$$\mathbb{E}\{(X - m_X)^3\}/\sigma_X^3 \quad >> \text{skewness}(X)$$

→ “asymmetry of the data around the sample mean” of the distribution

- The normalized 4th central moment is the *kurtosis*:

$$\mathbb{E}\{(X - m_X)^4\}/\sigma_X^4 \quad >> \text{kurtosis}(X)$$

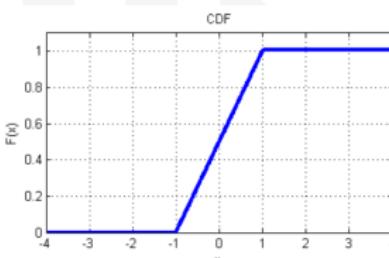
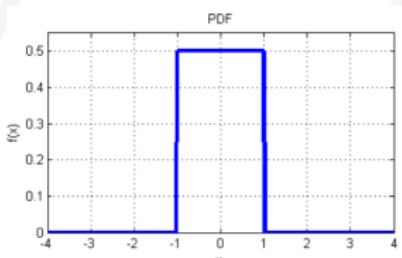
→ “outlier-prone” of the distribution

Uniform distribution:

- Equal probability of all values within bounds
- Matlab: `>> rand, pdf, cdf`
- Probability density function (PDF)

$$f_X(x) = \begin{cases} 0 & x < a \\ \frac{1}{b-a} & a \leq x \leq b \\ 0 & x > b \end{cases}$$

- Example: $a = -1, b = 1$



```

>> x = -4:0.1:4;
>> p = pdf('Uniform',x,-1,1); plot(x,p,'b')
>> c = cdf('Uniform',x,-1,1); plot(x,c,'b')
```

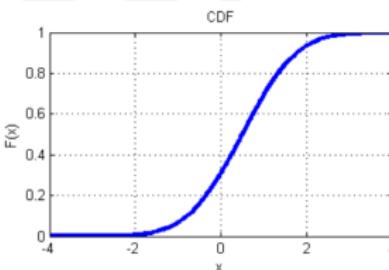
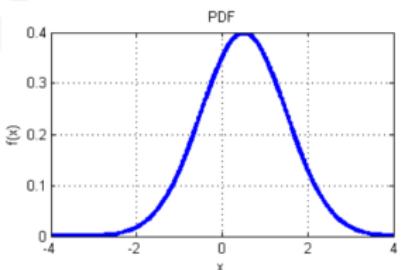
Exercise: Generate a random vector with 1000 samples from a uniform distribution and compute the first and higher moments. Discuss.

Gaussian (normal) distribution:

- Matlab: `>> randn, pdf, cdf`
- Probability density function (PDF)

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}} \exp\left(-\frac{1}{2} \frac{(x - \mu_X)^2}{\sigma_X^2}\right), \quad -\infty < x < \infty$$

- Denoted as: $x \sim \mathcal{N}(\mu_X, \sigma_X^2)$
- Example: $\mu_X = 0.5, \sigma_X^2 = 1$



```
>> x = -4:0.1:4;
>> p = pdf('Normal',x,0.5,1); plot(x,p,'b')
>> c = cdf('Normal',x,0.5,1); plot(x,c,'b')
```

Exercise: Generate a random vector with 1000 samples drawn from a normal distribution and compute the first and higher moments. Discuss.

Other important PDFs:

	Values	PDF	$E[X]$	$\text{var}(X)$	$\phi_X(\omega)$
Uniform	$a < x < b$	$\frac{1}{b-a}$	$\frac{1}{2}(a+b)$	$\frac{(b-a)^2}{12}$	$\frac{\exp(j\omega b) - \exp(j\omega a)}{j\omega(b-a)}$
Exponential	$x \geq 0$	$\lambda \exp(-\lambda x)$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	$\frac{\lambda}{\lambda - j\omega}$
Gaussian	$-\infty < x < \infty$	$\frac{\exp[-(1/(2\sigma^2))(x-\mu)^2]}{\sqrt{2\pi}\sigma^2}$	μ	σ^2	$\exp[j\omega\mu - \sigma^2\omega^2/2]$
Laplacian	$-\infty < x < \infty$	$\frac{1}{\sqrt{2\sigma^2}} \exp(-\sqrt{2/\sigma^2} x)$	0	σ^2	$\frac{2/\sigma^2}{\omega^2 + 2/\sigma^2}$
Gamma	$x \geq 0$	$\frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\lambda x)$	$\frac{\alpha}{\lambda}$	$\frac{\alpha}{\lambda^2}$	$\frac{1}{(1-j\omega/\lambda)^\alpha}$ [Johnson et al 1994]
Rayleigh	$x \geq 0$	$\frac{x}{\sigma^2} \exp[-x^2/(2\sigma^2)]$	$\sqrt{\frac{\pi\sigma^2}{2}}$	$(2-\pi/2)\sigma^2$	

Exercise: Play around in MATLAB with: pdf, cdf, mean, var

Correlation: The second joint moment of two random variables X and Y is the *correlation*:

$$R_{XY} = \mathbb{E}\{XY\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} xyf_X(x)f_Y(y)dxdy \quad >> \text{corr}(X, Y)$$

Covariance: The second joint (centralized) moment of two random variables X and Y is the *covariance*:

$$K_{XY} = \mathbb{E}\{(X - m_X)(Y - m_Y)\} \quad >> \text{cov}(X, Y)$$

Correlation and covariance:

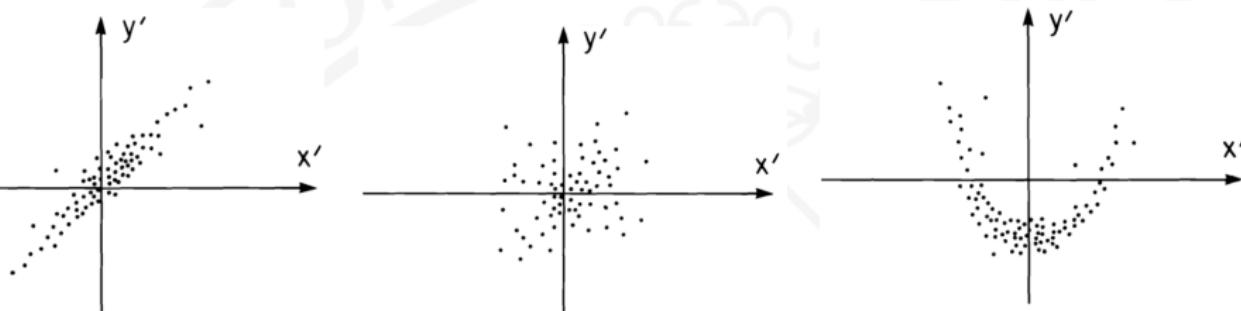
$$K_{XY} = R_{XY} - m_X m_Y$$

Correlation coefficient:

$$\rho = \frac{K_{XY}}{\sigma_X \sigma_Y} \quad -1 \leq \rho \leq +1 \quad >> \text{corrcoef}(X, Y)$$

- If $K_{XY} = 0$, then X and Y are *linearly uncorrelated*
- If $R_{XY} = 0$, then X and Y are orthogonal

Scatterplot: correlation and dependence



- Simple method to identify variable relations
- Simple transformations, e.g. $Y = X^2$, make the correlation coefficient useless
- `>> help scatter`
- `>> scatterhist`

Correlation matrix: For an n -tuple of random variables, $\mathbf{X} = [X_1, \dots, X_k]^\top$, there are k^2 pairs of random variables and associated correlations:

$$R_{X_i X_j} = \mathbb{E}\{X_i X_j\}, \quad i, j = 1, 2, \dots, k$$

This is a matrix of pairwise correlations:

$$\mathbf{R}_x = \mathbb{E}\{\mathbf{X}^\top \mathbf{X}\}$$

Covariance matrix: The matrix of covariances with (i, j) -th element:

$$K_{X_i X_j} = \mathbb{E}\{(X_i - m_{X_i})(X_j - m_{X_j})\},$$

which in matrix form is:

$$\mathbf{K}_x = \mathbf{R}_x - \boldsymbol{\mu}_x \boldsymbol{\mu}_x^\top,$$

where $\boldsymbol{\mu}_x$ is the n -tuple of means with element $\mathbb{E}\{X_i\}$.

Notion of convergence

If we measure two time samples $X = V(t)$ and $Y = V(t + \tau)$ being τ a small enough delay, we expect that X and Y to be correlated. If we repeat the experiment for an increasing number of n resistors, we obtain

$$\{\hat{\rho}_1, \hat{\rho}_2, \dots, \hat{\rho}_n\}$$

which should converge to the *true probabilistic correlation coefficient*, ρ .

We aim to assess

$$\lim_{n \rightarrow \infty} \hat{\rho}_n = \rho$$

Stochastic convergence

A sequence of random variables $\{X_n\}$ is actually a family of sequences of real numbers,

$$\{\{X_n(s)\} : s \in S\}$$

together with a sequence of joint probability distributions

$$\{F_{X_1 X_2 \dots X_n}\}.$$

There are four types of convergence:

- ① *Convergence almost surely:*

$$\lim_{n \rightarrow \infty} X_n(s) = X(s) \quad \forall s \in \tilde{S} \subseteq S, \quad P(\tilde{S}) = 1 \longrightarrow \text{Prob}\{\lim_{n \rightarrow \infty} X_n = X\} = 1$$

- ② *Convergence in MSE (aka expected square convergence):*

$$\lim_{n \rightarrow \infty} \mathbb{E}\{(X_n - X)^2\} = 0$$

- ③ *Convergence in Probability:*

$$\lim_{n \rightarrow \infty} \text{Prob}\{|X_n - X| > \varepsilon\} = 0, \quad \forall \varepsilon > 0$$

- ④ *Convergence in Distribution:*

$$\lim_{n \rightarrow \infty} F_{X_n}(x) = F_X(x)$$

Laws of large numbers

Imagine we actually repeat the experiment with n resistors of the same resistance and temperature and with voltmeters with the same bandwidth. Each set of n executions can be interpreted as either 1) n statistically independent experiments, or as 2) a composite experiment.

- ① *Weak law of large numbers:* The sequence of random variables $\{\bar{X}_n\}$ converges in probability to the nonrandom variable $P(A)$:

$$\lim_{n \rightarrow \infty} \text{Prob}\left\{ \left| \frac{K_n}{n} - P(A) \right| > \varepsilon \right\} = 1, \quad \forall \varepsilon > 0$$

- ② *Strong law of large numbers:*

$$\text{Prob}\left\{ \lim_{n \rightarrow \infty} \frac{K_n}{n} = P(A) \right\} = 1$$

```
>> X = randn(10,1); histfit(X)
>> X = randn(1000,1); histfit(X)
>> X = randn(10000,1); histfit(X)
```

Central limit theorem and the convergence of partial sums:

Consider

$$X_n = \sum_{i=1}^n Z_i,$$

and the standardized variables

$$Y_n = \frac{X_n - m_n}{\sigma_n},$$

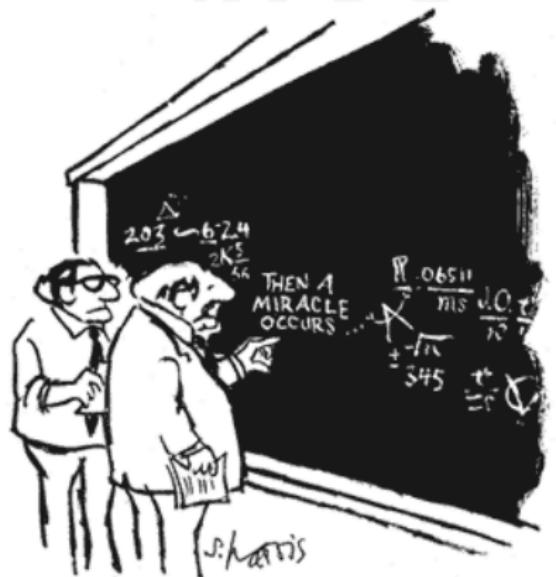
where m_n and σ_n^2 are the mean and variance of X_n . If $\{Z_i\}$ are independent and identically distributed (i.i.d) random variables, then Y_n converges in distribution to a Gaussian variable with zero mean and unity variance.

- Many phenomena are modeled in terms of Gaussian random variables
- The value of a variable (e.g. $V(t)$) is the result of a superposition of a large number of elementary effects (e.g. tiny voltage impulses).

```
>> X = randn(10,1); histfit(X)
>> X = randn(1000,1); histfit(X)
>> X = randn(10000,1); histfit(X)
```

Reviewed:

Sample point, sample space, sets, probability space, conditional probability, independence, random variable, correlation, covariance, distribution function, density function, Gaussian density, continuous/ discrete/ mixed random variable, probability mass function, joint distribution and density, multivariate Gaussian, functions/transformations of random variables, expectation, moments, characteristic function, conditional expectation, convergence, law of large numbers, central limit theorem, etc.



"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

Part 2: Discrete time random processes

Remember: Random variables

- The name random variable suggests a variable that takes on values randomly
- An observer measuring the amount of ‘noise’ on a communication link sees a *random variable*
- Mathematically, a random variable is neither random nor a variable
- A random variable is just a function mapping one sample space (part of a probability space) into another space (subset of the real-line space)

Random variables in signal processing

- A system transfers some ‘signal’ (of interest) through a noisy channel (electronic systems, medium of propagation, interfering signals)
- Signal and noise are *uncertain, unpredictable, random*
- No matter how much we know about the past, the future is hard to predict

Discrete-time random processes:

- A process is the result of an experiment
- Digital signal processing generates tons of examples:
 - speech,
 - visual signals (images, videos),
 - sonar and radar,
 - geophysical,
 - astrophysical,
 - biological signals, ...

Signal processing systems

- Basic operations: differentiation, integration/summation, multiplication, convolution, ...
- Both with (quasi) continuous (waveforms) and discrete-time signals (sequences)
- Probabilistic study of signals = study of averages over ensembles of waveforms or sequences
- The underlying probability theory = calculus of averages

What and how do we measure? Typically on a single member of the ensemble (a waveform/sequence) + averaging

- *Signal-to-noise error (SNR)*: mean/variance measured by time averaging
- *Channel equalizer*: which removes distortion → (time-averaged) MSE
- *Binary digital transmission system*: probability error (PE) is measured by computing the relative frequency of received bits in error over a long stream of bits

Example: The SNR problem in communication systems

- Study a communication system cannot be done looking at just one signal, but an ensemble of signal+noise processes
- We want to measure *expected values* (prob. params.) over the ensembles:

- **Signal-to-noise ratio (SNR):** relative strength of signal and noise

$$\text{SNR (dB)} = 10 \log \left(\frac{\sigma_x^2}{\sigma_n^2} \right), \quad x = s + n$$

- **Mean-square-error (MSE):** dissimilarity between a noisy signal and the clean version

$$\text{MSE} = \mathbb{E}\{(x - s)^2\}$$

- **Probability of error (PE):** likelihood of making an incorrect decision

$$\text{PE} = \mathbb{E}\{[\hat{s} \neq s]\}$$

Random processes

"Random processes are the probabilistic models of ensembles of waveforms and sequences"

Outline:

- ① Definition of a random process
- ② Temporal characteristics of random processes
 - Stationarity, WSS, and Ergodicity
 - Auto-correlation, auto-covariance
 - Cross-correlation, cross-covariance
 - Dependence
- ③ Spectral characteristics of random processes
 - Periodogram
 - Correlogram
 - Power spectral density (PSD)
- ④ Signal Processing applications
 - Interpolation
 - Noise-immunity
 - Signal detection, extraction and prediction
- ⑤ Examples of random processes
 - Bernoulli, Binomial, random walk
 - Markov, Wiener and Poisson processes
 - Autorregressive and moving average processes

Definition of random process

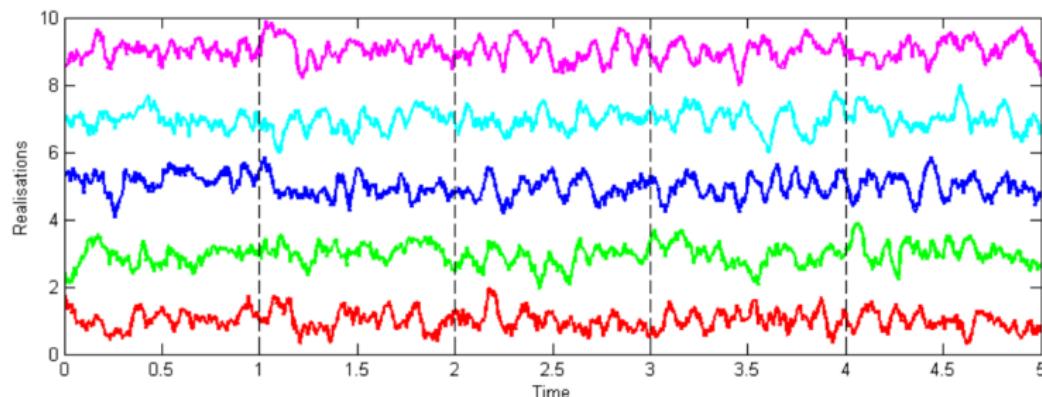
- A random process $X(t, s)$ is a random function of time t and a sample-point variable s
- $X(t, \cdot)$ is a function of sample points, i.e. a random variable
- $X(\cdot, s)$ is a function of time, i.e. a sample function

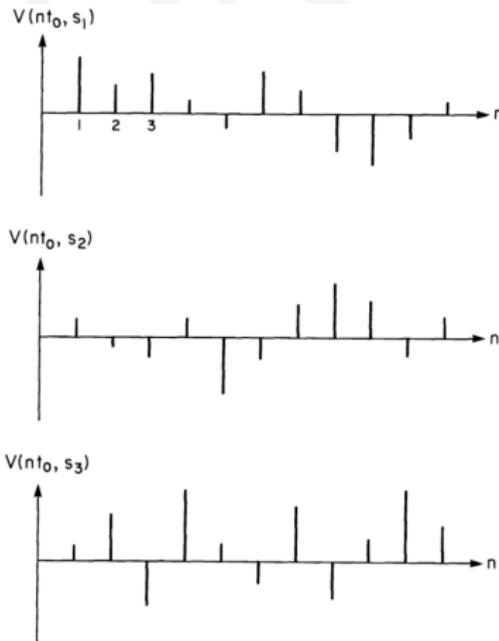
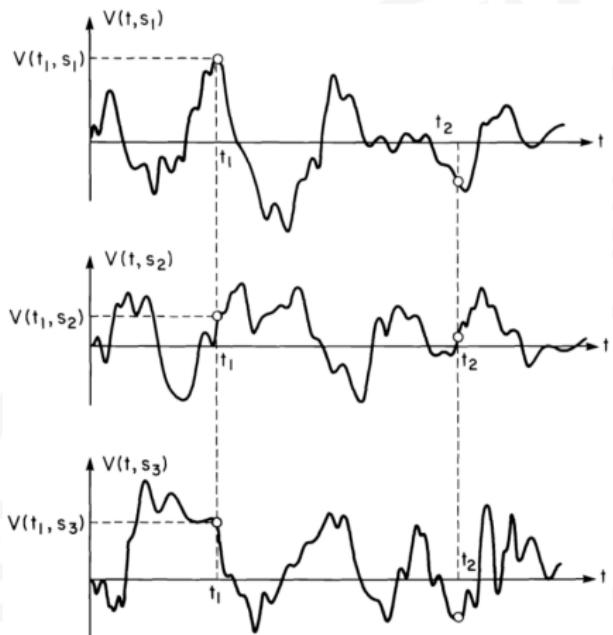
Intuition and notation for random processes

- Concept: Enlarging the random variable to include *time*
- Sometimes we use *stochastic process* instead of *random process*
- A random variable x becomes a function of the possible outcomes (values) s of an experiment and time t : $x(s, t)$
- The family of all such functions is called a random process, $X(s, t)$
- A random process becomes a random variable for fixed time

Ensemble and realization

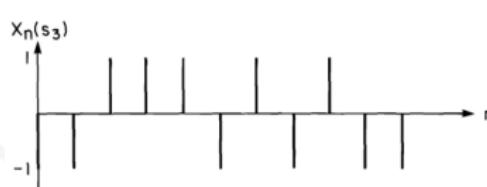
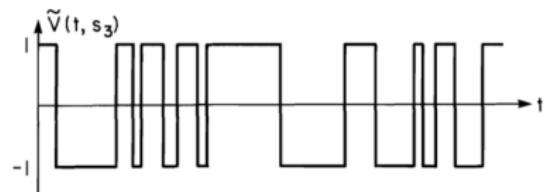
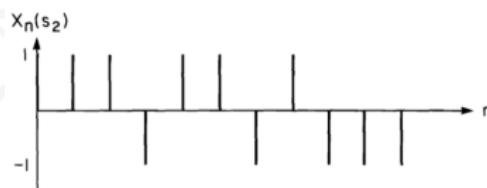
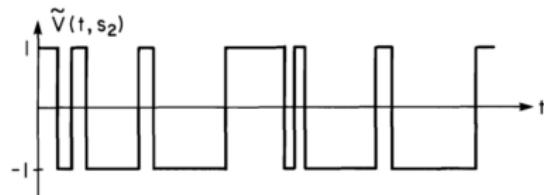
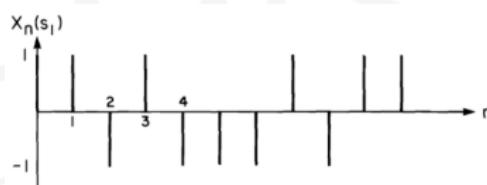
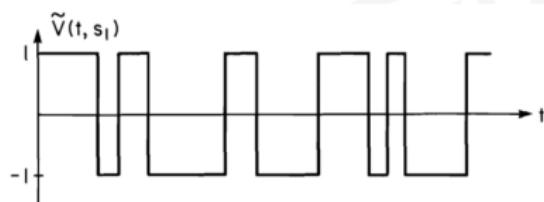
- $X(s, t)$ represents a family or ensemble of time functions
- Convenient short form $x(t)$ for specific waveform of the random process $X(t)$
- Each member time function is called a *realization*
- The complete collection of sample functions of a random process is called the *ensemble*





Statistical samples of a:

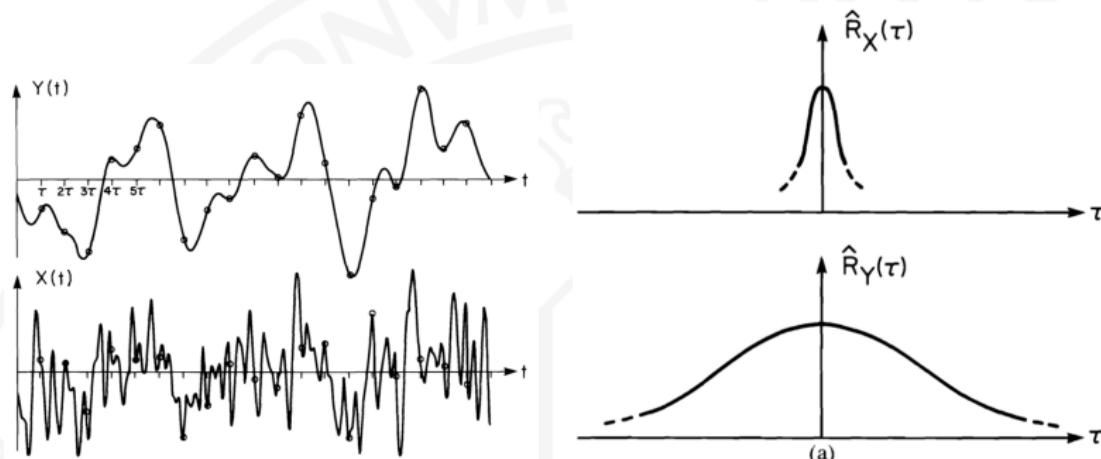
- Continuous-time random process
- Discrete-time random process



Statistical samples of a:

- Discrete-value, continuous-time random process
- Discrete-value, discrete-time random process

(Generalized) Harmonic Analysis Studies the *deterministic* (non-probabilistic) theory of random processes based on time-averages



- Empirical auto-correlation decreases with delay τ

$$R_X(\tau) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N X(n\tau + n)X(n\tau)$$

- $Y(t)$ is *narrow-band*: thermal noise at a lower temperature (less collisions)
- $X(t)$ is *wide-band*: thermal noise at a higher temperature (more collisions)

Empirical auto-correlation function (from a discrete-time average)

$$\hat{R}_X(\tau) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N X(n\tau + \tau) X(n\tau)$$

Autocorrelation function (from a continuous-time average):

$$\hat{R}_X(\tau) = \lim_{N \rightarrow \infty} \frac{1}{T} \int_{t=-T/2}^{T/2} X(t + \tau) X(t) dt$$

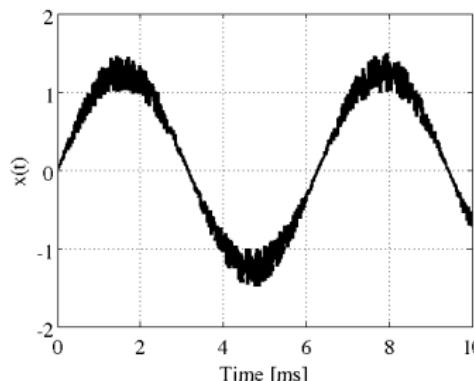
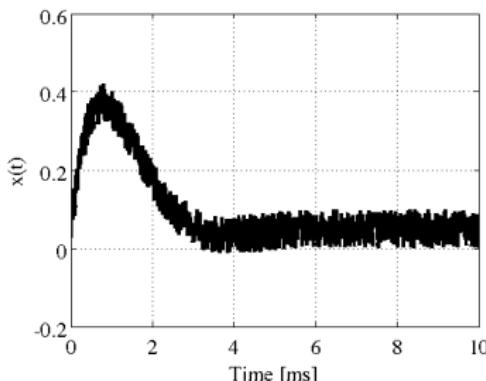
- Autocorrelation is related to the frequency composition of signals!
- We can study the frequency components via time-averages!
- Discrete/continuous autocorr are related for a finite (window) computation of the averaging/integration

Stationary process:

- "A stationary process (or strict(ly) stationary process or strong(ly) stationary process) is a stochastic process whose joint probability distribution does not change when shifted in time."
- Parameters such as the mean and variance, if they are present, also do not change over time and do not follow any trends
- Stationarity is useful in time series analysis

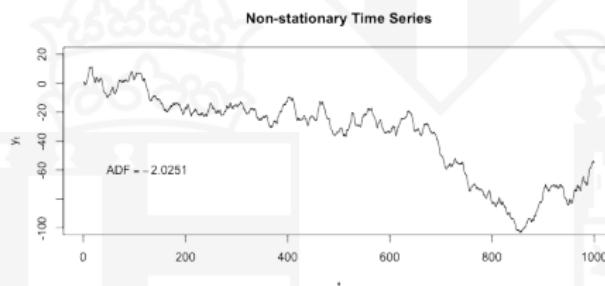
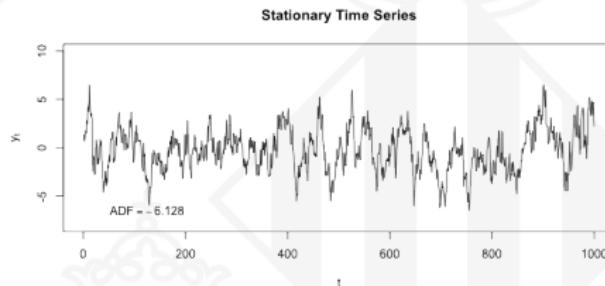
Cyclostationary process:

- "A cyclostationary process is a signal having statistical properties that vary cyclically with time".
- A cyclostationary process can be viewed as multiple interleaved stationary processes
- Examples: temperature, solar radiation, etc.



Stationary process, formally:

- Let $\{X_t\}$ be a stochastic process and let $F_X(x_{t_1+\tau}, \dots, x_{t_k+\tau})$ represent the cumulative distribution function of the joint distribution of $\{X_t\}$ at times $t_1 + \tau, \dots, t_k + \tau$
- Then, $\{X_t\}$ is said to be stationary if, for all k , for all τ , and for all t_1, \dots, t_k , $F_X(x_{t_1+\tau}, \dots, x_{t_k+\tau}) = F_X(x_{t_1}, \dots, x_{t_k})$
- Since τ does not affect $F_X(\cdot)$, F_X is not a function of time



Stationary process, examples:

- White noise is stationary
- The sound of a cymbal clashing, if hit only once, is not stationary because the acoustic power of the clash (and hence its variance) diminishes with time
- Some AR and MA processes may be either stationary or non-stationary, depending on the parameter values (poles inside/outside unit circle in z-domain)
- Let Y have a uniform distribution on $(0, 2\pi]$ and define the time series $\{X_t\}$ by

$$X_t = \cos(t + Y) \quad \text{for } t \in \mathbb{R}$$

Then $\{X_t\}$ is strictly stationary

Wide-sense stationarity (WSS): AKA weak-sense stationarity, covariance stationarity, or second-order stationarity

- WSS random processes only require that 1st moment and covariance do not vary with respect to time
- The mean function of a WSS continuous-time random process $x(t)$:

$$\mathbb{E}[x(t)] = m_x(t) = m_x(t + \tau) \text{ for all } \tau \in \mathbb{R}$$

→ the mean function $m_x(t)$ must be constant

- The autocovariance function of a WSS continuous-time RP $x(t)$:

$$\begin{aligned}\mathbb{E}[(x(t_1) - m_x(t_1))(x(t_2) - m_x(t_2))] &= C_x(t_1, t_2) = \\ &= C_x(t_1 + (-t_2), t_2 + (-t_2)) = C_x(t_1 - t_2, 0).\end{aligned}$$

→ the covariance function depends only on the difference between t_1 and t_2 , and only needs to be indexed by one variable rather than two variables:

$$C_x(t_1 - t_2, 0) \rightarrow C_x(\tau) \text{ where } \tau = t_1 - t_2.$$

- This implies that the autocorrelation depends only on $\tau = t_1 - t_2$:

$$R_x(t_1, t_2) = R_x(t_1 - t_2)$$

Weak or wide-sense stationarity (WSS), advantages:

- When processing WSS random signals with linear, time-invariant (LTI) filters, it is helpful to think of the correlation function as a linear operator
- Since it is a circulant operator (depends only on the difference between the two arguments), its eigenfunctions are the Fourier complex exponentials
- Additionally, since the eigenfunctions of LTI operators are also complex exponentials, LTI processing of WSS random signals is highly tractable—all computations can be performed in the frequency domain
- Thus, the WSS assumption is widely employed in signal processing algorithms

Jointly wide-sense stationarity:

Two processes X_t and Y_t are jointly WSS if each one is WSS and the cross-correlation depends only on the difference between time-indices:

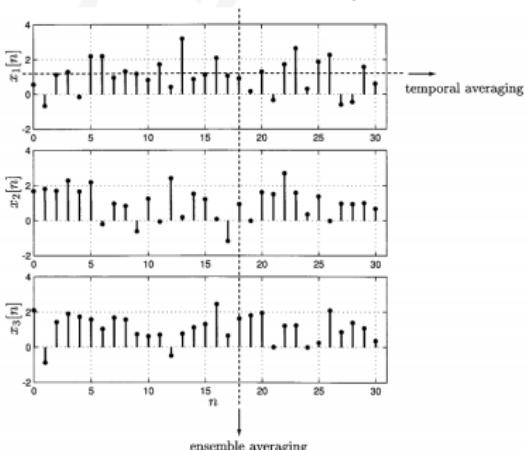
$$R_{XY}(\tau) = \mathbb{E}\{X_t Y_{t-\tau}\}$$

Ergodicity:

- An ergodic dynamical system has the same behavior averaged over time as averaged over the space of all the system's states (phase space)
- Ergodicity is where the ensemble average equals the time average
- Examples:
 - In physics, a system satisfies the ergodic hypothesis of thermodynamics
 - In statistics, a RP for which the time average of one sequence of events is the same as the ensemble average

Ergodicity:

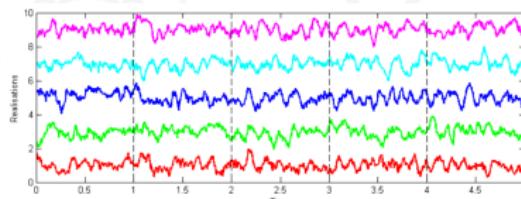
- When a random process is WSS, its mean does not depend on time
- Hence, the RVs $\{..., X(-1), X(0), X(1), ...\}$ all have the same mean
- At least as far as the mean is concerned, when we observe a realization of a random process, it is as if we are observing multiple realizations of the same random variable
- This suggests that we may be able to determine the value of the mean from a single infinite length realization
- If it is true that the temporal average converges to the true mean $\mu = 1$, then the *temporal averaging* is equivalent to *ensemble averaging* or that the “*random process is ergodic in the mean*”



- This property is of great practical importance since it assures us that by averaging enough samples of the realization, we can determine the mean of the random process
- A random process is ergodic in the autocorrelation if we can determine the autocorrelation by averaging enough autocorrelation samples of the realization

Ergodicity, example in electronics:

- Each resistor has thermal noise associated with it and it depends on the temperature
- Take N resistors (N should be very large) and plot the voltage across those resistors for a long period
- For each resistor you will have a waveform
- Calculate the average value of that waveform
- This gives you the time average
- You should also note that you have N waveforms as we have N resistors
- These N plots are known as an ensembles
- Now take a particular instant of time in all those plots and find the average value of the voltage
- That gives you the ensemble average for each plot
- If both ensemble average and time average are the same then it is ergodic.



(Discrete) ergodicity, summarizing:

- Not always possible to obtain different samples from the RP
- Sometimes we only have one sample!
- Can we infer the statistical properties of the process using just one sample from the process? If so, the process is ergodic.
- A process is ergodic if the mean is:

$$\langle X(n) \rangle = \frac{1}{2N+1} \sum_{n=-N}^N X(n) = \mathbb{E}\{X(n)\}$$

- A process is ergodic if the autocorrelation is:

$$\langle X(n)X(n-l) \rangle = \mathbb{E}\{X(n)X(n-l)\}$$

- Two processes X and Y are joint ergodic if:

$$\langle X(n)Y(n-l) \rangle = \mathbb{E}\{X(n)Y(n-l)\}$$

Remember the important definitions:

- Mean of a random process:

$$\mathbb{E}\{X(t)\} = m_X(t), \text{ where } m_X(\cdot) \text{ is a 'mean waveform'}$$

- Autocorrelation of a random process:

$$\mathbb{E}\{X(t_1)X(t_2)\} = R_X(t_1, t_2), \quad \mathbb{E}\{X(t)X(t + \tau)\} = R_X(t, t + \tau)$$

- Autocorrelation of a WSS random process:

$$R_X(\tau) = R_X(t, t + \tau), \quad R_X(\tau) = R_X(-\tau), \quad R_X(0) = \mathbb{E}\{X^2(t)\}$$

- Autocovariance of a random process:

$$\mathbb{E}\{[X(t_1) - m_X(t_1)][X(t_2) - m_X(t_2)]\} = K_X(t_1, t_2)$$

- Cross-correlation for two random processes:

$$\mathbb{E}\{X(t_1)Y(t_2)\} = R_{XY}(t_1, t_2)$$

- Cross-covariance for two random processes:

$$\mathbb{E}\{[X(t_1) - m_X(t_1)][Y(t_2) - m_Y(t_2)]\} = K_{XY}(t_1, t_2)$$

Expectation in MATLAB:

- Recall:

$$\mathbb{E}\{X\} = \int_{-\infty}^{\infty} \alpha f_X(\alpha) d\alpha$$

- In real life we don't have the PDF, just observations (samples)!
- In real life we never have all realizations, so we need to assume *ergodicity*!
- Given $X(n)$, approximate the ensemble average with the time average:

$$\mathbb{E}\{X\} \approx \frac{1}{N} \sum_{n=1}^N X(n)$$

PDF estimation in MATLAB:

- The PDF is estimated by the normalised histogram
`>> hist(x)`
`>> [counts,centers] = hist(x,nbin)`
`>> histfit(x)`
- The histogram gives directly the count of all different values per bin
- Normalise this, and we obtain the probability that any value can occur (density).
- This multiplied with the hit number of all possible values gives, naturally, the count of all values
- **Explore:** `>> help ksdensity`
- **Discuss:** the problems in multidimensional PDF estimation

Autocorrelation function estimation in MATLAB:

- The ACF contains information about the history of the random process
- Assuming a large time interval $2T$ and ergodicity (the RP is WSS)

$$R_X(t, t + \tau) = \mathbb{E}\{X(t, t + \tau)\} \approx \frac{1}{2T} \int_{t-\tau}^{t+\tau} x(t)x(t + \tau)dt \approx R_X(\tau),$$

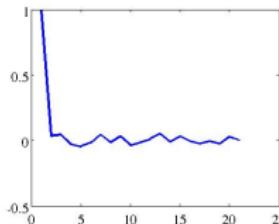
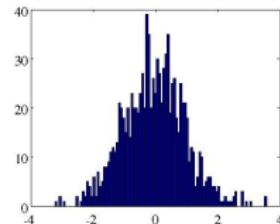
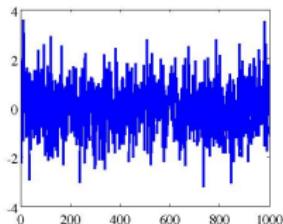
or in discrete notation

$$R_X(t, t + \tau) \approx \frac{1}{N} \sum_{n=1}^N x(n)x(n + k),$$

which is simply a convolution without reversing

- Autocorrelation in MATLAB

```
>> [acf, lags, bounds] = autocorr(y);
>> x=randn(1,1000); plot(x); hist(x,100); plot(autocorr(x));
```



Autocorrelation function estimation in MATLAB:

- The ACF contains information about the history of the random process
- Assuming a large time interval $2T$ and ergodicity (the RP is WSS)

$$R_X(t, t + \tau) = \mathbb{E}\{(t, t + \tau)\} \approx \frac{1}{2T} \int_{t-T}^{t+T} x(t)x(t + \tau)dt \approx R_X(\tau),$$

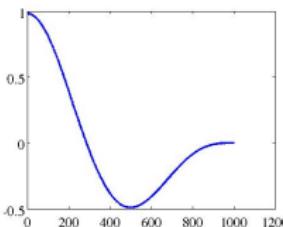
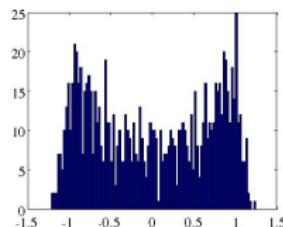
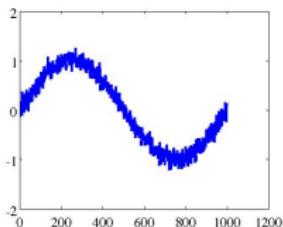
or in discrete notation

$$R_X(t, t + \tau) \approx \frac{1}{N} \sum_{n=1}^N x(n)x(n + k),$$

which is simply a convolution without reversing

- Autocorrelation in MATLAB

```
>> [acf, lags, bounds] = autocorr(y);
>> x=sin(1:(2*pi/1000):2*pi)+0.1*randn(1,1001);
```



- Play around with: `>> load sunspot.dat`

Cross-correlation function (XCF) estimation in MATLAB:

- The XCF contains information about the cross-history between two random processes

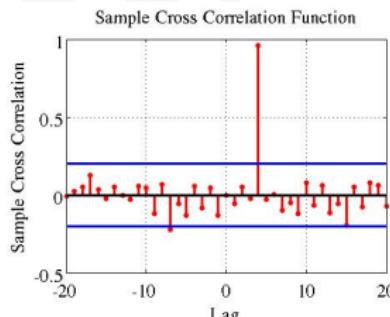
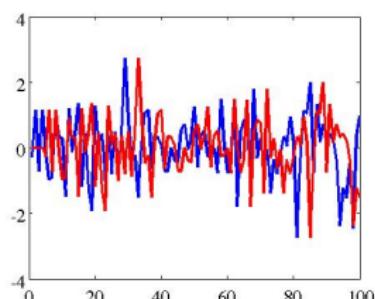
$$\mathbb{E}\{X(t_1)Y(t_2)\} = R_{XY}(t_1, t_2)$$

- Cross-correlation in MATLAB

```
>> [xcf, lags, bounds] = crosscorr(y1, y2);
```

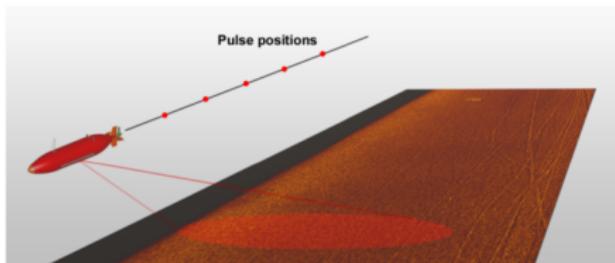
- Toy example in MATLAB

```
% Random sequence of 100 Gaussian deviates and a delayed
% version lagged by 4 samples
>> x = randn(100,1); % 100 Gaussian deviates N(0,1)
>> y = lagmatrix(x,4); % Delay it by 4 samples
>> y(isnan(y)) = 0; % Replace NaN's with zeros
>> plot(x,'b'); hold on, plot(y,'r')
>> crosscorr(x,y) % It should peak at the 4th lag
```



Real data collected by a sonar:

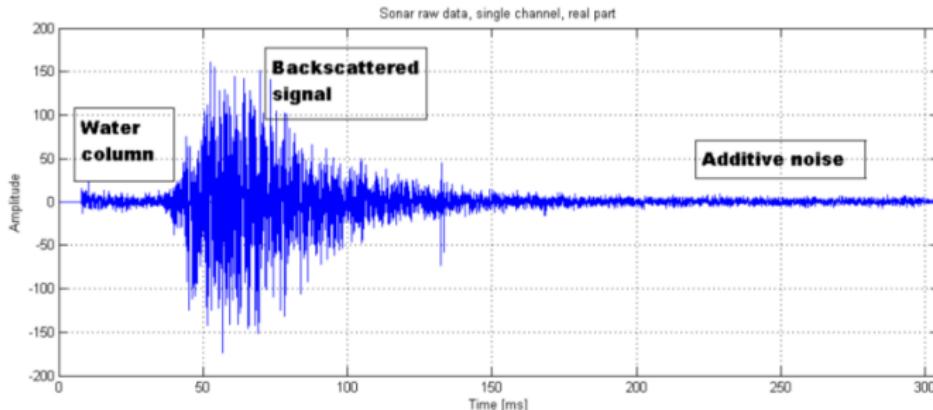
- The HUGIN autonomous underwater vehicle
- Wideband interferometric synthetic aperture sonar
- Transmitter that insonifies the seafloor with a LFM pulse
- Array of receivers that collects the echoes from the seafloor
- The signal scattered from the seafloor is considered to be random
- The signal consists of a signal part and additive noise



Question 1: is the process stationary?

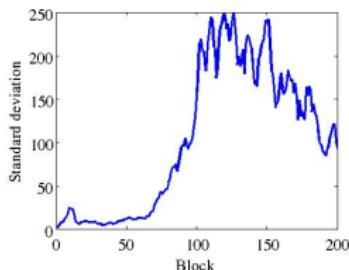
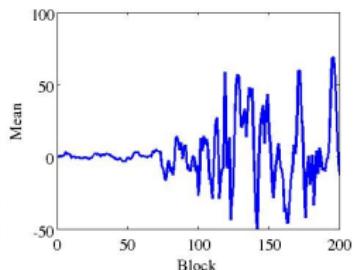
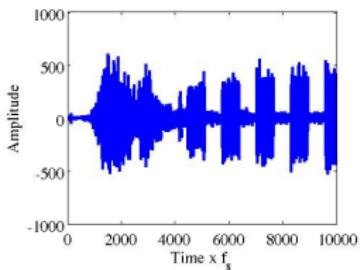
- Single channel timeseries from one ping
- Consider the collected data a random process.

```
>> load sonardata2;  
>> channel = 10;  
>> plot(1:10000,real(data(:,channel)));
```



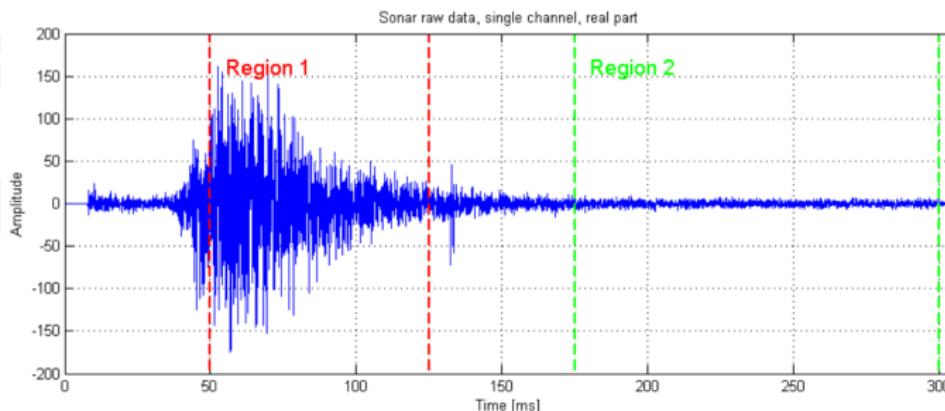
Question 1: is the process stationary?

```
nblocks = 200;
blocksize = 50;
step = (blocksize/4);
for n = 1:nblocks
    statarr(n,1) = mean( data( (n-1)*step+1:(n-1)*step+blocksize ) );
    statarr(n,2) = std( data( (n-1)*step+1:(n-1)*step+blocksize ) );
end
```



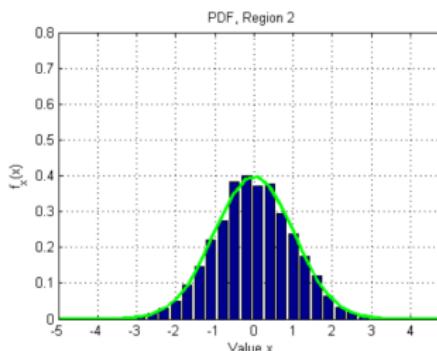
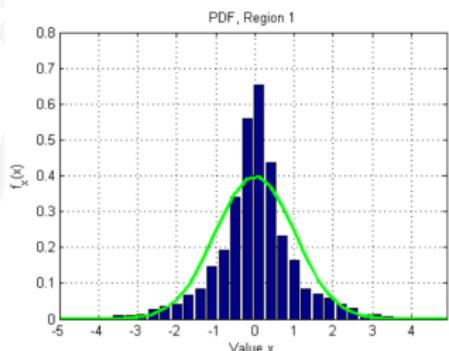
Question 1: is the process stationary? No!

- Divide into “similar” regions before we continue our statistical analysis
- Region 1: Backscattered signal from the seafloor
- Region 2: Additive noise



Question 2: Is the probability density function Gaussian?

- Approach: Compare the theoretical PDF with the estimated PDF (from the normalised histogram)
- Easier: `>> histfit`
- Play around with other statistics to assess deviation from a Gaussian



Question 2: PDF estimation is more complex than expected!

- The sonar data is complex!

$$x(t) = x_{Re}(t) + jx_{Im}(t) = ae^{j\phi(t)}$$

- The complex random sequence can be considered two independent random sequences (in a vector) with joint PDF
- We can check the PDF of the real and imaginary part separately
- If $x_{Re}(t)$ and $x_{Im}(t)$ are statistically independent, it can be shown that the PDF of the amplitude (or magnitude)

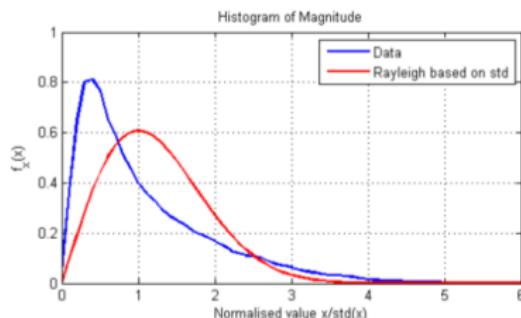
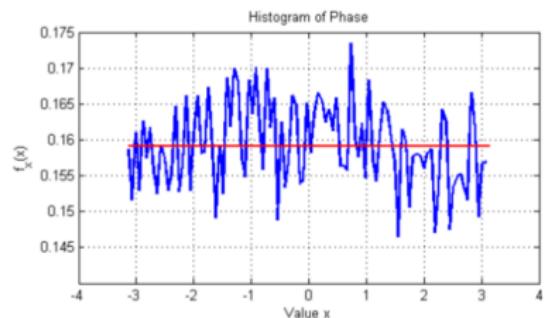
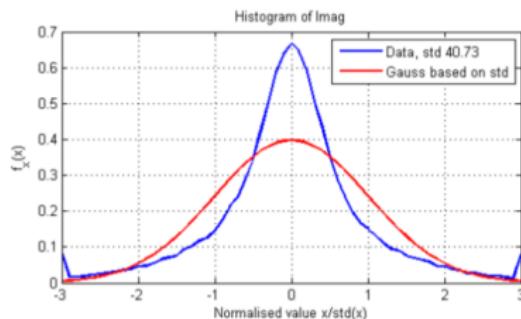
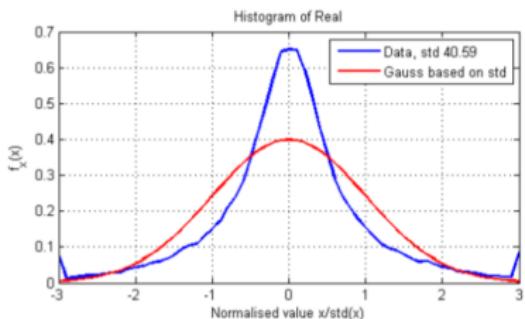
$$a(t) = \sqrt{x_{Re}(t)^2 + x_{Im}(t)^2}$$

should be a Rayleigh distribution, and that the PDF of the phase

$$\phi(t) = \tan^{-1}(x_{Im}/x_{Re})$$

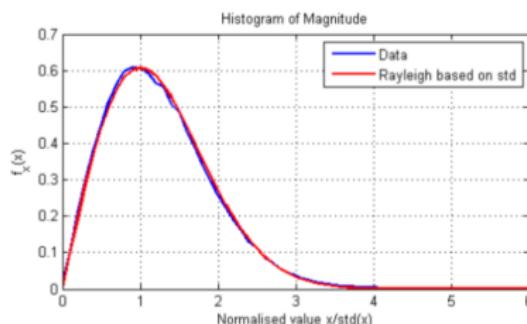
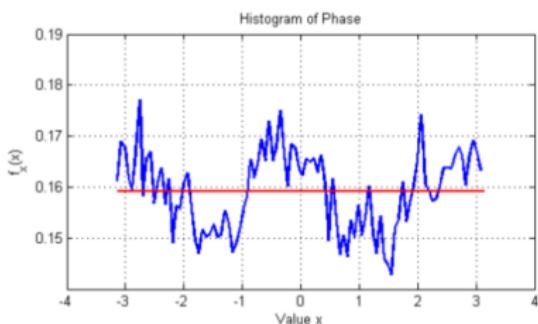
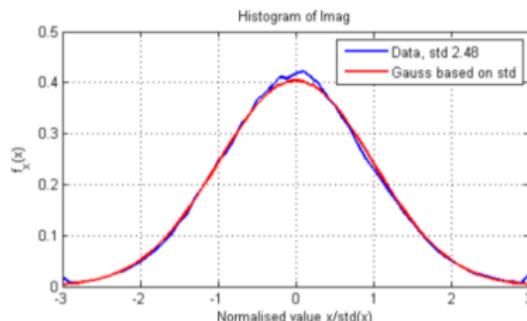
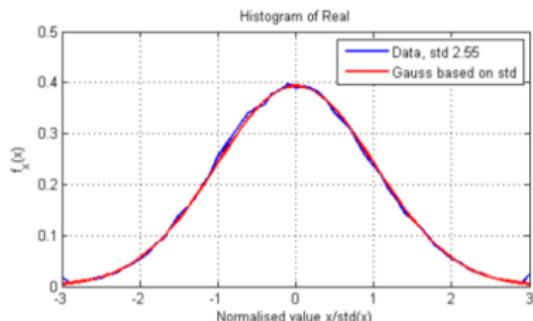
should be uniform

Question 2: PDF estimation in region 1



```
>> help pdf, raylpdf
```

Question 2: PDF estimation in region 2



```
>> help pdf, raylpdf
```

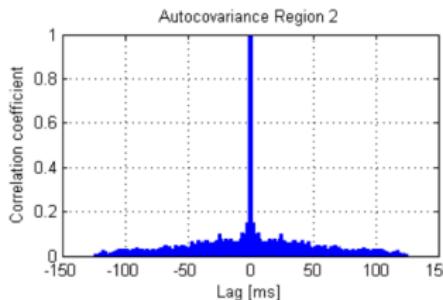
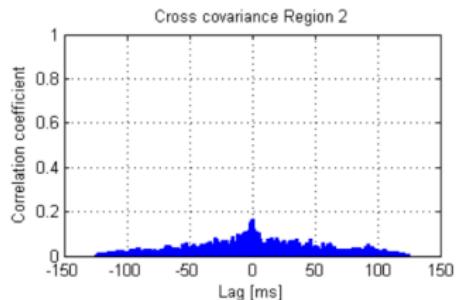
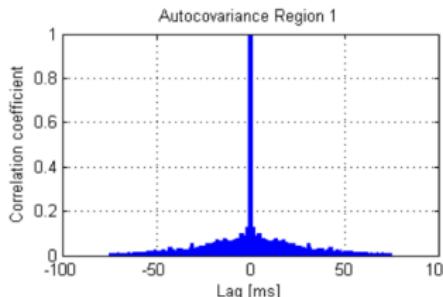
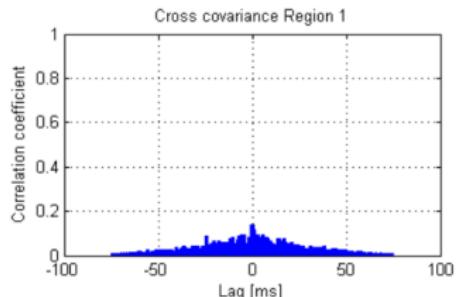
Question 2, conclusions

- In region 2, the real and imaginary part fits a Gauss well
- The phase is also uniform (all phase values are equally probable)
- The magnitude also fits well a Rayleigh distribution
- In region 1, this is not the case.
- The histogram indicates that the PDF is heavy tailed.
- This means that it is more likely to have spikes (large amplitude values) in the time-series than in a time-series with Gaussian PDF.
- This actually fits well the theory of acoustic scattering.

Discuss: What can an estimate of the PDF be used to?

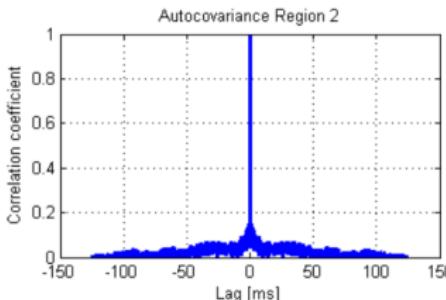
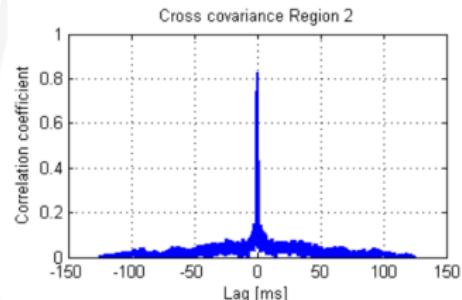
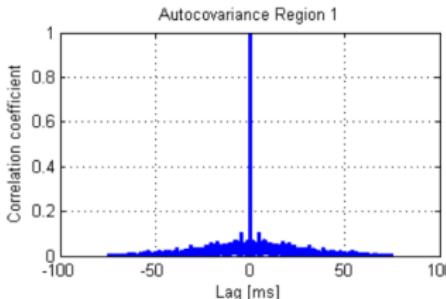
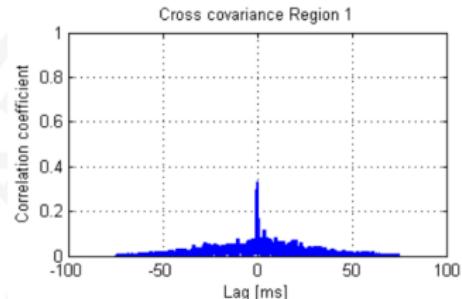
Question 3: are real and complex parts dependent?

- If the normalised cross-covariance is zero, the two processes are said to be uncorrelated
- Play around with `>> corr, corrcoef, cov,`



Question 3: are channels dependent?

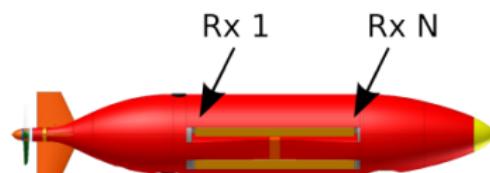
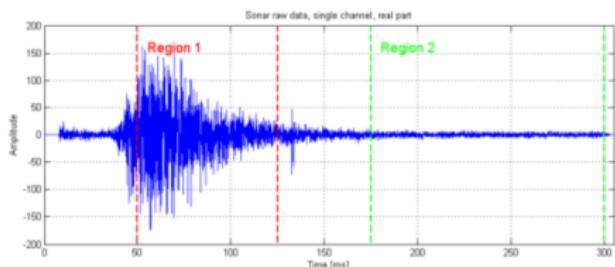
- If the normalised cross-covariance is zero, the two processes are said to be uncorrelated
- Play around with `>> corr, correcoef, cov,`



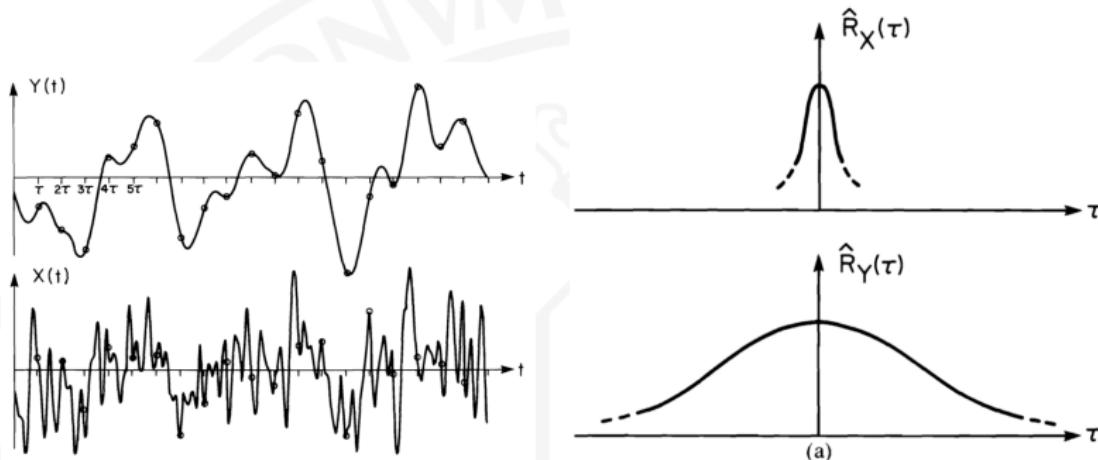
Question 3: are real and complex channels dependent?

- The real and imaginary part of the signal is uncorrelated
- The individual channels (receiver elements) are correlated
- What physical phenomenon could cause this?
- The channels are strongly correlated in region 2 (the noise region)

Discuss: Why is this?



Remember: The ACF characterizes the temporal properties of the signals
Now: What about the spectral properties?



- Empirical auto-correlation decreases with delay τ

$$R_X(\tau) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N X(n\tau + n)X(n\tau)$$

- $Y(t)$ is *narrow-band*: thermal noise at a lower temperature (less collisions, lower frequencies)
- $X(t)$ is *wide-band*: thermal noise at a higher temperature (more collisions, higher frequencies)

Fourier transform

- For a deterministic sequence $x_T(t)$, the Fourier transform is defined as

$$\mathcal{F}[X_T(t)] = \tilde{X}_T(f) = \tilde{X}_T(\omega) = \int_{-\infty}^{\infty} X_T(t) e^{-j2\pi ft} dt$$

- The Fourier transform is simply called the *spectrum*
- $\omega = 2\pi f$ is understood as angular frequency (if t is time)
- The inverse Fourier transform

$$X_T(f) := X_T(\omega) = \int_{-\infty}^{\infty} \tilde{X}_T(t) e^{+j2\pi ft} df$$

Periodogram or finite-time spectrum

$$\hat{R}_X(\tau) = \lim_{N \rightarrow \infty} \frac{1}{T} \int_{t=-T/2}^{T/2} X(t+n)X(t)dt$$

- Consider the finite segment:

$$X_T(t) = \begin{cases} X(t) & |t| \leq T/2 \\ 0 & |t| > T/2 \end{cases}$$

- Compute the Fourier transform:

$$\mathcal{F}[X_T(t)] = \tilde{X}_T(f) = \int_{-\infty}^{\infty} X_T(t) e^{-j2\pi ft} dt, \quad X_T(f) = \int_{-\infty}^{\infty} \tilde{X}_T(t) e^{+j2\pi ft} df$$

- $\tilde{X}_T(f)$ is the complex density of complex sinusoids
- Periodogram* is the (convenient) time-normalized squared magnitude:

$$(1/T)|\tilde{X}_T(f)|^2 = \int_{-\infty}^{\infty} R_X(\tau) e^{-j2\pi f\tau} d\tau$$

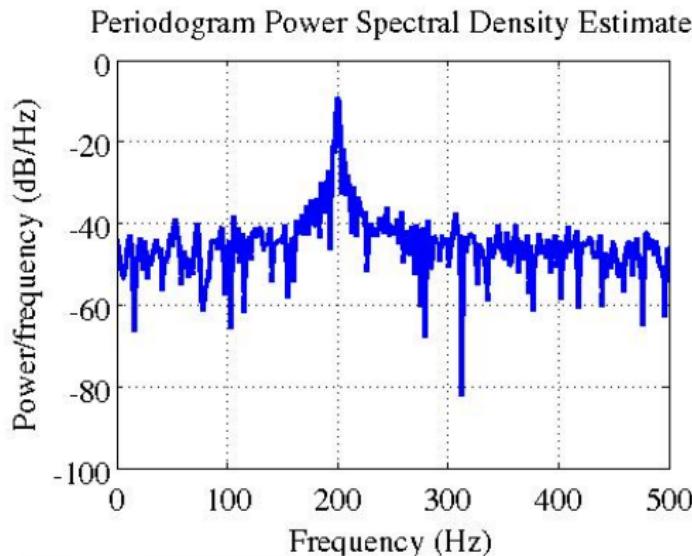
where

$$R_X(\tau) = \frac{1}{T} \int_{t=-T/2}^{T/2} X_T(t+n)X_T(t)dt$$

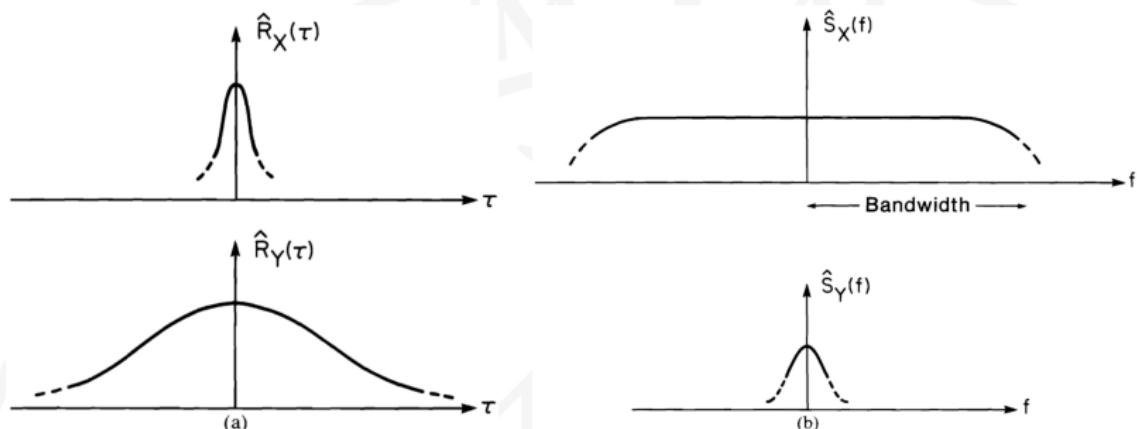
is the *correlogram*

Periodogram or finite-time spectrum

```
>> Fs = 1000;  
>> t = 0:1/Fs:.3;  
>> x = cos(2*pi*t*200)+0.1*randn(size(t));  
>> periodogram(x, [], 'onesided', 512, Fs)
```



Correlogram or finite-time autocorrelation



- **Correlogram:**

$$R_X(\tau) = \frac{1}{T} \int_{t=-T/2}^{T/2} X_T(t+n)X_T(t)dt = \frac{1}{T} \int_{-T/2}^{T/2-|\tau|} X(t+|\tau|)X(t)dt$$

- Kind of autocorrelation related to the frequency composition of the finite segment $X_T(t)$:

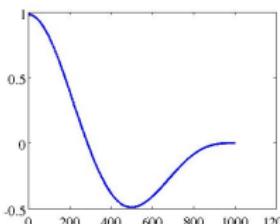
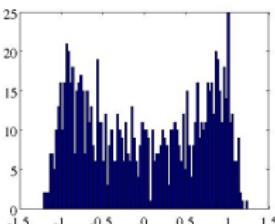
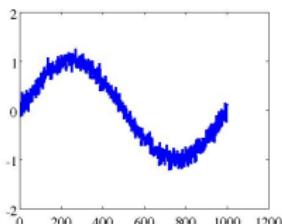
$$\lim_{T \rightarrow \infty} R_X(\tau)_T = \hat{R}_X(\tau)$$

- “The autocorrelation \hat{R}_X is related to the frequency composition of $X(t)$ through the Fourier transform”

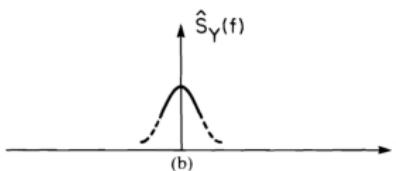
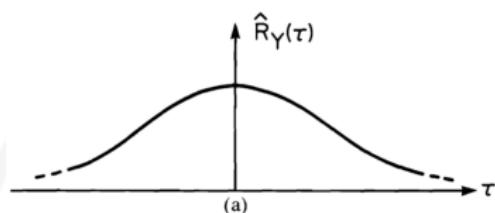
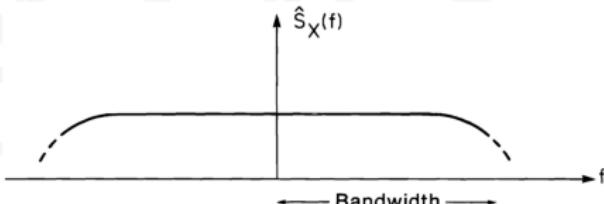
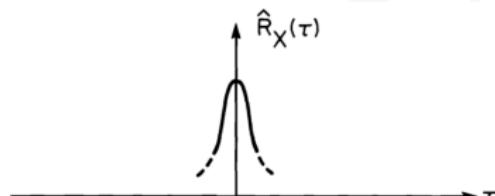
Correlogram or finite-time autocorrelation

- The autocorrelation function is an important diagnostic tool for analyzing time series in the time domain
- We use the autocorrelation plot, or correlogram, to better understand the evolution of a process through time by the probability of relationship between data values separated by a specific number of time steps
- The correlogram plots correlation coefficients on the vertical axis, and lag values on the horizontal axis
- A correlogram is not useful when the data contains a trend; data at all lags will appear to be correlated because a data value on one side of the mean tends to be followed by a large number of values on the same side of the mean. We must remove any trend in the data before you create a correlogram
- Explore:** >> diff, parcorr

```
>> [acf, lags, bounds] = autocorr(y);  
>> x=sin(1:(2*pi/1000):2*pi)+0.1*randn(1,1001);
```



Power spectral density (PSD):



- Problem: Since $(1/T)|\tilde{X}_T(f)|^2$ shows erratic behavior as $T \rightarrow \infty$, we do a sliding averaging (centered in u):

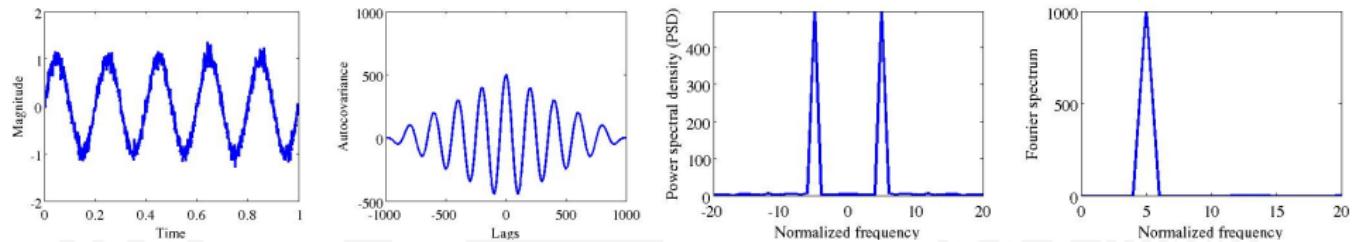
$$\hat{S}_X(f) = \lim_{T \rightarrow \infty} \lim_{U \rightarrow \infty} \frac{1}{U} \int_{-U/2}^{U/2} \frac{1}{T} |\tilde{X}_T(u, f)|^2 du = \int_{-\infty}^{\infty} \hat{R}_X(\tau) e^{-j2\pi f \tau} d\tau$$

Example: Power spectral density (PSD) for the waveform X is the frequency density of the time-averaged power that the voltage $X(t)$ would dissipate in a resistance:

$$\frac{1}{2\Delta} \langle P \rangle \approx \frac{\hat{S}_X(f)}{r} [\text{watts}], \quad \Delta : \text{small bandwidth}$$

Power spectral density (PSD) example: The power spectral density reveals frequency selective information

```
>> clear;clc;close all;
>> n = 1000; dt = 0.001; n = 1000; t= [0:n-1] * dt; f = 5;
>> x = sin(2*pi*f*t) + 0.1*randn(1,n);
>> [covx,lags] = xcov(x);
>> figure,plot(t,x); xlabel('Time'),ylabel('Magnitude')
>> figure,plot(lags,covx); xlabel('Lags'),ylabel('Autocovariance')
>> Sx = abs(fftshift( fft( x ) ));
>> nfreq = [-n/2:n/2-1]/n / dt;
>> figure,plot(nfreq,Sx);
>> xlabel('Normalized frequency'),ylabel('Power spectral density (PSD)')
>> st = exp( j*2*pi*f*t ); Sw = abs(fftshift(fft(st)));
>> figure,plot(nfreq,Sw);
>> xlabel('Normalized frequency'),ylabel('Fourier spectrum')
```



Power spectral density (PSD) in MATLAB:

>> pburg	PSD using Burg method
>> pcov	PSD using covariance method
>> peig	Pseudospectrum using eigenvector method
>> periodogram	PSD using periodogram
>> pmcov	PSD using modified covariance method
>> pmtm	PSD using multitaper method (MTM)
>> pmusic	Pseudospectrum using MUSIC algorithm
>> pwelch	PSD using Welch's method
>> pyulear	PSD using Yule-Walker AR method

- Spectral estimation will be a separate topic in this course ...
- We will see details of these algorithms and more examples ...

Transfer function of a filter

- If $X(t)$ is the input to a linear time-invariant filter with impulsive-response function $h(t)$, and $Y(t)$ is the output

$$Y(t) = \int_{-\infty}^{\infty} h(t-u)X(u)du = X(t) \otimes h(t),$$

then the input and output autocorrelations are related via convolution

$$\hat{R}_Y(\tau) = \int_{-\infty}^{\infty} \hat{R}_X(\tau-u)r_h(u)du = \hat{R}_X(\tau) \otimes r_h(\tau)$$

$$r_h(\tau) = \int_{-\infty}^{\infty} h(\tau+v)h(v)dv = h(\tau) \otimes h(-\tau)$$

- Convolution theorem for Fourier transforms allows to show:

$$\hat{S}_Y(f) = \hat{S}_X(f)|H(f)|^2$$

where

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

which is the *transfer function*.

- Analogy:

$$\tilde{Y}(f) = \tilde{X}(f)H(f)$$

where $\tilde{X}(f)$, $\tilde{Y}(f)$ are the Fourier transforms of the input and output waveforms (equivalent for the finite case, sequences and summations)

1: Interpolation of time-sampled waveforms

- Let $X(t)$ be a random (unpredictable) waveform
- Let $\{X(iT) : i = 0, \pm 1, \pm 2, \dots\}$ the time-sampled version
- Let be $p(t)$ an interpolating pulse
- The approximation to the waveform is:

$$X(t) \approx \hat{X}(t) = \sum_{-\infty}^{\infty} X(iT)p(t - iT)$$

- How well do we do it?

$$\text{Error power} = \text{MSE} = \langle [X(t) - \hat{X}(t)]^2 \rangle$$

- Nyquist-Shannon sampling theorem:** MSE=0 iff PSD is bandlimited to less than half the sampling rate:

$$\hat{S}_X(f) = 0, \quad |f| \geq B > \frac{1}{2T},$$

and the interpolating pulse is an appropriately designed bandlimited pulse:

$$p(t) = \frac{\sin(\pi t/T)}{\pi t/T}$$

2: Signal detection

- Detection of a finite-energy signal buried in noise is very important in signal processing (radar, sonar, communications, etc.)!
- Try to design detectors that maximize SNR
- SNR defined as the ratio of the detector output Y when the signal alone is present over the time-average power of the detector output when the noise alone is present

$$H(f) = \frac{S^*(f)e^{-j2\pi f t_0}}{\hat{S}_N(f)},$$

where S are the Fourier transforms of the signal and the noise

- This is known as the *matched filter*!
- Optimal detection statistic:

$$\text{SNR}_{max} = \int_{-\infty}^{\infty} \frac{|S(f)|^2}{\hat{S}_N(f)} df \approx \sqrt{\frac{T}{2} \int_{-\infty}^{\infty} \left(\frac{S_S(f)}{S_N(f)} \right)^2 df}$$

- Problem: How to estimate the noise spectrum? Let's see the signal extraction problem ...

3: Signal extraction

- Extract a random signal buried in noise:

$$X(t) = S(t) + N(t), \quad \hat{S}(t) = X(t) \otimes h(t)$$

- We want to determine the filter transfer function $H(f)$ that minimizes MSE:

$$\text{Error power} = \text{MSE} = \langle [S(t) - \hat{S}(t)]^2 \rangle$$

- We will see that:

$$H(f) = \frac{\hat{S}_S(f)}{\hat{S}_S(f) + \hat{S}_N(f)}$$

- This minimizes MSE:

$$\text{MSE}_{min} = \int_{-\infty}^{\infty} \frac{\hat{S}_S(f) \hat{S}_N(f)}{\hat{S}_S(f) + \hat{S}_N(f)} df$$

- High attenuation at the frequencies where noise dominates the signal power, and viceversa

4: Signal prediction

- Prediction of the future value of a time-discrete random process is very important (forecasting in economics, meteorology, bioengineering, electronics, ...)
- Let's use $\{X([k - i]T) : i = 0, 1, 2, \dots, n - 1\}$ to predict value $X([k + p]T)$ (p steps into the future!):

$$\hat{X}([k + p]T) = \sum_{i=0}^{n-1} h_i X([k - i]T)$$

- We want to determine the transfer function $H(f)$ that minimizes MSE:

$$\text{Error power} = \text{MSE} = \langle [X([k - i]T) - \hat{X}([k + p]T)]^2 \rangle$$

- We will see that the optimal n prediction coefficients satisfy:

$$\sum_{i=0}^{n-1} \hat{R}_X([j - i]T) h_i = \hat{R}_X([j + p]T), \quad j = 0, 1, 2, \dots, n - 1,$$

- This minimizes MSE:

$$\text{MSE}_{min} = \hat{R}_X(0) - \sum_{i=0}^{n-1} h_i \hat{R}_X([i + p]T)$$

Main conclusion: The PSD and the autocorrelation function play a fundamental role in signal processing applications!

Example 1: Bernoulli process: consider an infinite sequence of independent Bernoulli trials of a binary experiment, such as flipping a coin. The resultant sequence of event indicators:

$$x_n = \begin{cases} 1 & \text{success in } n\text{-th trial} \\ 0 & \text{failure in } n\text{-th trial} \end{cases}$$

- Example of a discrete-value, discrete-time random process
- Probability of success:

$$P\{X_n = 1\} = p$$

- Mean:

$$m_X(n) = p,$$

which is independent of time n

- Autocovariance:

$$K_X(n_1, n_2) = \begin{cases} p(1-p) & n_1 - n_2 = 0 \\ 0 & n_1 - n_2 \neq 0 \end{cases},$$

which depends only on time difference $n_1 - n_2$

Example 2: Binomial counting process: consider counting the number of successes in the Bernoulli process:

$$Y_n = \sum_{i=1}^n X_i$$

- The infinite sequence $\{Y_n\}$ is an example of a discrete-value, discrete-time random process
- Mean:

$$m_Y(n) = n p,$$

which depends on time n

- Autocovariance:

$$K_Y(n_1, n_2) = p(1 - p) \min\{n_1, n_2\},$$

where

$$\min\{n_1, n_2\} = \begin{cases} n_1 & n_1 - n_2 \leq 0 \\ n_2 & n_1 - n_2 \geq 0 \end{cases},$$

which depends on more than time difference $n_1 - n_2$

- Probability distribution for n Bernoulli trials yielding k successes:

$$p^k (1 - p)^{n-k}$$

- The total number of sequences is the *binomial coefficient*:

$$\frac{n!}{k!(n-k)!}$$

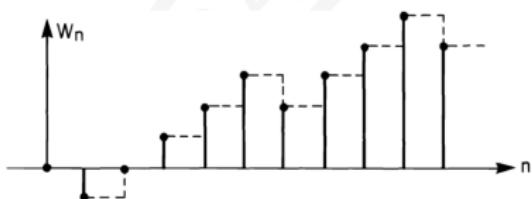
Example 3: Random-walking process: modify the Bernoulli process to ± 1 :

$$z_n = \begin{cases} +1 & \text{success in } n\text{-th trial} \\ -1 & \text{failure in } n\text{-th trial} \end{cases}$$

and consider the sum of these binary variables $W_n = \sum_{i=1}^n Z_i$

- The underlying process Z is related to the Bernoulli process X :

$$Z_i = 2(X_i - \frac{1}{2})$$



and then the random-walk process W is related to the binomial counting process Y by

$$W_n = 2Y_n - n$$

- Mean: $m_W(n) = n(2p - 1)$
- Autocovariance: $K_W(n_1, n_2) = 4p(1-p) \min\{n_1, n_2\}$
- The underlying Bernoulli process Z can be recovered from the random walk W by differencing:

$$Z_n = W_n - W_{n-1}$$

- The random walk is called an *independent-increment process*

Example 4: Random-amplitude sine wave random process: a continuous-time process:

$$X(t) = A \sin(\omega_o t + \theta),$$

for which ω_o and θ are non-random

- Mean:

$$m_X(t) = m_A \sin(\omega_o t + \theta),$$

- Autocovariance:

$$K_X(t_1, t_2) = \mathbb{E}\{A^2\} \sin(\omega_o t_1 + \theta)(\omega_o t_2 + \theta)$$

Exercise: compute the mean $m_Y(t)$ and autocovariance $R_Y(t_1, t_2)$ of a random-amplitude-and-phase sine wave process

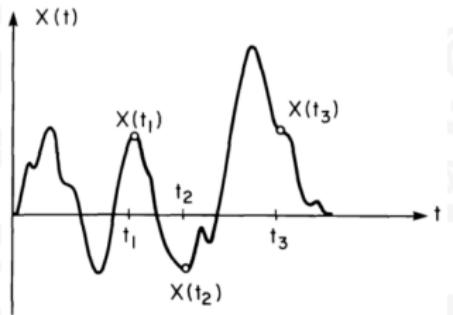
$$Y(t) = A \sin(\omega_o t + \theta),$$

where the random θ is independent of the random amplitude A and is uniformly distributed on the interval $[-\pi, \pi)$

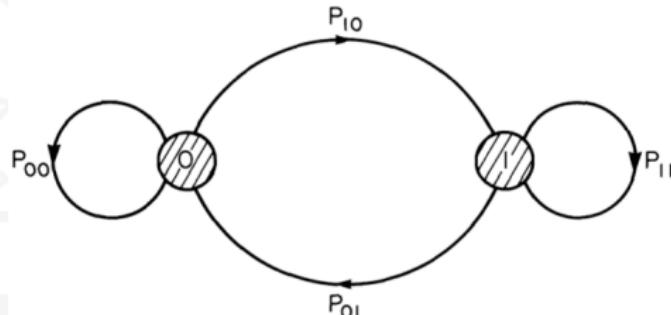
Example 5: Markov process: stochastic process that satisfies the Markov property

- The Markov property: if one can make predictions for the future of the process based solely on its present state just as well as one could knowing the process's full history
- Conditional on the present state of the system, its future and past are independent
- A Markov process can be thought of as 'memoryless' process
- The specific values x_1, x_2, x_3, \dots that can be taken on by the discrete random variables in a Markov chain are called *states*

Chapman-Kolmogorov



Binary digital system



Example 5a of Markov process: Gambling:

- Suppose that you start with \$10, and you wager \$1 on an unending, fair, coin toss indefinitely, or until you lose all of your money.
- If X_n represents the number of dollars you have after n tosses, with $X_0 = 10$, then the sequence $\{X_n : n \in [0, \infty)\}$ is a Markov process.
- If I know that you have \$12 now, then it would be expected that with even odds, you will either have \$11 or \$13 after the next toss.
- This guess is not improved by the added knowledge that you started with \$10, then went up to \$11, down to \$10, up to \$11, and then to \$12.
- The process described here is a Markov chain on a countable state space that follows a random walk

Example 5b of Markov process: A birth-death process:

- Suppose that you are popping one hundred kernels of popcorn, and each kernel will pop at an independent, exponentially-distributed time.
- Let X_t denote the number of kernels which have popped up to time t . Then this is a continuous-time Markov process.
- If after some amount of time, I want to guess how many kernels will pop in the next second, I need only to know how many kernels have popped so far.
- It will not help me to know when they popped, so knowing X_t for previous times t will not inform my guess.
- The process described here is an approximation of a Poisson process – Poisson processes are also Markov.

Example 5c of Markov process: Dynamical systems:

- Markov processes are present in the finite-order linear discrete-time systems described by the difference equation:

$$X_{n+1} = a_n X_n + Z_n,$$

where $\{Z_n\}$ is the excitation sequence, $\{X_n\}$ is the sequence of system states, and $\{a_n\}$ models the internal feedback.

- Future state X_{n+1} depends on only the current state, past states are irrelevant.
- Thus, if excitation $\{Z_n\}$ has no memory (i.e. sequence of independent random variables), then the sequence of states is a Markov process.
- In continuous-time dynamical systems happens the same:

$$\frac{dX}{dt} = a(t)X(t) + Z(t) \rightarrow dX(t) = a(t)X(t)dt + \underbrace{Z(t)dt}_{dW(t)},$$

where $dX(t)$ depends on only the current state $X(t)$ and if the increment $dW(t)$ is independent of the past, $X(t)$ is a Markov process

Example 6: Wiener process: is a continuous-time stochastic process, aka *Brownian motion process* very useful to model motion in gases and liquids, thermal noise in electrical conductors and various diffusions

Three conditions:

- The initial position is zero: $W(0) = 0$
- The mean is zero: $\mathbb{E}\{W(t)\} = 0$
- The increments of $W(t)$ are independent, stationary and Gaussian

- Unconditional probability density function:

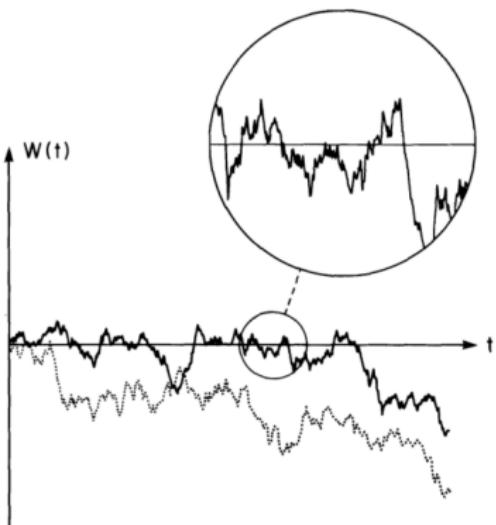
$$f_{W_t}(x, t) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}}$$

- The expectation is zero: $\mathbb{E}[W_t] = 0$.
- The variance is t :

$$\mathbb{V}(W_t) = \mathbb{E}[W_t^2] - \mathbb{E}^2[W_t] = \mathbb{E}[W_t^2] = t$$

- Covariance: $K(W_s, W_t) = \min(s, t)$
- Correlation:

$$R(W_s, W_t) = \frac{K(W_s, W_t)}{\sigma_{W_s} \sigma_{W_t}} = \sqrt{\frac{\min(s, t)}{\max(s, t)}}$$



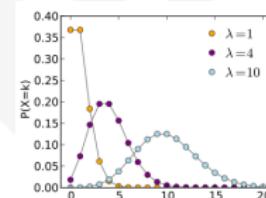
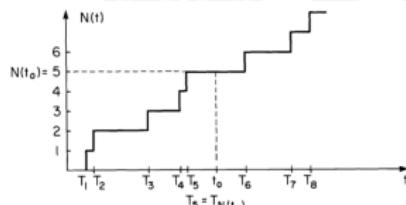
Example 7: Poisson process: stochastic process that counts the number of events and the time points at which these events occur in a given time interval.

- Good model of radioactive decay, shot noise in electronic devices, photon detection, telephone calls, and document retrieval
- Place at random m points in $[0, T]$, seek the probability $P_t(n)$ of the event that $n \leq m$ lie in the subinterval $[0, t]$, $t < T$
- Binomial distribution:

$$P_t(n) = \frac{m!}{m!(m-n)!} p^n (1-p)^{m-n}, \quad p = t/T : \text{prob. success}$$

- The Poisson theorem ($n \sim mp$, $\lim_{m \rightarrow \infty} (n/m) = p$):

$$P_t(n) \approx \frac{(mp)^n}{n!} e^{-mp} = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$



- Consequences: (1) number of points in two disjoint intervals are statistically independent; (2) probability that n_1 points lie in $[\tau_1, \tau_1 + t_1]$ and n_2 points lie in $[\tau_2, \tau_1 + t_2]$ is $P_{t_1}(n_1)P_{t_2}(n_2)$

Example 7a of a Poisson process: Shot noise

- Consider a vacuum-tube diode in which electrons emitted from the heated cathode are attracted to the anode
- Let the electron emission rate be temperature-limited
- Emission times is well modelled by a Poisson process
- The current through the diode resulting from these emissions:

$$X(t) = \sum_{i=1}^N (t) h(t - T_i), \quad t \geq 0,$$

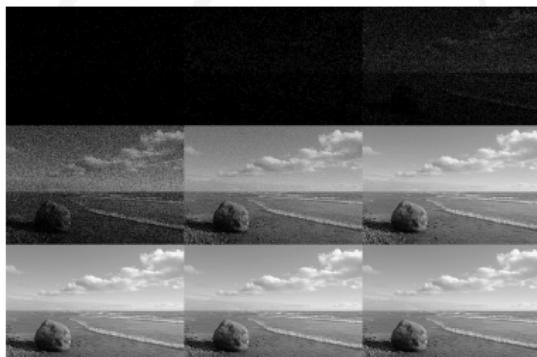
where $t = 0$ is the time at which the diode is energized, $N(t)$ is the number of emissions during $[0, t]$, $\{T_i\}$ are the emission times, and the form of the pulse h is a function of the cathode-anode geometry, temperature and voltage.

- Shot noise occurs also in other electronic devices: generation recombination noise in semiconductors, emission noise in PN devices, microwave tube noise, etc.

Example 7b of a Poisson process: Photon detection

- Photon counting in optical devices, where shot noise is associated with the particle nature of light
- For large numbers the Poisson distribution approaches a normal distribution
- Since the standard deviation of shot noise is equal to the square root of the average number of events N , the signal-to-noise ratio (SNR) is given by:

$$\text{SNR} = \frac{N}{\sqrt{N}} = \sqrt{N}$$



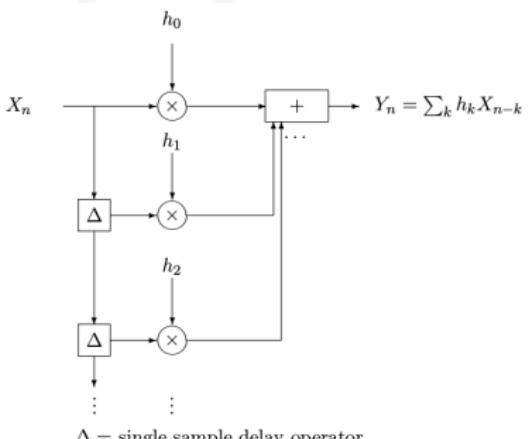
- Thus when N is very large, the SNR is very large as well, and any relative fluctuations in N due to other sources are more likely to dominate over shot noise
- However when the other noise source is at a fixed level, such as thermal noise, increasing N (the DC current or light level, etc.) can sometimes lead to dominance of shot noise

Example 8: MA: Moving average process:

- Many complicated random processes are well modeled as a linear operation on a simple process
- For example, a complicated process with memory might be constructed by passing a simple iid process through a linear filter
- If X_n inputs a linear system described by a convolution, there is a δ -response h_k such that the output process Y_n is given by

$$Y_n = \sum_k X_{n-k} h_k$$

- A linear filter like this is called a moving-average filter since the output is a weighted running average of the inputs



- If only a finite number of the h_k are not zero, then the filter is called a finite-order moving-average filter (or an FIR filter, for "finite impulse response")
- The order of the filter is equal to the maximum minus the minimum value of k for which the h_k are nonzero. For example, if $Y_n = X_n + X_{n-1}$, we have a first-order moving-average filter
- The associated transfer function is stable, "all zeros" filter

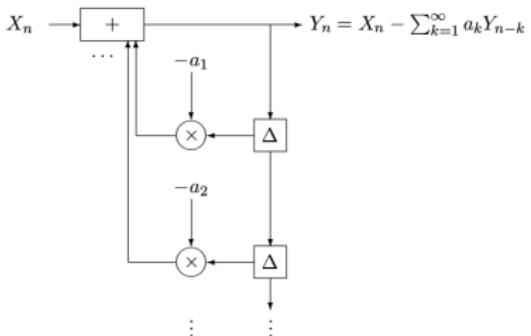
Example 9: AR: Autorregressive process:

- Another form of difference equation describing a linear system is obtained by convolving the outputs to get the inputs instead of vice versa
 - For example, the output process may satisfy a difference equation of the form

$$X_n = \sum_k a_k Y_{n-k}$$

- For convenience it is usually assumed that $a_0 = 1$ and $a_k = 0$ for negative k and hence that the equation can be expressed as

$$Y_n = X_n - \sum_{k=1}^{\infty} a_k Y_{n-k}$$

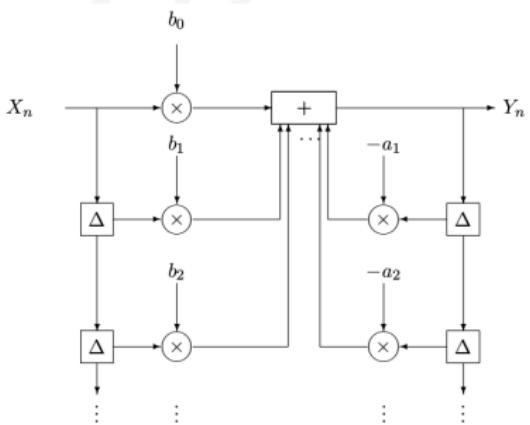


- The numbers $\{a_k\}$ are the regression coefficients, and the filter is called *auto-regressive* (or an IIR filter, for “infinite impulse response”)
 - The order of the filter is equal to the maximum minus the minimum value of k for which the a_k are nonzero
 - The associated transfer function may not be stable, “all poles” filter

Example 10: ARMA: Autorregressive and moving average process:

- Combination of AR + MA
- ARMA filters are said to be finite-order if only a finite number of the a_k 's and b_k 's are not zero
- The output process may satisfy a (finite) difference equation of the form

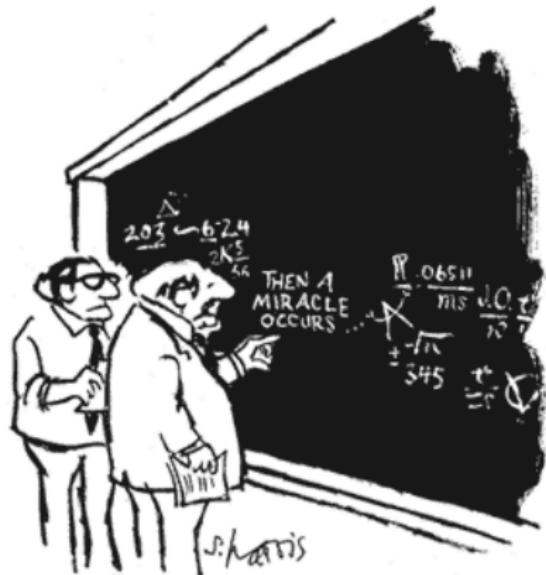
$$Y_n = \sum_{k=1}^P a_k Y_{n-k} + \sum_{k=0}^Q b_k X_{n-k}$$



- One can often describe a linear system by any of these filters, and hence one often chooses the simplest model for the desired application
- Occam's razor and parsimonious
- An ARMA filter representation with only three nonzero a_k and two nonzero b_k would be simpler than either a pure AR or pure MA representation, which would in general require an infinite number of parameters

Reviewed:

Random variables, continuous/discrete process, SNR, MSE, PE, duality between probability models of ensembles of waveforms/sequences and random processes, applications in signal processing (interpolation, signal detection, extraction, prediction), examples of processes (Bernoulli, Binomial, random walk, Markov, Wiener, Poisson, AR/MA/ARMA), types of random processes, mean, autocorrelation, autocovariance, cross-correlation, cross-covariance, stationarity, WSS, Ergodicity, etc.



"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

Part 3: Spectral estimation

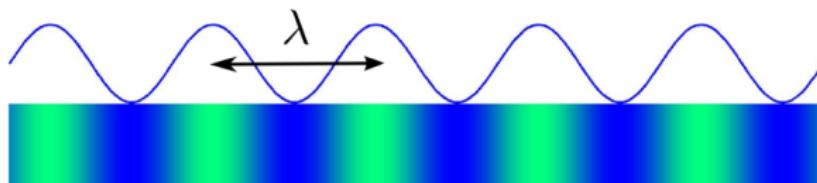
Basic Physics: travelling waves

- Travelling waves are efficient information carriers
- Examples: electromagnetic, acoustic (pressure waves), seismic (shear waves), optical (light)
- When do we use waves and need frequency domain representations:
 - wireless communications
 - audio, music
 - imaging: radar, sonar, seismics



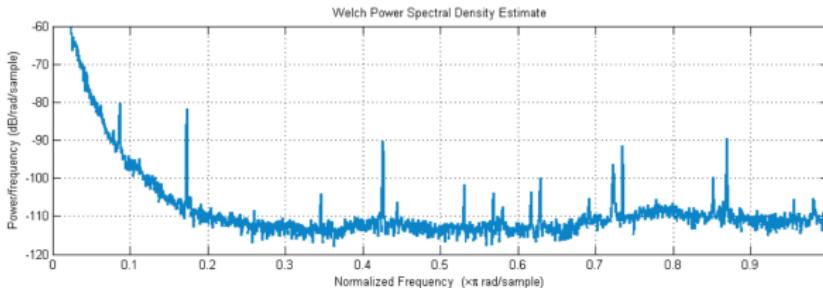
Waves and frequency representation

- The waves can be described by their frequency coverage
- Parameters to characterise the waves are:
 - Wave period T [s]
 - Frequency $f = 1/T$ [Hz]
 - Angular frequency $\omega = 2\pi f$ [rad/s]
 - Wavelength $\lambda = c/f$ [m]
 - Wavenumber $k = 2\pi/\lambda$ [1/m]
 - Phase velocity c [m/s]



Applications of spectral estimation

- Vibration analysis, resonance characterisation, harmonic analysis
- Signal analysis: Classify signals: NB, BB, LP, HP, non-stationary...
- System identification: Identify LTI system transfer functions
- Linear prediction, filtering, detection: Spectrum determines optimum methods
- Signal compression, audio/video, voice encoding/decoding
- Beamforming/Direction finding/imaging



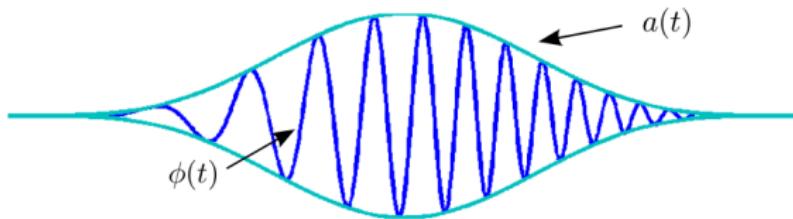
Waves and frequency representation

- Narrow band signal description

$$s(t) = a(t)\cos(\omega_o t + \phi(t)) = \Re\{a(t)e^{j(\omega_o t + \phi(t))}\}$$

where $\omega_o = 2\pi f_o$ is called the center frequency

- $a(t)$ and $\phi(t)$ are often assumed as slowly varying (compared to the wave period)



The Fourier transform

- For a deterministic sequence $x_T(t)$, the Fourier transform is defined as

$$\mathcal{F}[X_T(t)] = \tilde{X}_T(f) = \tilde{X}_T(\omega) = \int_{-\infty}^{\infty} X_T(t) e^{-j2\pi ft} dt$$

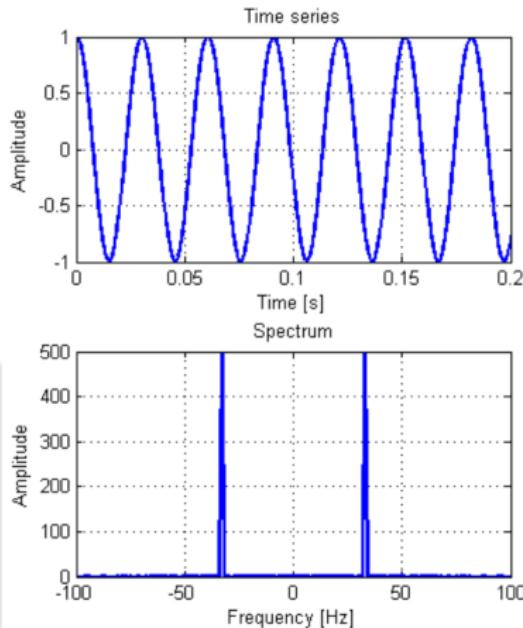
- The Fourier transform is simply called the *spectrum*
- $\omega = 2\pi f$ is understood as angular frequency (if t is time)
- The inverse Fourier transform

$$X_T(f) := X_T(\omega) = \int_{-\infty}^{\infty} \tilde{X}_T(t) e^{+j2\pi ft} df$$

The Fourier transform of a deterministic signal

- The frequency coverage is related to the sampling interval as $f_{tot} = 1/\delta t$
- The frequency discretization is related to the time series length as $\delta f = 1/t_{tot} = 1/N\delta t$

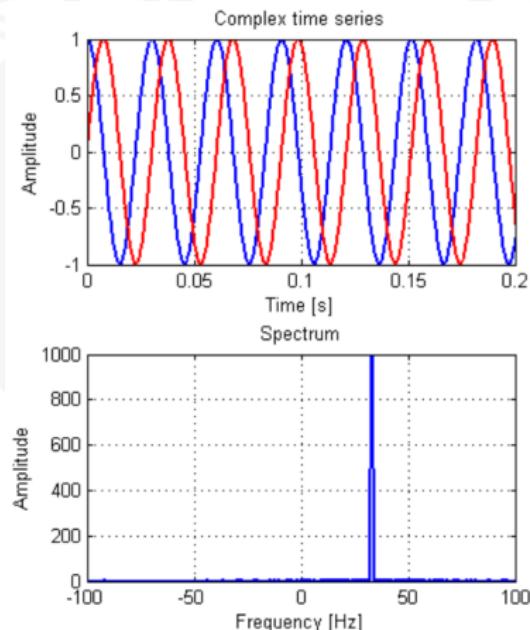
```
dt = 0.001;
n = 1000;
t = [0:n-1] * dt; f = 33;
st = cos( 2*pi*f*t );
Sw = abs(fftshift(fft(st)));
faxe = [-n/2:n/2-1]/n / dt;
figure, plot(t,st)
figure, plot(faxe,Sw)
```



The Fourier transform of a deterministic **complex** signal

- The Fourier transform of a real sequence results in a real symmetric spectrum
- The Fourier transform of a complex sequence results in a complex unsymmetric spectrum

```
dt = 0.001;
n = 1000;
t = [0:n-1] * dt; f = 33;
st = exp( j*2*pi*f*t );
Sw = abs(fftshift(fft(st)));
faxe = [-n/2:n/2-1]/n / dt;
figure, plot(t,st)
figure, plot(faxe,Sw)
```



The power spectrum

- A random process is an ensemble of discrete-time signals
- Assume that the random process is Wide Sense Stationary
- The autocorrelation of a WSS random process is a deterministic function of delay (only)
- The Fourier transform of the autocorrelation function is the power spectrum or the power spectral density (Einstein-Wiener-Khintchine)

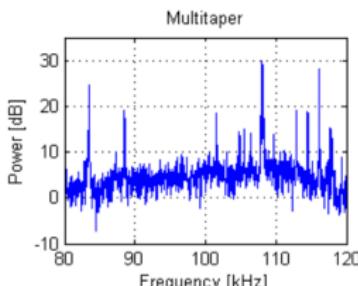
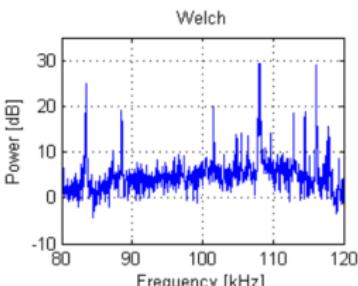
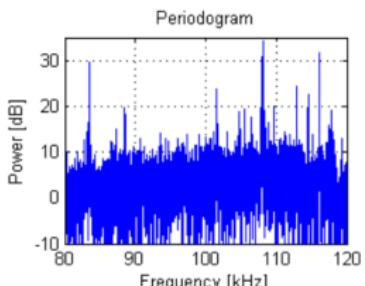
$$P_X(\omega) = \int_{-\infty}^{\infty} R_X(\tau) e^{-j\omega\tau} d\tau$$

$$R_X(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} P_X(\omega) e^{j\omega\tau} d\omega$$

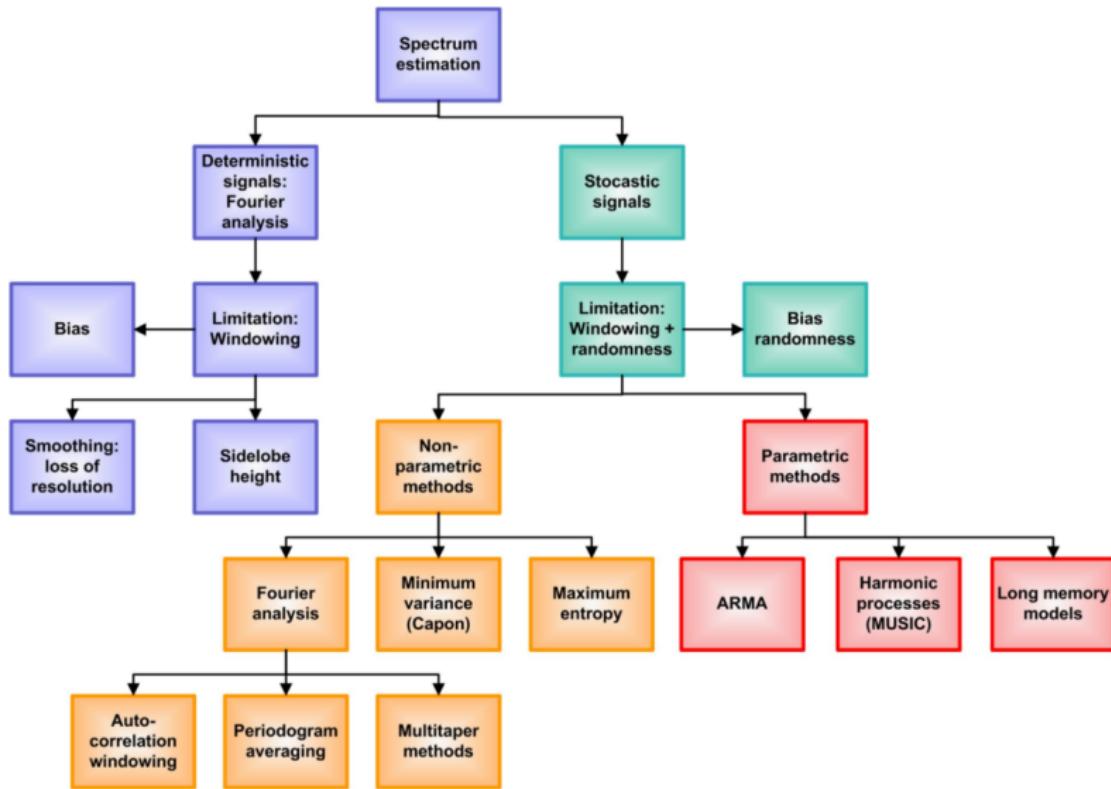
Spectral estimation

- PSD estimation \approx XCORR estimation
- The basic method: the periodogram!
- Performance measures: Bias, variance, spectral resolution
- Many methods that improve performance:

>> pburg	PSD using Burg method
>> pcov	PSD using covariance method
>> peig	Pseudospectrum using eigenvector method
>> periodogram	PSD using periodogram
>> pmcov	PSD using modified covariance method
>> pmtm	PSD using multitaper method (MTM)
>> pmusic	Pseudospectrum using MUSIC algorithm
>> pwelch	PSD using Welch's method
>> pyulear	PSD using Yule-Walker AR method



Overview of spectrum estimation techniques



The periodogram

- The periodogram is simply the discrete Fourier transform of the biased estimator of the autocorrelation sequence

$$P_X(\omega) = \sum_{k=-N+1}^{N-1} \hat{R}_X(k) e^{-j\omega k}$$

- We introduce a window function and rewrite the autocorrelation

$$x_N(n) = w_R(n)x(n)$$

with the window function

$$w_R(n) = \begin{cases} 1 & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases}$$

- The autocorrelation sequence becomes then

$$\hat{R}_X(k) = \frac{1}{N} \sum_{n=-\infty}^{\infty} x_N(n+k)x_N^*(n) = \frac{1}{N} x_N(k) * x_N^*(-k)$$

The periodogram – alternative form

- Taking the Fourier transform and applying the convolution theorem ("convolution becomes multiplication in the other domain"), the periodogram becomes

$$P_X(\omega) = \frac{1}{N} X_N(\omega) * X_N^*(\omega) = \frac{1}{N} |X_N(\omega)|^2,$$

where

$$X_N(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n} = \sum_{n=0}^{N-1} x(n) e^{-j\omega n}$$

is the discrete Fourier transform of the random sequence

- Note the difference between the two different spectral estimates: One has $2N - 1$ output frequencies, the other has N output frequencies

Spectral estimation: which ACF estimator to use?

Estimating the power spectral density (PSD) is equivalent to estimating the autocorrelation function (ACF).

→ Which estimator do we choose?

- The asymptotically unbiased (but still biased) estimator of the autocorrelation is:

$$\hat{R}_x^b(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} x_N(n+k)x_N^*(n)$$

- Why not choose the unbiased estimator (superscript u)

$$\hat{R}_x^u(k) = \frac{1}{N} \sum_{n=0}^{N-1-|k|} x_N(n+|k|)x_N^*(n), \quad |k| < N$$

Spectral estimation: which ACF estimator to use?

For many stationary random processes of practical interest, the mean square error (MSE) is

$$\text{MSE}(\hat{R}_X^b(k)) := \mathbb{E}\{(\hat{R}_X^b - \hat{R}_X)^2\} < \mathbb{E}\{(\hat{R}_X^u - \hat{R}_X)^2\} := \text{MSE}(\hat{R}_X^u(k))$$

- MSE is a quality measure – Low MSE is good
- We recall that the MSE is related to the variance and bias as

$$\text{MSE}(\hat{\theta}) = \underbrace{\mathbb{V}(\hat{\theta})}_{\text{variance}} + \underbrace{(\mathbb{E}(\hat{\theta}))^2}_{\text{bias}^2}$$

- This means that reducing the bias increases the variance for a given mean square error — bias-variance dilemma

Spectral estimation: which ACF estimator to use?

- **Variance:** Consider maximum lag $k = N - 1$, then

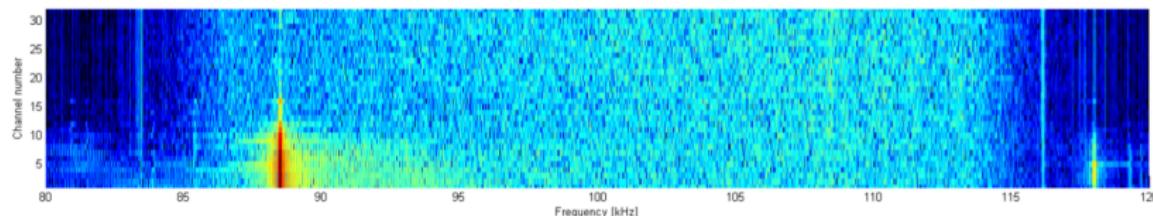
$$\hat{R}_X^b(N-1) = \frac{1}{N} x(N-1)x^*(0)$$

while

$$\hat{R}_X^u(N-1) = x(N-1)x^*(0)$$

for which we see that the variance of $\hat{R}_X^u(N-1)$ is N^2 times larger than the variance of $\hat{R}_X^b(N-1)$

- Is all about variance of the estimator?
- The performance of a spectral estimator can be characterised by several different measures:
 - Bias and spectral leakage
 - Frequency resolution
 - Variance



Bias and spectral leakage

- The periodogram is the Fourier transform of the estimated autocorrelation sequence

$$P_X(\omega) = \sum_{k=-N+1}^{N-1} \hat{R}_X(k) e^{-j\omega k}$$

where

$$\hat{R}_X^u(k) = \frac{1}{N} \sum_{n=0}^{N-1-|k|} x(n+|k|)x^*(n)$$

- Even this unbiased estimator is biased!

$$\mathbb{E}\{\hat{R}_X^u(k)\} = \frac{1}{N} \sum_{n=0}^{N-1-|k|} \mathbb{E}\{x(n+|k|)x^*(n)\} = \frac{N-|k|}{N} \mathbb{E}\{\hat{R}_X(k)\}$$

- The bias reduces with N

Bias and spectral leakage

- We write this as

$$\mathbb{E}\{\hat{R}_X^u(k)\} = w_B(k)R_X(k)$$

where $w_B(k)$ is called a Bartlett window (triangular shape)

$$w_B(n) = \begin{cases} \frac{N-|k|}{N} & |k| \leq N \\ 0 & |k| > N \end{cases}$$

- Using the convolution theorem, this becomes

$$\mathbb{E}\{\hat{P}_X(\omega)\} = \frac{1}{2\pi} W_B(\omega) * P_X(\omega)$$

where W_B is the Fourier transform of w_B , which is a sinc squared

$$W_B(\omega) = \left(\frac{\sin(N\omega/2)}{N \sin(\omega/2)} \right)^2$$

Bias and spectral leakage

- The expected value of the periodogram is the true power spectrum convolved with a sinc squared
- The periodogram is a biased spectral estimator
- It is however, asymptotically unbiased since

$$\lim_{N \rightarrow \infty} \mathbb{E}\{\hat{P}_X(\omega)\} = P_X(\omega)$$

- This does not mean that everything's fine!

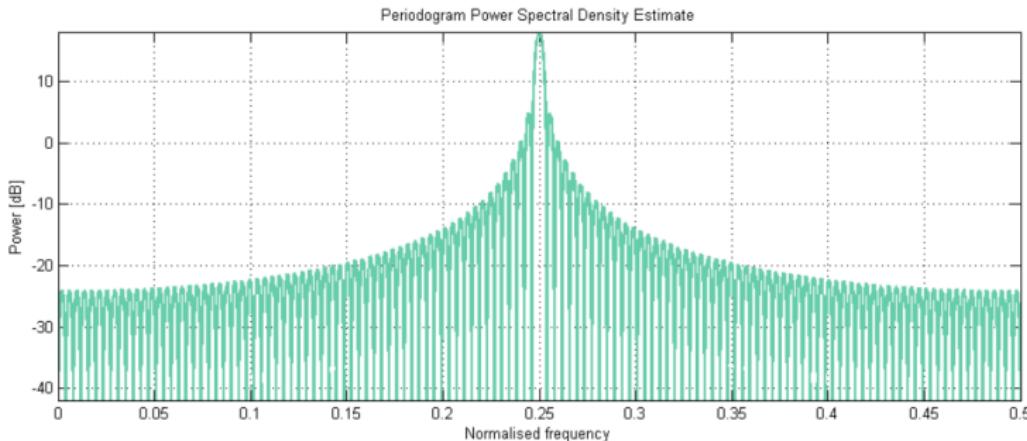
"... for processes with spectra typical of those encountered in engineering, the sample size must be extraordinarily large for the periodogram to be reasonable unbiased.

(1982)

Thomson

Bias example: single sinusoid without noise (deterministic)

- The very slowly fall-off of the sinc-pattern causes the bias
- This is also referred to as spectral leakage
- Example: Autoregressive Moving Average (ARMA) process Spectral leakage is especially evident when data records are short



Bias example: random signal

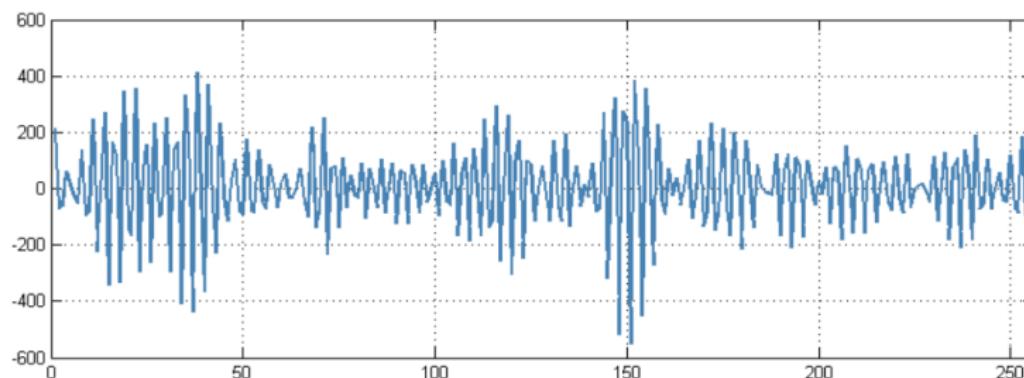
- Remember PDS: An ARMA process may be generated by filtering white noise with a linear shift-invariant filter that has a rational system function
- Algorithm: construct b and a coefficients
- Theoretical spectrum is then calculated by using MATLAB freqz
- ARMA model:

```
NZ = 1024;  
b = poly( [-0.8, 0.97*exp(j*pi/4), 0.97*exp(-j*pi/4), ...  
0.97*exp(j*pi/6), 0.97*exp(-j*pi/6) ] );  
a = poly( [ 0.8, 0.95*exp(j*3*pi/4), 0.95*exp(-j*3*pi/4), ...  
0.95*exp(j*2.5*pi/4), 0.95*exp(-j*2.5*pi/4) ] );  
b = b*sum(a)/sum(b);  
[h,faxe_m] = freqz(b,a,NZ);  
faxe_m = faxe_m / (2*pi);  
P_model = abs(h).2;  
n=1024; % Data size  
M=100; % Leading size (transient)  
w=randn(M+n,1); % WGN sequence  
x=filter(b,a,w); % Apply filter on WGN sequence  
x=x(M+1:M+n); % remove transient
```

Bias example: random signal

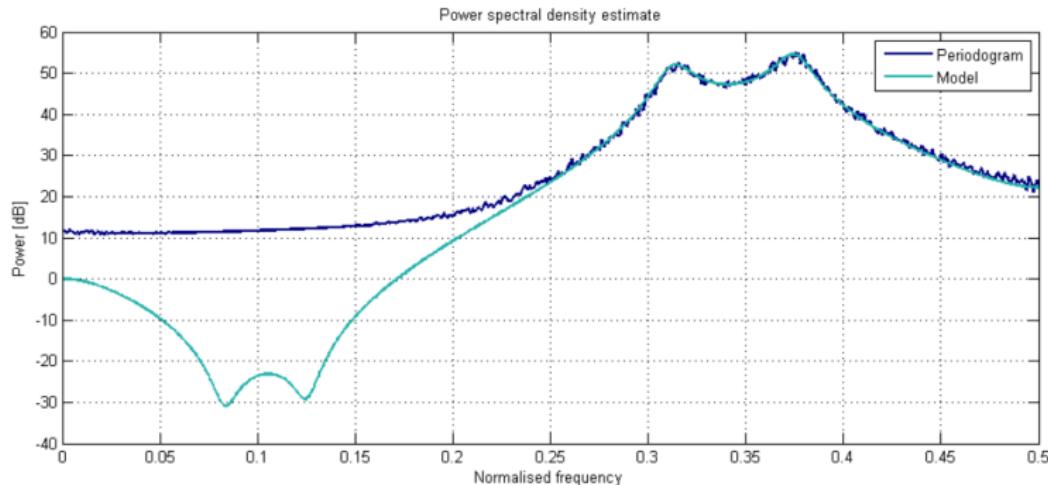
- Now we filter a WGN throughout the filter

```
n=1024; % Data size  
M=100; % Leading size (transient)  
w=randn(M+n,1); % WGN sequence  
x=filter(b,a,w); % Apply filter on WGN sequence  
x=x(M+1:M+n); % remove transient
```



Bias example: random signal

- We want to investigate estimator bias which implies that we must suppress estimator variance
- We average 40 realizations to reduce the variance



Window size and resolution

- The periodogram is based on the autocorrelation sequence

$$\hat{R}_X(k) = \frac{1}{N} \sum_{n=-\infty}^{\infty} x_N(n+k)x_N^*(n) = \frac{1}{N} x_N(k) * x_N^*(-k)$$

where $x_N(n) = w_R(n)x(n)$ and $w_R = 0$ outside the data interval

- This is in effect applying a rectangular window on the data
- In Fourier domain (using the convolution theorem)

$$X_N(\omega) = X(\omega)W_R(\omega)$$

- The Fourier transform of a rectangular window is a sinc

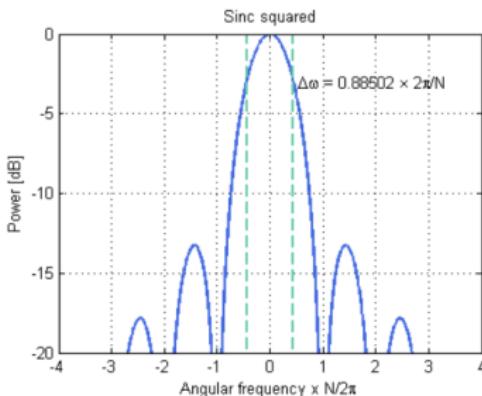
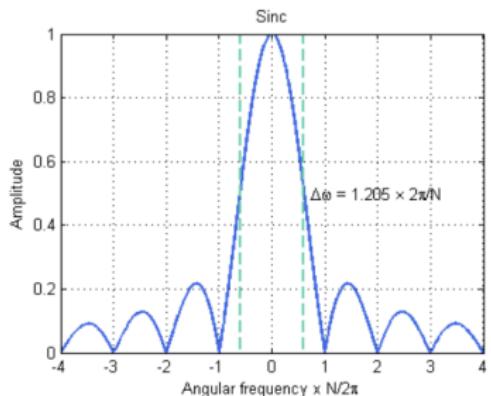
$$W_R(\omega) = \frac{\sin(N\omega/2)}{N \sin(\omega/2)}$$

Window size and resolution: single sinusoid without noise (deterministic)

- The frequency resolution is the smallest distance two different signals are displaced (in frequency domain) and still resolved
- This is related to the main-lobe width (simple to approximate)

$$\Delta\omega \approx \frac{2\pi}{N} \leftrightarrow \Delta f \approx \frac{f_s}{N}$$

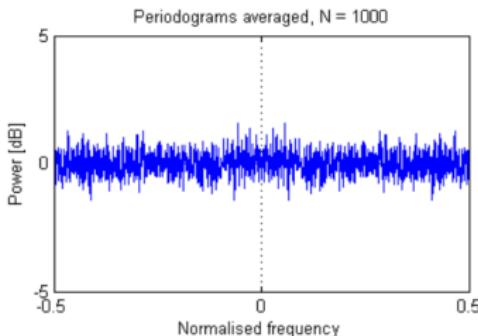
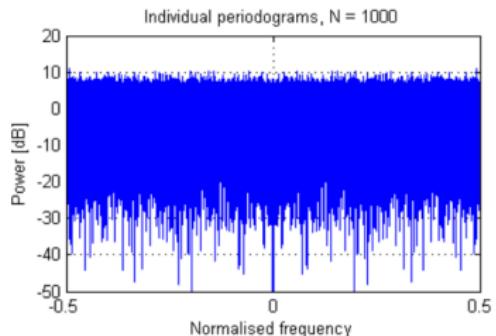
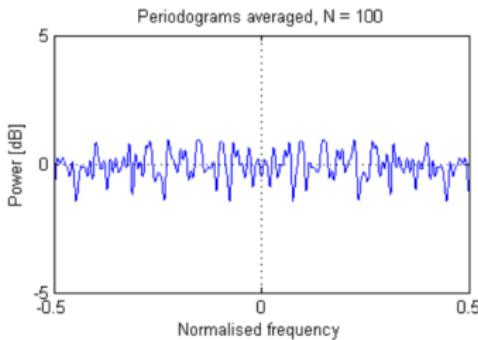
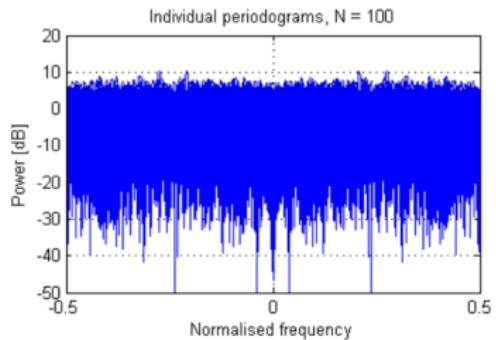
where f_s is the sampling frequency and N is the data window size



Variance

- The variance of the periodogram spectral estimator is (after rather complicated calculations)
- The variance does not approach zero as the data length N increases
- The periodogram is not a *consistent estimator* (i.e. converges in some sense to the true value)
- Why does the variance not decrease with increasing N ?
- Increasing N means increasing the number of individual frequencies (instead of increasing the accuracy of each frequency)

Periodogram variance WGN with $N = 100$ and $N = 1000$, and 100 realizations



Classical spectral estimation

- The periodogram spectral estimator suffers from bias and variance
- Classical spectral estimation is all about improving Fourier based spectral estimation techniques
- Three different approaches to improve performance:
 - ➊ Bias reduction by 'tapering': The modified periodogram
 - ➋ Variance reduction by 'smoothing' (averaging):
 - Welch-Bartlett method
 - Blackman-Tukey method
 - ➌ Bias-Variance reduction by 'smoothing+averaging': the multitapping

Method 1: The modified (windowed) periodogram

- The main contributor to bias is the sinc-pattern caused by the rectangular (on the data) / triangular (on the ACF) window
- We can reduce the bias by applying another window function:

$$\hat{P}_X^{(modified)}(\omega) = \frac{1}{NU} |X_N(\omega)|^2 = \frac{1}{NU} \left| \sum_{n=-\infty}^{\infty} x(n)w(n)e^{-j\omega n} \right|^2$$

where

$$U = \frac{1}{NU} \sum_{n=0}^{N-1} |w(n)|^2$$

is a factor to ensure that $\hat{P}_X(\omega)$ is asymptotically unbiased

Method 1: The modified (windowed) periodogram

- Following the previous calculations, we find that the bias becomes

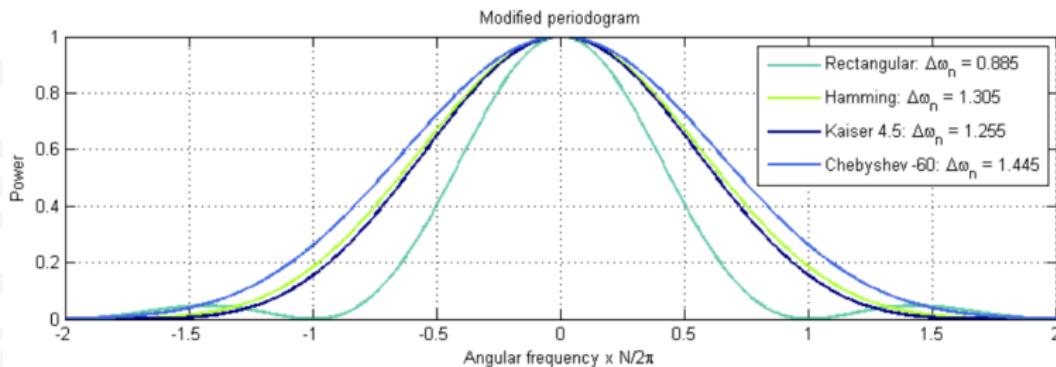
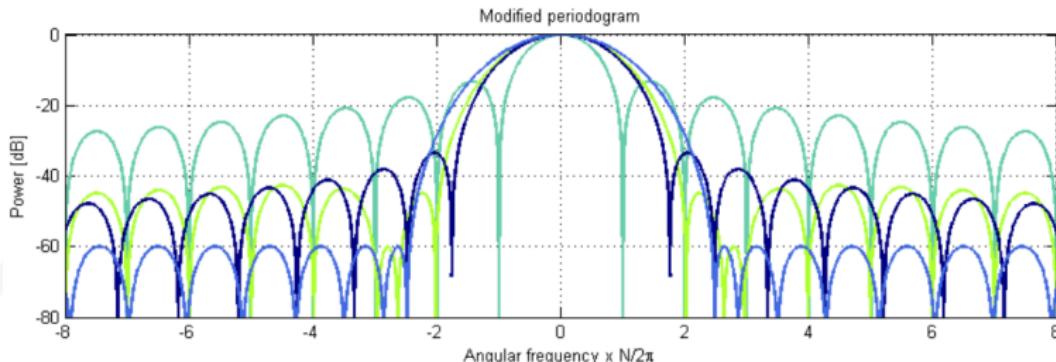
$$\mathbb{E}\{\hat{P}_X(\omega)\} = \frac{1}{2\pi NU} P_X(\omega) * |W(\omega)|^2$$

- The choice of window provides a trade-off between bias and spectral resolution
- The choice of window does not affect the estimator variance
- The modified periodogram is (also) not a consistent estimate of the power spectrum
- The window is characterised by the main-lobe 3dB-width, peak sidelobe level and integrated sidelobe level
- Windowing is also called “*tapering*”

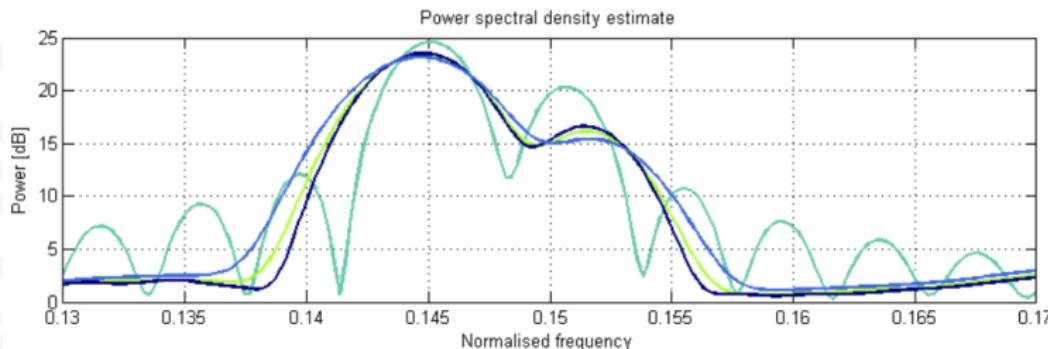
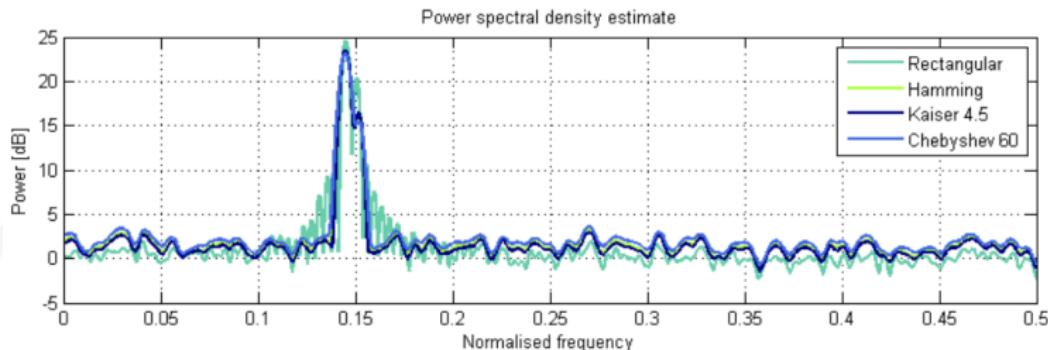
Method 1: Examples of windows (tapers)

- There are a lot of windows with different characteristics
 - Hamming: easy to implement - decent performance
 - Kaiser: optimised to minimise energy outside mainlobe. Parameter choice to trade resolution vs sidelobe suppression. Medium difficulty in implementation
 - Chebyshev: optimised to control peak sidelobe level. Parameter choice gives directly (flat) sidelobe level. Difficult to implement
- See also `wvtool` for visualization of different windows

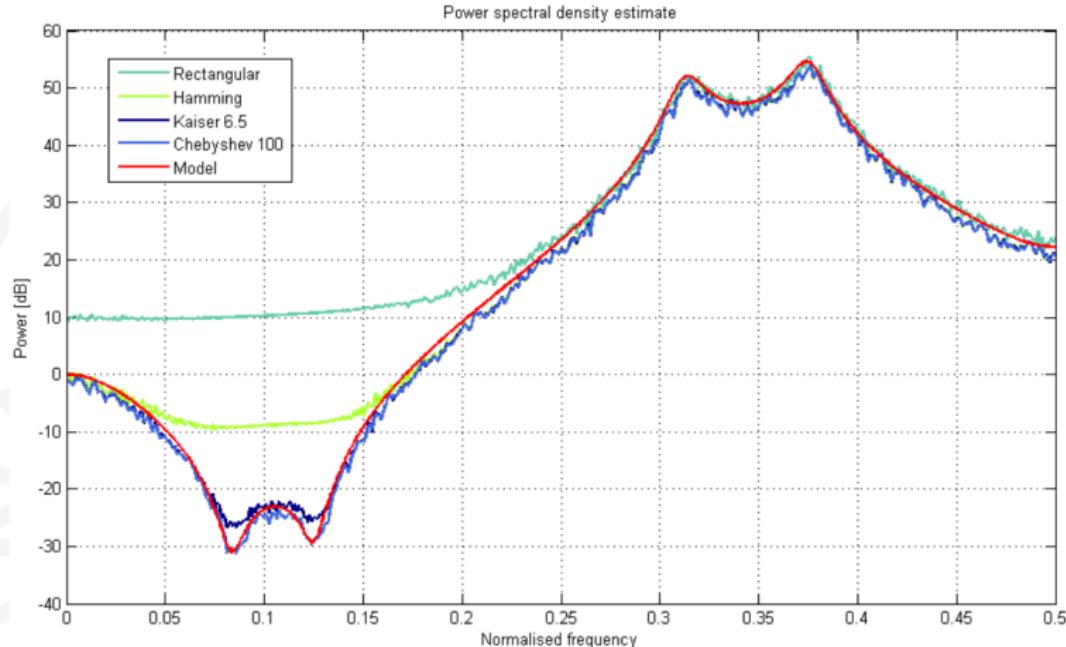
Method 1: Examples of windows (tapers)



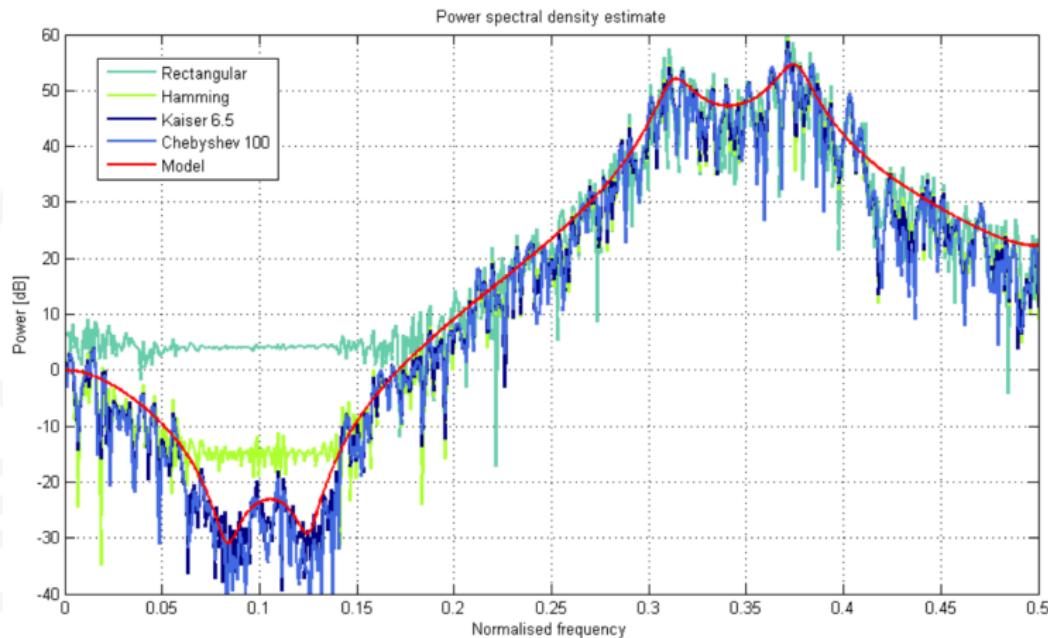
Method 1: Examples of windows (tapers) Resolution test: Two closely spaced sinusoids at $f = [0.145, 0.150]$ in WGN



Method 1: Examples of windows (tapers) Bias test: Worst case ARMA process (40 realizations averaged)



Method 1: Examples of windows (tapers) We still have problems with the variance (single realization)



Method 2: Reducing variance with periodogram averaging

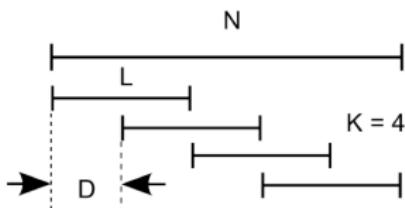
- A simple straight forward approach to reduce variance is as follows:
 - ➊ Divide the sequence into segments
 - ➋ Estimate the PSD of each segment
 - ➌ Average (incoherently) the estimates to reduce variance
- The technique obviously reduces the spectral resolution since each PSD estimate uses fewer samples
- The reduction in variance is obviously related to the number of estimates averaged
- We are going to review two classical methods:
 - Bartlett's method: non-overlapping periodograms
 - Welch's method: overlapping modified periodograms

Method 2: Welch's method to reduce variance via averaging

- Divide the total sequence of N samples into segments of size L , offset each segment by D points into a total of K segments such that

$$N = L + D(K - 1)$$

- See MATLAB's command `>> buffer, tapdelay`



- The estimator is defined as

$$\hat{P}_X^{(welch)}(\omega) = \frac{1}{KLU} \sum_{i=0}^{K-1} \left| \sum_{n=0}^{L-1} x(n + iD) w(n) e^{-j\omega n} \right|^2 = \frac{1}{K} \sum_{i=0}^{K-1} \hat{P}_X^{(m,i)}(\omega)$$

where

$$U = \frac{1}{NU} \sum_{n=0}^{N-1} |w(n)|^2$$

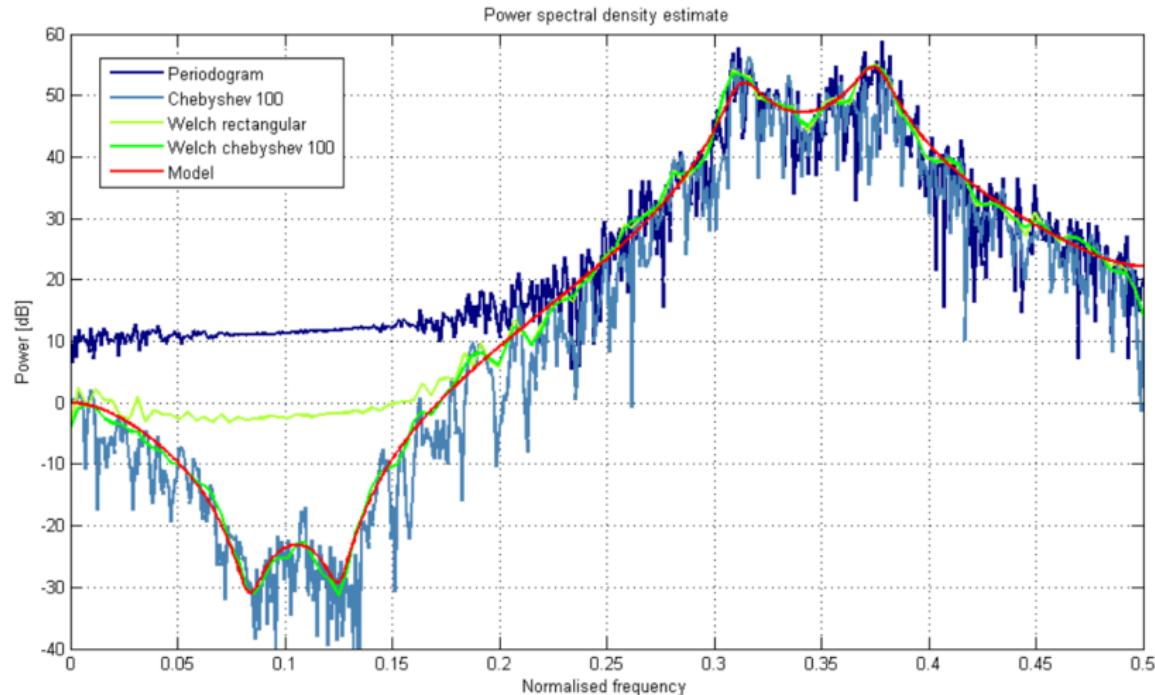
where $w(n)$ is a window of choice and $\hat{P}_X^{(m,i)}(\omega)$ is the i -th modified periodogram

Method 2: Welch's method, properties

- The bias and resolution of the estimator is given by the modified periodogram that is applied on each segment
- The variance becomes dependent on the overlap, window type and number of segments
- For 50% overlap and a Bartlett window

$$\mathbb{V}\{\hat{P}_X^{(welch)}(\omega)\} \approx \frac{9L}{16N} P_X^2(\omega)$$

- The variance decreases with increasing N
- The estimator is consistent
- The estimator is asymptotically unbiased, since the modified periodogram estimator is asymptotically unbiased

Method 2: Welch's method Single realization of ARMA process

Method 2: Blackman-Tukey method of periodogram smoothing

- We realise that the variance in the autocorrelation estimate increases with increasing absolute lag
- We apply a window on the ACF to suppress the elements that contribute to the variance

$$\hat{P}_X^{(Blackman-Tukey)}(\omega) = \sum_{k=-M}^M \hat{R}_X(k)w(k)e^{-j\omega k}, \quad M \leq N - 1$$

- Again, the convolution theorem states that the PSD becomes

$$\hat{P}_X^{(Blackman-Tukey)}(\omega) = \frac{1}{2\pi} \hat{P}_X^{(b)}(\omega) * W(\omega)$$

where W is the Fourier transform of w

- Similar to the modified periodogram but different results

Method 2: Blackman-Tukey method of periodogram smoothing

- The bias of the Blackman-Tukey spectral estimator is

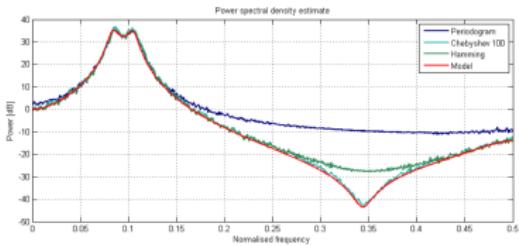
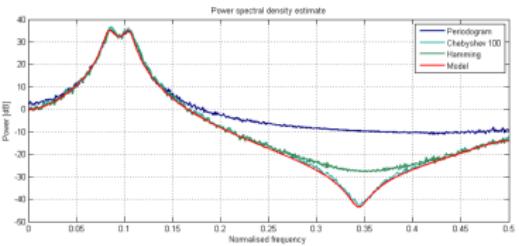
$$\mathbb{E}\{\hat{P}_X^{(\text{Blackman-Tukey})}(\omega)\} = \frac{1}{2\pi} \mathbb{E}\{\hat{P}_X^{(b)}(\omega)\} * W(\omega)$$

- After some maths, we find the variance (see Hayes) to be

$$\mathbb{V}\{\hat{P}_X^{(\text{Blackman-Tukey})}(\omega)\} \approx P_X^2(\omega) \frac{1}{N} \sum_{k=-M}^M w^2(k), \quad N \gg M \gg 1 * W(\omega)$$

- This estimator is consistent. The variance reduces with increasing N
- There is a trade-off (again) between bias and variance:
 - M should be large to minimise bias
 - Large M , however, increases the variance

Review of spectral estimation methods

Bias reduction, windows	Variance reduction, averaging
Windows: Rectangular, Hamming, Kaiser, Chebyshev,	Welch's, Blackman-Tukey, etc.
Bias can be reduced by applying tapering (or windowing)	Variance can be reduced by averaging multiple modified periodograms
At the cost of loss in spectral resolution	At the cost of loss in spectral resolution
Does not affect variance	Does not affect bias
Bias due to sidelobes is referred to as spectral leakage	More or less equivalent to spectral smoothing
	
	

Let's combine the best of both worlds!

Method 3: Multitaper spectral estimation

- Inspired by the success of tapering and averaging, one could construct a new spectral estimator as follows:
 - ① Construct several different tapers of size N (full size)
 - ② Ensure that the tapers are properly designed orthogonal functions
 - ③ Produce modified periodograms using each taper (with low bias)
 - ④ Average (with or without weighting) to reduce variance
- First suggested by Thomson in 1982
- This estimator is consistent. The variance reduces with increasing N
- Windows based on Discrete Prolate Spheroidal Sequences (DPSS)
- MATLAB function `>> pmtm` for multitaper spectral estimator
- MATLAB function `>> dpss` to produce the tapers

Method 3: Multitaper spectral estimation

- Assume a sequence of data $x(n)$ of size N , and a set of K different tapers
- The multitaper spectral estimator is

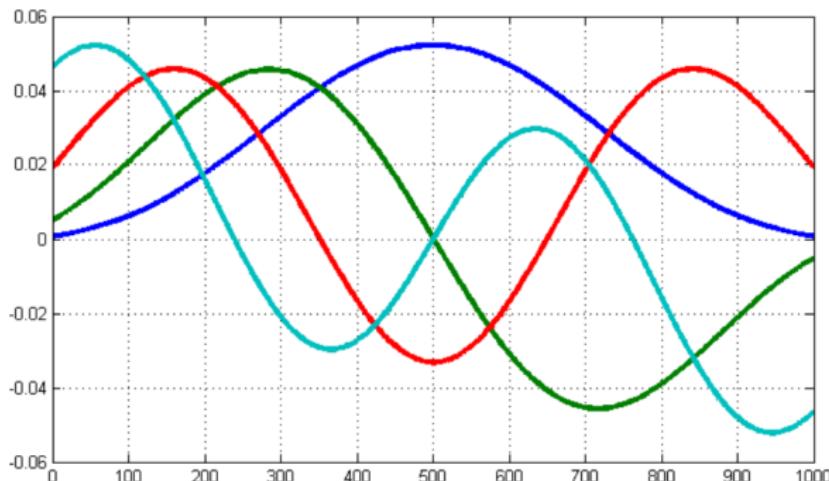
$$\hat{P}_x^{(m,k)}(\omega) = \frac{1}{N} \left| \sum_{n=0}^{N-1} \left| x(n) w_k(n) e^{-j\omega n} \right| \right|^2$$

$$\hat{P}_x^{(\text{multitaper})}(\omega) = \frac{1}{K} \sum_{k=0}^{K-1} \hat{P}_x^{(m,k)}(\omega)$$

- Each taper $w_k(n)$ must have low sidelobe level to prevent bias
- The individual modified periodograms $\hat{P}_x^{(m,k)}(\omega)$ must be pairwise uncorrelated with common variance

Method 3: Discrete Prolate Spheroidal Sequences (DPSS)

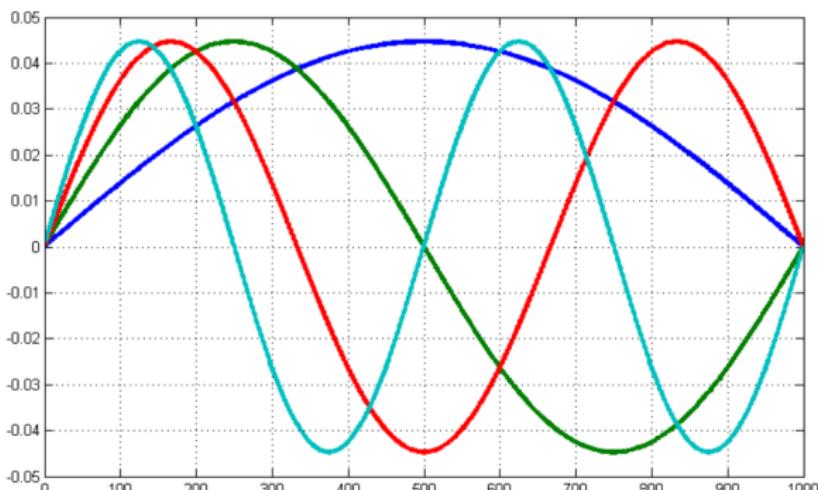
- Discrete Prolate Spheroidal Sequences (DPSS) are optimal tapers
- DPSS are, however, complicated to construct (from scratch)
- Simple in MATLAB: `>> [e,v] = dpss(1000,2);`



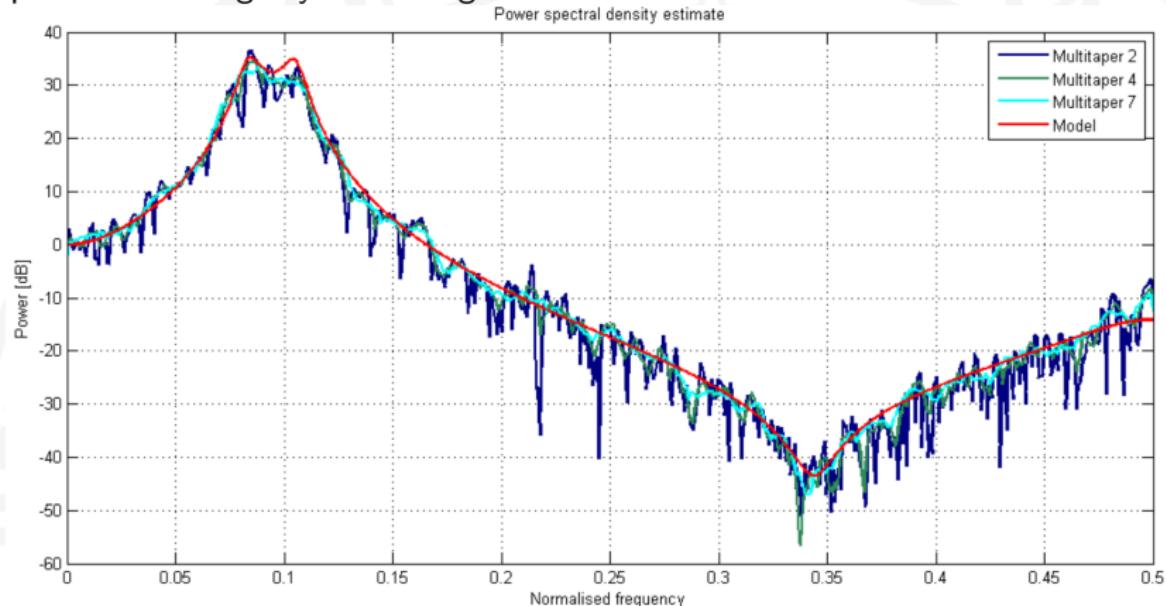
Method 3: Poor-mans multitapers

- A simpler set of orthonormal tapers can be constructed from sinusoidal tapers

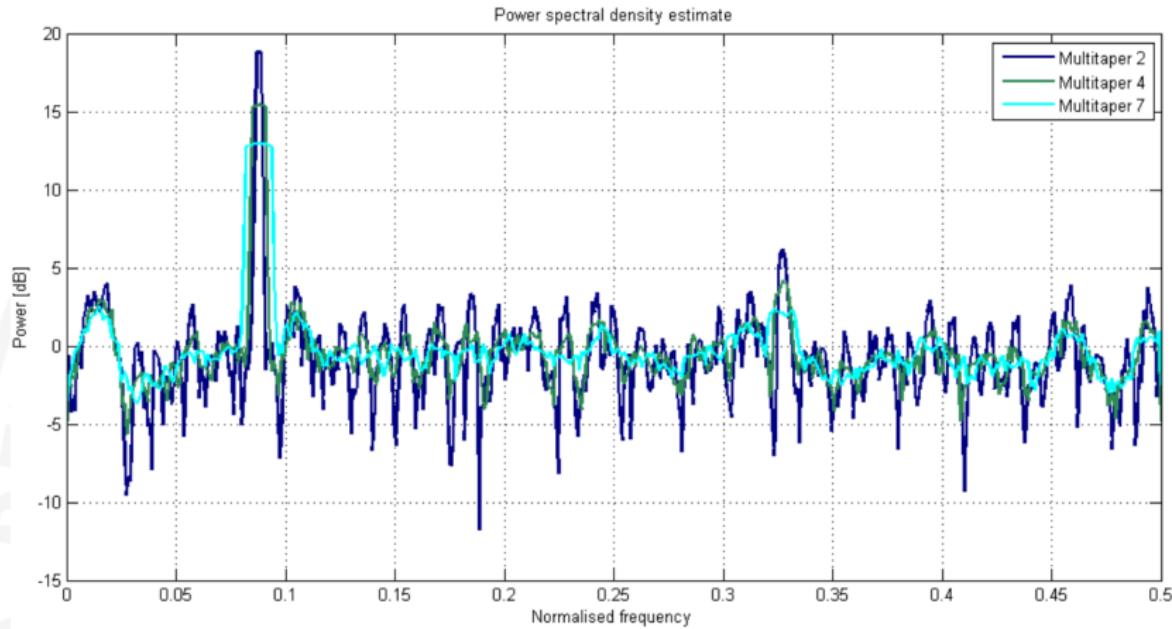
$$w_k(n) = \sqrt{\frac{2}{N+1}} \sin\left(\frac{\pi(k+1)(n+1)}{N+1}\right)$$



Method 3: Performance of multitaper spectral estimation: ARMA random process with large dynamic range



Method 3: Performance of multitaper spectral estimation: Single sinusoid in White Gaussian Noise



Parametric density estimation methods:

- Previous nonparametric methods based on windowing/tapering of the signal assume that the signal is null outside the window
- Let's fit a parametric model for the signal so we get rid of such assumption
- This should improve the resolution if the model is correct (too rigid, too flexible, overfitted, order/lag size)
- Several processes can model a discrete-time signal (aka time-series):
 - AutoRegressive (AR)
 - Moving Average (MA)
 - AutoRegressive and Moving Average (ARMA)
 - Sum of harmonics (complex sinusoids)
 - Multiple Signal Classification (MUSIC)
- We need an accurate estimate of model parameters
- All of them are based on autocorrelation and partial correlation

Four approaches:

- AutoRegressive, AR(p):

$$X(n) = \sum_{k=1}^p a_k Y(n-k)$$

- Moving Average, MA(q):

$$Y(n) = \sum_{k=0}^q b_k X(n-k)$$

- AutoRegressive and Moving Average, ARMA(p, q):

$$Y(n) = \sum_{k=1}^p a_k Y(n-k) + \sum_{k=0}^q b_k X(n-k)$$

- Sum of harmonics (complex sinusoids) in noise:

$$Y(n) = \sum_{k=1}^M A_k e^{j2\pi f_k n} + X(n)$$

where Y_n is the observed output of the system, X_n is the unobserved input of the system (zero mean white Gaussian noise process with unknown variance), and a_k, b_k, A_k are coefficients to be estimated

Spectrum Estimation with AR Models:

- AutoRegressive, AR(p):

$$X_n = \sum_{k=1}^p a_k Y(n-k)$$

- PSD of the process is given by:

$$P_{\text{AR}}(f) = \frac{\sigma^2}{|1 + \sum_{k=1}^p a_k e^{-j2\pi fk}|^2}$$

- We need to estimate a_k and the noise variance σ^2

How?

The autocorrelation, or the Yule-Walker method

- Pre-multiply by $x^*(n - k)$ and take expectations. After some maths, Yule-Walker solution:

$$\hat{\mathbf{R}}\mathbf{a} = -\hat{\mathbf{r}},$$

where $\hat{\mathbf{R}}$ is a $p \times p$ matrix

$$\hat{\mathbf{R}} = \begin{pmatrix} \hat{r}(0) & \hat{r}(-1) & \dots & \hat{r}(-p+1) \\ \hat{r}(1) & \hat{r}(0) & \dots & \hat{r}(-p+2) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{r}(p-1) & \hat{r}(p-2) & \dots & \hat{r}(0) \end{pmatrix} \quad \hat{\mathbf{r}} = (\hat{r}(1), \hat{r}(2), \dots, \hat{r}(p))^\top$$

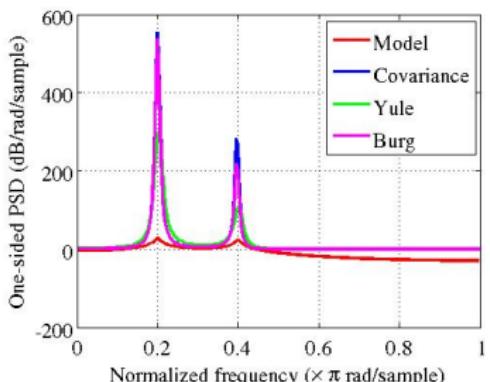
- Parameters $\hat{\mathbf{a}} = -\hat{\mathbf{R}}^{-1}\hat{\mathbf{r}}$
- Noise variance: $\hat{\sigma}^2 = \hat{r}(0) + \sum_{k=1}^p a_k \hat{r}(k)$

Several alternative methods with AR models:

- Yule-Walker method >> pyulear
- Covariance method >> pcov
- Burg method >> pburg

Spectrum Estimation with AR Models

```
% Define AR filter coefficients  
a = [1 -2.2137 2.9403 -2.1697 0.9606];  
[H,w] = freqz(1,a,256); % AR filter freq response  
% Scale to make one-sided PSD  
Hp = plot(w/pi,20*log10(2*abs(H)/(2*pi)), 'r'); hold on;  
x = filter(1,a,randn(256,1)); % AR system output  
Pcov = pcov(x,4,511); Pyulear = pyulear(x,4,511); [Pburg,W] =  
pburg(x,4,511);  
plot(W/pi,20*log10(Pcov), 'b'); plot(W/pi,20*log10(Pyulear), 'k');  
plot(W/pi,20*log10(Pburg), 'm')  
xlabel('Normalized frequency ( $\times \pi$  rad/sample)')  
ylabel('One-sided PSD (dB/rad/sample)')  
legend('Model', 'Covariance', 'Yule', 'Burg'); grid
```



Spectrum Estimation with AR Models: choosing p

- Akaike Information Criterion (AIC):

$$\text{AIC}(k) = N \log \hat{\sigma}_k^2 + 2k, \quad k : \text{order}$$

- Bayesian Information Criterion (BIC)
- Minimum Description Length (MDL) principle:

$$\text{MDL}(k) = N \log \hat{\sigma}_k^2 + k \log N, \quad k : \text{order}$$

When to use AR-based spectrum estimation?

- The AR-based spectrum estimation methods show very good performance if the processes are narrowband and have sharp peaks in their spectra
- Also, many good results have been reported when they are applied to short data records

Spectrum Estimation with MA Models:

- Moving Average, $MA(q)$:

$$Y(n) = \sum_{k=0}^q b_k X(n-k)$$

- PSD of the process is given by:

$$P_{MA}(f) = \sigma^2 \left| 1 + \sum_{k=1}^q b_k e^{-j2\pi fk} \right|^2$$

- One can show that $r(k) = 0$, $\forall |k| > q$, so:

$$P_{MA}(f) = \sum_{k=-q}^q r(k) e^{-j2\pi fk}$$

- We need to estimate r_k , which is nonlinear
- Durbin proposed an approximate procedure that is based on a high order AR approximation of the MA process:
 - Data are modeled by an AR model of order L , where $L \gg q$
 - Coefficients are estimated using the AR equation
 - Sequence $\{\hat{a}_1, \dots, \hat{a}_L\}$ is fitted with an $AR(q)$ model, whose parameters are also estimated using the autocorrelation method
 - Estimated coefficients $\{\hat{b}_1, \dots, \hat{b}_q\}$ are then used in $P_{MA}(f)$

Spectrum Estimation with ARMA Models:

- AutoRegressive and Moving Average, ARMA(p, q), $M = p + q$:

$$Y(n) = \sum_{k=1}^p a_k Y(n-k) + \sum_{k=0}^q b_k X(n-k)$$

- PSD of the process is given by:

$$P_{\text{ARMA}}(f) = \sigma^2 \frac{|1 + \sum_{k=1}^q b_k e^{-j2\pi fk}|^2}{|1 + \sum_{k=1}^p a_k e^{-j2\pi fk}|^2}$$

- The ML estimates of the ARMA coefficients are difficult to obtain
- We usually resort to methods that yield suboptimal estimates

$$\begin{pmatrix} \hat{r}(q) & \hat{r}(q-1) & \dots & \hat{r}(q-p+1) \\ \hat{r}(q+1) & \hat{r}(q) & \dots & \hat{r}(q-p+2) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{r}(M-1) & \hat{r}(M-2) & \dots & \hat{r}(M-p) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} + \begin{pmatrix} \epsilon_{q+1} \\ \epsilon_{q+2} \\ \vdots \\ \epsilon_M \end{pmatrix} = \begin{pmatrix} \hat{r}(q+1) \\ \hat{r}(q+2) \\ \vdots \\ \hat{r}(M) \end{pmatrix}$$

or

$$\hat{\mathbf{R}}\mathbf{a} + \boldsymbol{\epsilon} = -\hat{\mathbf{r}},$$

where ϵ_i is a term that models the errors in the Yule-Walker equations

Spectrum Estimation with ARMA Models:

- The Yule-Walker expression now is:

$$\hat{\mathbf{R}}\mathbf{a} + \boldsymbol{\epsilon} = -\hat{\mathbf{r}},$$

where ϵ_i is a term that models the errors in the Yule-Walker equations

- The least-squares estimates of \mathbf{a} :

$$\hat{\mathbf{a}} = (\hat{\mathbf{R}}^H \hat{\mathbf{R}})^{-1} \hat{\mathbf{R}}^H \hat{\mathbf{r}}$$

- Once the AR coefficients are estimated, we can filter the observed data and obtain a sequence that is approximately modeled by an MA(q) model
- From the data $y(n)$ we can estimate the MA PSD and obtain the PSD estimate of the data $x(n)$:

$$P_{\text{ARMA}}(f) = \frac{\hat{P}_{\text{MA}}(f)}{|1 + \sum_{k=1}^p a_k e^{-j2\pi fk}|^2}$$

Pisarenko Harmonic Decomposition (PHD) method:

- Sum of harmonics (complex sinusoids) in noise:

$$Y(n) = \sum_{k=1}^M A_k e^{j2\pi f_k n} + X(n),$$

where f_k is the frequency of the k -th complex sinusoid, A_k is the complex amplitude: $A_k = |A_k|e^{j\phi_k}$, and $X(n)$ is a sample of a zero mean white noise

- The PSD of the process is a sum of the continuous spectrum of the noise and a set of impulses with area $|A_k|^2$ at the frequencies f_k :

$$P(f) = \sum_{k=1}^m |A_k|^2 \delta(f - f_k) + P_\epsilon(f)$$

- Pisarenko found that the frequencies of the sinusoids can be obtained from the eigenvector corresponding to the smallest eigenvalue of the autocorrelation matrix: `>> [V Lambda] = eigs(R, 'ascend');`

Pisarenko Harmonic Decomposition (PHD) method:

- ① Estimate the $(m+1) \times (m+1)$ autocorrelation matrix

$$\mathbf{R} = \sum_{i=1}^m (\lambda_i + \sigma^2) \mathbf{v}_i \mathbf{v}_i^H + \sum_{i=m+1}^M \sigma^2 \mathbf{v}_i \mathbf{v}_i^H,$$

provided it is known that the number of complex sinusoids is m , where $\{\lambda_i\}_{i=1}^m$ are the nonzero eigenvalues of \mathbf{R} with associated eigenvectors \mathbf{v}_i .

- ② Evaluate the minimum eigenvalue λ_{m+1} and the eigenvectors of \mathbf{R} .
- ③ Set the white-noise power to $\hat{\sigma}^2 = \lambda_{m+1}$, estimate the frequencies of the complex sinusoids from the peak locations of $\hat{P}_{PHD}(f)$ in

$$\hat{P}_{PHD}(f) = \frac{1}{|\mathbf{X}^H(f)\mathbf{v}_{m+1}|^2}$$

- ④ Compute the powers of the complex sinusoids $P_i = |A_i|^2$ solving a problem with m linear equations
- ⑤ Substitute the estimated parameters in

$$P(f) = \sum_{k=1}^m |A_k|^2 \delta(f - f_k) + P_\epsilon(f)$$

Pisarenko's method is not used frequently in practice because its performance is much poorer than the performance of some other signal and noise subspace based methods developed later

Multiple Signal Classification (MUSIC):

- A procedure very similar to Pisarenko's proposed by Schmidt (late 1970s)
- Suppose again that the process $\{Y(n)\}$ is described by m complex sinusoids in white noise
- Eigendecompose the correlation matrix: we actually assume that the m largest eigenvalues span the signal subspace, and the remaining eigenvectors, the noise subspace
- Estimate the noise variance from the $M - m$ smallest eigenvalues:

$$\hat{\sigma}^2 = \frac{1}{M-m} \sum_{i=m+1}^M \lambda_i$$

and the frequencies from the peak locations of the pseudospectrum

$$P_{MUSIC}(f) = \frac{1}{\sum_{i=m+1}^M |\mathbf{X}^H \mathbf{v}_i(f)|^2}$$

- The powers of the complex sinusoids and the parameters are estimated as in Pisarenko's

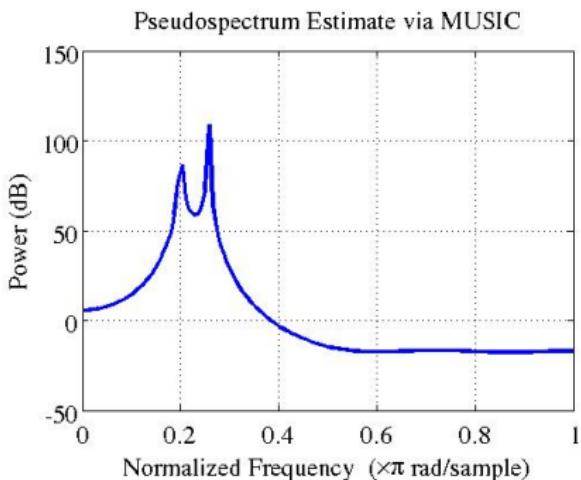
Improving MUSIC: the Eigenvector method (EV)

- MUSIC has better performance than Pisarenko's method because of the introduced averaging via the extra noise eigenvectors
- The averaging reduces the statistical fluctuations present in Pisarenko's pseudospectrum, which arise due to the errors in estimating the autocorrelation matrix
- These fluctuations can further be reduced by applying the Eigenvector method, which is a modification of MUSIC and whose pseudospectrum is given by:

$$P_{EV}(f) = \frac{1}{\sum_{i=m+1}^M |\frac{1}{\lambda_i} \mathbf{X}^H \mathbf{v}_i(f)|^2}$$

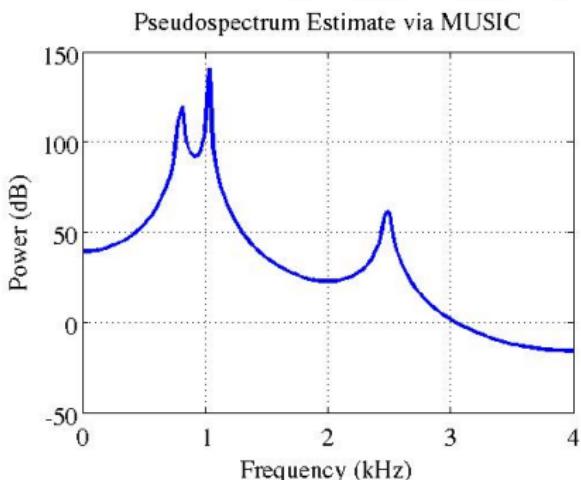
MUSIC example 1: This example analyzes a signal vector x , assuming that two real sinusoidal components are present in the signal subspace. In this case, the dimension of the signal subspace is 4 because each real sinusoid is the sum of two complex exponentials

```
>> randn('state',0);
>> n = 0:199;
>> x = cos(0.257*pi*n) + sin(0.2*pi*n) + 0.01*randn(size(n));
>> pmusic(x,4) % Set p to 4 because two real inputs
```



MUSIC example 2: This example analyzes the same signal vector x with an eigenvalue cutoff of 10% above the minimum. Setting $p(1) = \text{Inf}$ forces the signal/noise subspace decision to be based on the threshold parameter $p(2)$. Specify the eigenvectors of length 7 using the `nwin` argument, and set the sampling frequency fs to 8 kHz:

```
>> randn('state',0);
>> n = 0:199;
>> x = cos(0.257*pi*n) + sin(0.2*pi*n) + 0.01*randn(size(n));
>> pmusic(x,[Inf,1.1],[],8000,7); % Window length = 7
```



AR-based PSD estimation: Advantages and shortcomings:

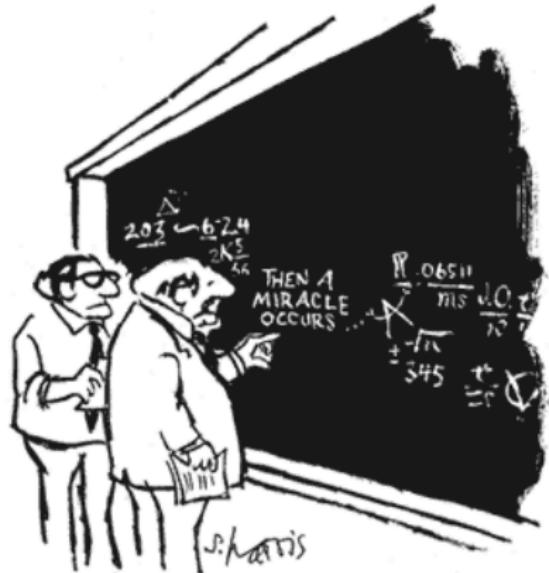
	Burg	Covariance	Modified Covariance	Yule-Walker
Characteristics	Does not apply window to data	Does not apply window to data	Does not apply window to data	Applies window to data
	Minimizes the forward and backward prediction errors in the least squares sense, with the AR coefficients constrained to satisfy the L-D recursion	Minimizes the forward prediction error in the least squares sense	Minimizes the forward and backward prediction errors in the least squares sense	Minimizes the forward prediction error in the least squares sense (also called autocorrelation method)
Advantages	High resolution for short data records	Better resolution than Y-W for short data records (more accurate estimates)	High resolution for short data records	Performs as well as other methods for large data records
	Always produces a stable model	Able to extract frequencies from data consisting of p or more pure sinusoids	Able to extract frequencies from data consisting of p or more pure sinusoids	Always produces a stable model
		Does not suffer spectral line-splitting		
Disadvantages	Peak locations highly dependent on initial phase	May produce unstable models	May produce unstable models	Performs relatively poorly for short data records
	May suffer spectral line-splitting for sinusoids in noise, or when order is very large	Frequency bias for estimates of sinusoids in noise	Peak locations slightly dependent on initial phase	Frequency bias for estimates of sinusoids in noise
	Frequency bias for estimates of sinusoids in noise		Minor frequency bias for estimates of sinusoids in noise	
Conditions for Nonsingularity		Order must be less than or equal to half the input frame size	Order must be less than or equal to 2/3 the input frame size	Because of the biased estimate, the autocorrelation matrix is guaranteed to be positive-definite, hence nonsingular

Spectral estimation in MATLAB:

Method	Description	Functions
Periodogram	Power spectral density estimate	periodogram
Welch	Averaged periodograms of overlapped, windowed signal sections	pwelch , cpsd , tfestimate , mscoh
Multitaper	Spectral estimate from combination of multiple orthogonal windows (or "tapers")	pmtm
Yule-Walker AR	Autoregressive (AR) spectral estimate of a time-series from its estimated autocorrelation function	pyulear
Burg	Autoregressive (AR) spectral estimation of a time-series by minimization of linear prediction errors	pburg
Covariance	Autoregressive (AR) spectral estimation of a time-series by minimization of the forward prediction errors	pcov
Modified Covariance	Autoregressive (AR) spectral estimation of a time-series by minimization of the forward and backward prediction errors	pmcov
MUSIC	Multiple signal classification	pmusic
Eigenvector	Pseudospectrum estimate	peig

Reviewed:

Spectral estimation, Fourier transform, power spectrum, parametric vs non-parametric spectral estimation, ACF windowing, periodogram averaging, multitaper methods, ARMA modeling, etc.



"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

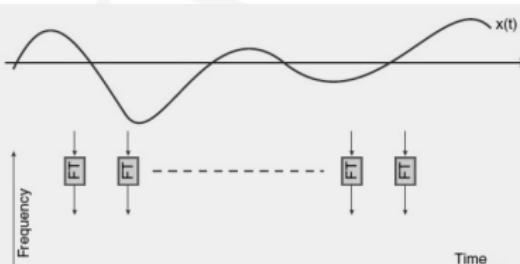
Part 4: Signal decomposition and transforms

Spectral analysis of non-stationary signals

- What happens when the signals are non-stationary?
- The autocorrelation function is no longer a function of lag only
- Non-trivial problem
- Simple intuitive approach:
 - Break the timeseries into segments that are locally WSS
 - Estimate the spectrum for each segment
- This is then a time-frequency representation
- Different approaches to time-frequency analysis:
 - Gabor filtering/transform
 - Short Time Fourier Transform
 - Second order time-frequency analysis (Cohen class)
 - Wavelet analysis
 - Spectrogram

Gabor transform is a ‘local Fourier Transform’:

- The Gabor transform, named after Dennis Gabor, is a special case of the short-time Fourier transform.
- It is used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time.
 - The function to be transformed is first multiplied by a Gaussian function (window) and then transformed with a Fourier transform to derive the time-frequency analysis
 - The window function means that the signal near the time being analyzed will have higher weight
 - The Gabor transform of a signal $x(t)$ is defined by this formula:



- The Gabor transform is invertible:

$$x(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G_x(\tau, f) e^{j2\pi t f} df d\tau$$

Gabor transform implementation:

- The Gaussian function has infinite range and it is impractical for implementation
- However, a level of significance can be chosen (for instance 0.00001) for the distribution of the Gaussian function.

$$\begin{cases} e^{-\pi a^2} \geq 0.00001; & |a| \leq 1.9143 \\ e^{-\pi a^2} < 0.00001; & |a| > 1.9143 \end{cases}$$

- Outside these limits of integration $|a| > 1.9143$, the Gaussian function is small enough to be ignored
- Thus the Gabor transform can be satisfactorily approximated as

$$G_x(t, f) = \int_{-1.9143+t}^{1.9143+t} e^{-\pi(\tau-t)^2} e^{-j2\pi f\tau} x(\tau) d\tau$$

- This simplification makes the Gabor transform practical and realizable
- The window function width can also be varied to optimize the time-frequency resolution tradeoff replacing:

$$-\pi(\tau - t)^2 \rightarrow -\pi\alpha(\tau - t)^2$$

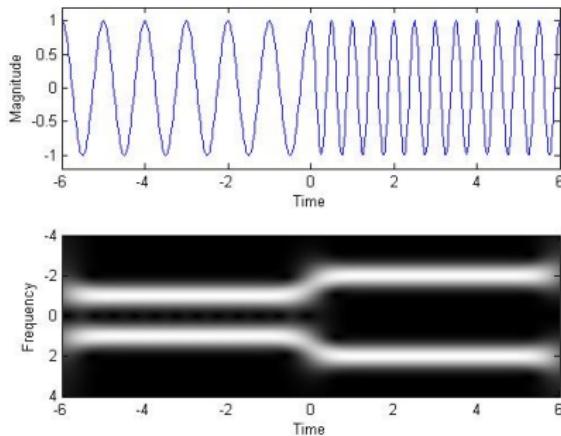
- Selection of α is critical to achieve good frequency resolution in short time-windows (and viceversa) \rightarrow Heisenberg principle

Properties of the Gabor transform:

Signal	Gabor transform	Remarks
$x(t)$	$G_x(t, f) = \int_{-\infty}^{\infty} e^{-\pi(\tau-t)^2} e^{-j2\pi f\tau} x(\tau) d\tau$	-
$a \cdot x(t) + b \cdot y(t)$	$a \cdot G_x(t, f) + b \cdot G_y(t, f)$	Linearity
$x(t - t_0)$	$G_x(t - t_0, f) e^{-j2\pi f t_0}$	Shifting
$x(t)e^{j2\pi f_0 t}$	$G_x(t, f - f_0)$	Modulation

Example of the Gabor transform: Adding the frequency axis we can detect different time-dependent components in the signal

$$x(t) = \begin{cases} \cos(2\pi t) & \text{for } t \leq 0, \\ \cos(4\pi t) & \text{for } t > 0. \end{cases}$$



Example of time-frequency representation

Musical score for "Meno mosso." featuring piano, cello, bassoon, oboe, and strings. The score consists of three staves. The top staff features lyrics in Russian and German. The middle staff features lyrics in German. The bottom staff features lyrics in German. The score includes dynamic markings such as *p*, *f*, *mf*, and *ff*. The piano part includes bassoon and oboe parts.

Мено моссо.

Что Ах - ах - я
Was thou ich denn.
Не зна - я
O Him-melt!
Ес - я
Wedn mein

Мено моссо.

Ах - бишь ты ме - я,
Hol - der du mich liebst,
ес - ле - бишь ты ме - я,
wenn du wirk -lich so mich liebst,
ско - рый скор -
komm her, kne nie -
на ко -
der theurer

Gel. Gel.

Fl. Vcl. Cl. Cbr.

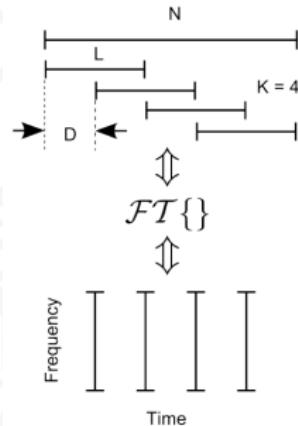
Short time Fourier transform (STFT)

- Divide the total sequence of N data samples into segments of size L , and offset each segment by D points into a total number of K segments such that $N = L + D(K - 1)$
- The Short Time Fourier Transform is

$$SFTF(i, \omega) = \sum_{n=0}^{L-1} x(n + iD)w(n)e^{-j\omega n}$$

where $w(n)$ is a window of choice

- Note the similarity with Welch's method of periodogram averaging
- Also known as sliding window Fourier transform and spectrogram



Short time Fourier transform (STFT)

The STFT can be generalised to use any Fourier based spectral estimator

- Example: Spectrogram based on the modified periodogram:

$$\hat{P}_X(t, \omega) = \frac{1}{L} \left| \sum_{n=0}^{L-1} x(n + iD) w(n) e^{-j\omega n} \right|^2, \quad t = (i + D/2)\delta t$$

where δt is the sampling frequency

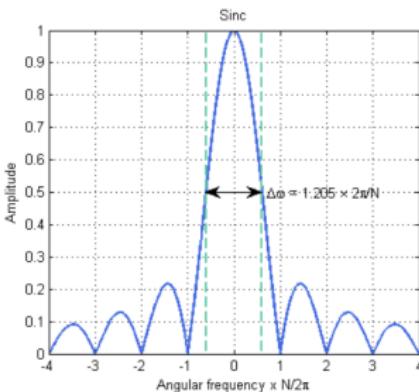
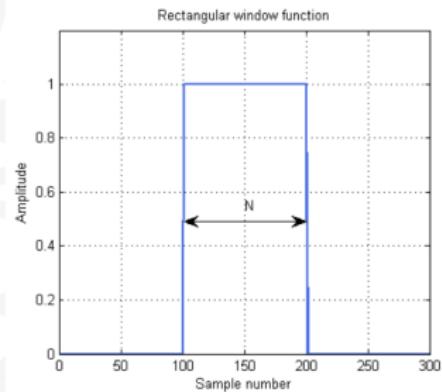
- See `spectrogram` and `specgramdemo` in MATLAB
- Note: STFTs should really be viewed as a stack of individual spectral estimates. In order to construct a proper density $\hat{P}_X(t, \omega)$ as a function of time and frequency, energy conservation has to be taken into account. This is not the case for STFT-based representations.

Time and frequency resolution in STFT

- The *uncertainty principle* states that the time duration Δt and frequency bandwidth $\Delta\omega$ are related by

$$\Delta t \Delta \omega \geq \frac{1}{2}$$

- A fundamental property of the Fourier transform pair $s(t)$ and $S(\omega)$
- First derived by Heisenberg in 1927 in quantum mechanics
- Example: rectangular time window and sinc frequency window

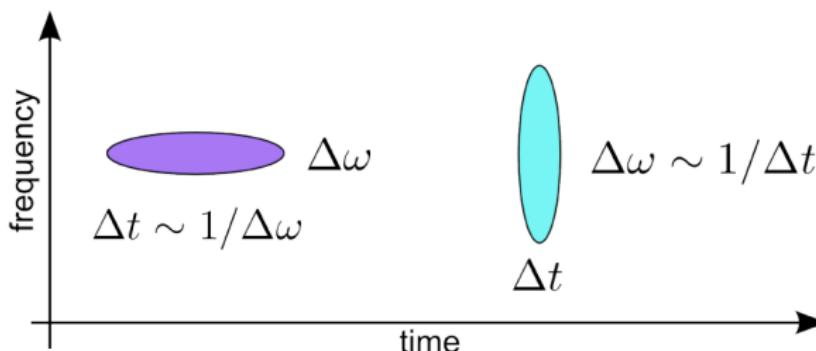


Werner Heisenberg
theoretical physicist
Nobel prize 1932
From wikipedia

Time and frequency resolution in STFT

The time-frequency resolution relation leads to the following:

- Higher frequency resolution requires larger time duration and thereby lower time resolution
- Higher time resolution requires shorter time duration and thereby lower frequency resolution



Time-frequency analysis generalizes Gabor analysis

- The Gabor transform of a signal $x(t)$:

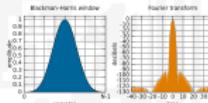
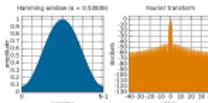
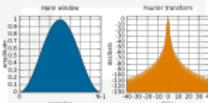
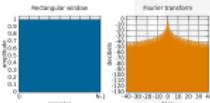
$$G_x(t, f) = \int_{-\infty}^{\infty} e^{-\pi(\tau-t)^2} e^{-j2\pi f\tau} x(\tau) d\tau$$

- Time-frequency analysis:

$$G_x(t, f) = \int_{-\infty}^{\infty} w(t - \tau) e^{-j2\pi f\tau} x(\tau) d\tau$$

- Windows (the same as in Fourier-based spectral analysis):

- Rectangular: $w(t) = 1$
- Hann (Hanning): $w(t) = 0.5 (1 - \cos(\frac{2\pi t}{N-1}))$
- Hamming: $w(t) = \alpha - \beta \cos(\frac{2\pi t}{N-1})$
- Blackman: $w(t) = a_0 - a_1 \cos(\frac{2\pi t}{N-1}) + a_2 \cos(\frac{4\pi t}{N-1})$
- Blackman-Harris: $w(t) = a_0 - a_1 \cos(\frac{2\pi t}{N-1}) + a_2 \cos(\frac{4\pi t}{N-1}) - a_3 \cos(\frac{6\pi t}{N-1})$



Example: STFT of FM signal

- Consider a simple deterministic mono-component signal

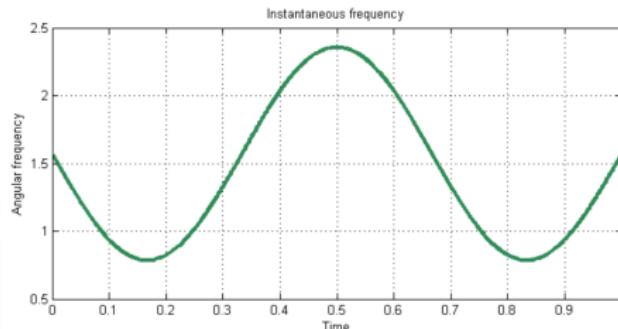
$$s(t) = a(t)e^{j\phi(t)}, \quad \phi(t) = \omega_0 t + \frac{a_1}{\omega_1} \cos(\omega_1 t)$$

- The instantaneous frequency is defined as

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

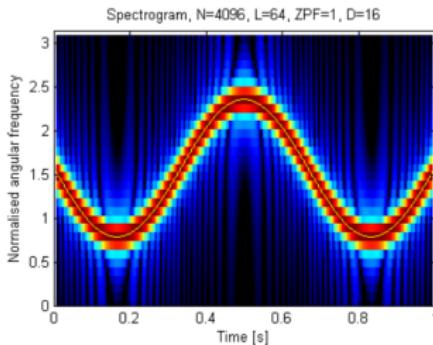
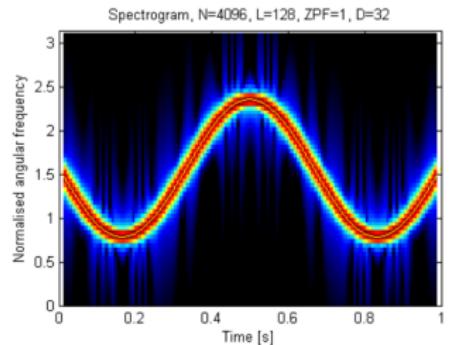
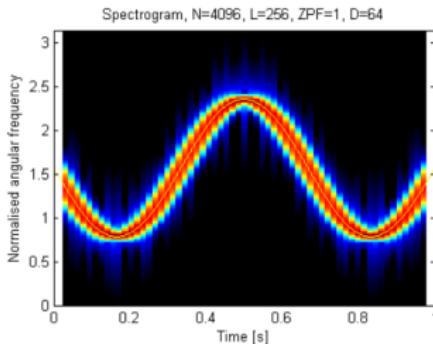
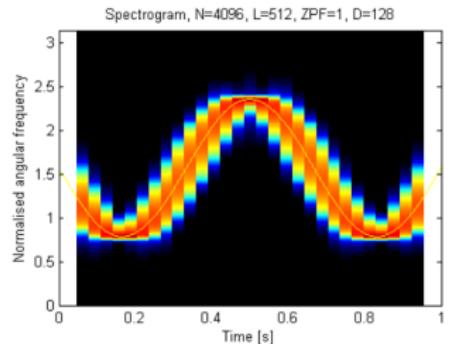
- For our signal, the IF becomes

$$\omega_{IF}(t) = \omega_0 - a_1 \sin(\omega_1 t)$$



Example: STFT of FM signal

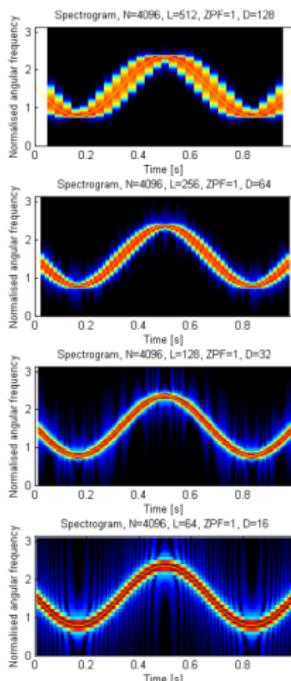
The effect of choosing segment size L



Example: STFT of FM signal

Comments on choosing segment size:

- The first spectrogram has a long time window $L = 512$
- During the window length, the frequency changes
- This causes smearing which appears as poorer resolution in the frequency domain
- The last spectrogram has a short time window $L = 64$
- Here, we observe “true” lowering of the spectral resolution due to window length



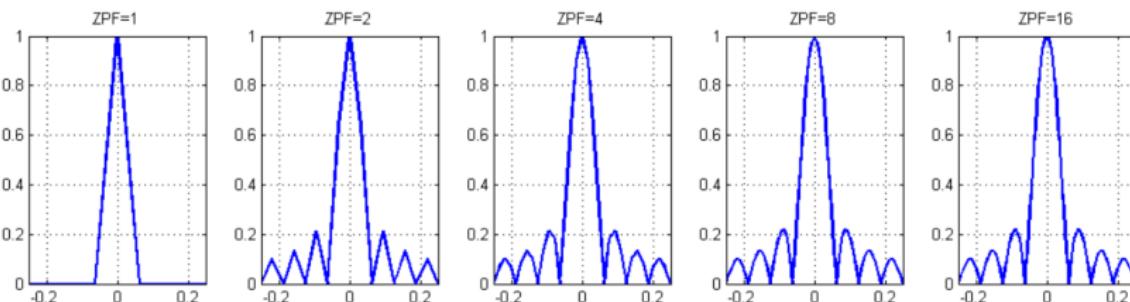
This example shows the importance of window length in the time-frequency representation

A dirty trick: Zeropadding

A simple trick to get smoother spectral representation is zeropadding:

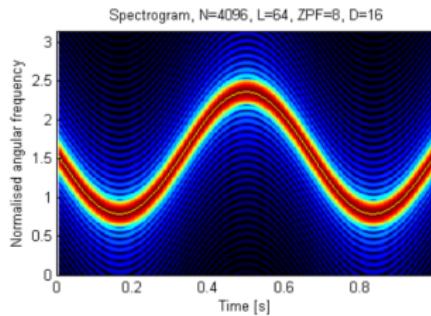
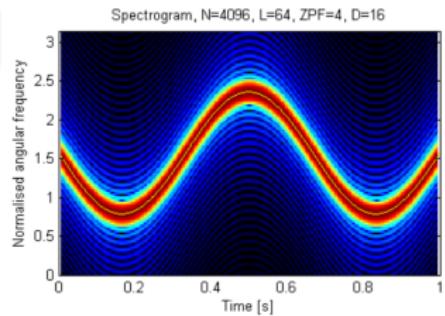
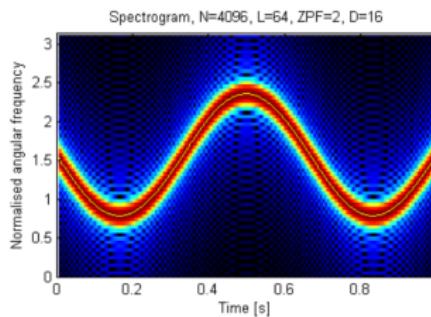
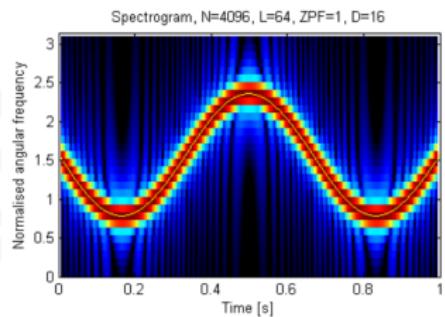
- Zeropadding is adding zeros in the sequence to be FT'ed
- Done in MATLAB for you: `fft(x, N)` will zeropad the sequence x to a total of N elements before FT is applied

```
>> N = 16; ii = 3;  
>> NN = N * 2(ii-1);  
% Number of samples (including zeropadding)  
>> xax = [-NN/2:NN/2-1]/NN; % Proper x-axis for plotting  
>> X = 1/N*abs(fftshift(fft( ones(N,1), NN ))); % Rectangular window
```



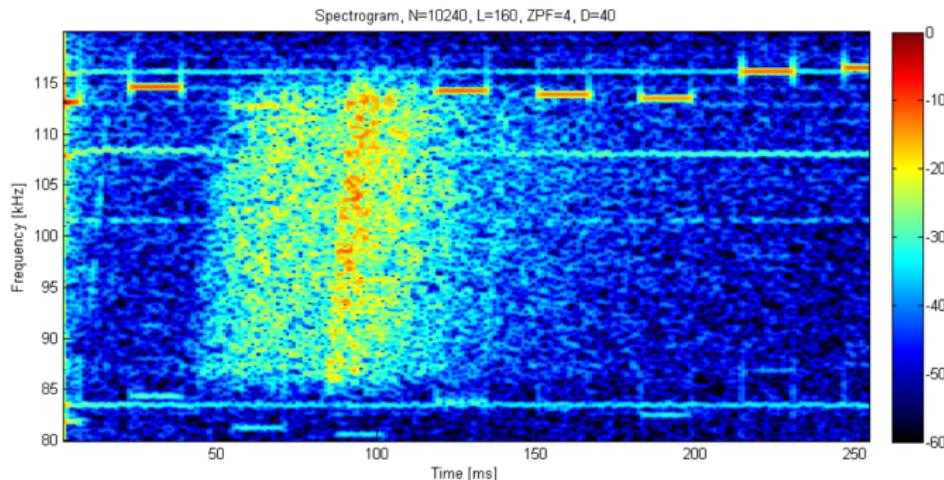
A dirty trick: Zeropadding

Example: STFT of FM signal

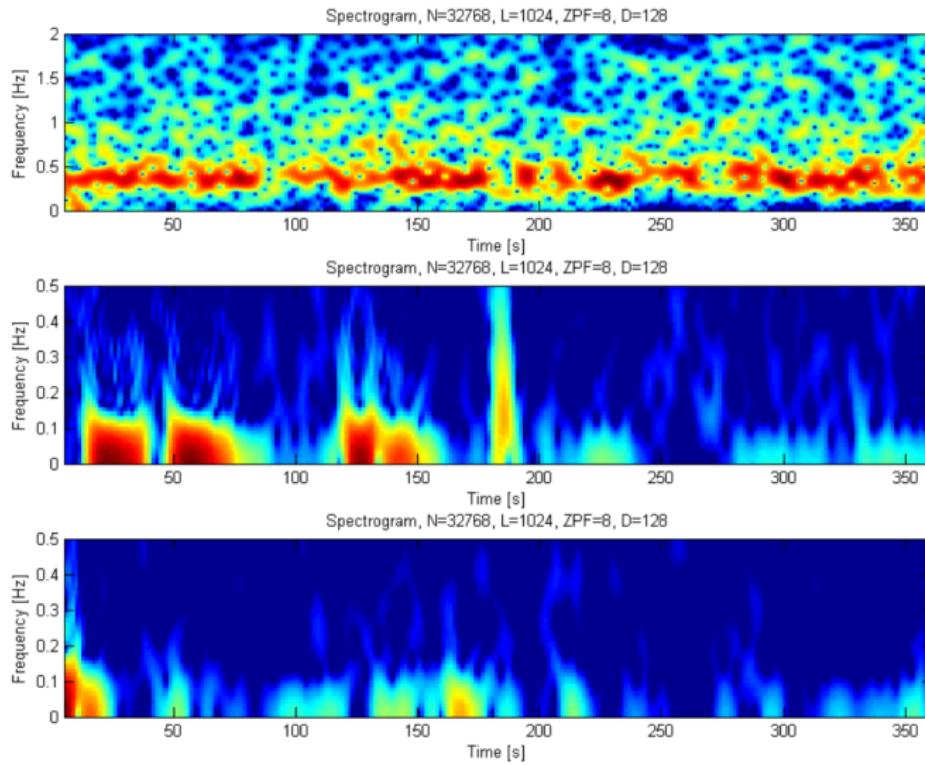


Time-frequency representation of sonar data

Short Time Fourier Transform of single ping of sonar rawdata Modified periodogram with Kaiser 4.5 window and zeropadding

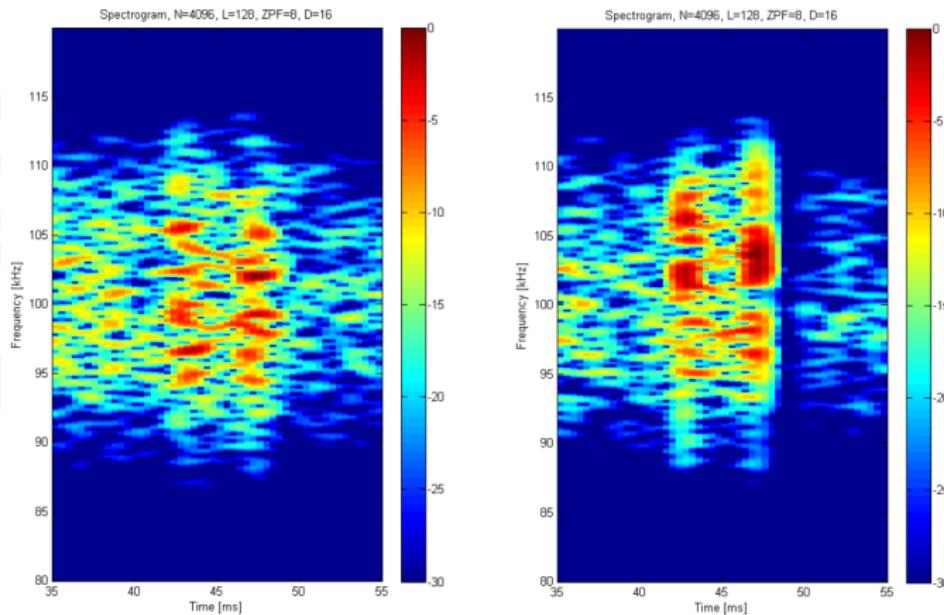


Time-frequency representation of sonar data Short Time Fourier Transform of navigation data



Time-frequency representation of sonar data

STFT of sonar data before and after beamforming



Second order time-frequency representations

- The uncertainty principle limits directly the ability to resolve transient frequencies in the STFT
- How do we approach this?
- We capture the time variation (non-stationarity) into a time-varying autocorrelation function

$$R_X(t, \tau) = x(t + \tau/2)x(t - \tau/2)$$

and directly transform into time-frequency domain

- The Wigner-Ville distribution does this

$$W(t, \omega) = \int x(t + \tau/2)x(t - \tau/2)e^{-j\omega\tau} d\tau$$

- Introduced by Wigner in 1932 in quantum mechanics, and introduced to signal analysis by Ville in 1948

Second order Time-frequency representations

- This approach has a number of desirable properties
 - It obtains “full” resolution for LFM type signals
 - It is energy preserving (and as such a proper distribution)
 - It does however, produce cross terms (ghosts)
- A generalised form (referred to as Cohen’s class) is

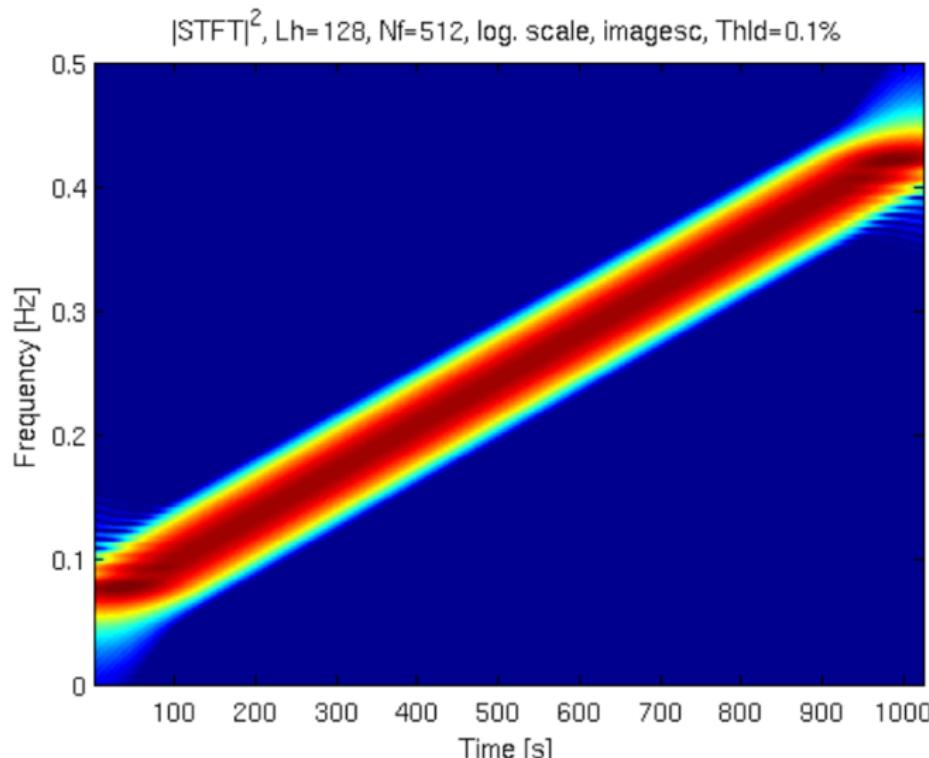
$$C(t, \tau) = W(t, \omega)\Phi(t, \omega)$$

where the kernel function $\Phi(t, \omega)$ can be chosen

- The generalised form can describe any time-frequency representation (including the STFT)
- By choosing the kernel function, the cross terms can be suppressed at the cost of loss in resolution
- MATLAB Toolbox, and Octave Toolbox: <http://tftb.nongnu.org/>

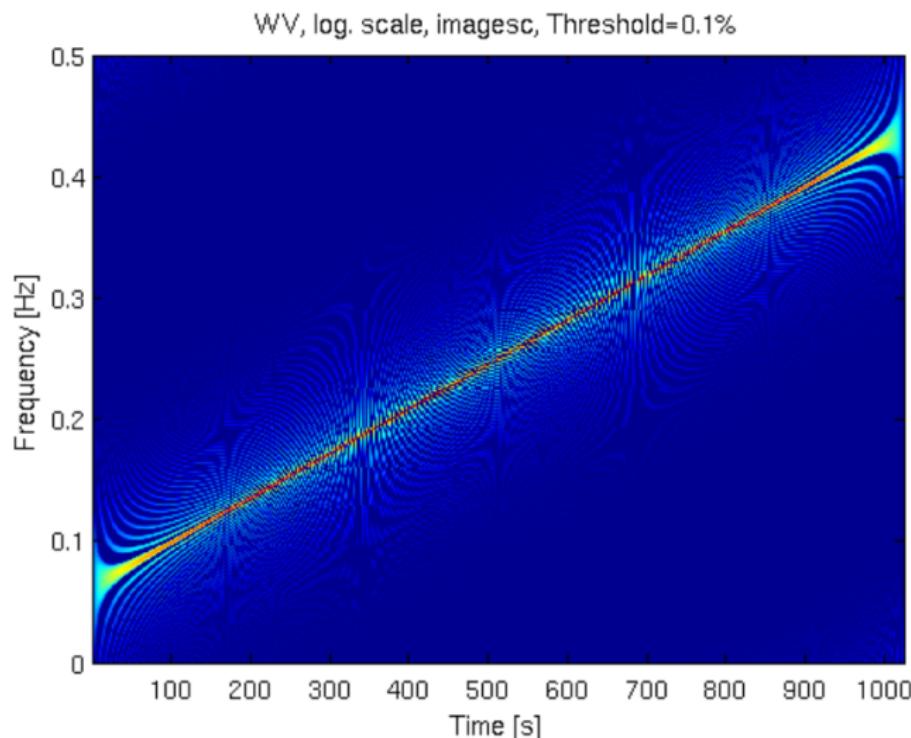
Example: Time-frequency representations

Example: STFT of LFM signal



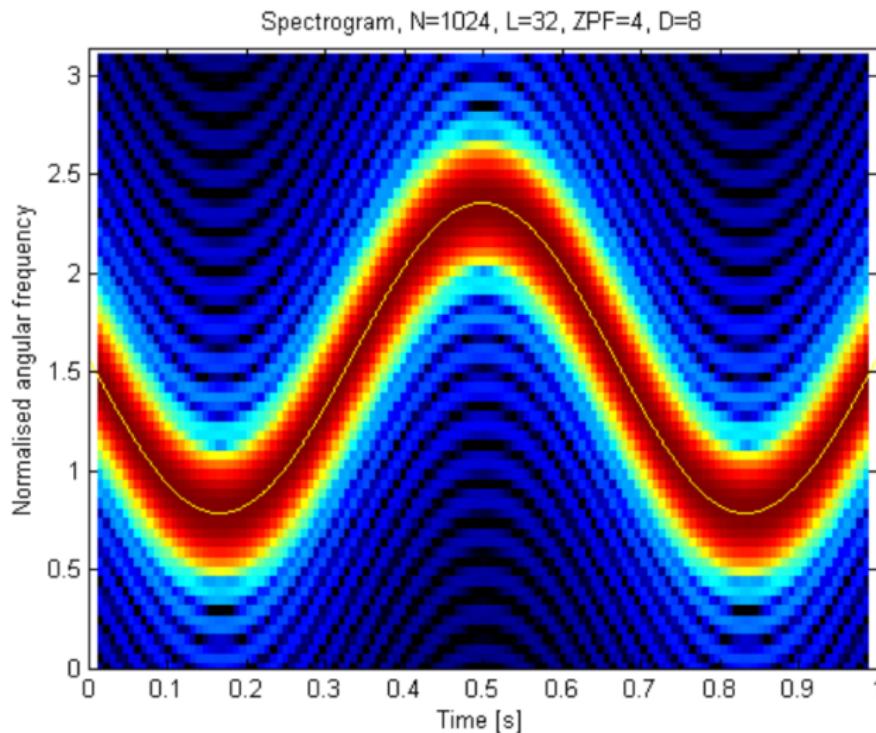
Example: Time-frequency representations

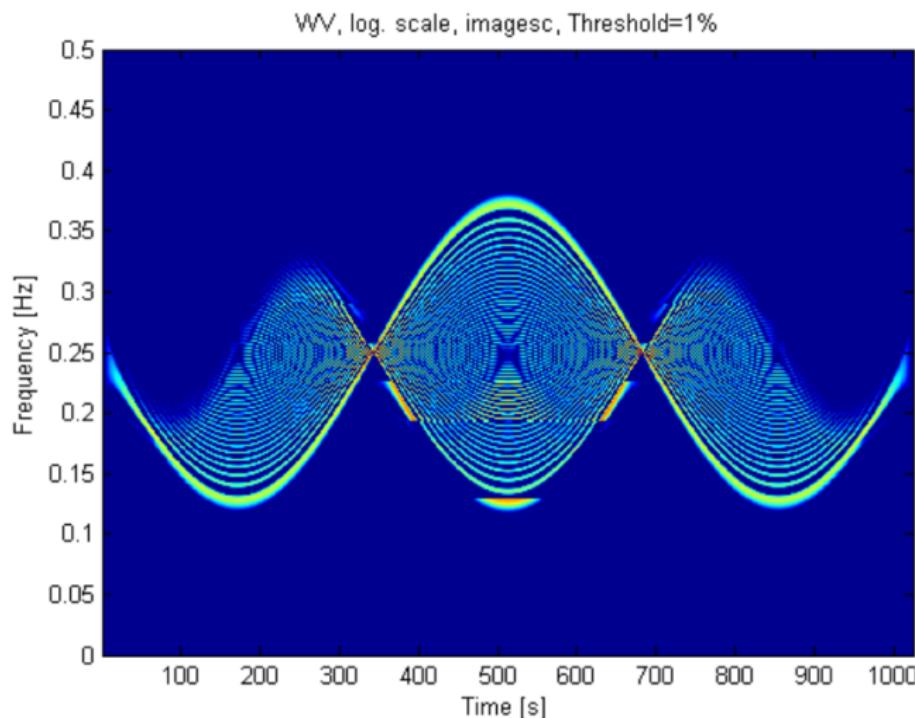
Example: Wigner-Ville distribution of LFM signal



Example: Time-frequency representations

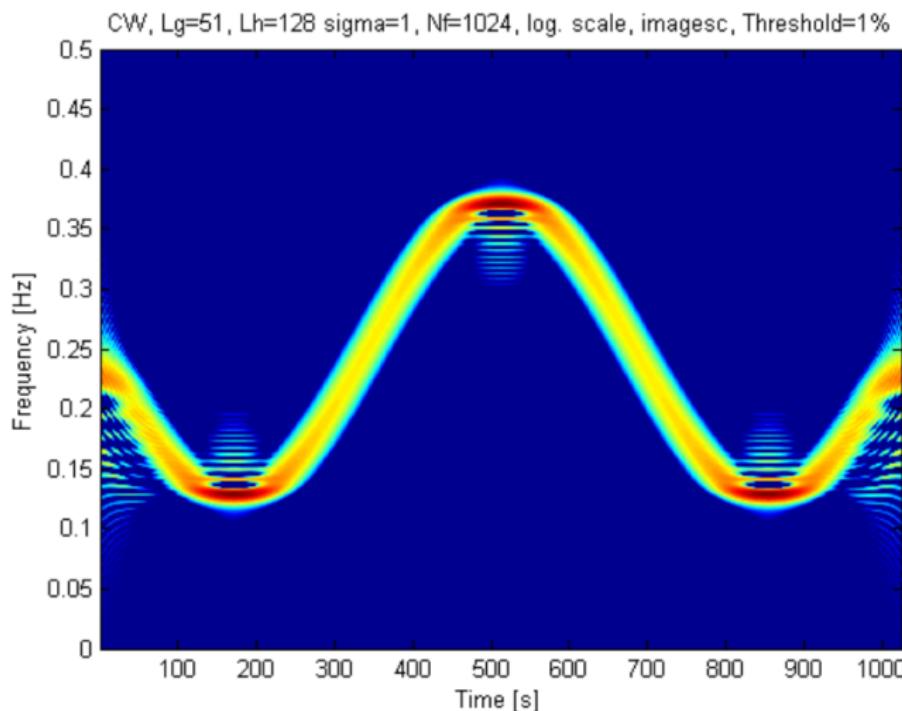
Example: STFT of FM signal



Example: Time-frequency representations**Example: Wigner-Ville distribution of FM signal**

Example: Time-frequency representations

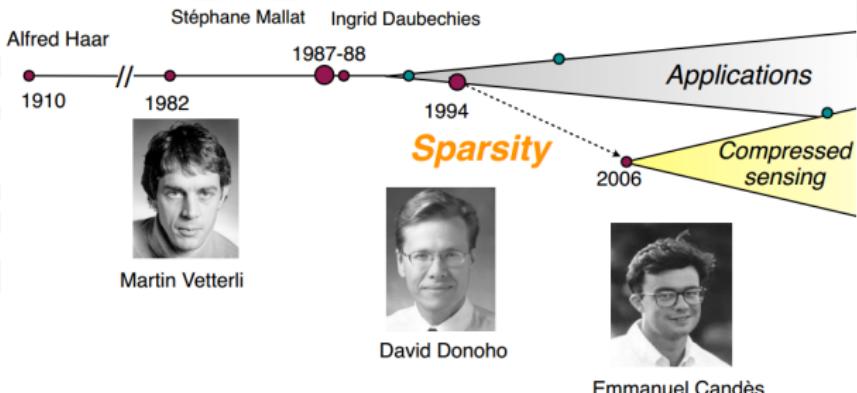
Example: Choi-Williams distribution of FM signal



Wavelets



- B-spline wavelets (1991)
- Wavelets in medicine and biology (CRC 1996)
- Fractional wavelets (EPFL 2000)



- 1946, Denis Gabor: STFT with Gaussian windows
- 1982, Jean Morlet: geophysics application, propose to replace the modulation by the dilation of a fixed function
- 1984, Alex Grossmann: link between Morlet's wavelet and coherent states in quantum physics + link with frame theory
- 1985, Yves Meyer (Gauss Prize 2010): link with harmonic analysis and establishment of mathematical foundations for a wavelet theory + discovery of the first orthonormal wavelet basis (1986)
- followers . . . : S. Mallat, I. Daubechies, R. Coifman, A. Cohen, . . .

Wavelets applications:

- All started in seismic signals analysis (events occur at different time and frequency regions ... and scales!)
- Soon become a standard technique for many *change detection* problems
- Wavelets appropriate for detecting changes, discontinuities, trends, etc
- They capture/describe the signal statistics with few components/coefficients: ideal for signal/image coding/compression and denoising/restoration

Wavelets main properties:

- Wavelets are invertible transforms
- Wavelets have two main parameters: scale and shift translation; more flexible than Fourier to study local behaviors of the signal!
- The basis functions in wavelets are time-limited (in Fourier sin/cos are extended $\pm\infty$)

Wavelets generalize time-frequency analysis and Gabor analysis

- The Gabor transform of a signal $x(t)$:

$$G_x(t, f) = \int_{-\infty}^{\infty} e^{-\pi(\tau-t)^2} e^{-j2\pi f\tau} x(\tau) d\tau$$

- Time-frequency analysis:

$$G_x(t, f) = \int_{-\infty}^{\infty} w(t - \tau) e^{-j2\pi f\tau} x(\tau) d\tau$$

- Wavelet analysis:

$$G_x(t, f) = \int_{-\infty}^{\infty} w(t - \tau) e^{-j2\pi f\tau} x(\tau) d\tau$$

but now the window is

$$w(t - \tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} x(t) \underbrace{\psi\left(\frac{t - \tau}{s}\right)}_{\text{mother}} dt$$

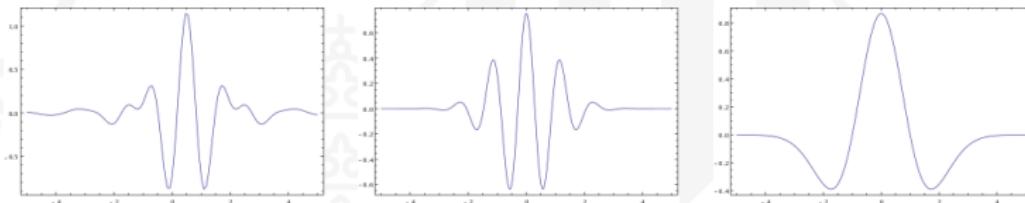
- The wavelet transform is simply a kind of correlation function between the mother wavelet $\psi(t)$, scaled and shifted, and the input signal

Continuous wavelet transforms:

- Wavelet analysis:

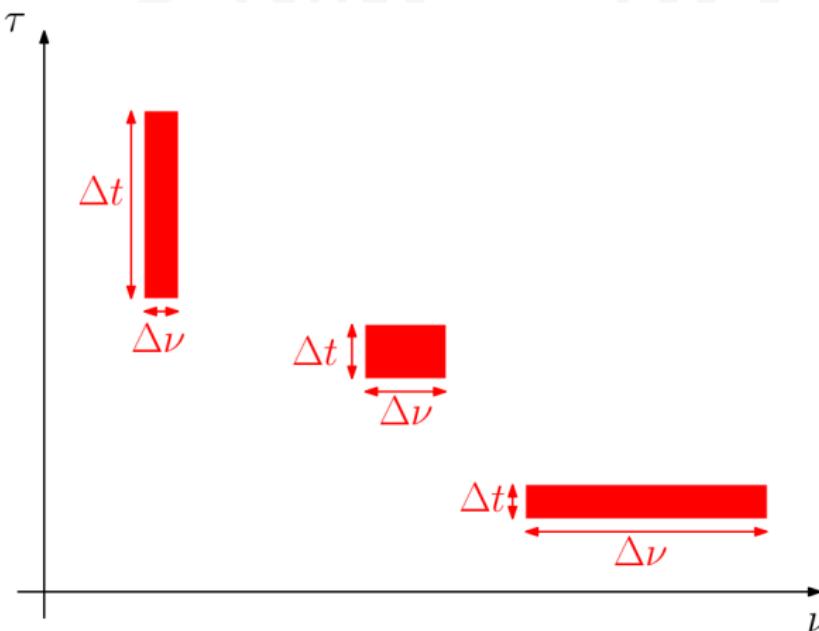
$$G_x(t, f) = \int_{-\infty}^{\infty} w(t) e^{-j2\pi f \tau} x(\tau) d\tau, \quad w(t) = \frac{1}{\sqrt{s}} \underbrace{\int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-\tau}{s}\right) dt}_{\text{mother}}$$

- The wavelet transform is simply a kind of correlation function between the mother wavelet $\psi(t)$, scaled and shifted, and the input signal
- Standard *mother wavelets*: Meyer, Morlet, Mexican hat:



- Intuition: The scale factor s will control the 'shape' of the mother wavelet: $s > 1$ dilates the wavelet and $s < 1$ compresses the wavelet in time
- This property is not shared by the Gabor or the Time-Frequency transforms

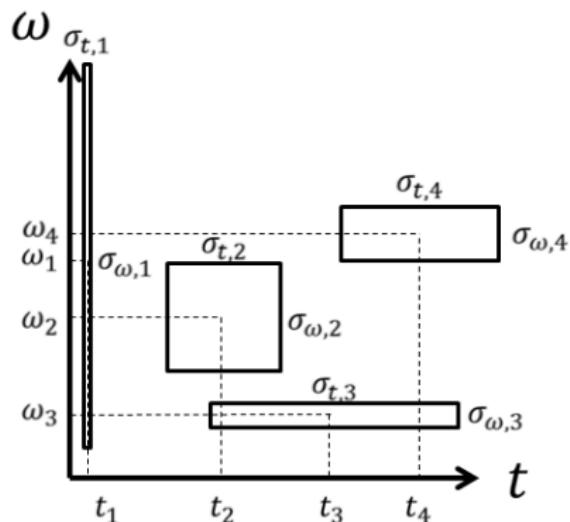
What do we gain with all this?



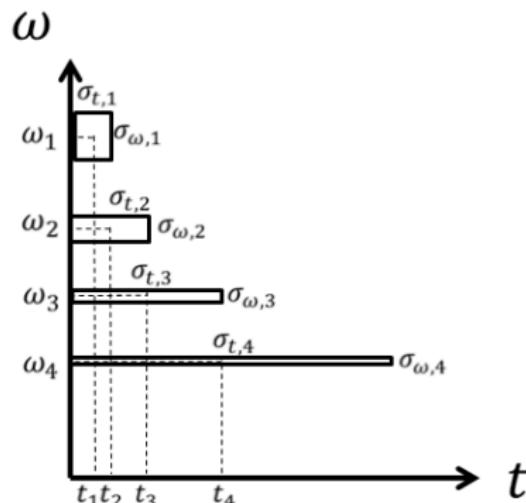
- Fast changes: low frequency resolution, high time resolution
- Slow changes: high frequency resolution, low time resolution

What do we gain with all this?

Four **distinct** STFT time-frequency atoms

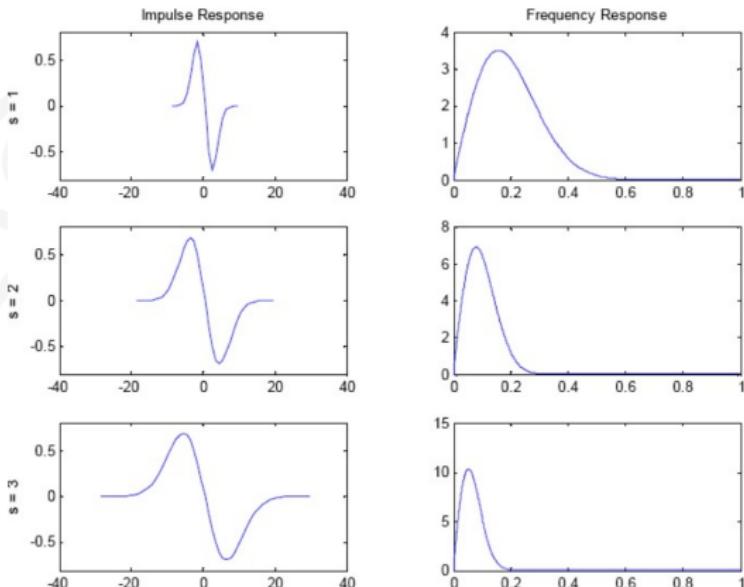


Set of one wavelet + children's multiresolutional time-frequency atoms

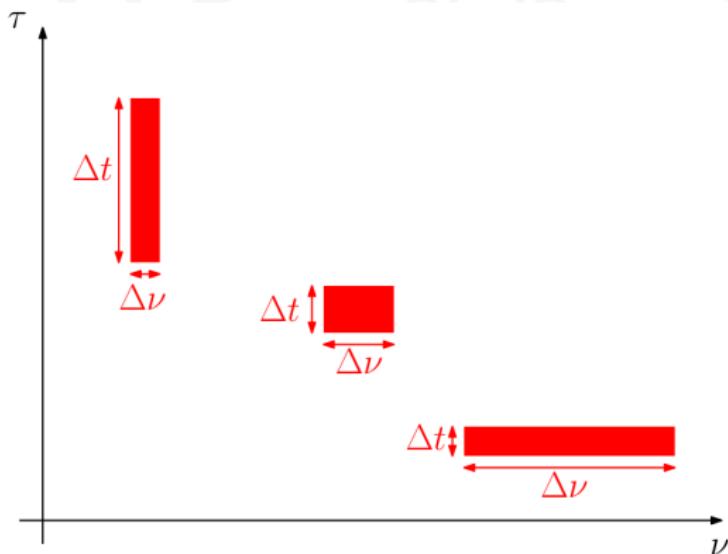


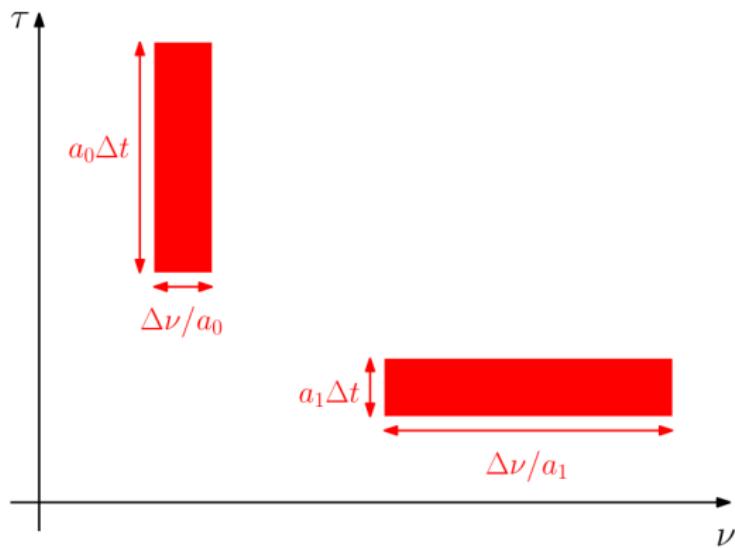
- Fast changes: low frequency resolution, high time resolution
- Slow changes: high frequency resolution, low time resolution

Intuition on the scale parameter:



- Low (time) scales is equivalent to study low frequency components, i.e. the rough features of the signal
- High (time) scales is equivalent to study high frequency components, i.e. the details in the signal
- There's a tradeoff between time scale and frequency resolution too!

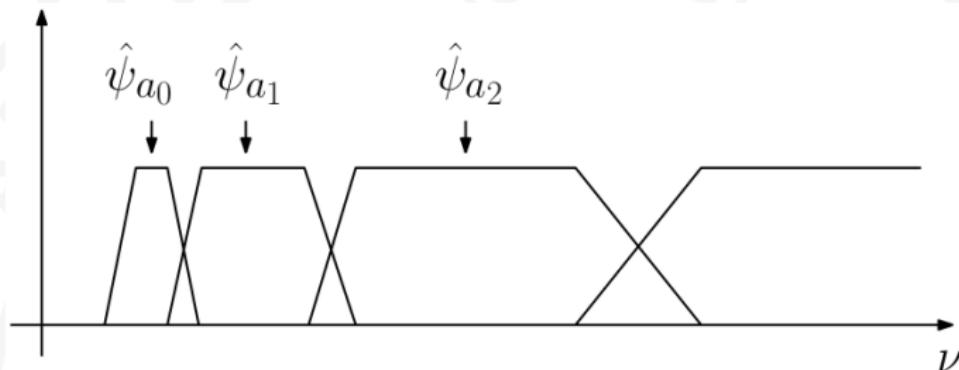
Time-frequency plane tiling:

Time-frequency plane tiling:

The discrete wavelet transform:

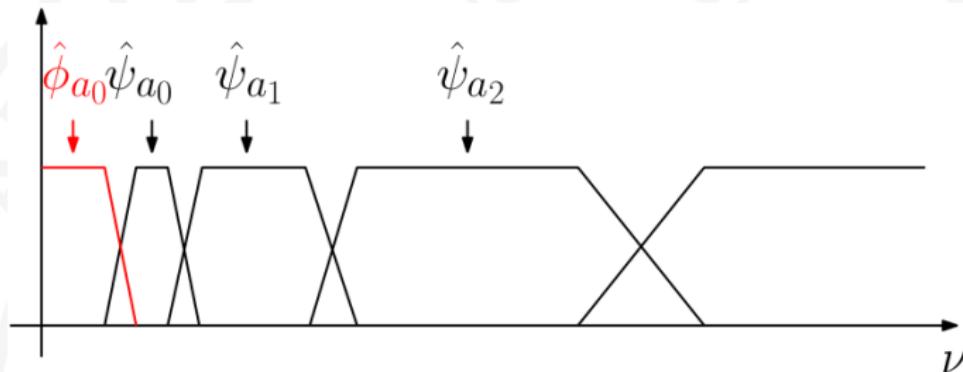
- When the signal is not continuous, just discretize the wavelets:
 - Sampled signals ($\text{timestep} = N - 1$).
 - Discrete scales: $(s_j, u_k) = \{2^j, k \cdot 2^j | j, k \in \mathbb{Z}\}$
 - Example: $N = 512$ samples and take $j = 3$, we study relations for $s = 8$ at positions $n = 8, 16, 32, \dots, 512$
- MATLAB: `wavedemo`

The wavelet transform is a band-pass filtering:



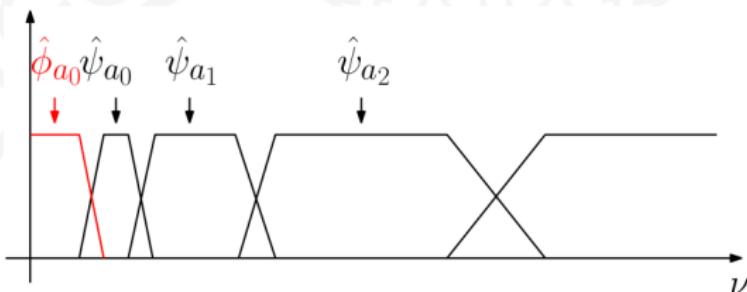
We cannot get the zero frequency

The wavelet transform is a band-pass filtering:



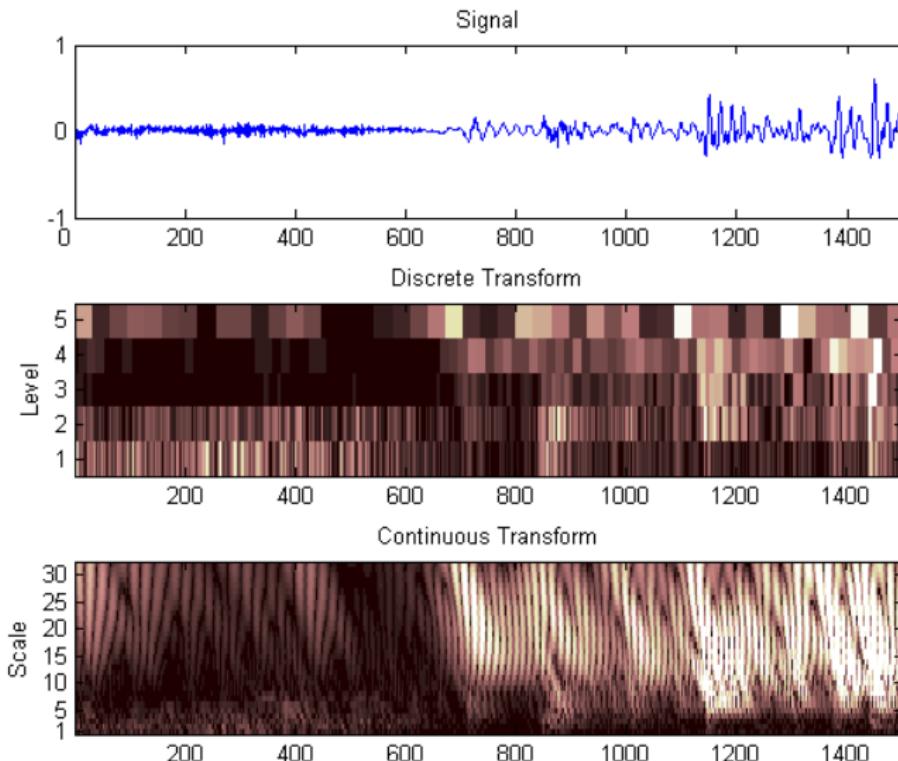
The missing part is obtained with the scaling function

The wavelet transform is a band-pass filtering:

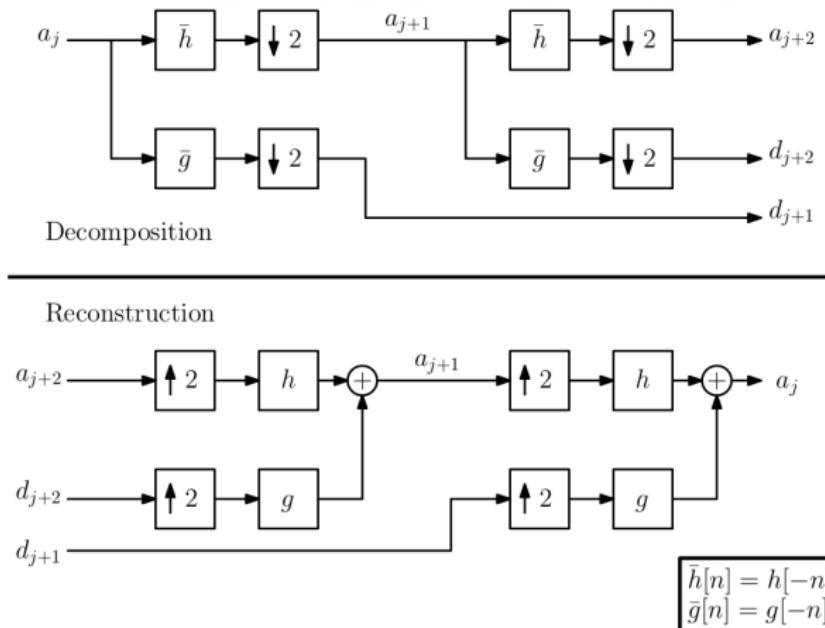


- WT are essentially a filter bank with different central frequencies and widths that increase with f
- WT can be casted as a spectral analyzer

Scaleogram or scalogram: visual method of displaying a wavelet transform: x representing time, y representing scale, and z representing wavelet coefficient value

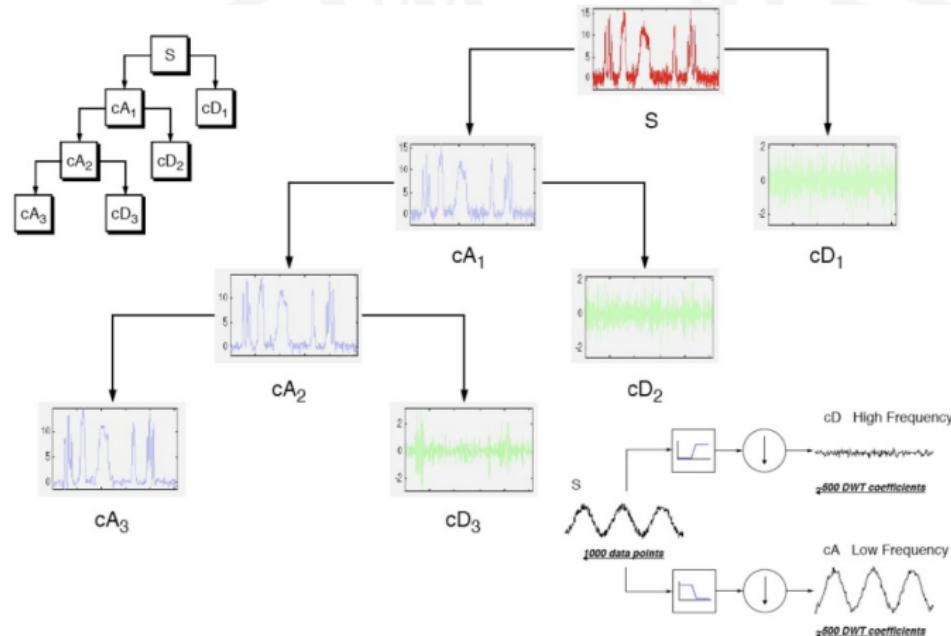


The discrete wavelet transform is a multiresolution spectral analyzer:



- DWT decompose the signal in ‘approximation’ (low f) and ‘detail’ (high f) coefficients
- Reconstruction of the signal from the coefficients is trivial; just reverse the operations

The DWT is a multiresolution spectral analyzer:



```
s = sin(20.*linspace(0,pi,1000)) + 0.5.*rand(1,1000);
[cA,cD] = dwt(s,'db2');
ss = idwt(cA,cD,'db2'); % Full reconstruction
ss = idwt(cA,zeros(1,501),'db2'); % Inverse using the LF approximation)
ss = idwt(zeros(1,501),cD,'db2'); % Inverse using the HF approximation)
```

Important wavelet features:

- Simple, fast implementation: Mallat's filterbank algorithm
- Mathematical properties: Riesz basis, vanishing moments,...
- Good modeling of the organization of the primary visual system

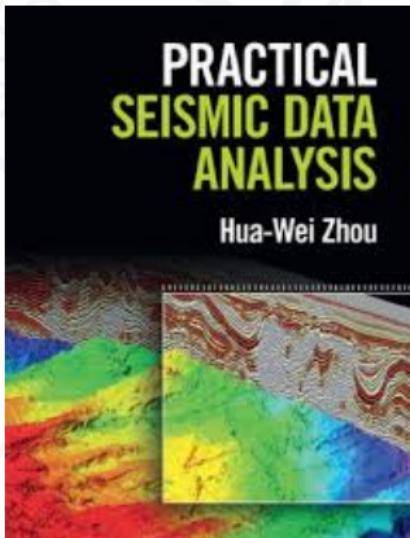
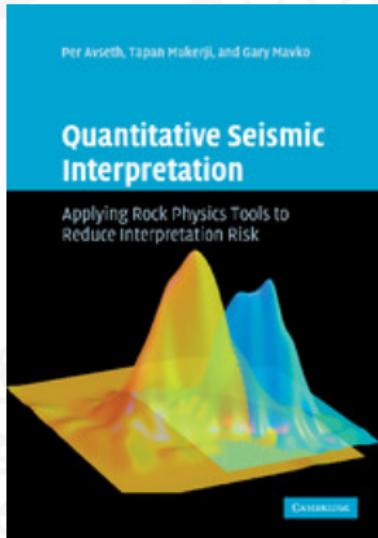
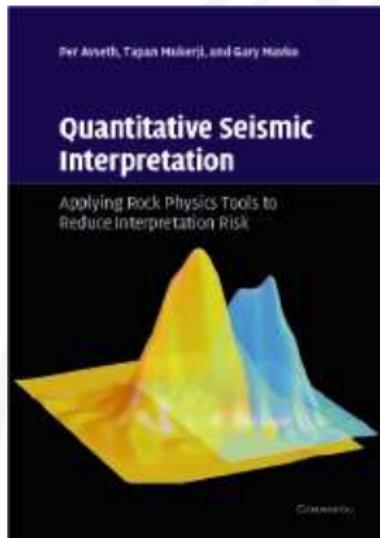
Many successful applications:

- Data compression
- Filtering, denoising
- Fusion
- Detection and feature extraction
- Inverse problems: wavelet regularization

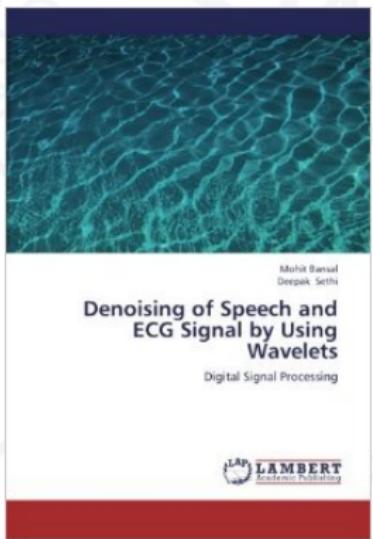
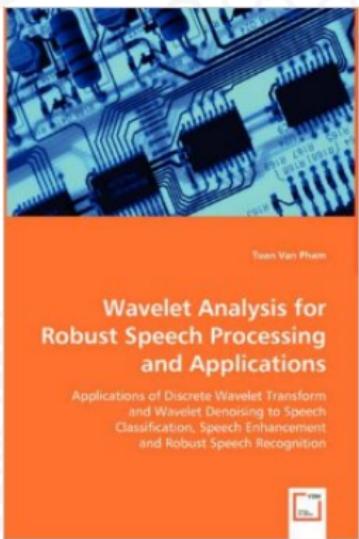
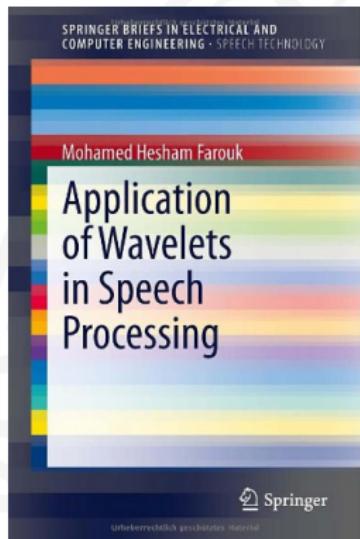
Current topics in wavelet research and “compressed sensing”

- Better wavelet dictionaries (frames): steerable wavelets, ...
- Better (model-based) regularization schemes
- Automatic parameter adjustment (e.g., scale-dependent threshold)
- Addressing harder inverse problems

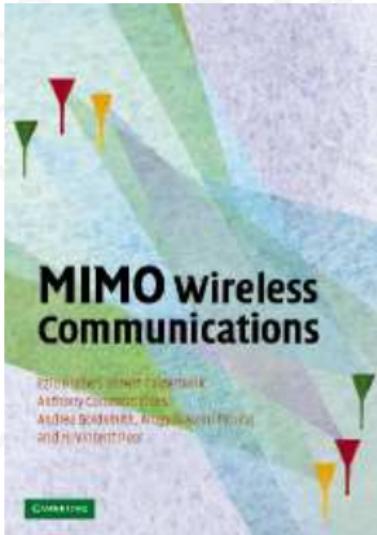
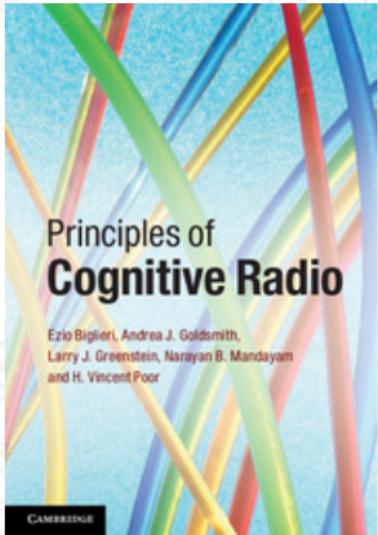
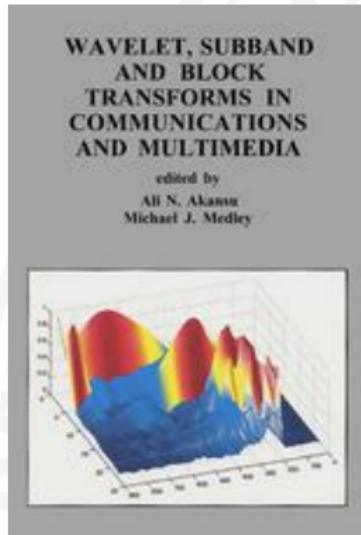
Seismic signal processing



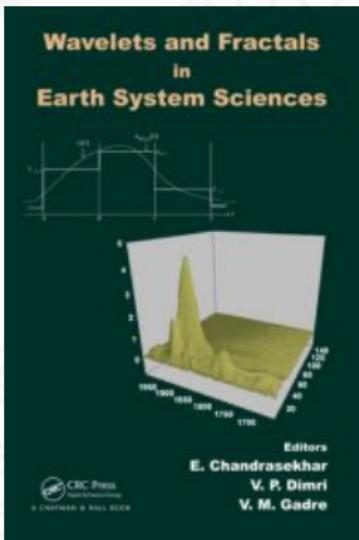
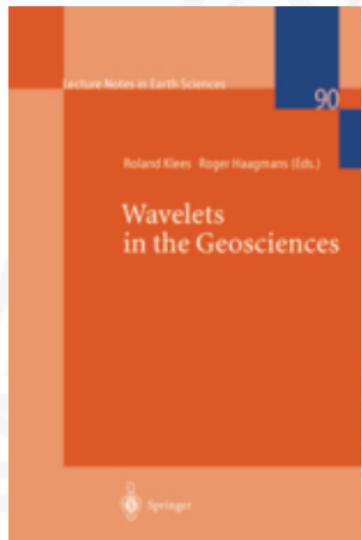
Audio processing



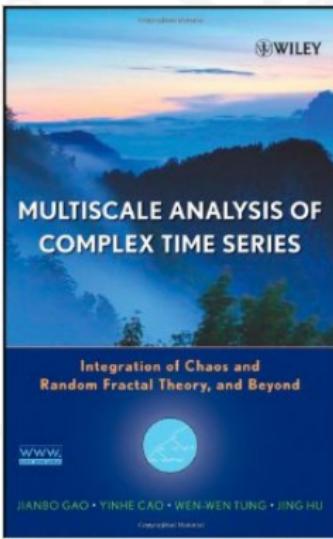
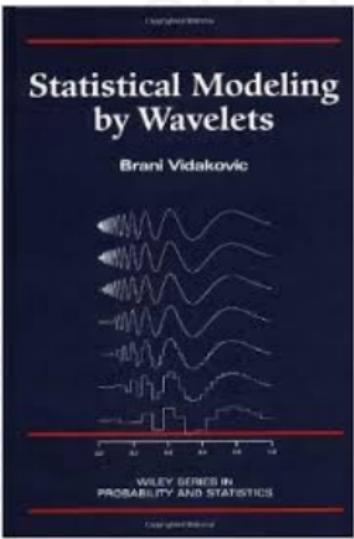
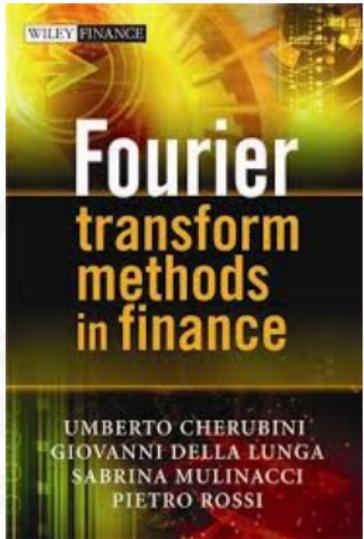
Communications



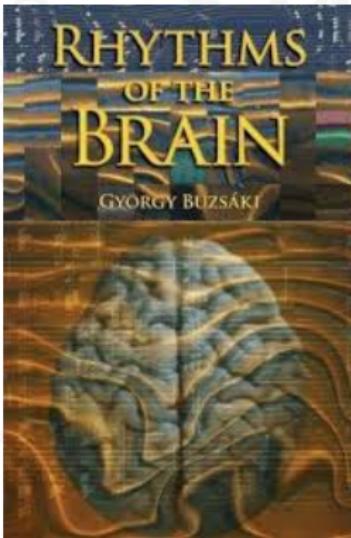
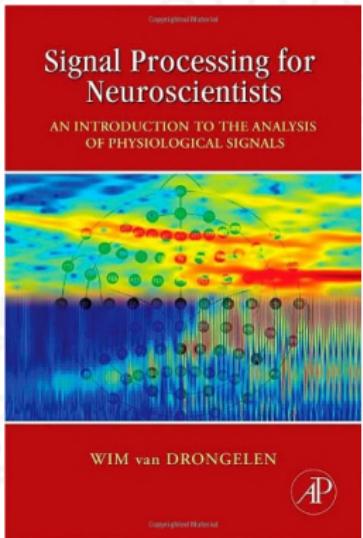
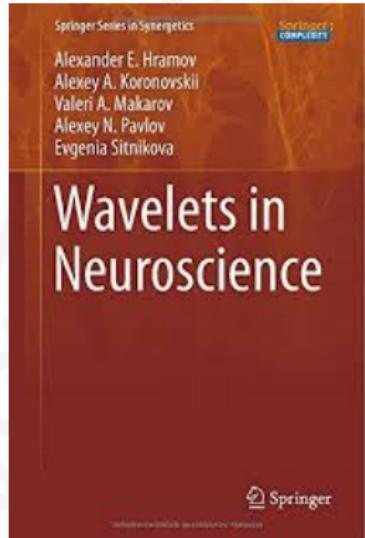
Geosciences

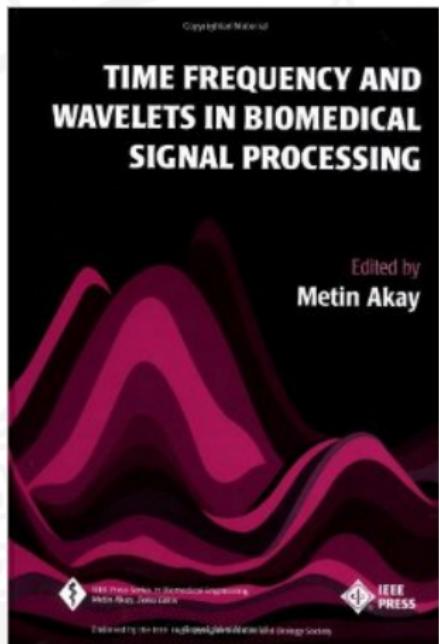


Times series analysis

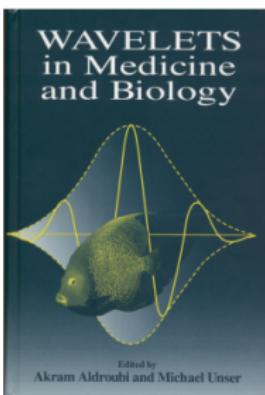


Neuroscience





Bioengineering



Wavelets in medical imaging: Survey 1991-1999

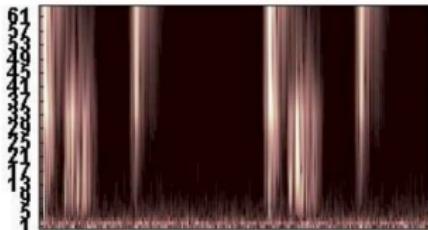
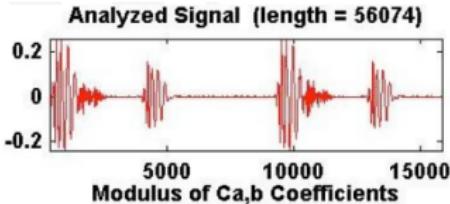
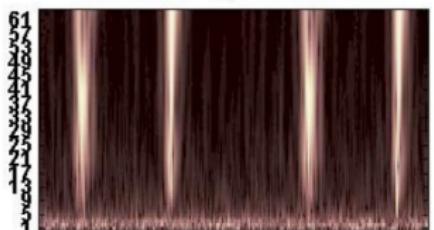
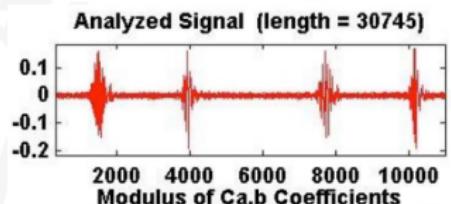
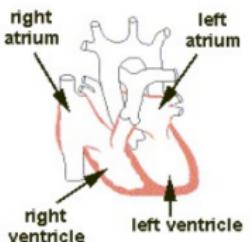
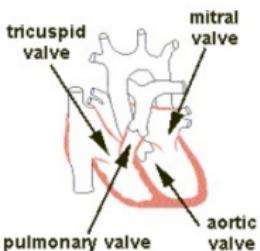
References

- Unser and Aldroubi, *Proc IEEE*, 1996
- Laine, *Annual Rev Biomed Eng*, 2000

- Special issue, *IEEE Trans Med Im*, 2003

Image processing task	Application / modality	Principal Authors
Image compression	<ul style="list-style-type: none"> • MRI • Mammograms • CT • Angiograms, etc... 	Angelis 94; DeVore 95; Manduca 95; Wang 96; etc ...
Filtering	<ul style="list-style-type: none"> <i>Image enhancement</i> • Digital radiograms • MRI • Mammograms • Lung X-rays, CT 	Laine 94, 95; Lu, 94; Qian 95; Guang 97; etc ...
	<ul style="list-style-type: none"> <i>Denoising</i> • MRI • Ultrasound (speckle) • SPECT 	Weaver 91; Xu 94; Coifman 95; Abdel-Malek 97; Laine 98; Novak 98, 99
Feature extraction	<ul style="list-style-type: none"> <i>Detection of micro-califications</i> • Mammograms 	Qian 95; Yoshida 94; Strickland 96; Dhawan 96; Baoyu 96; Heine 97; Wang 98
	<ul style="list-style-type: none"> <i>Texture analysis and classification</i> • Ultrasound • CT, MRI • Mammograms 	Barman 93; Laine 94; Unser 95; Wei 95; Yung 95; Busch 97; Mojsilovic 97
	<ul style="list-style-type: none"> <i>Snakes and active contours</i> • Ultrasound 	Chuang-Kuo 96
Wavelet encoding	<ul style="list-style-type: none"> • Magnetic resonance imaging 	Weaver-Healy 92; Panych 94, 96; German 96; Shimizu 96; Jian 97
Image reconstruction	<ul style="list-style-type: none"> • Computer tomography • Limited angle data • Optical tomography • PET, SPECT 	Olson 93, 94; Peyrin 94; Walnut 93; Delaney 95; Sahiner 96; Zhu 97; Kolaczyk 94; Raheja 99
Statistical data analysis	<ul style="list-style-type: none"> <i>Functional imaging</i> • PET • fMRI 	Ruttimann 93, 94, 98; Unser 95; Feilner 99; Raz 99
Multi-scale Registration	<ul style="list-style-type: none"> <i>Motion correction</i> • fMRI, angiography <i>Multi-modality imaging</i> • CT, PET, MRI 	Unser 93; Thévenaz 95, 98; Kybic 99
3D visualization	<ul style="list-style-type: none"> • CT, MRI 	Gross 95, 97; Muraki 95; Kamath 98; Horbelt 99

Bioengineering: Murmур detection: healthy vs pathologic



Bioengineering: Sleep phase detection from EEG recordings



Bioengineering: Removing noise from fMRI images

MAGNETIC RESONANCE IN MEDICINE 21, 288-295 (1991)

COMMUNICATIONS

Filtering Noise from Images with Wavelet Transforms

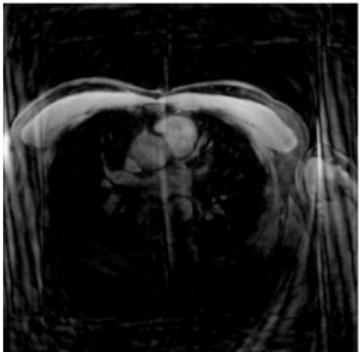
J. B. WEAVER,* YANSUN XU,* D. M. HEALY, JR.,† AND L. D. CROMWELL*

*Department of Radiology, Dartmouth-Hitchcock Medical Center; and †Department of Mathematics, Dartmouth College, Hanover, New Hampshire 03755

Received April 12, 1991

A new method of filtering MR images is presented that uses wavelet transforms instead of Fourier transforms. The new filtering method does not reduce the sharpness of edges. However, the new method does eliminate any small structures that are similar in size to the noise eliminated. There are many possible extensions of the filter. © 1991 Academic Press, Inc.

L_2 regularization (Laplacian)



ℓ_1 wavelet regularization

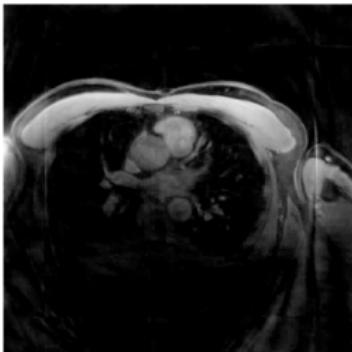


Image processing

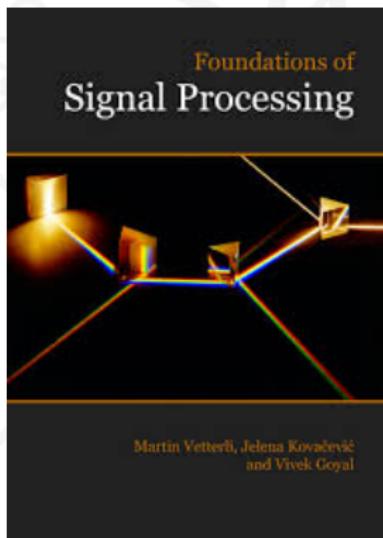
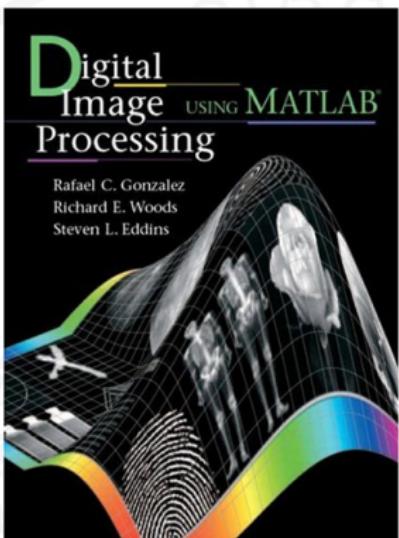
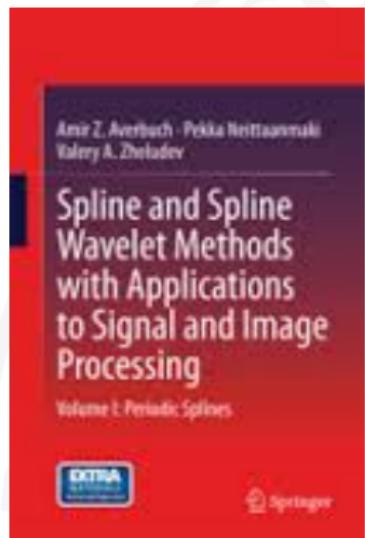
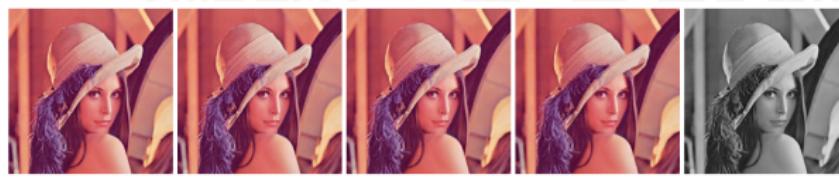
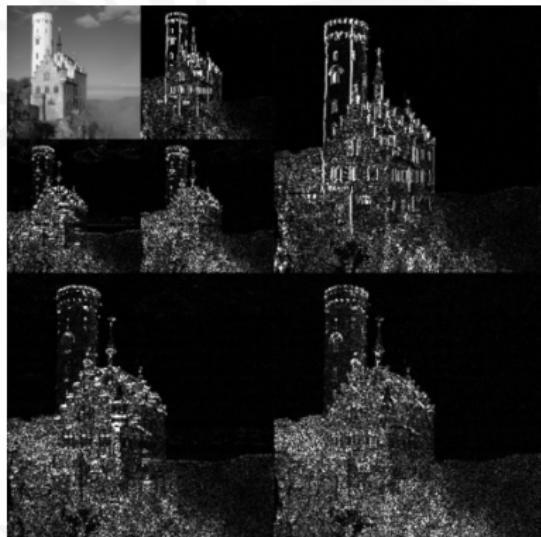


Image processing: coding/compression



5 bpp

3 bpp

2.25 bpp

1.75 bpp

2 bpp

Image processing: denoising/restoration

Noisy Image (0.46)



HT (0.67)



ST (0.66)



BG (0.67)



Image processing: image fusion

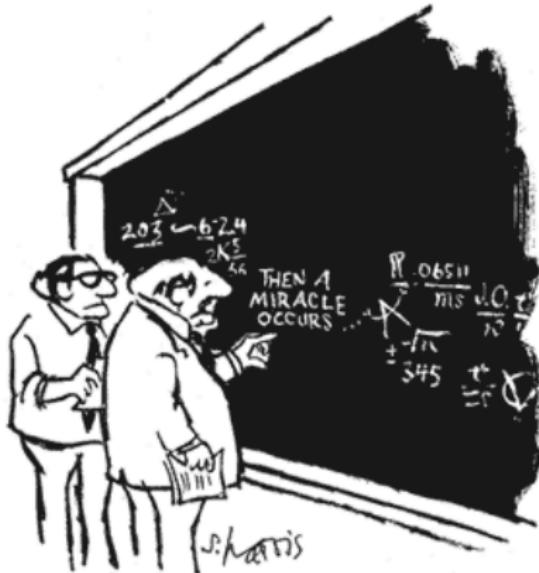


Image processing: multi-resolution fusion



Reviewed:

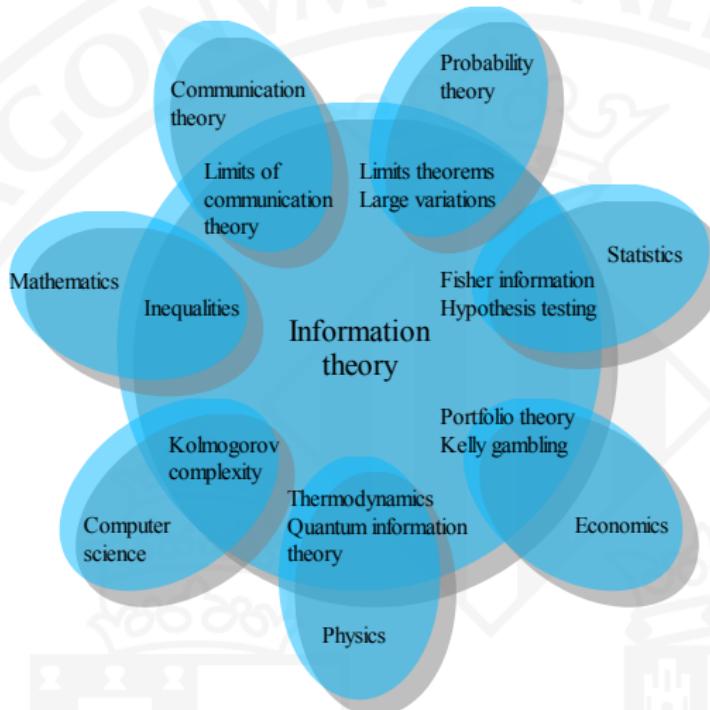
Non-stationary signals, Gabor filter and the Heisenberg uncertainty principle, Time-frequency analysis, Short Time Fourier Transform (STFT), Spectrogram, Uncertainty principle, Instantaneous frequency, Second order time-frequency relations, wavelets, multiresolution analysis, applications to signal/image processing.



"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

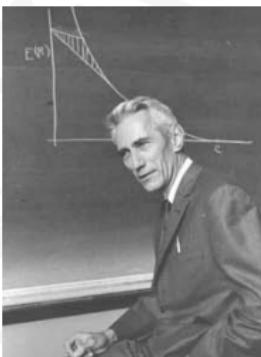
Part 5: Introduction to Information Theory

Information theory framework



Introduction

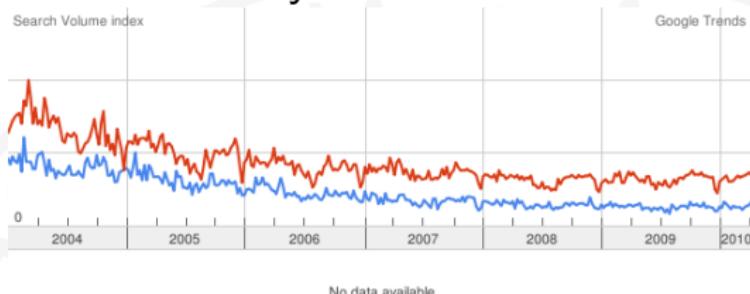
- *"Information theory is a branch of applied mathematics and electrical engineering involving the quantification of information."*
- **Claude E. Shannon** (1948) finds fundamental limits on signal processing operations, such as compressing data and reliably storing and communicating data



Tons of applications:

- statistical inference and machine learning
- signal/image processing: natural language processing, compression, estimation, ...
- communication: routing, transmission, networks, ...
- bio-things: neurobiology, bioinformatics, neuroscience, bioengineering, ...
- eco-things: ecology, remote sensing, environmental monitoring, ...
- physics: thermal, quantum computing, ...
- security: plagiarism detection, cryptography, ...

Resources on information theory



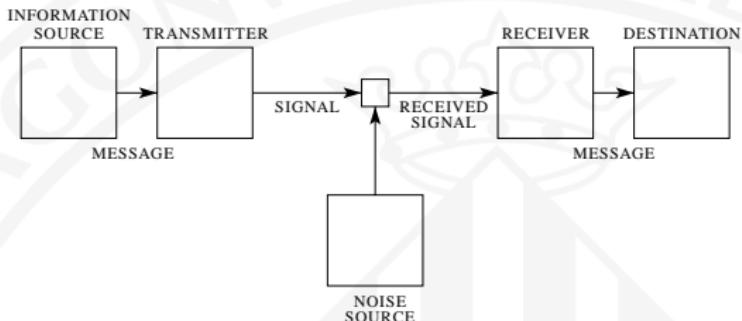
- Thomas M. Cover & Joy A. Thomas, *Elements of Information Theory*, Wiley & Sons, 1991
- David J.C. MacKay, *Information theory, inference and learning algorithms*, Cambridge University Press, 2004. Free at
<http://www.inference.phy.cam.ac.uk/mackay>
- IEEE Transactions on Information Theory
- <http://videolectures.net/>
- http://en.wikipedia.org/wiki/Information_theory
- <http://www.inference.phy.cam.ac.uk/mackay/>
- http://www.youtube.com/watch?v=z2Whj_nL-x8

5 equations that changed Science

- ① 2nd Newton law: $F = ma$
- ② Maxwell-Faraday equation: $\Delta E = -\frac{dB}{dt}$
- ③ Einstein's mass-energy equivalence: $E = mc^2$
- ④ Nyquist-Shannon theorem: $F_{sampling} \geq 2 \times B$
- ⑤ Shannon-Hartley equation: $C = B \log_2(1 + SNR)$

TWO OUT OF 5!

Communication Channels



Examples:

- Voice — AIR — ear
- spacecraft — VACUUM — Earth
- modem1 — WIRE — modem2
- file — HDD — file
- transmitted signal — CHANNEL — received signal (=transm.+noise)

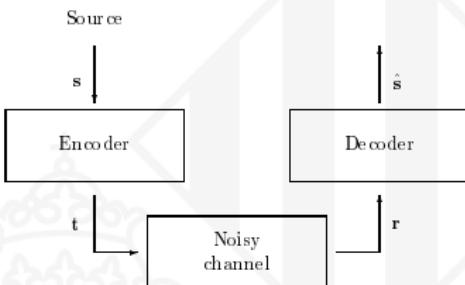
Main concern: 'reliable communication over unreliable channel'

Solutions to the communication problem

- *Physical solution*

- thicker films
- higher magnetic field \vec{B}
- more bandwidth
- more \$!!!

- *System solution*

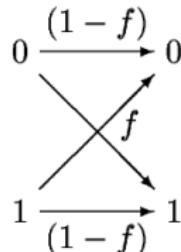


- The encoder adds redundancy
- The channel adds noise
- The decoder decodes s and n , hence it does *inference*

Problem 1. The binary symmetric channel (BSC)

$$\begin{array}{ccc} x & \xrightarrow{\quad} & 0 \\ & \times & \\ & \xleftarrow{\quad} & y \\ 1 & & 1 \end{array} \quad \begin{array}{llll} P(y=0|x=0) & = & 1-f; & P(y=0|x=1) & = & f; \\ P(y=1|x=0) & = & f; & P(y=1|x=1) & = & 1-f. \end{array}$$

- Probability graph of the flip
- f is the probability of a wrong flip

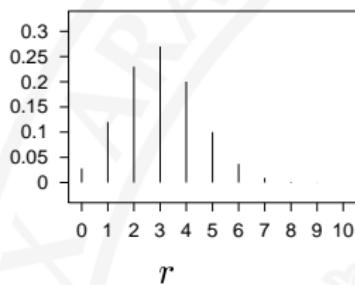
Problem 1: The binary symmetric channel ...

Q: A file of $N = 10\,000$ bits is stored on this disc drive (with $f = 0.1$), then read.

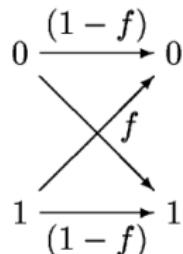
Roughly how many bits are flipped?

$$\boxed{} \pm \boxed{}$$

The binomial distribution



- “The binomial (Bernoulli) distribution is the discrete probability distribution of the number of successes/failures in a sequence of n independent yes/no experiments each with probability p ”
- We have $n = 10000$, $f = p = 0.1$ then ...
- Mean: $\bar{x} = np = 1000$
- Variance:
$$\sigma_x^2 = np(1 - p) = 900 \rightarrow \sigma = 30$$
- Solution: $\bar{x} \pm \sigma_x = 1000 \pm 30$
- Sometimes: $\bar{x} \pm 2\sigma_x = 1000 \pm 60$

Problem 2: The binary symmetric channel ...

Q: To make a successful business selling 1 Gigabyte disc drives, how small does the flip probability f need to be?

The binomial distribution

- Successful means no error for the living time of the device
- We have $n = 1 \text{ Gb} = 10^9 \cdot 8 \text{ bits}$, then ...
 - **Trivial solution:** if we want the HDD live forever without error, then ...

$$\bar{x} \pm \sigma_x = 0 \pm 0$$

- **Realistic solution:**

$$f = \frac{1}{1\text{Gbyte/day} \times 365\text{days/year} \times 5\text{years} \times 10^6\text{customers}} \approx 10^{-19}$$

- The standard in HDD and storage devices is $f = 1/10^{18}$ errors !
- Let's look for just $f = 1/10^{15}$ errors !

Repetition code ' R_3 '

- A trick for building a successful encoder is repetition!
- Example of the repetition code ' R_3 ':

Source sequence s	Transmitted sequence t
0	000
1	111

- We transmit the source message $s = 0\ 0\ 1\ 0\ 1\ 1\ 0$ over a binary symmetric channel (BSC) with noise level $f = 0.1$ using R_3 .
- A possible noise vector n and received vector $r = t + n$:

$$\begin{array}{ccccccc}
 s & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
 t & \overbrace{000} & \overbrace{000} & \overbrace{111} & \overbrace{000} & \overbrace{111} & \overbrace{111} & \overbrace{000} \\
 n & 000 & 001 & 000 & 000 & 101 & 000 & 000 \\
 \hline
 r & 000 & 001 & 111 & 000 & 010 & 111 & 000
 \end{array}$$

- How to decode this received vector to obtain a good estimate of s ?

Ideas for a decoder

- Possibility 1: read the middle and discard the rest

Received r	Estimated \hat{s}
111	1
110	1
101	0
000	0
...	...

- Possibility 2: majority vote: 'find the hypothesis about s that involves least flips'

Received r	Estimated \hat{s}
111	1
110	1
101	1
000	0
...	...

- Possibility 3: learn a neural network or SVM → overfitting!

Possibility 4: use Bayes' theorem with a reasonable prior

$$p(s = 0) = p(s = 1) = 0.5$$

$$\begin{aligned} p(s = 0|r = 011) &= \\ \frac{p(r = 011|s = 0)p(s = 0)}{p(r = 011|s = 0)p(s = 0) + p(r = 011|s = 1)p(s = 1)} &= \\ \frac{(1-f)f \cdot f \cdot \frac{1}{2}}{(1-f)f \cdot f \cdot \frac{1}{2} + f \cdot (1-f) \cdot (1-f) \cdot \frac{1}{2}} &= \dots = f = 0.1 \end{aligned}$$

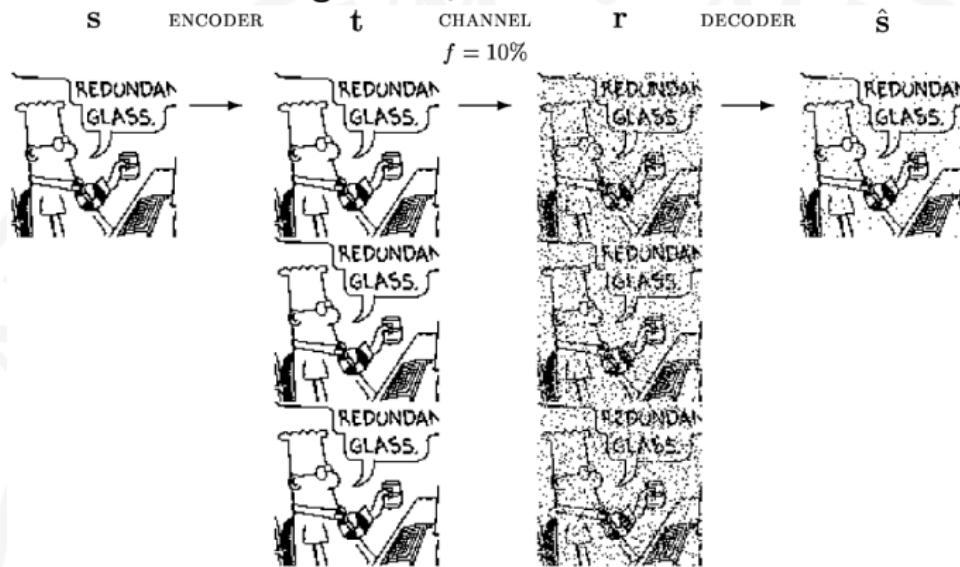
where the likelihood is (bits not to be flipped) \times (flip) \times (flip)

Possibility 2 revisited: the majority vote encoder

s	0	0	1	0	1	1	0
t	0 0 0	0 0 0	1 1 1	0 0 0	1 1 1	1 1 1	0 0 0
n	0 0 0	0 0 1	0 0 0	0 0 0	1 0 1	0 0 0	0 0 0
r	0 0 0	0 0 1	1 1 1	0 0 0	0 1 0	1 1 1	0 0 0
\hat{s}	0	0	1	0	0	1	0
corrected errors			*				
undetected errors					*		

- Quite robust to noise: repetition is great!
- Not all errors are corrected: depends on block sizes too...

Illustration of the decoding with R_3 encoder ...



- The error probability is dominated by the probability that two bits in a block of three are flipped, which scales as f^2
- In the case of the BSC with $f = 0.1$, the R_3 code has a probability of error after decoding of $p \approx 0.03$ per bit

Majority vote decoder in R_2

- Probability of $p(\mathbf{s} \neq \hat{\mathbf{s}})$ in R_2 for $f < 1$:

$$p_{R_2} \approx f^2 + f(1-f) + (1-f)f \approx 2f$$

- Probability of $p(\mathbf{s} \neq \hat{\mathbf{s}})$ in R_3 for $f < 1$:

$$p_{R_3} \approx \underbrace{f^3}_{3\text{flips}} + \underbrace{f(1-f)f + ff(1-f) + (1-f)ff}_{2\text{flips}} \approx 3f^2$$

- MATLAB:

```
>> f=0.1  
>> p2 = f*f + 2*(1-f)*f = 0.1900  
>> p3 = f*f*f + 3*f*f*(1-f) = 0.0280
```

- Why not going further and increase the repetition to R_n ?

Repetition code R_N

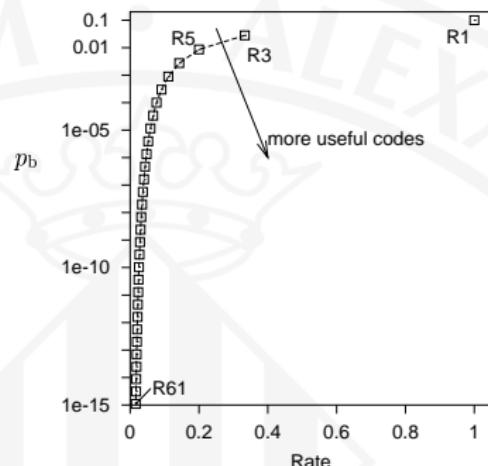
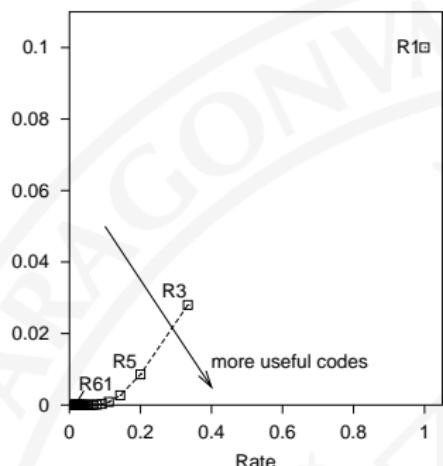
- Probability of $p(s \neq \hat{s})$ in R_N for $f < 1$ is dominated by the probability that $[N/2]$ bits are flipped:

$$p_b \approx 2^N (f(1-f))^{N/2} = (4f(1-f))^{N/2}$$

- Setting this equal to the required value of $p_b = 10^{-15}$, we find that:

$$N \approx 2 \frac{\log_2(10^{-15})}{\log_2(4f(1-f))} = 68$$

- Better estimate without approx.: $N \approx 61$ to get $p_b = 10^{-15}$
- So ... a trick would be to hide in a big box 60 hard disk drives! :)



- We have a nice encoder: repetition gives rise to zero error probability!
- Problem: we use the channel n times or send three times more information
- Distortion = error
- Rate = efficiency: $\frac{\text{\#bits to be sent}}{\text{\#times we use the channel}}$
- This is the rate-distortion problem

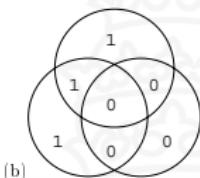
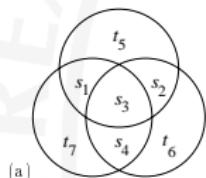
Redundancy without repetition?

- Parity checks: 'add a bit for checking' (mod_2). Here the rate is $R = 3/4$

Source r ($k = 3$)	transmitted t ($N = 4$)
111	1111
110	1100
101	1010
...	...

- Hamming (7,4)-codes:

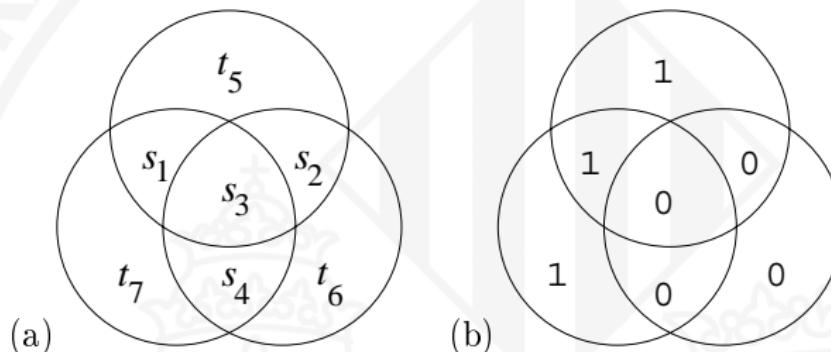
- 'Add three parity checks', i.e map $\mathbb{R}^4 \rightarrow \mathbb{R}^7$:
- The number of '1' must even in each circle!
- Linearity property: $1000 + 0111 = 1111$, i.e. $s_1 + s_2 = t_1 + t_2$



Source r ($k = 3$)	transmitted t ($N = 4$)
1000	1000101
0111	0111010
1111	1111111
...	...

Let's encode/decode with (7,4)-Hamming code

- $s = 1000101 \rightarrow r = 1100101 \rightarrow \hat{s} = 1000101$
- Steps:
 - Take the diagram and ask about fulfilment
 - All circles must be happy if no flip occurs
 - This is called the 'syndrome'
 - The decoder locates the common bit between circles and unflips it!



- Property 1: for one bit flip, H(7,4) can detect the error and correct it
- Property 2: for more than 2 errors, $p(s_i \neq \hat{s}_i) \approx 9f^2$

More on Hamming code ...

- The Hamming code is a linear code, it can be written compactly in matrix notation:

$$\mathbf{t} = \mathbf{s}\mathbf{G}$$

where \mathbf{G} is the *generator matrix* of the code,

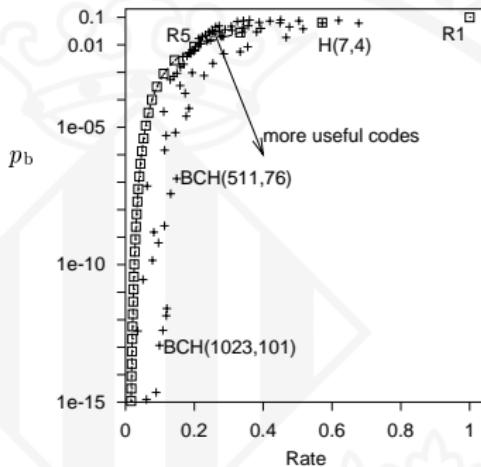
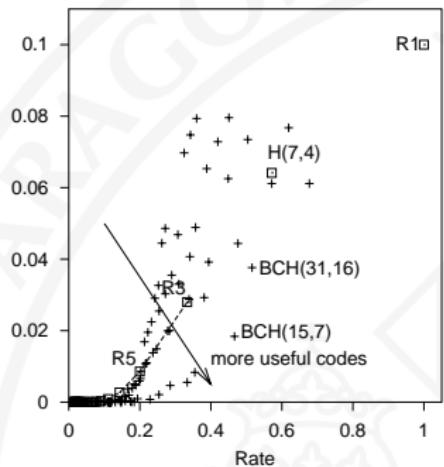
$$\mathbf{G}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

- The encoding operation uses mod_2 arithmetic:

$$1 + 1 = 0$$

$$0 + 1 = 1$$

- The rows are like the four basis vectors lying in a 7D binary space
- The 16 codewords are obtained by linear combinations of these vectors
- Linear algebra very useful to solve the so-called *maximum-likelihood decoder*



Reprinted with corrections from *The Bell System Technical Journal*,
Vol. 27, pp. 379–423, 623–656, July, October, 1948.

A Mathematical Theory of Communication

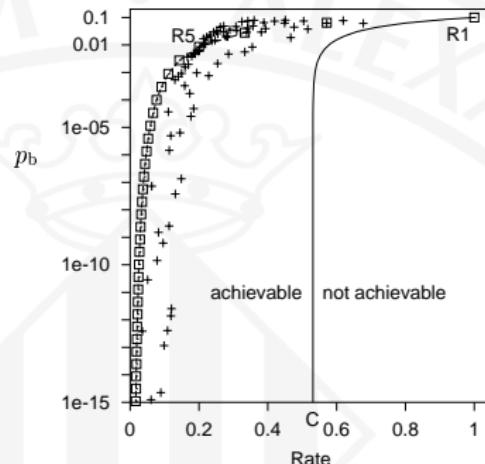
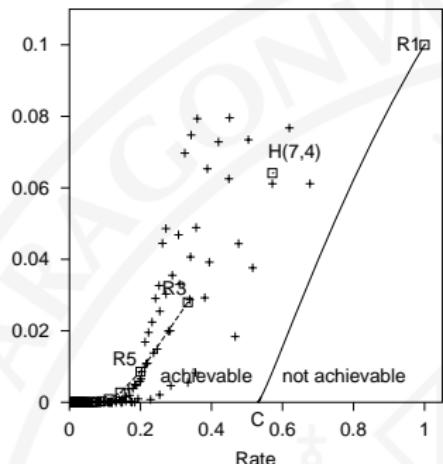
By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist¹ and Hartley² on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

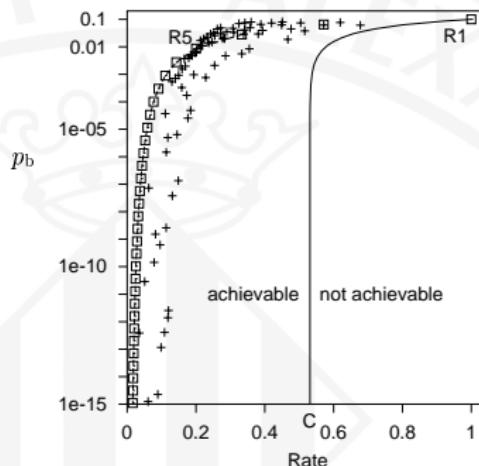
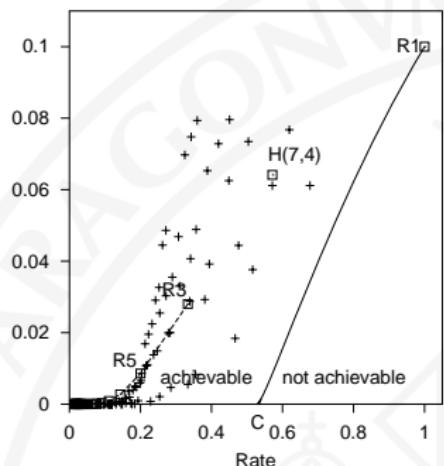
The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

Shannon's noisy channel coding theorem



- After WW-2: The achievability curve should cross the origin, i.e. '*no free lunch*' or '*no pain, no gain*'
- Shannon:** "For any channel, reliable (virtually error-free) communication is possible at rates up to C "
- Intuition:** 'one can design an encoder such that any $R < C$ '
- Watch out!**: nothing said about channel complexity (non-constructive th.)

Shannon's noisy channel coding theorem



For the binary symmetric channel (BSC) and $f = 0.1$:

$$C_{\text{BSC}} = 1 - H_2(f) \approx 0.53$$

$$H_2(f) = f \cdot \log_2 \left(\frac{1}{f} \right) + (1-f) \log_2 \left(\frac{1}{1-f} \right)$$

Shannon's noisy channel coding theorem

Problem: Suppose we want to sell 1Gbyte hdd with a $p_b = 10^{-15}$

- Gustau: We showed that 60 **noisy disk drives** would meet the specification to achieve 1 **high-quality disk drive**
—forget about the money, the size and trying to convince the client!
- Shannon:

'What performance are you trying to achieve? 10^{-15} ? You don't need *sixty* disk drives: you can get that performance with just *two* disk drives (since $1/2$ is less than 0.53). The capacity for $f = 0.1$ is 0.53 , so the number of disk drives needed at capacity is $1/0.53 = 1.88$. And if you want $p_b = 10^{-18}$, or 10^{-21} , or 10^{-24} or anything, you can get there with two disc drives too!'

- Gustau:
'Are you kidding me? your theorem is only useful for sequences of block codes with ever increasing blocklengths, and to achieve that rate you should use blocklengths bigger than 1 Gbyte!'
- Shannon:
'I agree: you cannot do it with such tiny disk drives but... if you had two noisy terabyte drives, you could make a single high-quality terabyte from them'
- Gustau:
'Ummmm... you're right!'

How to compress a redundant file?

e.g., $N = 1000$ tosses of a bent coin with $p_1 = 0.1$

Key questions:

- Compressibility of a dataset!?
 - How to measure the information content?
 - How much compression could be expected?
 - Why fixed rates for transmission/storage? Speech, images, time series, ...

Entropy codes ...

- are simple to implement
 - put more effort in low probability events ...
 - are used jointly with scalar quantizers (to conserve the average bit rate)
 - based on **INFORMATION**

What's information?

- Information is the reduction of uncertainty
- Some (informal) axioms:
 - ① if something is certain its uncertainty = 0
 - ② uncertainty should be maximum if all choices are equally probable
 - ③ uncertainty (information) should add for independent sources



How to measure information content?

- Let X be a random variable whose outcome x takes values in $\{a_1, \dots, a_L\}$ with probabilities $\{p_1, \dots, p_L\}$
- Shannon's information content for the outcome $x = a_i$:

$$H(x = a_i) = \log_2 \left(\frac{1}{P(x = a_i)} \right) = \log_2 \left(\frac{1}{p_i} \right)$$

is a sensible measure of information content

- The entropy

$$H(X) = \sum_i p_i \log_2 \left(\frac{1}{p_i} \right) = - \sum_i p_i \log_2(p_i)$$

is a sensible measure of expected (**average**) information content

- Entropy is measured in:
 - bits (**binary digits**) if base 2 log is used
 - nats (**natural digits**): natural (base e) log.
- Good things to do, but not the only one!

Measuring information content ...

- How many bits needed to compress your data?
- Shannon's information content:

$$H(x = a_i) = \log_2 \left(\frac{1}{P(x = a_i)} \right)$$

- Example: Observe a sequence '...00000100' with $p_1 = 0.1$ (or $p_0 = 0.9$):

$$H(x = 1) = \log_2 \left(\frac{1}{0.1} \right) = 3.3 \text{ bits}$$

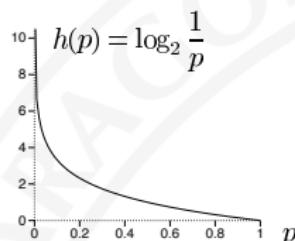
$$H(x = 0) = \log_2 \left(\frac{1}{0.9} \right) = 0.15 \text{ bits}$$

- Intuition:

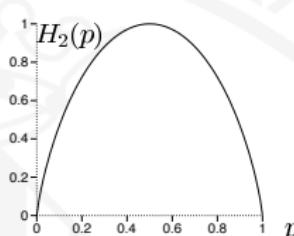
- The '1' has less information, you don't get too much *surprised* with a 0!
- You don't *learn* too much with a 0!
- The '1' is more improbable, more surprising, more informative!

Axiom 1-2: Information and uncertainty

- Consider a binary random variable that can take two values with probabilities p and $1 - p$.



p	$h(p)$	$H_2(p)$
0.001	10.0	0.011
0.01	6.6	0.081
0.1	3.3	0.47
0.2	2.3	0.72
0.5	1.0	1.0



- MATLAB:

```
>> p=hist(x,b);
>> h=log2(1./p);
>> H=p.*log2(1./p) + (1-p).*log2(1./(1-p));
>> figure(1),plot(p,h)
>> figure(2),plot(p,H)
```

- Improbable events are more informative, but less frequent on average
- The entropy satisfies the two first axioms
 - observation of a certain event carries no information
 - maximum information is carried by uniformly probable events

Axiom 3: Information under independence

- What about more than one variable?
- Example: we learn two variables $\{x, y\}$ that are independent, then

$$P(x, y) = P(x)P(y)$$

- Shannon's information content is:

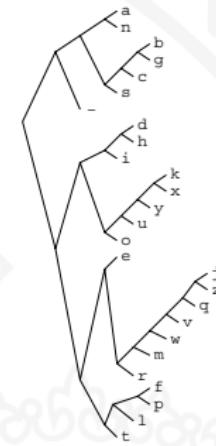
$$H(x, y) = \log_2 \left(\frac{1}{P(x, y)} \right) = \log_2 \left(\frac{1}{P(x)} \right) + \log_2 \left(\frac{1}{P(y)} \right) = H(x) + H(y)$$

- **Additive property:** If variables are independent, the information content is the sum of their informations!

Huffman algorithm

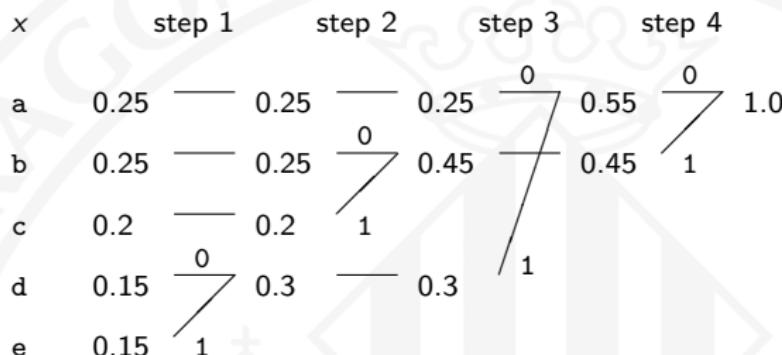
- Huffman (1952), “A Method for the Construction of Minimum-Redundancy Codes”
- Huffman coding is an entropy encoding algorithm used for lossless data compression
- Huffman encoding gives the optimal compression for any distribution
- Huffman coding uses a specific method for choosing the representation for each symbol
- The most common characters use shorter strings of bits
- It is optimal if the representation rates are preserved

a_i	p_i	$\log_2 \frac{1}{p_i}$	l_i	$c(a_i)$
a	0.0575	4.1	4	0000
b	0.0128	6.3	6	001000
c	0.0263	5.2	5	00101
d	0.0285	5.1	5	10000
e	0.0913	3.5	4	1100
f	0.0173	5.9	6	111000
g	0.0133	6.2	6	001001
h	0.0313	5.0	5	10001
i	0.0599	4.1	4	1001
j	0.0006	10.7	10	1101000000
k	0.0084	6.9	7	1010000
l	0.0335	4.9	5	11010
m	0.0235	5.4	6	110101
n	0.0596	4.1	4	0001
o	0.0689	3.9	4	1011
p	0.0192	5.7	6	111001
q	0.0008	10.3	9	1101000001
r	0.0508	4.3	5	10101
s	0.0567	4.1	4	0011
t	0.0706	3.8	4	1111
u	0.0334	4.9	5	10101
v	0.0069	7.2	8	11010001
w	0.0119	6.4	7	1101001
x	0.0073	7.1	7	1010001
y	0.0164	5.9	6	101001
z	0.0007	10.4	10	1101000001
-	0.1928	2.4	2	01



Example 1: Huffman algorithm

Let $A_X = \{ a, b, c, d, e \}$
 and $P_X = \{ 0.25, 0.25, 0.2, 0.15, 0.15 \}$.



The codewords are obtained by reverse concatenation, $C = \{00, 10, 11, 010, 011\}$.

a_i	p_i	$H(p_i)$	l_i	$c(a_i)$
a	0.25	2.0	2	00
b	0.25	2.0	2	10
c	0.2	2.3	2	11
d	0.15	2.7	3	010
e	0.15	2.7	3	011

Example 2: Huffman algorithm in MATLAB

```
>> help huffmandict
>> symbols = [1:5]
>> prob = [.3 .3 .2 .1 .1]
>> dict = huffmandict(symbols,p); % Create the dictionary.
>> hcode = huffmanenco(sig,dict); % Encode the data.
>> dhsig = huffmandeco(hcode,dict); % Decode the code.
```

Symbol codes, recap.

- Simple way to compress things
- Everything in the alphabet will be given a simple word
- Essentially, Huffman gives short codes to most probable things
- Huffman makes optimal symbol codes! (not trivial to show)

Notes

- The receiver has to know how to decode: either having a table or to know the encoding rule (e.g. a header bit)
- How to decode? go from top of the tree to the leaves
- Vast literature on error correcting codes (flips corrections)
- There are some cases where compression and encoding are merged (e.g. Mackay-Nils code)

Problems with Huffman codes

- Huffman coding is optimal for a symbol-by-symbol coding (i.e. a stream of unrelated symbols)
- Symbol coding fails for extreme distributions!
- What if the PDF changes?
 - not identically distributed, (e.g. 'a' is far much more common than 'z')
 - not independent (e.g., 'cat' is more common than 'cta')
 - over time, context-dependent, adaptive (learning), ...
- Arithmetic coding and Lempel-Ziv-Welch (LZW) coding often have better compression capability

Solutions for Huffman codes

- Grouping symbols can help in changing environments
- Block-wise Huffman coding solves changes in repetition rates
- Huffman coding is widely used because of its simplicity, high speed and patent-free
- Huffman coding is often used as a 'back-end' to PKZIP, JPEG and MP3 compression

Run-length encoding (RLE)

- Simple way to encode things
- Runs (repetitive sequences) of data are stored as a single data value and count, rather than as the original run:

WWWWWWWWWWWWBWWWWWWWWWWWWBWWWWWWWW

WWWWWWWWWWWWWWWWBWWWWWWWWWWWWWWWW

is encoded as

12W1B12W3B24W1B14W

- The run-length code represents the original 67 characters in only 18!
- It is also useful for binary streams
- It is well suited to palette-based iconic images
- Common formats for RLE: Truevision TGA, PackBits, PCX and ILBM.
- JPEG uses it with the coefficients remaining after transform and quantization
- RLE is used in faxes!
- RLE also applied to low-quality audio signals, just after a predictive filter

"Entropy is a measure of how organized or disorganized a system is: Gain of entropy eventually is nothing more nor less than loss of information"

Entropy in thermodynamics

- entropy is measured in [J/K] Joules/Kelvin
- machines are basically energy conversion devices
- Greek $/\varepsilon\nu\tau\rho\omega\pi\alpha/$ means 'conversion', 'change'
- systems tend to progress to higher entropy, change, conversion

Entropy in statistical mechanics

- entropy is a measure of the number of ways to arrange a system
- measure of 'disorder' (the higher the entropy, the higher the disorder)
- amount of order, disorder, and/or chaos in a system

Entropy in other fields of science

- Ecological entropy is a measure of biodiversity
- Social entropy is a measure of the natural decay within a social system
- Neurological entropy is the likelihood of patient's consciousness

Remember: How to measure information content?

- Let X be a random variable whose outcome x takes values in $\{a_1, \dots, a_L\}$ with probabilities $\{p_1, \dots, p_L\}$
- Shannon's information content for the outcome $x = a_i$:

$$H(x = a_i) = \log_2 \left(\frac{1}{P(x = a_i)} \right) = \log_2 \left(\frac{1}{p_i} \right)$$

is a sensible measure of information content

- The entropy

$$H(X) = \sum_i p_i \log_2 \left(\frac{1}{p_i} \right) = - \sum_i p_i \log_2(p_i)$$

is a sensible measure of expected (**average**) information content

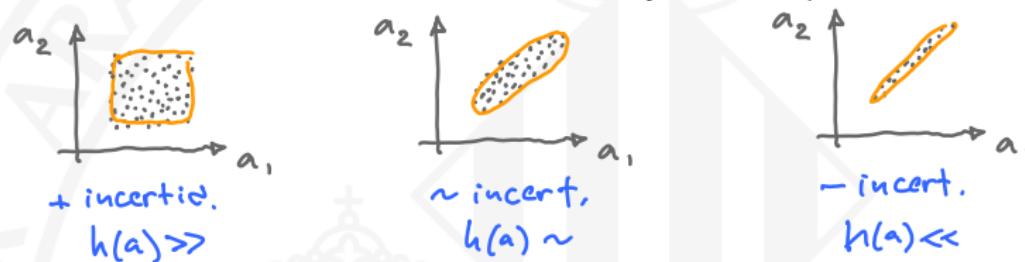
- Entropy is measured in:
 - bits (**binary digits**) if base 2 log is used
 - nats (**natural digits**): natural (base e) log.
- Good things to do, but not the only one!

Shannon's information, intuitively

- Shannon's information or entropy of a vector \mathbf{a} with PDF $P(\mathbf{a})$

$$H(\mathbf{a}) = \int P(\mathbf{a}) \log_2 \left(\frac{1}{P(\mathbf{a})} \right) d\mathbf{a} = - \int P(\mathbf{a}) \log_2(P(\mathbf{a})) d\mathbf{a}$$

- **Intuition:** 'entropy is related to the PDF volume'
- **Intuition 2:** 'more volume, more uncertainty, more surprise'



- **Interesting properties:**

- ① Entropy of a unidimensional Gaussian: $H(\mathbf{a}) = \frac{1}{2} \ln(2\pi e \sigma^2)$
- ② Entropy of a Gaussian depends on the volume $|\Sigma|$:

$$H(\mathbf{a}) = \frac{1}{\log(2)} \ln((2\pi e)^{d/2} |\Sigma|^{1/2})$$

- ③ The \mathcal{N} distrib has the highest H among all distrib. with Σ
- `entropies.m`, `hgu.m`

Change in entropy under transformations

- Given $F : \mathbf{a} \in \mathbb{R}^d \rightarrow \mathbf{b} \in \mathbb{R}^d$, then

$$H(\mathbf{a}) \rightarrow H(\mathbf{b}) = H(\mathbf{a}) + \mathbb{E}[\log_2 |\nabla F(\mathbf{a})|]$$

- For the demo, first remember:

- The differential in volume in the transformed domain depends on the Jacobian of the transform, $d\mathbf{b} = |\nabla F(\mathbf{a})|d\mathbf{a}$
- Remember PDFs under transforms: $P(\mathbf{b}) = P(\mathbf{a})|\nabla F(\mathbf{a})|^{-1}$

$$\begin{aligned} H(\mathbf{b}) &= - \int P(\mathbf{b}) \log_2(P(\mathbf{b})) d\mathbf{b} = \\ &= - \int P(\mathbf{a}) |\nabla F(\mathbf{a})|^{-1} \log_2(P(\mathbf{a}) |\nabla F(\mathbf{a})|^{-1}) |\nabla F(\mathbf{a})| d\mathbf{a} = \\ &= - \int P(\mathbf{a}) \log_2(P(\mathbf{a})) d\mathbf{a} - \int P(\mathbf{a}) \log_2(|\nabla F(\mathbf{a})|^{-1}) d\mathbf{a} = \\ &= H(\mathbf{a}) + \mathbb{E}[\log_2 |\nabla F(\mathbf{a})|] \end{aligned}$$

- Orthogonal transforms (rotations) conserve entropy!**
- htransforms.m**

Entropy (negatively biased) estimation in MATLAB

```
function H = entropy(p)

    p = p/sum(p); % Empirical estimate of the distribution
    idx = p~=0;
    H = -sum(p(idx).*log2(p(idx))));
```

Entropy estimation with MM correction in MATLAB

```
% MLE estimator with Miller-Maddow correction
```

```
function H = entropy_mm(p)

    c = 0.5 * (sum(p>0)-1)/sum(p); % Miller-Maddow correction
    p = p/sum(p); % Empirical estimate of the distribution
    idx = p~=0;
    H = -sum(p(idx).*log2(p(idx))) + c;
```

- hbias.m

Entropy estimation: toy example

If $X = a$ (with $p_a = 1/2$), $X = b$ (with $p_b = 1/4$), $X = c$ (with $p_c = 1/8$), and $X = d$ (with $p_d = 1/8$). The entropy of X is

$$H(X) = -\frac{1}{2} \log_2 \left(\frac{1}{2} \right) - \frac{1}{4} \log_2 \left(\frac{1}{4} \right) - \frac{1}{8} \log_2 \left(\frac{1}{8} \right) - \frac{1}{8} \log_2 \left(\frac{1}{8} \right) = \frac{7}{4} \text{ bits}$$

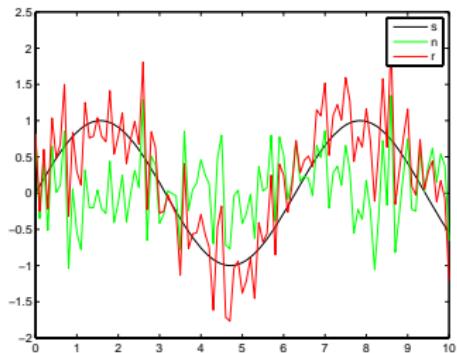
Note: An efficient question would be 'is $X = a?$ ' because it splits the probability in 0.5-0.5, a second best would be 'is $X = b?$ ' and so on...

The resulting expected number of binary questions required to know X is 1.75. It can be demonstrated that the expected number of questions lies between

$$H(X) \leq L \leq H(X) + 1$$

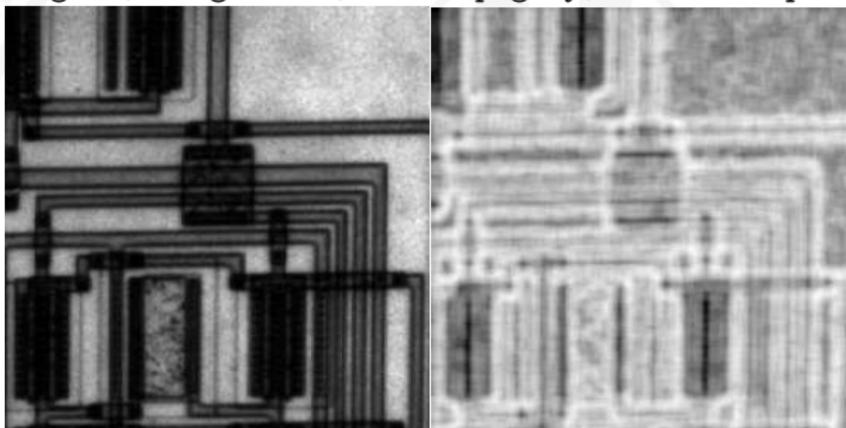
Entropy estimation in MATLAB: clean vs noisy signals.

```
>> help entropy  
>> s = sin(0:0.01:10);  
>> n = 0.75*randn(size(s));  
>> r = s + n;  
>> plot(0:0.1:10,s,'k',0:0.1:10,n,'g',0:0.1:10,r,'r')  
>> corrcoef(s,n) = -0.0396  
>> entropy(s) = 1.53, entropy(n) = 1.71, entropy(r) = 2.05  
>> entropy(s)+entropy(n) = 3.2457  
>> jointentropy(s,n) = 3.2411  
>> jointentropy(s,r) = 3.2678
```



Entropy estimation in MATLAB: feature extraction.

```
>> help entropy  
>> help entropyfilt  
>> I = imread('circuit.tif');  
>> E = entropy(I);  
>> J = entropyfilt(I);  
>> figure, imagesc(I), colormap gray; axis off square  
>> figure, imagesc(J), colormap gray; axis off square
```



Entropy estimation for time series processing

- http://www.tech.plym.ac.uk/spmc/links/sp/sp_entropy.html
- <http://www.nbb.cornell.edu/neurobio/land/PROJECTS/Complexity/index.html>
- <http://www.mpiiks-dresden.mpg.de/~tisean/>
- <http://www.mathworks.com/matlabcentral/fileexchange/3102>

Joint entropy $H(X, Y)$ The joint entropy $H(X, Y)$ of a pair of discrete random variables (X, Y) with a joint distribution $p(x, y)$ is:

$$H(X, Y) = - \sum_x \sum_y p(x, y) \log_2(p(x, y))$$

If X and Y are independent: $p(x, y) = p(x)p(y)$ and the Shannon information content is:

$$H(X, Y) = \log_2 \left(\frac{1}{P(x, y)} \right) = \log_2 \left(\frac{1}{P(x)} \right) + \log_2 \left(\frac{1}{P(y)} \right) = H(x) + H(y)$$

Conditional entropy The conditional entropy is the average uncertainty remaining about x if we have observed y :

$$H(X|Y) = - \sum_x \sum_y p(x, y) \log_2(p(x|y)) = H(X, Y) - H(Y)$$

Relation between joint and conditional entropies The entropy of a pair of random variables is the entropy of one plus the conditional entropy of the other.

$$H(X, Y) = H(X) + H(Y|X)$$

Corollaries:

- If X and Y are independent $H(X|Y) = H(X)$
- $H(Y|X) \neq H(X|Y)$ but $H(X) - H(X|Y) = H(Y) - H(Y|X)$

ON INFORMATION AND SUFFICIENCY

BY S. KULLBACK AND R. A. LEIBLER

The George Washington University and Washington, D. C.

1. Introduction. This note generalizes to the abstract case Shannon's definition of information [15], [16]. Wiener's information (p. 75 of [18]) is essentially the same as Shannon's although their motivation was different (cf. footnote 1, p. 95 of [16]) and Shannon apparently has investigated the concept more completely. R. A. Fisher's definition of information (intrinsic accuracy) is well known (p. 709 of [6]). However, his concept is quite different from that of Shannon and Wiener, and hence ours, although the two are not unrelated as is shown in paragraph 2.

R. A. Fisher, in his original introduction of the *criterion of sufficiency*, required "that the statistic chosen should summarize the whole of the relevant information supplied by the sample," (p. 316 of [5]). Halmos and Savage in a recent paper, one of the main results of which is a generalization of the well known Fisher-Neyman theorem on sufficient statistics to the abstract case, conclude, "We think that confusion has from time to time been thrown on the subject by . . . , and (c) the assumption that a sufficient statistic contains all the information in only the technical sense of 'information' as measured by variance," (p. 241 of [8]). It is shown in this note that the information in a sample as defined herein, that is, in the Shannon-Wiener sense cannot be increased by any statistical operations and is invariant (not decreased) if and only if sufficient statistics are employed. For a similar property of Fisher's information see p. 717 of [6], Doob [19].

We are also concerned with the statistical problem of discrimination ([3], [17]), by considering a measure of the "distance" or "divergence" between statistical populations ([1], [2], [13]) in terms of our measure of information. For the statistician two populations differ more or less according as to how difficult it is to discriminate between them with the best test [14]. The particular measure of

Kullback-Leibler divergence (KLD)

- The KLD measures differences (a kind of 'distance') between PDFs
- Definition: Given two PDFs $P(\mathbf{a})$ and $Q(\mathbf{a})$, the KLD between them is

$$D_{KL}(P(\mathbf{a})\|Q(\mathbf{a})) = \int P(\mathbf{a}) \log \left(\frac{P(\mathbf{a})}{Q(\mathbf{a})} \right) d\mathbf{a}$$

KLD properties

- $D_{KL} \geq 0$
- $D_{KL} = 0$ iff $P(\mathbf{a}) = Q(\mathbf{a})$

Watch out!

- D_{KL} is not a distance!
- A distance $d(\cdot\|\cdot)$ must fulfil three conditions:
 - Positiveness: $d(x\|y) \geq 0$ $d(x\|y) = 0$ iff $x = y$:
 - Triangle inequality: $d(x\|z) \geq d(x\|y) + d(y\|z)$:
 - Symmetry: $d(x\|y) = d(y\|x)$:

Kullback-Leibler divergence (KLD), ctd'

- The KLD measures differences (a kind of 'distance') between PDFs
- Definition: Given two PDFs $P(\mathbf{a})$ and $Q(\mathbf{a})$, the KLD between them is

$$D_{KL}(P(\mathbf{a})\|Q(\mathbf{a})) = \int P(\mathbf{a}) \log \left(\frac{P(\mathbf{a})}{Q(\mathbf{a})} \right) d\mathbf{a}$$

Property 1: Pythagoras in KLD

Given P, Q , there exists R such that:

$$D_{KL}(P(\mathbf{a})\|Q(\mathbf{a})) = D_{KL}(P(\mathbf{a})\|R(\mathbf{a})) + D_{KL}(R(\mathbf{a})\|Q(\mathbf{a}))$$

Property 2: KLD is invariant under invertible affine transforms

Given $\mathbf{F} : \mathbf{b} = \mathbf{G}\mathbf{a} + \mathbf{n}$, and $\nabla\mathbf{F} = \mathbf{G}$

$$\begin{aligned} D_{KL}(P(\mathbf{b})\|Q(\mathbf{b})) &= \int P(\mathbf{b}) \log \left(\frac{P(\mathbf{b})}{Q(\mathbf{b})} \right) d\mathbf{b} = \\ &\int P(\mathbf{a}) |\nabla\mathbf{G}|^{-1} \log \left(\frac{P(\mathbf{a}) |\nabla\mathbf{G}|^{-1}}{Q(\mathbf{a}) |\nabla\mathbf{G}|^{-1}} \right) |\nabla\mathbf{G}| d\mathbf{a} = D_{KL}(P(\mathbf{a})\|Q(\mathbf{a})) \end{aligned}$$

Example: Check assymetry of KLD Let $\mathcal{X} = \{0, 1\}$ and consider two distributions p and q on \mathcal{X} . Let $p(0) = 1 - r$, $p(1) = r$, $q(0) = 1 - s$, $q(1) = s$. Then

$$D_{KL}(p\|q) = (1 - r) \log \frac{1 - r}{1 - s} + r \log \frac{r}{s}$$

and

$$D_{KL}(q\|p) = (1 - s) \log \frac{1 - s}{1 - r} + s \log \frac{s}{r}$$

If $s = r$, then $D_{KL}(p\|q) = D_{KL}(q\|p) = 0$.

If $r = 1/2$ and $s = 1/4$, then $D_{KL}(p\|q) = 0.21$ bits and $D_{KL}(q\|p) = 0.18$ bits.

In general $D_{KL}(p\|q) \neq D_{KL}(q\|p)$!

Example: Check Pythagoras and KLD under rotations `KLDiv.m`, `JSDiv.m`, `kldproperties.m`

Cross-entropy The cross entropy for two distributions \mathbf{p} and \mathbf{q} over the same probability space:

$$H(\mathbf{p}, \mathbf{q}) = H(\mathbf{p}) + D_{KL}(\mathbf{p} \parallel \mathbf{q}) \rightarrow D_{KL}(\mathbf{p} \parallel \mathbf{q}) = H(\mathbf{p}, \mathbf{q}) - H(\mathbf{p})$$

Intuition: divergence is the difference of volume between PDFs

Demo:

$$\begin{aligned} H(\mathbf{p}, \mathbf{q}) &= - \sum_i \mathbf{p} \log_2(\mathbf{q}) = - \sum_i \mathbf{p} \log_2\left(\frac{\mathbf{p}\mathbf{q}}{\mathbf{p}}\right) = \\ &- \left[\sum_i \left(\mathbf{p} \log_2(\mathbf{p}) + \mathbf{p} \log_2\left(\frac{\mathbf{q}}{\mathbf{p}}\right) \right) \right] = H(\mathbf{p}) + D_{KL}(\mathbf{p} \parallel \mathbf{q}) \end{aligned}$$

Consequence: For discrete \mathbf{p} and \mathbf{q} this means:

$$H(\mathbf{p}, \mathbf{q}) = - \sum_i \mathbf{p} \log_2(\mathbf{q}) \neq H(\mathbf{q}, \mathbf{p}) = - \sum_i \mathbf{q} \log_2(\mathbf{p})$$

Statistical independence

- **Definition:** Components in vector \mathbf{a} are statistically independent if the joint PDF can be ‘factorized’:

$$P_{\mathbf{a}}(\mathbf{a}) = \prod_{i=1}^d P_{a_i}(a_i) = P_{a_1}(a_1)P_{a_2}(a_2) \cdots P_{a_d}(a_d)$$

- **Intuition 1:** look at the conditional PDF: “*Statistical independence means $P(a_i|a_{j \neq i}) = P(a_i)$ since observing (knowing) a_j does not convey any information on a_i* ”:

$$P(a_i|a_j) = \frac{P(a_i, a_j)}{P(a_j)} = (\text{factorization}) = \frac{P(a_i)P(a_j)}{P(a_j)} = P(a_i)$$

- **Intuition 2:** look at the KLD and assume you can factorize $P_{\mathbf{a}}(\mathbf{a}) = \prod_{i=1}^d P_{a_i}(a_i)$, then

$$D_{KL}(P(\mathbf{a}) \| \prod_{i=1}^d P_{a_i}(a_i)) = \int P(\mathbf{a}) \log \left(\frac{P(\mathbf{a})}{\prod_{i=1}^d P_{a_i}(a_i)} \right) d\mathbf{a} = \int P(\mathbf{a}) \log(1) = 0$$

Mutual information or ‘dependence’

The mutual information of two discrete random variables x and y can be defined as:

$$I(x, y) = \sum_x \sum_y p(x, y) \log \left(\frac{p(x, y)}{p_1(x)p_2(y)} \right)$$

where $p(x, y)$ is the joint probability distribution function of x and y , and $p_1(x)$ and $p_2(y)$ are the marginal probability distribution functions of x and y respectively.

Intuitions

- Mutual information measures the information that x and y share
- I measures how much knowing one of these variables reduces our uncertainty about the other.
- If x and y are independent, then knowing x does not give any information about y and vice versa, so $I = 0$
- If $x = y$, all information conveyed by x is shared by y : knowing x determines the value of y and vice versa, so I is the uncertainty contained in x or y alone, i.e. the entropy of x or y

Mutual information or ‘dependence’ The mutual information of two discrete random variables x and y can be defined as:

$$I(x, y) = \sum_x \sum_y p(x, y) \log \left(\frac{p(x, y)}{p_1(x)p_2(y)} \right)$$

where $p(x, y)$ is the joint probability distribution function of x and y , and $p_1(x)$ and $p_2(y)$ are the marginal probability distribution functions of x and y respectively.

/ measures independence

$I(x; y) = 0$ iff x and y are independent random variables.

Demo: if x and y are independent, then $p(x, y) = p(x)p(y)$, and therefore:

$$\log \left(\frac{p(x, y)}{p_1(x)p_2(y)} \right) = \log(1) = 0$$

/ properties $I(x; y) \geq 0$ and symmetric $I(x; y) = I(y; x)$

The big picture

$$H(X, Y)$$

$$H(X)$$

$$H(Y)$$

$$H(X | Y)$$

$$I(X; Y)$$

$$H(Y|X)$$

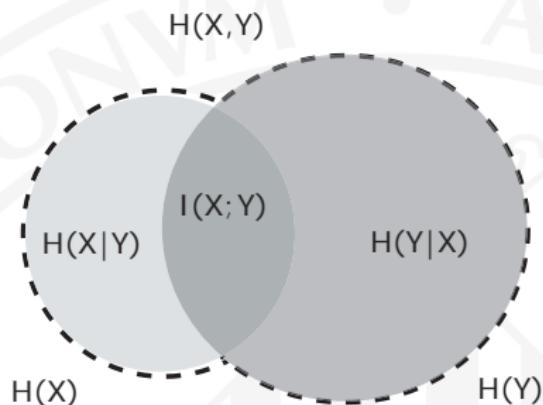
(from David MacKay's book)

$H(X, Y)$ is the **joint entropy** of X, Y

$H(X|Y)$ is the **conditional entropy** of X given Y

$I(X; Y)$ is the **mutual information** between X and Y

The big picture II: basic relations



- ① $I(X; Y) = H(X) - H(X|Y)$
- ② $I(X; Y) = H(Y) - H(Y|X)$
- ③ $H(X, Y) = H(X) + H(Y) - I(X; Y)$
- ④ $I(X; X) = H(X) - H(X|X) = H(X)$
- ⑤ $I(Y; X) = I(X; Y)$
- ⑥ $I(X; Y) \geq 0$, and $I(X; Y) = 0$ iff $X \perp Y$

Multi-information

The mutual information property

$$H(X, Y) = H(X) + H(Y) - I(X; Y)$$

leads to

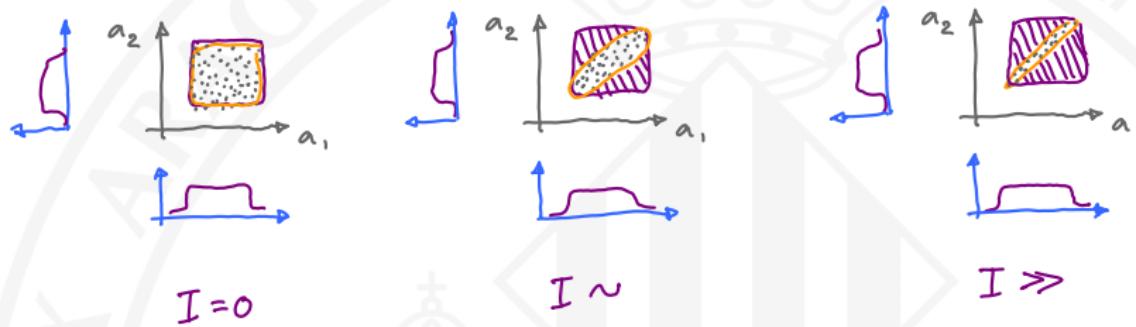
$$I(X; Y) = H(X) + H(Y) - H(X, Y)$$

and can be generalized to multi-dimensional spaces:

$$I(\mathbf{a}) = \sum_{i=1}^d H(a_i) - H(\mathbf{a})$$

Intuition on mutual information

* Intuición: $I = \sum_x h_x - h \equiv$ diferencia entre el volumen del producto de marginales frente al volumen de la conjunta



Cuanto mayor es la relación entre las variables mayor es la diferencia entre los volúmenes (entropías)



Property 1: Information cannot hurt!

The mutual information is positive by definition:

$$I(X; Y) = H(Y) - H(Y|X) \geq 0 \quad \rightarrow \quad H(Y) \geq H(Y|X)$$

Property 2: / with Gaussian random variables If you assume x and y are Gaussian random variables [Cardoso03]:

$$I(x, y) = -\frac{1}{2} \log \left(\frac{|\mathbf{C}|}{|\mathbf{C}_{xx}| |\mathbf{C}_{yy}|} \right), \quad \text{where } \mathbf{C} = \begin{pmatrix} \mathbf{C}_{xx} & \mathbf{C}_{xy} \\ \mathbf{C}_{xy}^\top & \mathbf{C}_{yy} \end{pmatrix}$$

See it this way: given $(X, Y) \sim \mathcal{N}(0, \mathbf{C})$ correlated Gaussian variables with \mathbf{C}

$$\mathbf{C} = \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

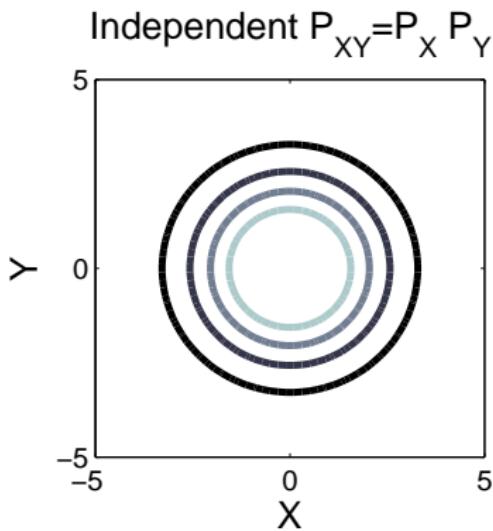
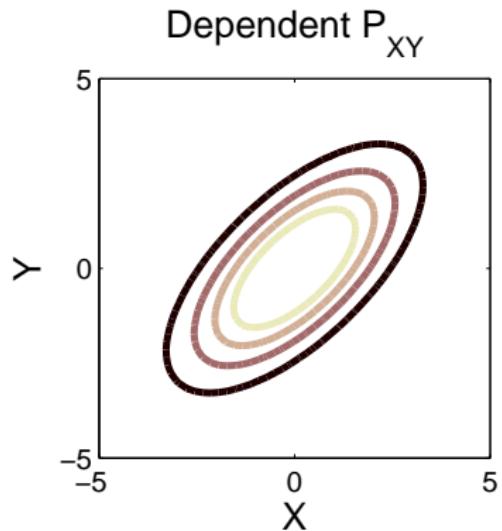
$$H(X) = H(Y) = 0.5 \log(2\pi e \sigma^2)$$

$$H(X, Y) = 0.5 \log((2\pi e)^2 |\mathbf{C}|) = 0.5 \log((2\pi e)^2 \sigma^4 (1 - \rho^2))$$

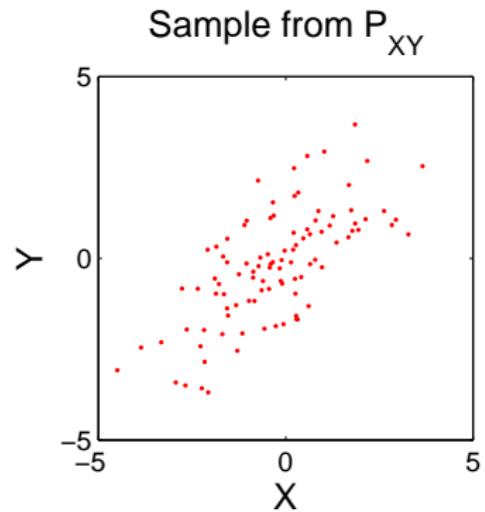
$$I(X; Y) = H(X) + H(Y) - H(X, Y) = -0.5 \log(1 - \rho^2)$$

- gaussianmutual.m

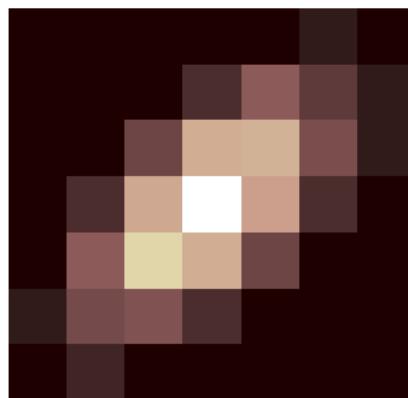
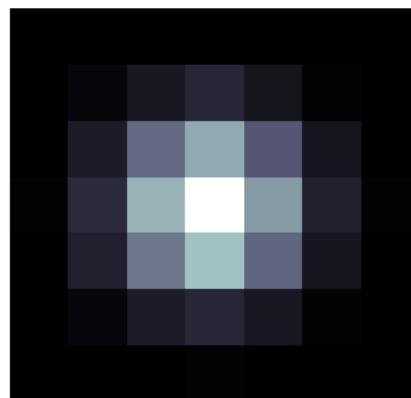
- Given distribution \Pr , test $\mathcal{H}_0 : \Pr = \Pr_x \Pr_y$
- Continuous valued, multivariate: $\mathcal{X} := \mathbb{R}^d$ and $\mathcal{Y} := \mathbb{R}^{d'}$



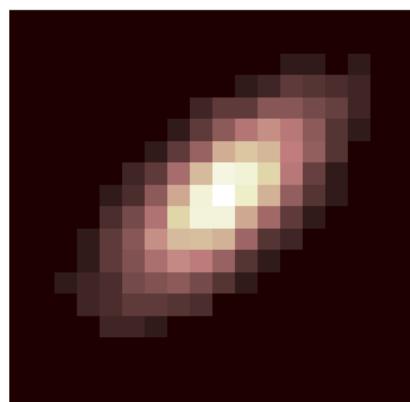
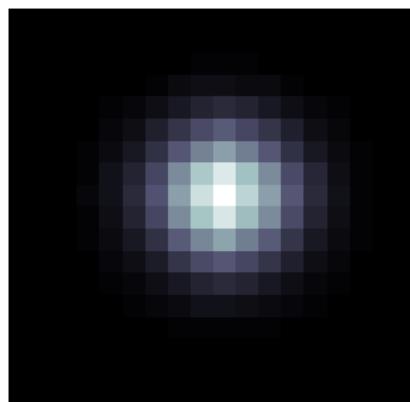
- Given distribution \Pr , test $\mathcal{H}_0 : \Pr = \Pr_x \Pr_y$
- Finite sample observed $(X_1, Y_1), \dots, (X_n, Y_n)$



- Given distribution \Pr , test $\mathcal{H}_0 : \Pr = \Pr_X \Pr_Y$
- Partition space \mathcal{X} into m_n bins, space \mathcal{Y} into m'_n bins

Discretized empirical P_{XY} Discretized empirical $P_X P_Y$ 

- Given distribution \Pr , test $H_0 : \Pr = \Pr_x \Pr_y$
- Refine partition m_n, m'_n for increasing n

Discretized empirical P_{XY} Discretized empirical $P_X P_Y$ 

Histogram-based mutual estimation and the curse of dimensionality

- In high dimensional problems, the space is typically empty ... :(
- The curse of dimensionality [Fukunaga78]
- We need much more samples, n , to fill in the space as d increases
- Assuming $n = b^2$ for b bins, $s = b^2 \cdot d$:

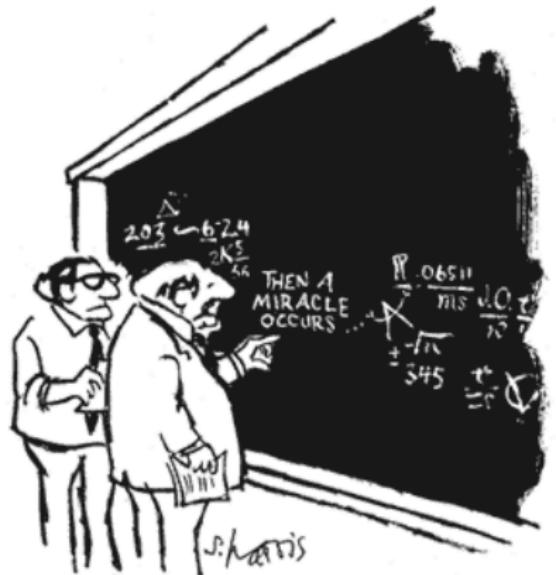
d	s	Memory [Bytes]
1	11	968
2	14641	117.128
3	1771561	14.172.488
4	214358881	1.714.871.048
5	25.937.000.000	HELP MEMORY
6	3.138.400.000.000	HELP MEMORY

Source code for estimating mutual information

- MATLAB does not have a function to do it! :(
- Several toolboxes available:
 - <http://www.mathworks.com/matlabcentral/fileexchange/14888-mutual-information-computation>
 - <http://www.cs.rug.nl/~rudy/matlab/>
 - <http://www.bioss.ac.uk/~dirk/software/MutInf/>
 - <http://www.physik3.gwdg.de/tstool/>
 - <http://www.klab.caltech.edu/~kraskov/MILCA/>

Reviewed:

Information theory, main quantities, entropy, divergence, mutual information, channels, communication errors, capacity, applications in signal and image processing, etc.



"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."