# Introduction to Biomedical Engineering

## Biosignal processing

Kung-Bin Sung

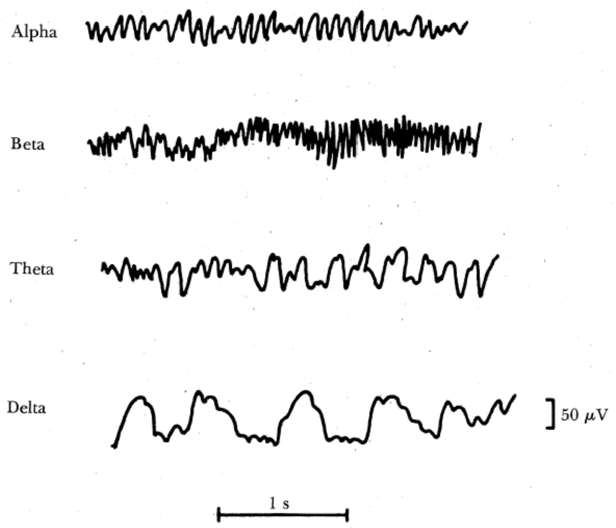
# Outline

Chapter 10: Biosignal processing

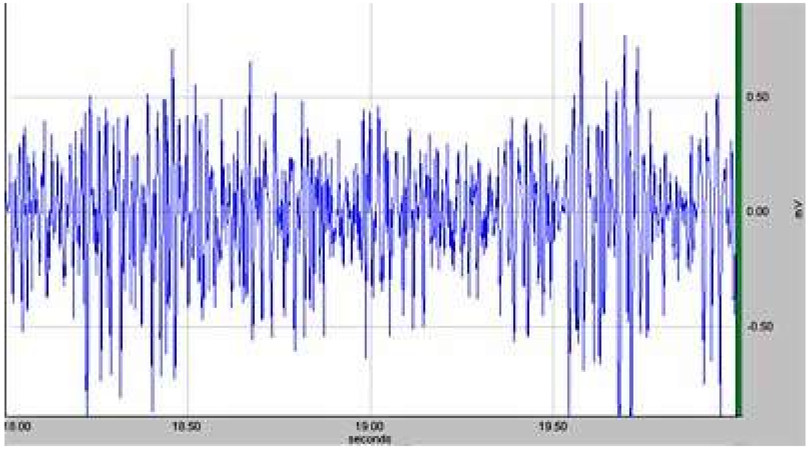
- Characteristics of biosignals
- Frequency domain representation and analysis
  - Fourier series, Fourier transform, discrete Fourier transform
  - Digital filters
- Signal averaging
- Time-frequency analysis
  - Short-time Fourier transform
  - Wavelet transform
- Artificial neural networks

# Example biosignals

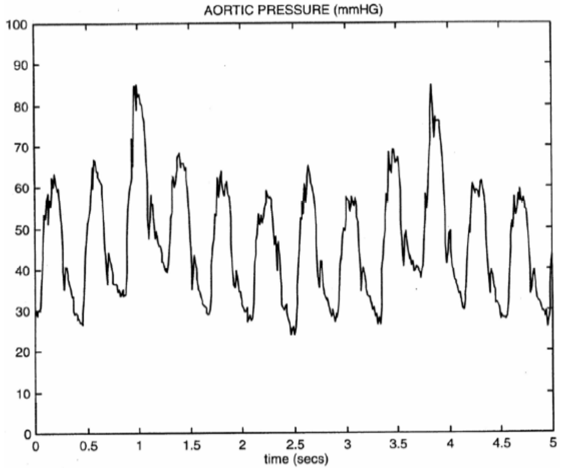
EEG



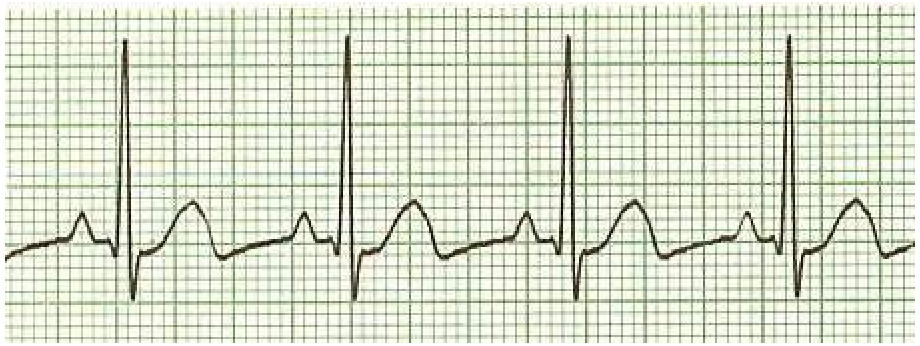
EMG



Blood pressure



ECG



# Characteristics of (bio)signals

- Continuous vs. discrete

  $x(t)$ continuous variables such as time and space

  $x(n)$ $n$ =0, 1, 2, 3… $\Rightarrow n$ is an integer

  sampled at a finite number of points

  we will deal with discrete signals in this module (a subset of digital signal processing)

- Deterministic vs. random
  - Deterministic: can be described by mathematical functions or rules

  Periodic signal is an example of deterministic signals

  $x(t) = x(t + nT)$ $\Rightarrow$ repeats itself every T units in time, T is the period

  ECG and blood pressure both have dominant periodic components

# Characteristics of biosignals

Random (stochastic) signals

- Contain uncertainty in the parameters that describe them, therefore, cannot be precisely described by mathematical functions

- Often analyzed using statistical techniques with probability distributions or simple statistical measures such as the mean and standard deviation

- Example: EMG (electromyogram)

Stationary random signals: the statistics or frequency spectra of the signal remain constant over time. It is the stationary properties of signals that we are interested in

Real biological signals always have some unpredictable noise or change in parameters $\Rightarrow$ not completely deterministic

# Signal processing

We want to extract useful information from measured data

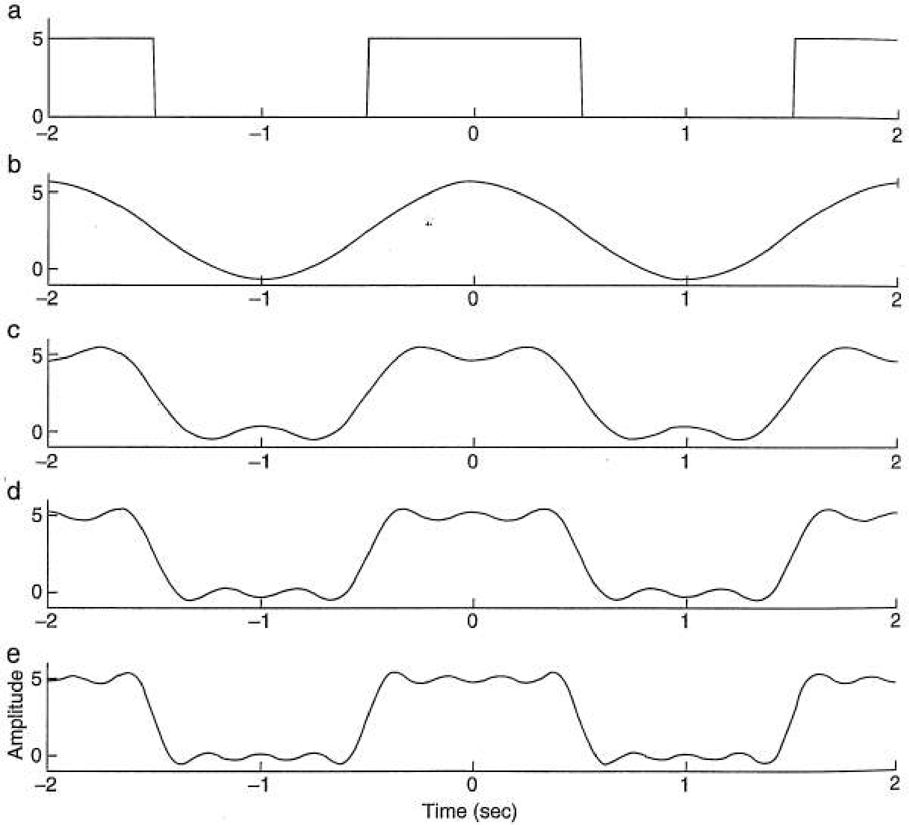
- Noise reduction and signal enhancement
- Signal conditioning
- Feature extraction
- Pattern recognition
- Classification such as diagnosis
- Data compression
- and more…

# Frequency domain representation of signals

Fourier's theory: a complex waveform can be approximated to any degree of accuracy with simpler functions

Example: a periodic square wave of period T can be represented by summing sinusoids with proper amplitudes and frequency

$$x(t) = a_0 + \sum_{m=1}^{\infty} (a_m \cos m\omega_0 t + b_m \sin m\omega_0 t)$$



Number of harmonics

1

2

3

4

$$a_0 = \frac{1}{T}\int_T x(t)dt$$

$$a_m = \frac{2}{T}\int_T x(t)\cos(m\omega_0 t)dt$$

$$b_m = \frac{2}{T}\int_T x(t)\sin(m\omega_0 t)dt$$

# Frequency domain representation of signals

But most biosignals are not periodic

As an expansion of the Fourier series in the previous slide, the Fourier integral or Fourier transform (FT) of a continuous signal is defined as

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

ω is continuous frequency, and X(ω) has complex values whose magnitude represent the magnitude of the frequency component at ω

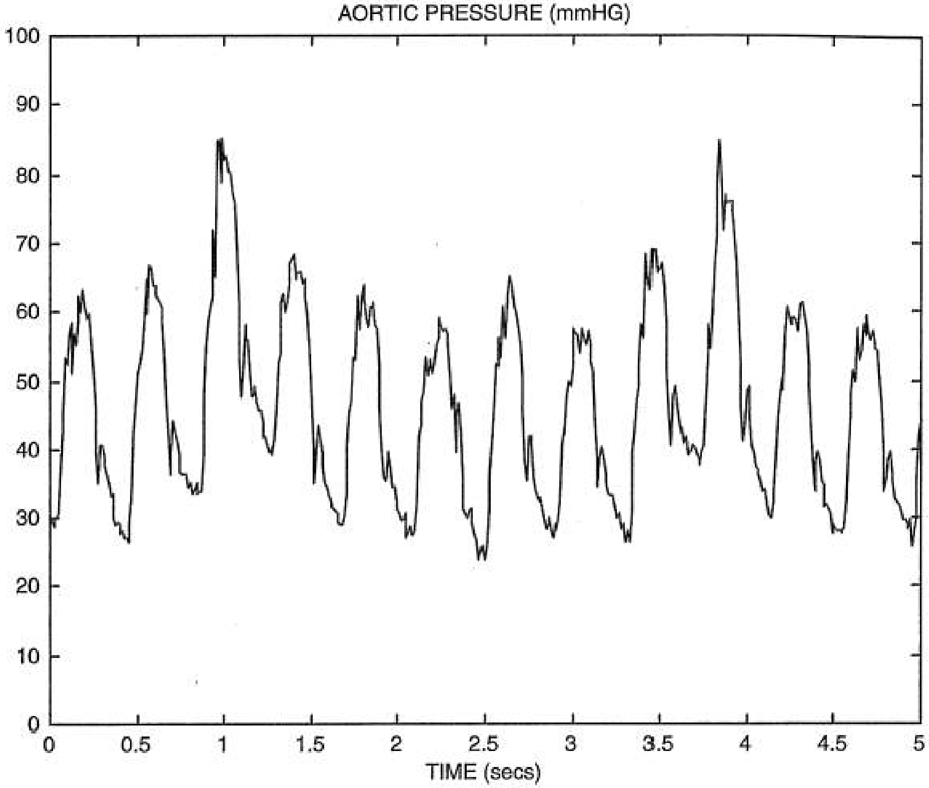
The original (time-domain) signal can be completely recovered by the inverse Fourier transform (IFT)

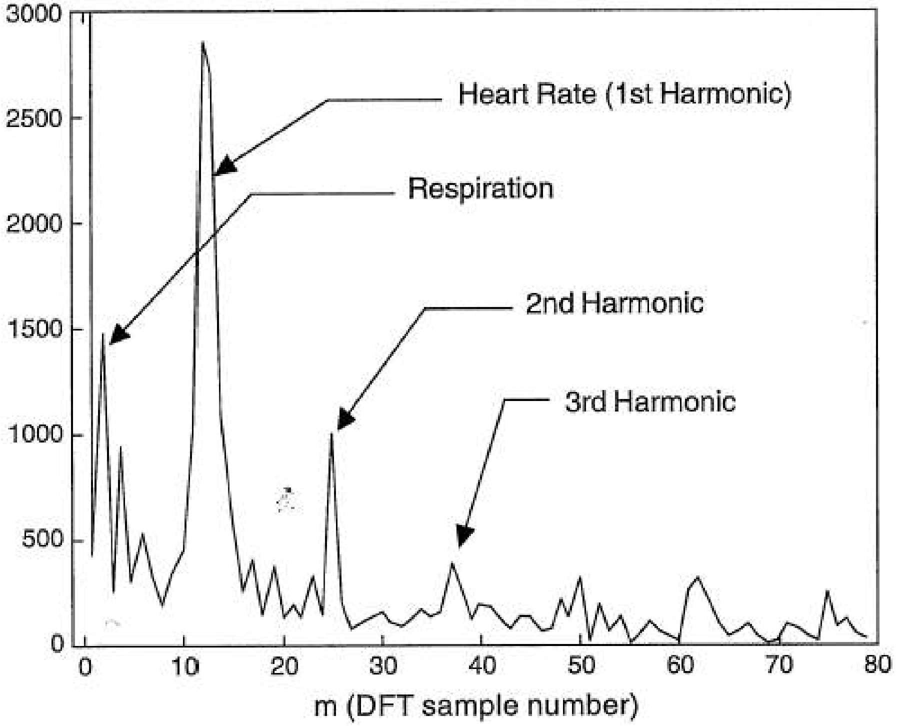
$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega$$

# Frequency domain representation of signals

Example: blood pressure waveform (sampled at 200 points/s)

Fourier domain (magnitude)





In practice, the signal is discrete in time and magnitude, and a discrete version of Fourier transform is carried out to get the Fourier (frequency) domain representation

# Discrete Fourier transform (DFT)

Recall that the input is a discrete signal, which is basically a series of numbers

$x(n)$ $\quad n$ =0, 1, 2, …, $N$-1

The discrete Fourier transform (DFT) of the discrete signal is

$$X(m) = \sum_{k=0}^{N-1} x(k) e^{-j\frac{2\pi mk}{N}} \qquad m = 0, 1, \ldots, N\text{-}1$$

Similarly, an inverse discrete Fourier transform is of this form:

$$x(k) = \frac{1}{N} \sum_{m=0}^{N-1} X(m) e^{-j\frac{2\pi mk}{N}}$$

Note that number of data points in $x(n)$ and $X(m)$ are always the same

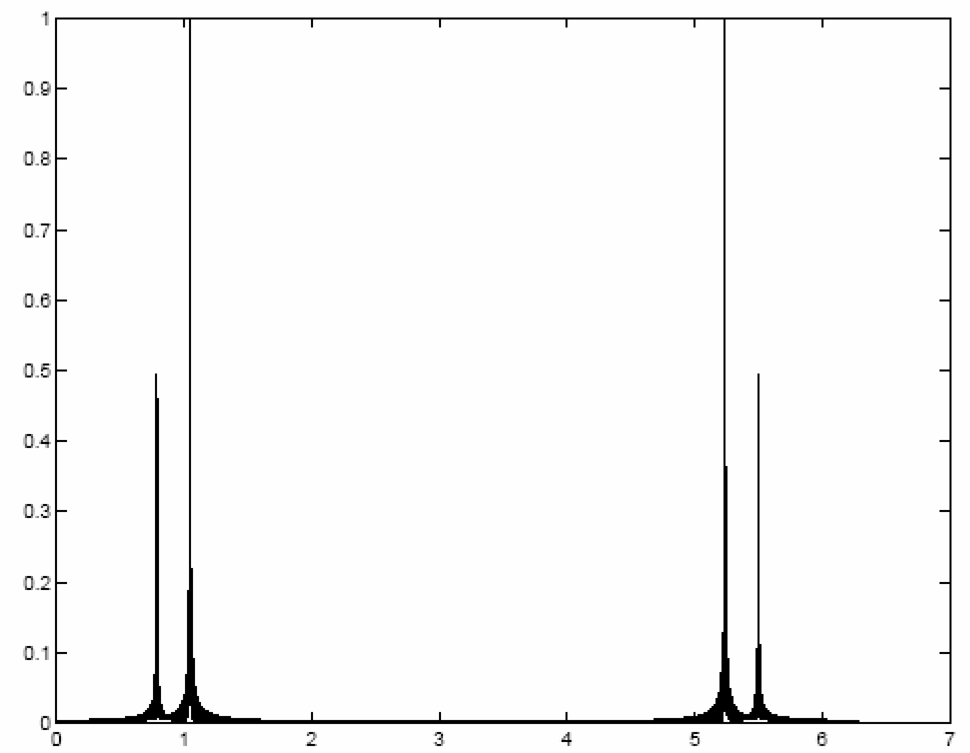
The “frequency” in the Fourier domain is not very straightforward and needs a closer look

# Discrete Fourier transform

Example problem 10.13 $x(n) = \sin(\frac{\pi}{4} n) + 2\cos(\frac{\pi}{3} n)$

Frequency of the signal

The magnitude of its DFT:



$m = 0$ $m = N\text{-}1$

0 (rad) Frequency (rad) 2π (rad)

Total number of data points $N$

$\Rightarrow$ The step size in frequency is

$$\frac{2\pi}{N} \text{ (rad)}$$

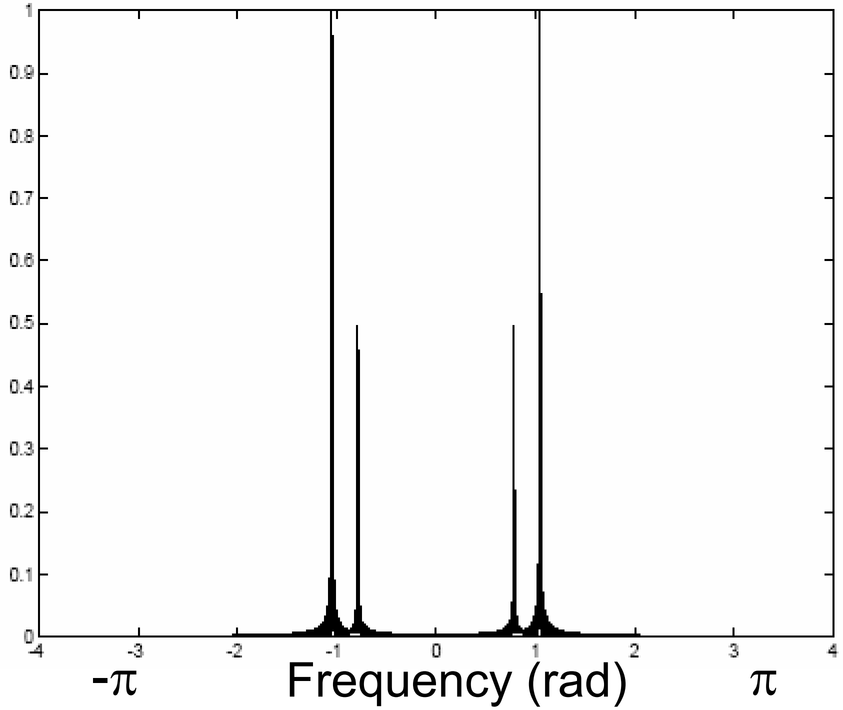
The corresponding frequency $\omega$ of the $X(m)$ series is 0~2π

# Discrete Fourier transform

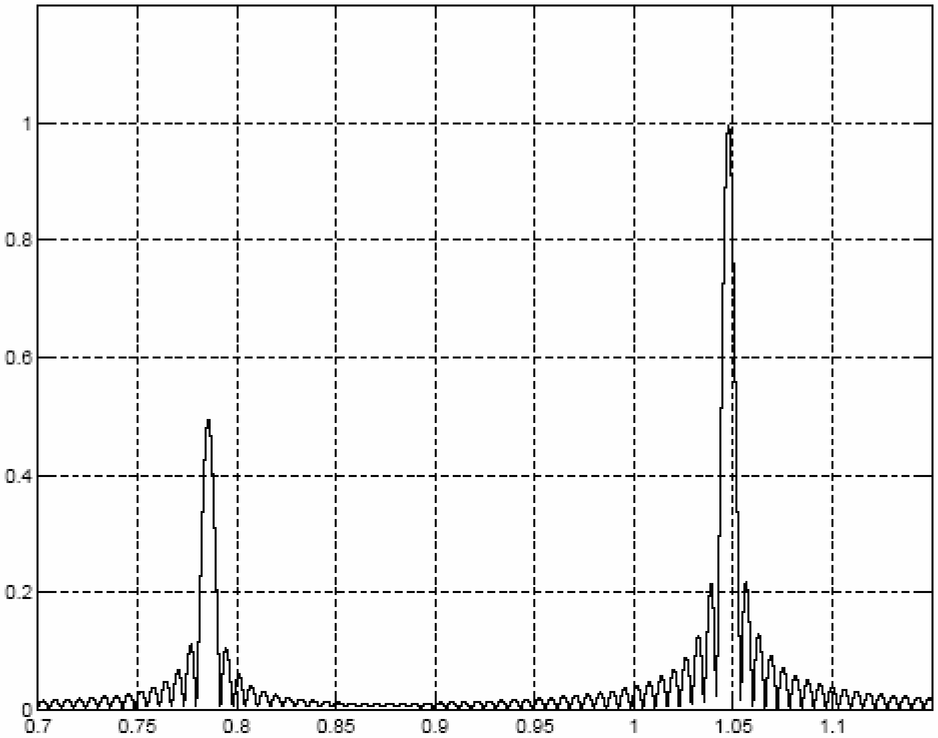
Example problem 10.13

For any real-valued signal, its Fourier transform has symmetric values with a center of ω=0. Conventionally, the DC (ω=0) component is plotted in the middle ⇒ switch the left and right halves of DFT ("*fftshift*" function in Matlab)

DFT magnitude after "*fftshift*"



zoom in around the peaks



Frequency (rad)

# Discrete Fourier transform

- Another important feature of DFT is that it is periodic with a period of $2\pi$
- Due to the symmetric values of DFT, it is sufficient to show only the frequency range $0 \sim \pi$

Relationship between frequency-domain sequence and the time domain signal

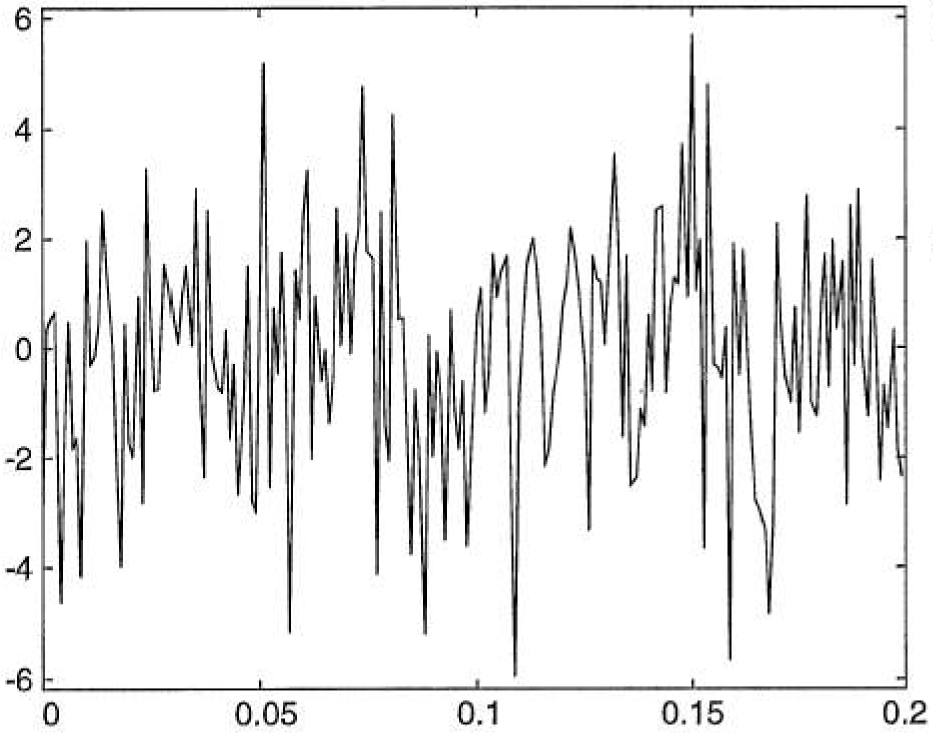
For a signal that is sampled at a sampling frequency $f_s$ (Hz)

Its DFT has a frequency range of $-\pi \sim \pi$ (rad) which corresponds to

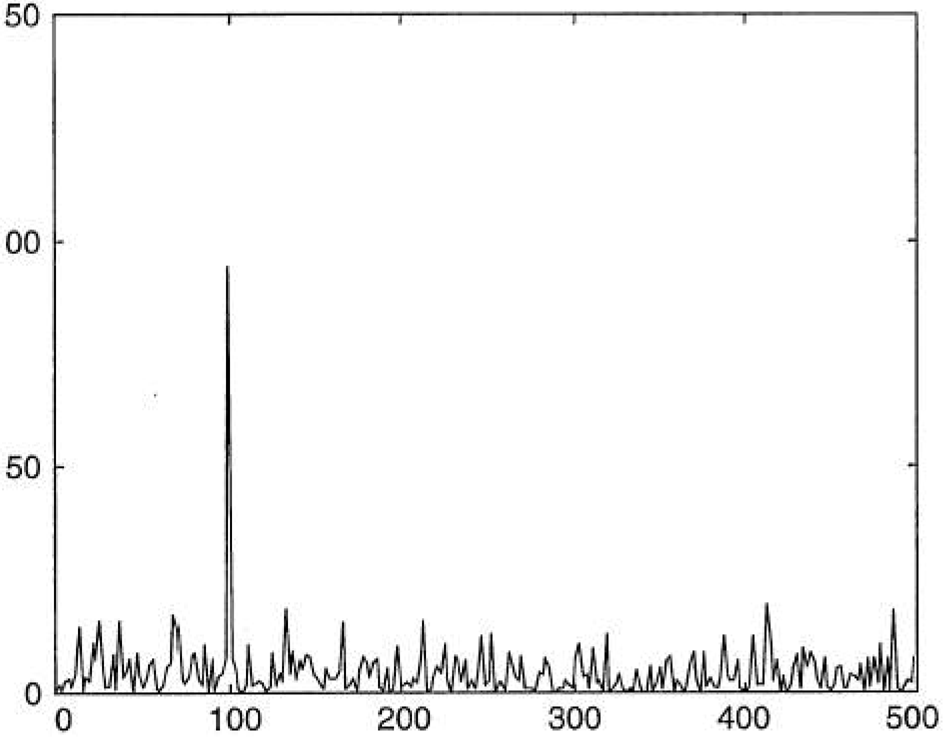
$$-\frac{1}{2} f_s \sim \frac{1}{2} f_s \quad \text{(Hz)}$$

# Example: extracting frequency-domain information in the signal

Time-domain signal (100Hz sine wave with random noise)



Frequency-domain (magnitude)



# Frequency domain analysis – comments

- Fourier transform describes the global frequency content of the signal
  - At each frequency $\omega$, the magnitude of FT represents the amount of that frequency contained in the signal
  - At each frequency , the phase of FT measures the location (relative shift) of that frequency component. However, the phase information is more difficult to interpret and less often used than the magnitude
- Methods that provide time-frequency information of the signal
  - Short-time Fourier transform
  - Wavelet transform

(we will come back to these later)

# Digital filters

As in the analog case, digital filters can be implemented in the frequency domain

$$\begin{tikzcd}
\text{time-domain} & \text{frequency-domain} \\
x(t) \arrow[r, "\text{Fourier transform}"'] & X(\omega) \arrow[d] \\
\text{filtered signal} & X(\omega) \cdot H(\omega) \arrow[l, "\text{inverse Fourier transform}"']
\end{tikzcd}$$

$H(\omega)$ is the transfer function of the filter

<u>For discrete signals</u>

$$\begin{tikzcd}
\text{time-domain} & \text{frequency-domain} \\
x(n) \arrow[r, "\text{DFT}"'] & X(m) \arrow[d] \\
\text{filtered signal} & X(m) \cdot H(m) \arrow[l, "\text{inverse DFT}"']
\end{tikzcd}$$

$H(m)$ is a discrete transfer function sampled from $H(\omega)$

# Digital filters

Alternatively, digital filters can be implemented in the time-domain

The general form of a real-time digital filter (difference equation)

$$y(k) = \sum_{\enclose{circle}{m=0}}^{M} b_m x(k-m) - \sum_{\enclose{circle}{m=1}}^{M} a_m y(k-m)$$

$y(k)$ output of the current time $k$

$x(k)$ input of the current time $k$

$y(k-1)$ and $x(k-1)$ are output and input of the previous data point

- It is real-time because it does not need the value of any “future” sample
- Can be calculated quite easily

Infinite impulse response (IIR) filter: impulse response has an infinite number of nonzero points

Example: $y(k) = \frac{1}{2} x(k) + \frac{1}{2} \enclose{circle}{y(k-1)}$ (depends on value of previous output)

# Digital filters

Finite impulse response (FIR) filter: impulse response has a finite number of nonzero points

Example: $$y(k) = \frac{1}{3}x(k) + \frac{1}{3}x(k-1) + \frac{1}{3}x(k-2)$$

The frequency-domain characteristics of digital filters can be analyzed by using $z$ transform which is analogous to the Laplace transform in the continuous

$z$ transform for discrete signals $$X(z) = \sum_{k=0}^{\infty} x(k) z^{-k}$$

Laplace transform for continuous signals $$L\{x(t)\} = X(s) = \int_0^{\infty} x(t) e^{-st} dt$$

Unfortunately, we don’t have time to talk more about $z$ transform and its use in digital filtering

# Signal averaging

Averaging can be considered as a low-pass filter since high frequency components will be attenuated by averaging

For most biological signals there is a random noise superimposed on the quantity of interest

$$y_i(t) = x(t) + n(t)$$

$y_i(t)$ is the measured signal; subscript i indicates multiple measurements are obtained

$x(t)$ is the deterministic component, assuming it exists

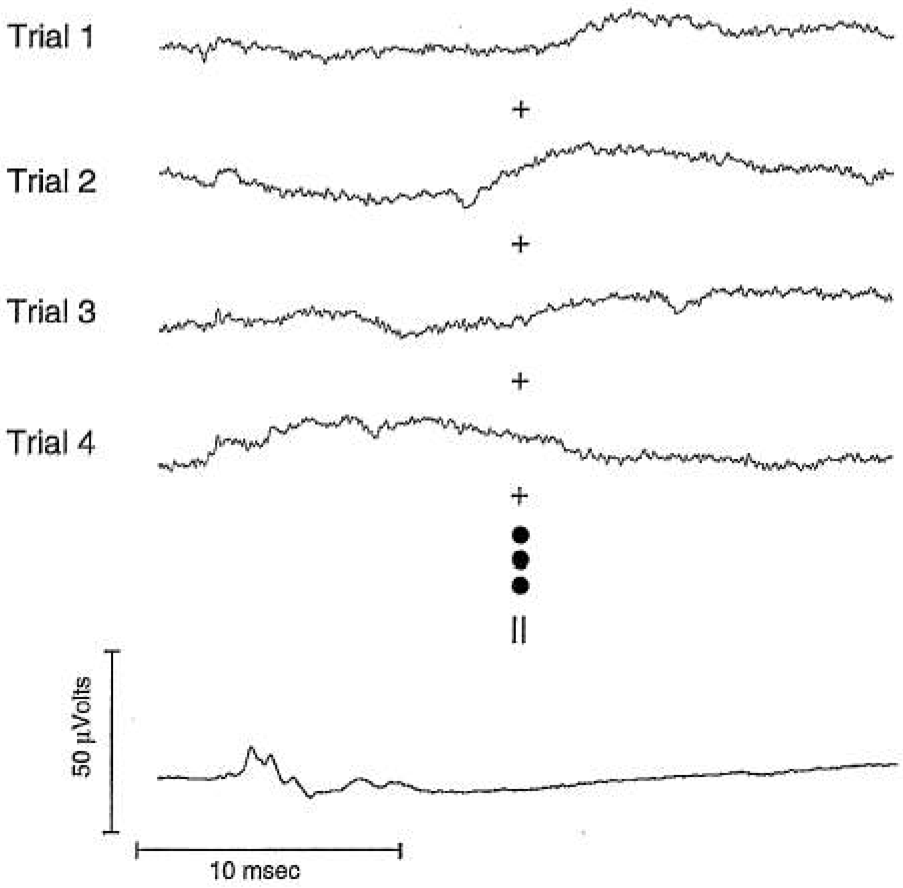
$n(t)$ is random noise

If we take the average of measurements from $N$ separate trials

$$\overline{y}(k) = \frac{1}{N}\sum_{i=1}^{N} y_i(k) \quad \Rightarrow \quad \overline{y}(k) = x(k) + \frac{1}{N}\sum_{i=1}^{N} n(k)$$

If the noise is purely random the error of measurement will decrease as the number of trials $N$ gets larger

# Signal averaging – example



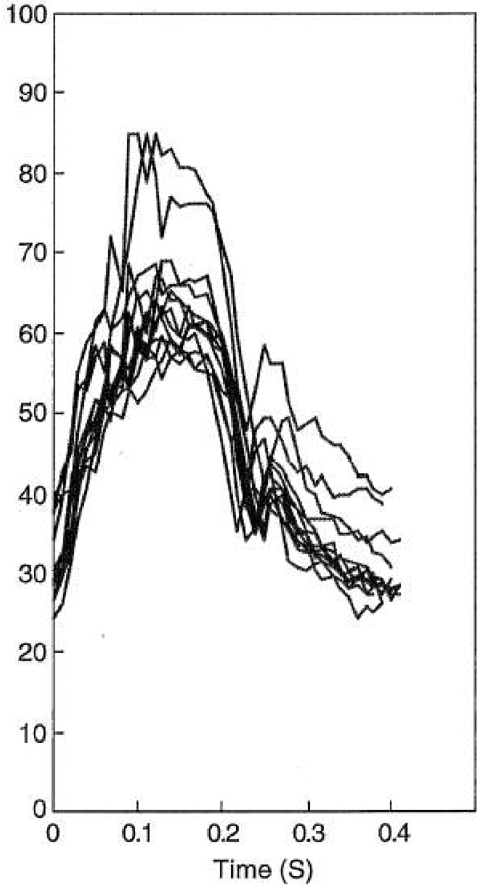
Auditory response averaged from 1000 trials (measurements) to reduce the effects of noise

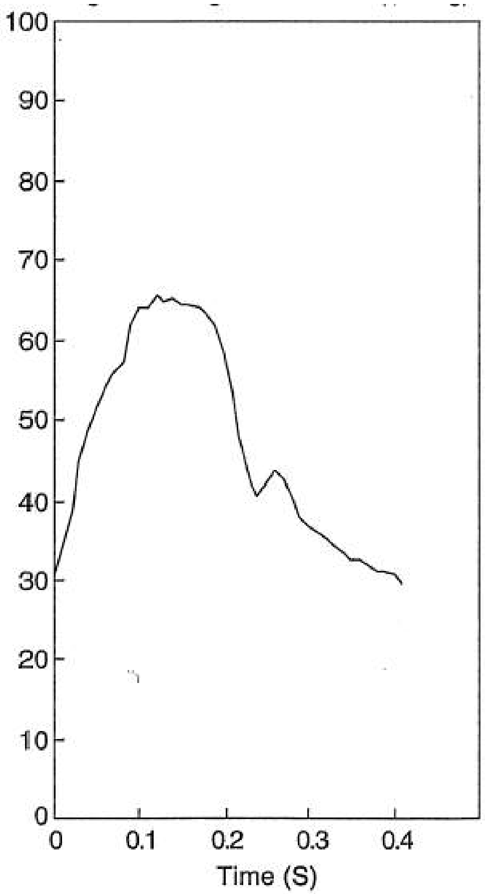
# Signal averaging – example

Use the blood pressure data (slide 9) as an example

Note that the blood pressure waveform is approximately periodic

Overlay of many periods of the pressure waveform





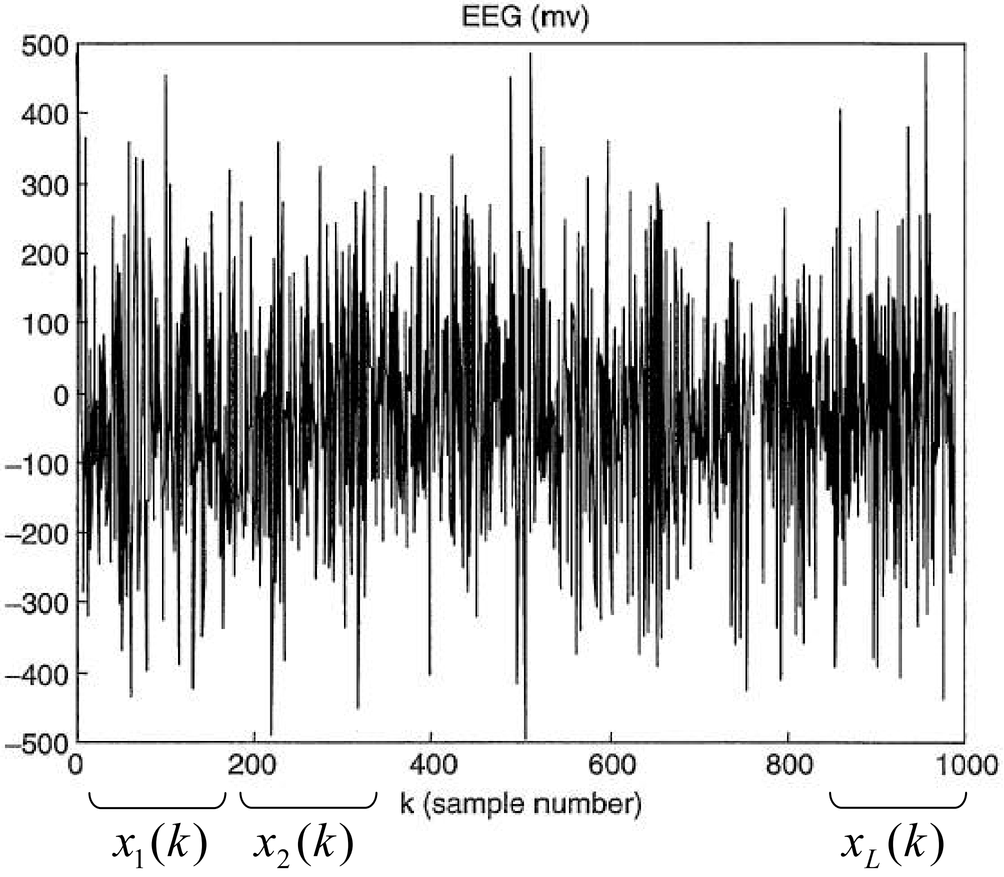
Average over all the periods

From the average waveform, we can get many useful parameters such as the maximum and minimum pressures, derivative of pressure rise during systole, and rate of decay during diastole

# Signal averaging – example

For signals that are more random (aperiodic), signal averaging in the frequency domain may be useful

Example: EEG signal is aperiodic and the frequency of the signal is of interest because it indicates the activity level of the brain

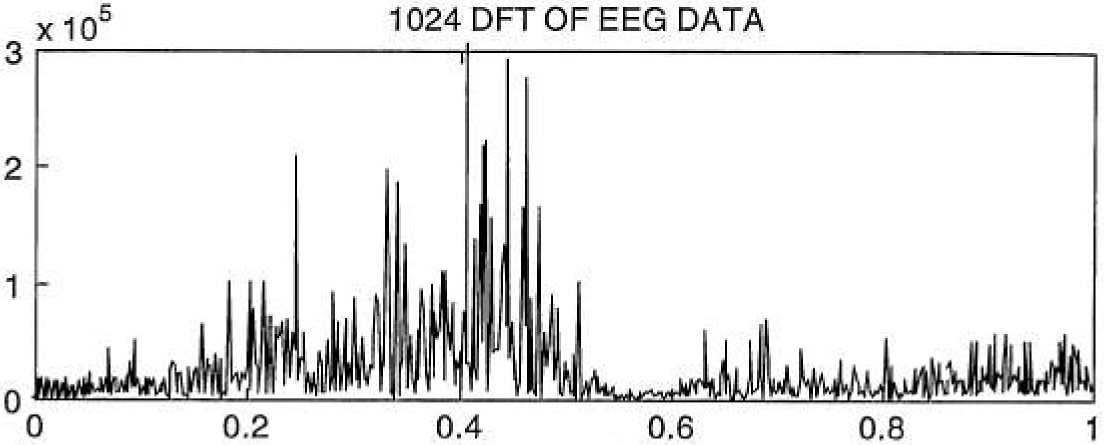


multiple segments of data

$$\begin{tikzcd}
x_i(k) \quad i=1,2,\ldots,L \arrow[d, "\text{DFT}"] & \\
X_i(m) \arrow[d, "|\bullet|^2"] & \\
P_i(m) \arrow[r, "\text{average}"'] & \overline{P}(m)
\end{tikzcd}$$

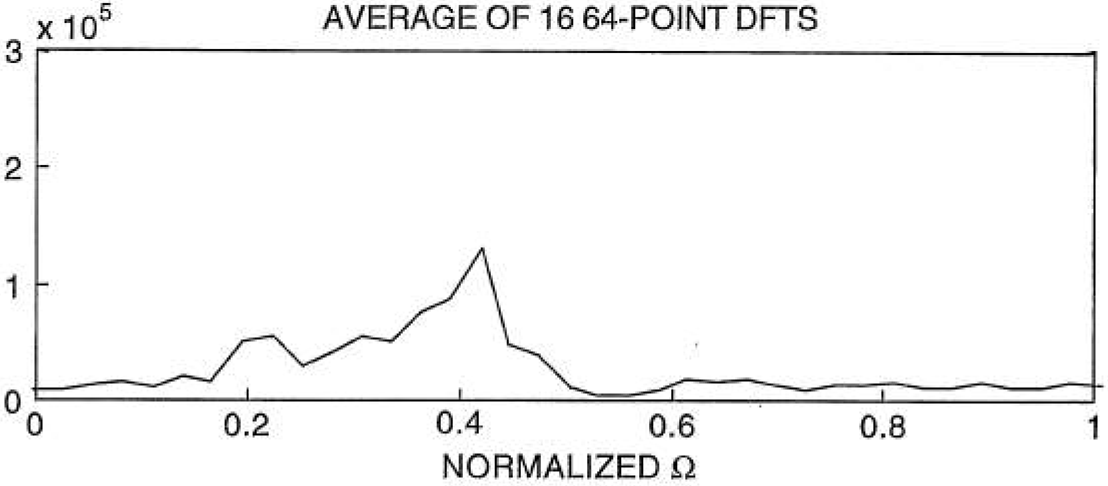
# Signal averaging (cont’d)

DFT (magnitude) of EEG data from previous slide



Raw data is divided into 16 segments, each containing 64 points

Average in the frequency domain

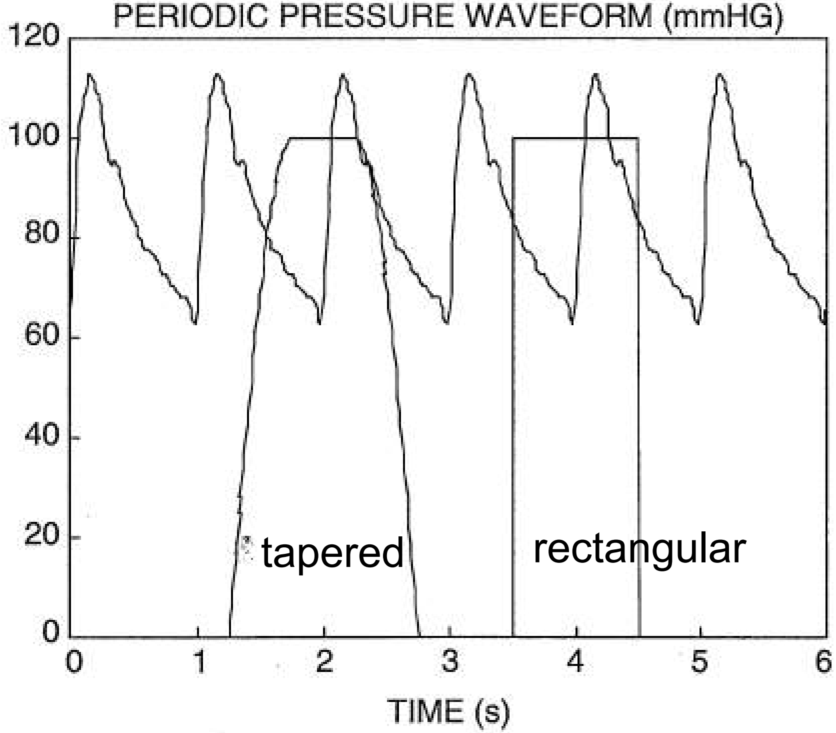


# Time-frequency analysis

To capture the “local” frequency characteristics of the signal, the short-time Fourier transform (STFT) can be used and is defined as

$$X(\omega, a) = \int_{-\infty}^{\infty} x(t) g(t-a) e^{-j\omega t} dt$$

where $g(t)$ is a window function which has a limited time span. $a$ is the amount of shift of the window function, therefore we can obtain the FT of the signal and know where in time it occurs



In practice, a sharp window is not the best choice due to the rippling effects it causes

# STFT – comments

However, the short-time Fourier transform has two major shortcomings:

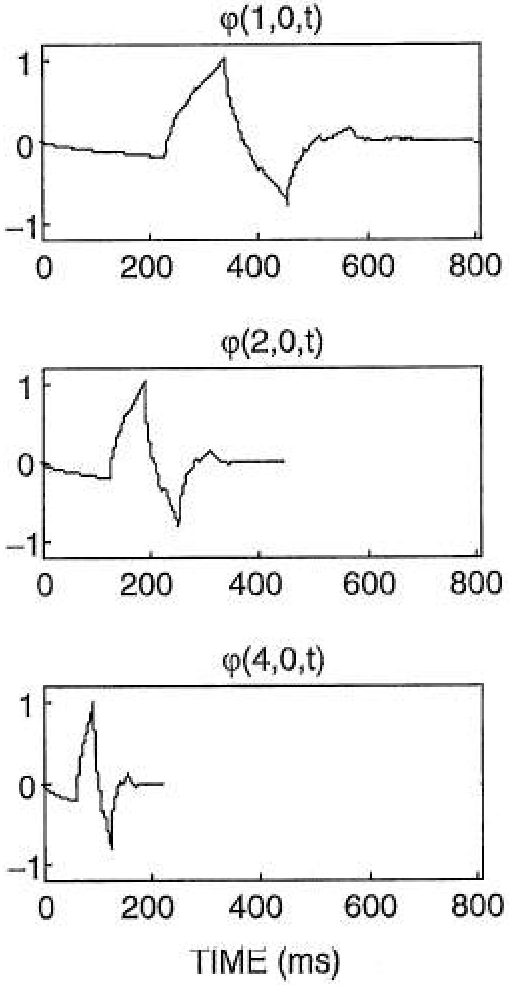
• The window length is fixed throughout the analysis. We are not able to capture events with different durations (without manually adjusting the length of window).

• The sinusoidal functions used in FT to model the signal are infinite in time. However, the characteristics of the signal we are interested in are local in time. Moreover, in DFT, the signal is assumed to be periodic, which is typically not the case. ⇒ it would be more advantageous to use time-limited basis functions (instead of sinusoids) to decompose and analyze the time-limited events.

# Wavelet transform (WT)

From the previous slide, we want to use basis functions that are time-limited and not periodic ⇒ choose a "mother wavelet"

To address the problem of fixed window widths, the concept of "scaling" is used

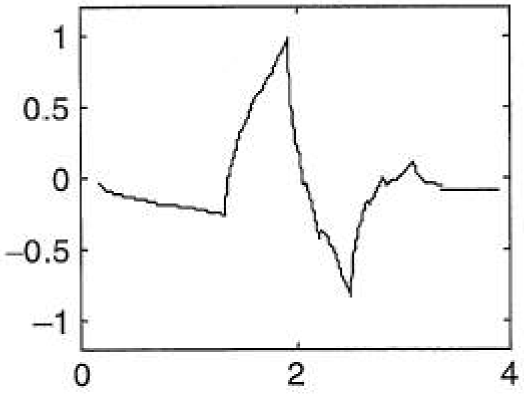


Original (mother) wavelet

Scaled by a factor of 1/2

Scaled by a factor of 1/4

The Daubechies (db) wavelet



The mother wavelet is scaled in time to create a series of "high-frequency" components similar to the harmonics in sinusoidal decomposition (Fourier series)

# Wavelet transform

The continuous wavelet transform can be expressed as

$$C(a,s)=\int_{-\infty}^{\infty}x(t)\varphi(\frac{t-a}{s})dt$$

where $a$ is the shifting factor and $s$ is the scaling factor

$\varphi(t)$ is the mother wavelet

Therefore, the basis functions in WT are the shifted and scaled versions of the mother wavelet

Every choice of mother wavelet gives a particular WT $\Rightarrow$ very important to choose most suitable mother wavelet for a particular task

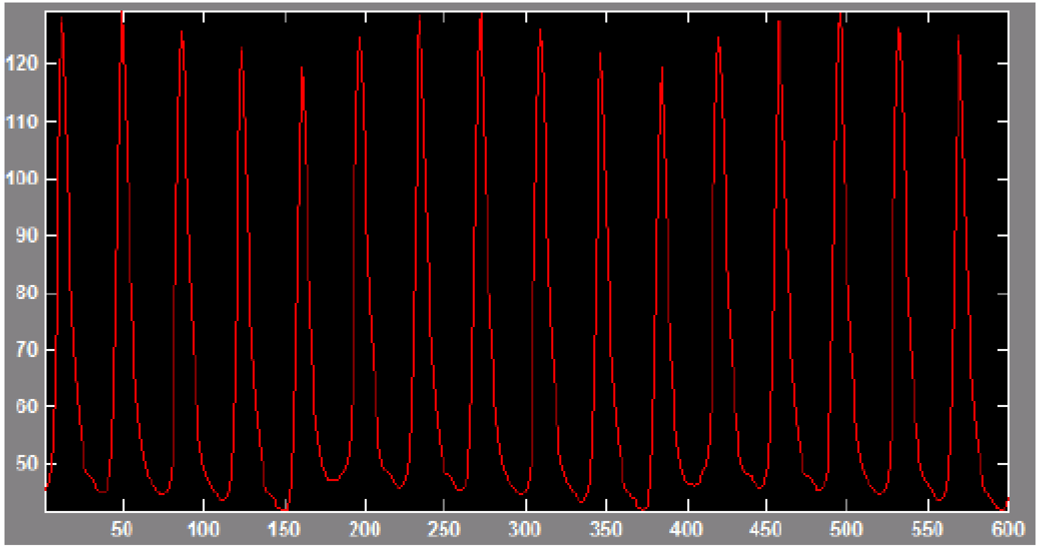
Rules of thumb: (a) use complex mother wavelets for complex signals

(b) Mother wavelet resembles the general shape of the signal to be analyzed

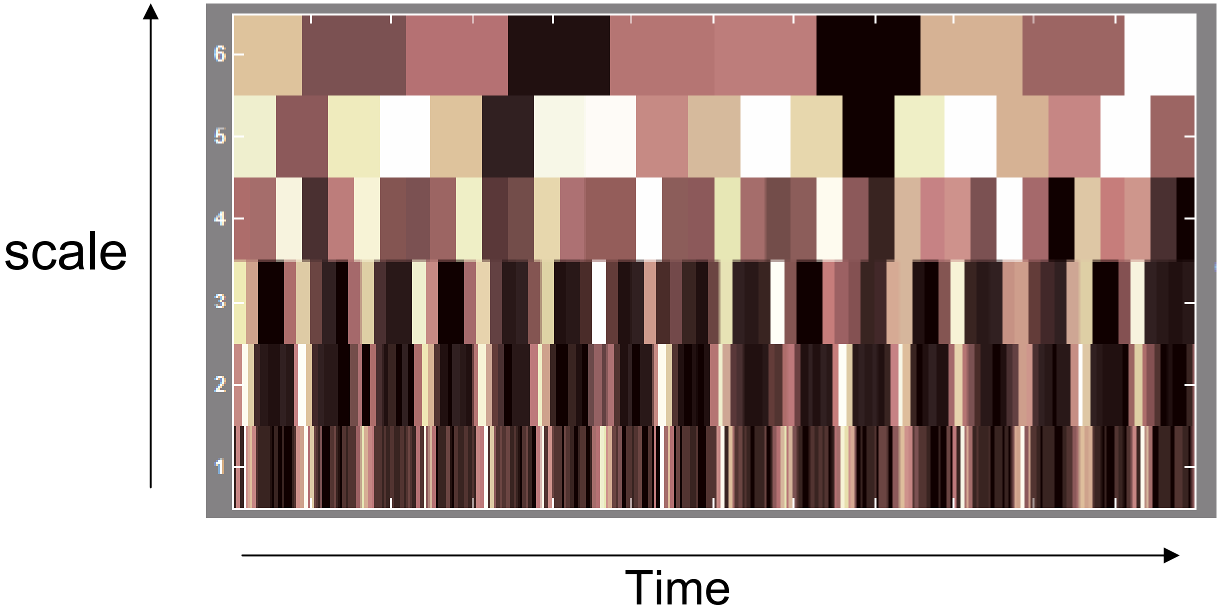
In practice, discrete wavelet transforms dealing with discrete signals are implemented by using digital low-pass and high-pass filters to decompose the signal into a series of “approximation” and “detail” components

# Wavelet transform – example

An example of WT results using Matlab



Raw data



WT coefficients

# Wavelet transform – example

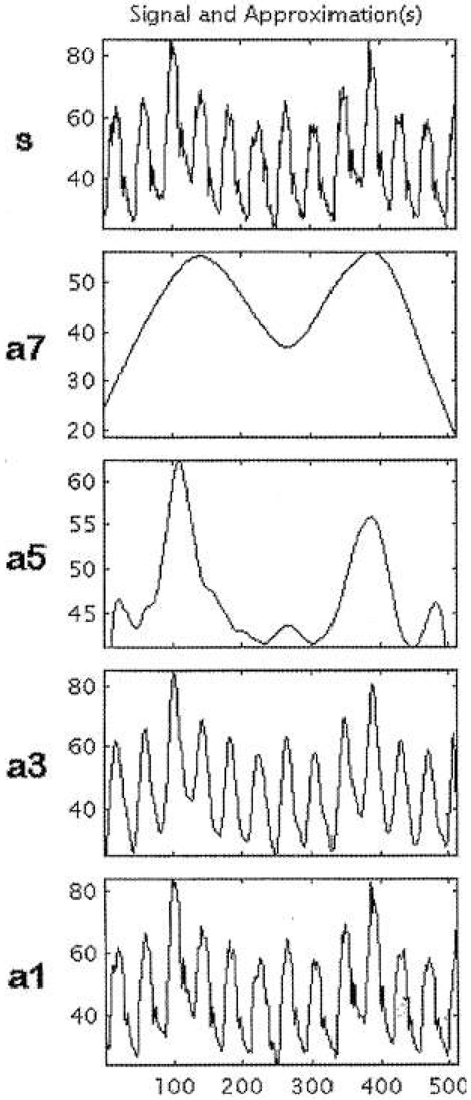
Blood pressure data, decomposition and reconstruction at different level of scales

approximations (low frequency)

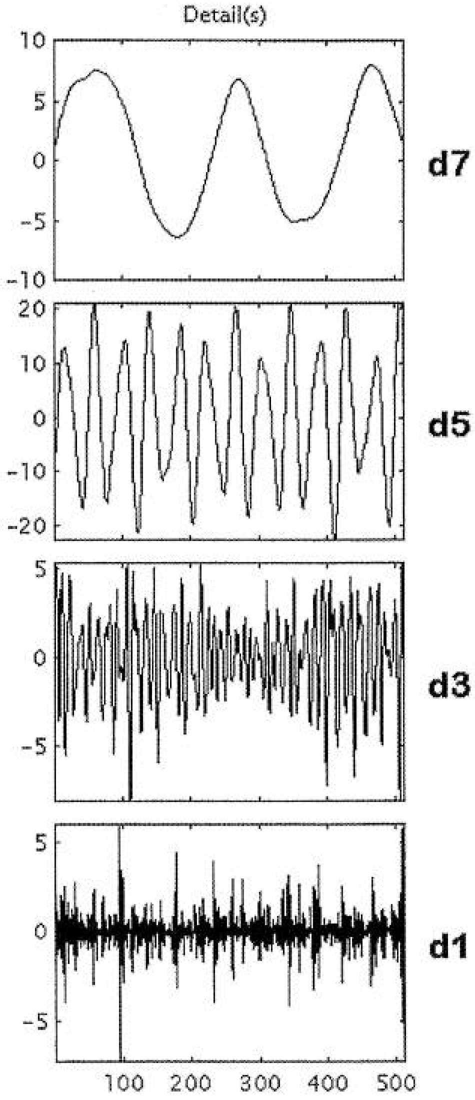
Raw data

Respiration

Heart beat



details (high frequency)



# Wavelet transform

- Applications of WT in biomedical signal and image processing: filtering, denoising, compression, and feature extraction

# Artificial neural networks (ANN)

The human brain is a complex, non-linear, highly parallel information processing system

The human brain consists of 100 billions of brain cells (neurons) that are highly interconnected

ANNs are computational methods inspired by the formation and function of biological neural structures

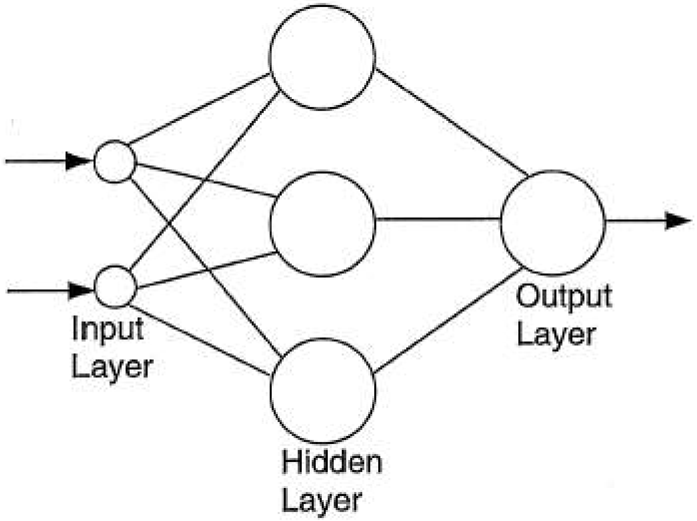
ANNs consist of much less number of neurons

ANNs are designed to learn from examples to recognize certain inputs and to produce a particular output for a given input

ANNs are commonly used for pattern detection and classification

# Artificial neural networks

Simple example of a multilayer neural network



Input layer: number of neurons equals to number of inputs

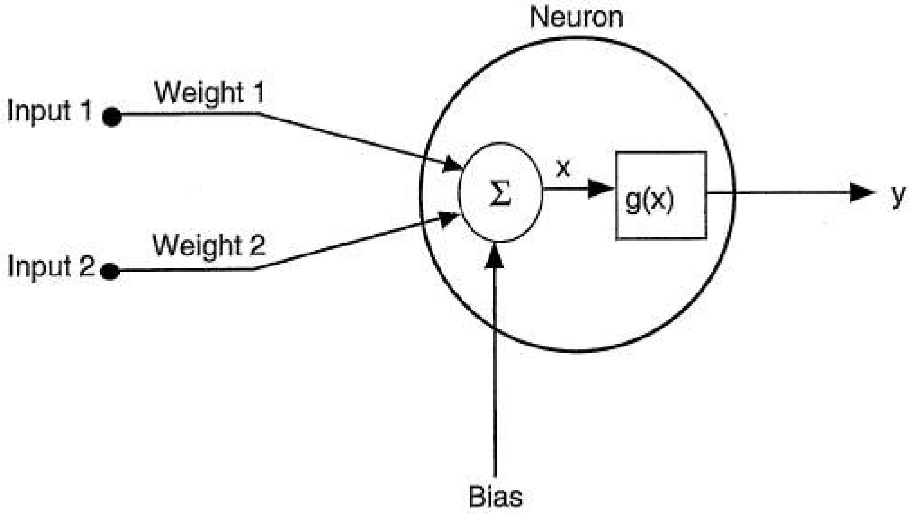
Output layer: number of neurons equals to number of outputs

There can be an arbitrary number of hidden layers, each of which can have an arbitrary number of neurons

The connections between neurons are represented in “weights”

# Artificial neural networks

Relationship between the inputs and the output of a neuron



$$x = bias + \sum_i input_i \cdot weight_i$$

$$y = g(x)$$

The function g(x) is called activation function to mimic the excitation of a biological neuron. Some examples of activation function include

Threshold function (T = threshold)

$$y = g(x) = \begin{cases} 1 & \text{if } x > T \\ 0 & \text{if } -T < x < T \\ -1 & \text{if } x < -T \end{cases}$$

Sigmoid function

$$y = g(x) = \frac{1}{1 + e^{-x}}$$

# Artificial neural networks

Training of the ANN (supervised learning)

- The goal of training is to teach it how to determine the particular class each input pattern belongs to
- The weights are initialized by random numbers
- Each example with a known class is fed into the ANN and the weights are updated iteratively to produce a better classification
- In backpropagation neural networks, the weights of the neural network are updated through the propagation of error (difference between the output and the target) from the output backward to the neurons in the hidden layers and input layers
- The training process terminates when the total error at the output no longer improves or a preset number of iterations has been passed
- Once the weights are found, the ANN is uniquely defined and ready to be used to analyze unknown data inputs

Note that ANNs require many training examples to learn a pattern. If the number of examples is not sufficient for a problem ⇒ overfit the training examples without learning the true patterns

# Artificial neural networks

- ANNs do not need any assumption about the distribution of data
- ANNs are nonlinear, therefore are suitable for solving complex nonlinear classification and pattern recognition problems
- ANNs with <u>unsupervised learning</u> (the outputs for given input examples are not known) have also been used in biosignal processing (ex. self-organizing feature maps networks)

# References

- Biomedical Signal and Image Processing, by Kayvan Najarian and Robert Splinter
  - Ch5: Wavelet Transform
  - Ch7.7: Neural Networks