Tfy-99.275 Lecture 3

Processing of non-stationary signals

Stationarity

defined as a quality of a process in which the statistical parameters of the process do not change with time

- A *truly stationary (or strongly stationary)* process has all higher-order moments constant (implying constant mean, variance, skewness, kurtosis,.. of the sample value distribution)
- For practical purposes - a *weakly stationary* process has a constant mean and variance

intermezzo – what are moments?

- moments of a distribution, p(x)

$$\text{1st order moment}: E[x] = \int_{-\infty}^{\infty} x \cdot p(x) \cdot dx = \bar{x}$$

$$\text{2nd order moment}: E[x^2] = \int_{-\infty}^{\infty} x^2 \cdot p(x) \cdot dx$$

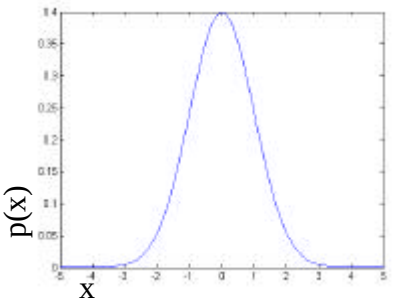
- in practice we often use *central* moments

$$\text{central moment of order } n: \; m_{nc} = E\left[(x - E[x])^n\right] = E\left[(x - \bar{x})^n\right]$$

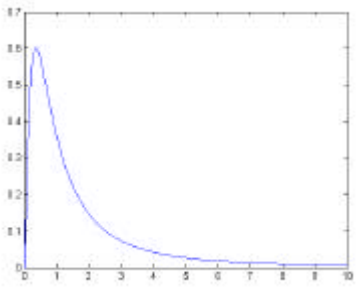
2nd order central moment is variance

$$\text{skewness factor}: \frac{m_{3c}}{(m_{2c})^{\frac{3}{2}}}$$ 'asymmetry' (=0 for symmetric distributions)

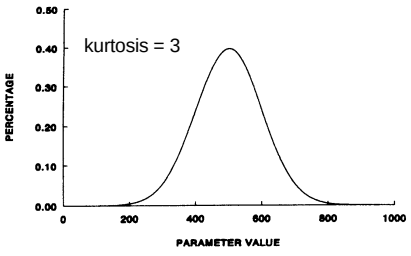
$$\text{kurtosis factor}: \frac{m_{4c}}{(m_{2c})^{2}}$$ indicates how 'outlier-prone' the distribution is



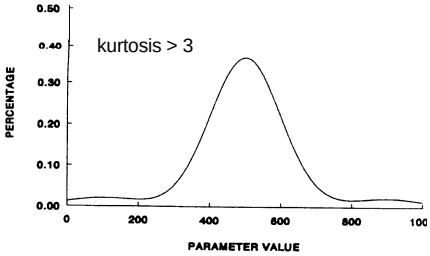
Gaussian with mean=0 and variance=1
For all Gaussians skewness=0 and kurtosis=3



A positively skewed distribution
(positive->distribution stretches
out further to the right than to the left,
negative would be the other way around)



Gaussian distribution



distribution with outliers

Stationarity

- strict stationary processes are never seen in practice and are discussed mainly for their mathematical properties
- stationarity is a relative term, not an absolute definition (e.g., dependent on the time scale you are using for analysing data)
- ‘close enough’/’weak’ stationarity is used in many applications to make life easier

(Non)Stationarity in Physiological measurements

- Measurements from living beings are typically nonstationary
- We thus have to be aware that the methods we use are applicable to changing situations

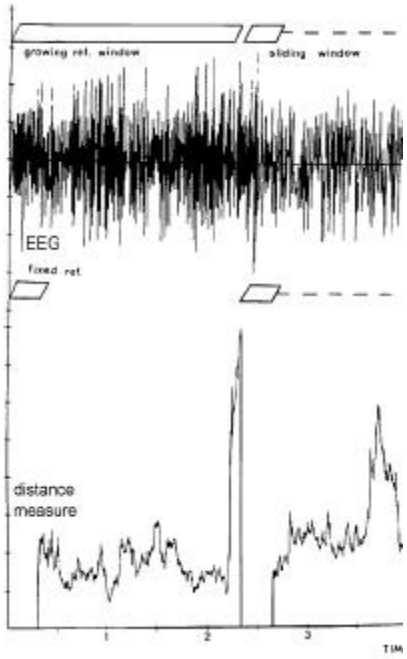
Recall: Short-Term Fourier Transform (STFT)

- Fourier techniques assume stationarity of the signal, most signals (also biomedical ones) are not stationary $\rightarrow$ divide signal into 'almost stationary' segments ('segmentation')
- the duration of each segment has to be determined by using *a priori* information or by examining the signal's local characteristics

$$STFT(f,t) = FT(x(t) \cdot w(t-t))$$

- perform Fourier transform on a window of data that slides along the time axis $\rightarrow$ time-frequency function that describes the frequency distribution near time τ.

- Time-frequency resolution trade-off: shorter segments (narrower windows) give better resolution in the time domain but increase the window width in the frequency domain and thus lead to poorer frequency resolution.
- In highly non-stationary signals usually equal length windows are used
- For many other signals, variable duration windows are used (e.g., EEG signals windows in the order of 5 to 30 s) – many different schemes exist to obtain such variable-sized windows



example
estimation of variable duration windows

Compare statistics (e.g., mean, variance,..) of data in a sliding window with those of a data in a reference window.
If the difference increases to exceed a preset threshold a segment border is identified.

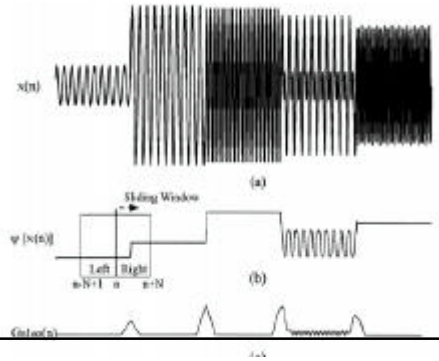
The reference window can either be constant or grow in time.

other example: non-linear energy operator (NLEO) (Agarwal et al)

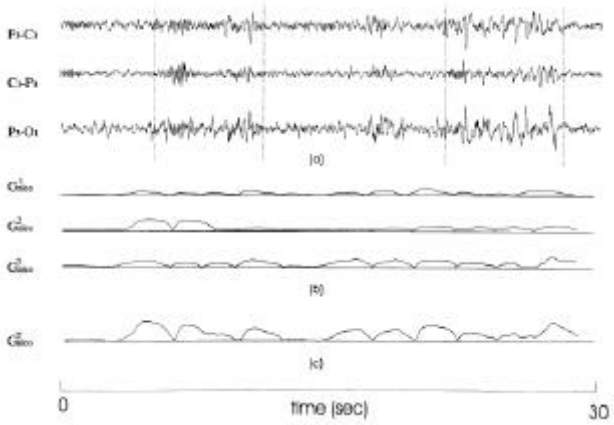
$$\Psi(n) = x(n-1)x(n-2) - x(n)x(n-3).$$

- calculate for samples at window positions n and n+1 and use as segmentation criterion:

$$G_{\text{nleo}}(n) = \sum_{m=n-N+1}^{n} \Psi(m) - \sum_{m=n+1}^{n+N} \Psi(m) \quad (2)$$



multichannel version uses summation of individual Gnleo values

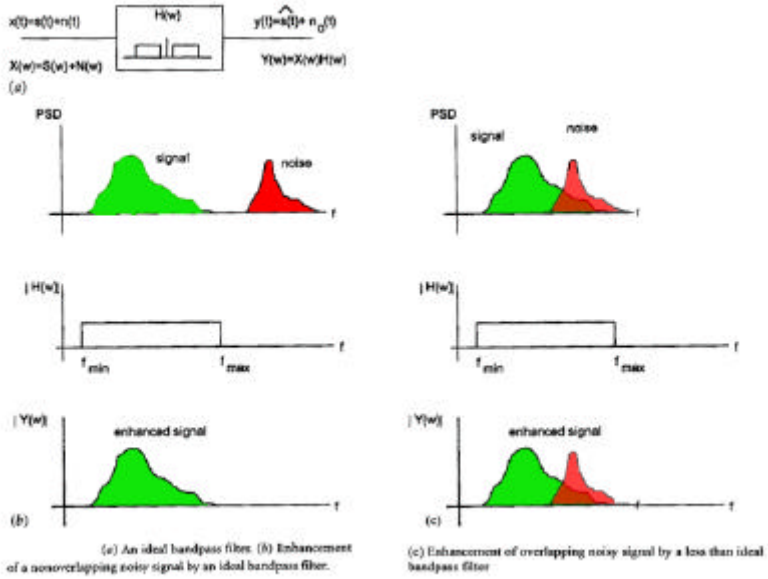


Filtering

- *optimal filtering*: obtain best possible filter, usually using assumptions about stationarity and *a priori* knowledge
- *time-varying (adaptive) filtering*: obtain filters that aim to achieve optimal filtering in changing environments and in environments in which the signal properties are not known beforehand

Optimal filtering

- problem of overlap of signals and noise in the frequency domain



(a) An ideal bandpass filter. (b) Enhancement of a nonoverlapping noisy signal by an ideal bandpass filter. (c) Enhancement of overlapping noisy signal by a less than ideal bandpass filter

Optimality Criterion

- To reach a filter that is optimally suited to deal with this kind of situations requires the definition of an optimality criterion
- The optimality criterion has to be chosen on the basis of what one wants to achieve with processing the data: different criteria lead to different optimal filters

Examples of some often used criteria

- minimization of mean squared error (MSE): $\min E\left[|e|^2\right]$
- maximization of the signal-to-noise ratio: max SNR
- least squares error criterion (LSE): $\min \sum_n |e_n|^2$

MSE uses a stochastic point of view, LSE regards the time series as deterministic and uses specific sequences of the record

Minimisation of the mean squared error (Wiener filtering)

- problem: develop a filter that estimates for time, $t+x$, the value of the signal $s(t+x)$ on the basis of a measurement $x(t)$, that contains signal, $s(t)$, and noise, $n(t)$. The filter output is $y(t)$. The case with $x=0$ is called smoothing, the case for $x>0$ is called prediction. We will assume $x=0$
- We can develop an IIR or FIR filter – in this example we'll use an FIR* filter with output: $y(t)=\sum_{l=0}^{L} h(l)\cdot x(t-l)$
- Define the error of the filter as the difference between the filter's output and its desired output $e(t) = s(t) - y(t)$

*FIR filters are a convenient choice because then we don't have to worry about stability issues

The sum of squared errors, ε

$$
\begin{aligned}
e &= \sum_{t=0}^{N} e^2(t) = \sum_{t=0}^{N} [s(t) - y(t)]^2 \\
&= \sum_{t=0}^{N} \left[s(t) - \sum_{l=0}^{L} h(l) \cdot x(t-l) \right]^2 \\
&= \sum_{t=0}^{N} s^2(t) - 2\sum_{l=0}^{L} h(l) \cdot r_{sx}(l) + \sum_{l=0}^{L} \sum_{m=0}^{L} h(l) \cdot h(m) \cdot r_{xx}(l-m)
\end{aligned}
$$

This is a quadratic function of FIR filter coefficients with the autocorrelation function (acf) of *x* and the crosscorrelation function (ccf) between *s* and *x* in it.

Find the minimum of the sum of squared errors

$\frac{\partial e}{\partial h(l)} = 0$ leads to

$$\sum_{l=0}^{L} h(l) \cdot r_{xx}(l-m) = r_{sx}(m)$$

This is the **Wiener-Hopf equation**, it represents a series of L+1 equations that must be solved simultaneously to find the optimal filter coefficients

$$
\begin{bmatrix}
r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(L) \\
r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(L-1) \\
\vdots & \vdots & \ddots & \vdots \\
r_{xx}(L) & r_{xx}(L-1) & \cdots & r_{xx}(0)
\end{bmatrix}
\begin{bmatrix}
h(0) \\ h(1) \\ \vdots \\ h(L)
\end{bmatrix}
=
\begin{bmatrix}
r_{sx}(0) \\ r_{sx}(1) \\ \\ r_{sx}(L)
\end{bmatrix}
$$

note: this is a symmetrical matrix with a so-called Toeplitz structure, Matlab can set it up quickly for you using the toeplitz() function

A more compact way of writing

$$\mathbf{Rh} = \mathbf{r}_{sx}$$

the solution for the FIR coefficients is then

$$\mathbf{h} = \mathbf{R}^{-1}\mathbf{r}_{sx}$$

note that we need to have information about the signals s and x in order to solve this!

Alternatively, by using the definition of the (cross) power spectral density as the Fourier transform of the ccf and acf we can write the Wiener-Hopf equation as:

$$H(w)\cdot S_{xx}(w) = S_{sx}(w)$$

or,

$$H(w) = \frac{S_{sx}(w)}{S_{xx}(w)} = \frac{S_{sx}(w)}{S_{ss}(w) + S_{nn}(w)}$$

Optimal (Wiener) Filter

- if signal and noise are uncorrelated and either the signal or the noise has zero mean, the frequency response of the Wiener filter becomes:

$$H_{opt}(w) = \frac{S_{ss}(w)}{S_{ss}(w) + S_{nn}(w)} = \frac{S_{ss}(w)}{S_{xx}(w)}$$

big problem: this requires *a priori* knowledge of the PSD of both the input signal and the desired signal, which is often not available

Wiener Filter

- Note: the solution presented here is not ’optimally-optimal’ due to the fact that we restricted our search to find a causal and discrete version of the filter. The best solution is given if we allow the filter to be non-causal (and non-discrete).
- a method that is called *a posteriori* Wiener filtering is sometimes used to circumvent the problem of needing *a priori* knowledge: this method relies on estimations of the PSD's to generate an 'estimated' optimal filter

Adaptive filtering

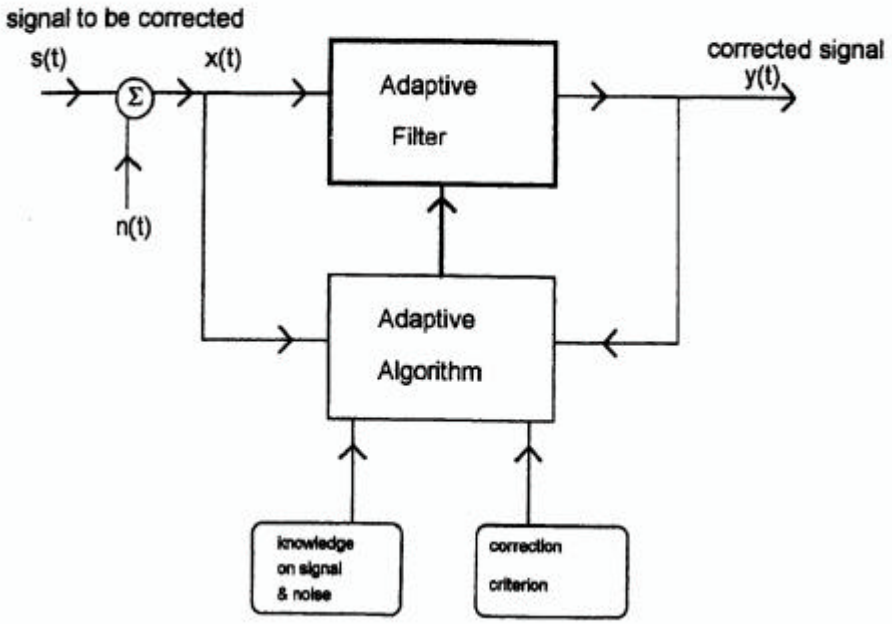
- optimal filters: stationary signals with known PSD. In reality this information is often not available and the signals are not stationary.
- processing non-stationary signals requires a filter that 'learns' signal properties and adjusts itself continuously to perform optimally under changing circumstances: adaptive filters.
- many similarities with adaptive algorithms in control theory
- typically, adaptive filters perform poorly in the initial phase as they will have to 'learn' the knowledge that is not *a priori* available

General Structure of an Adaptive Filter

- Three main parts:
  1. performance index/criterion: preferably using something 'easy' like minimising squared errors

  2. adaptive algorithm that changes the parameters
  non-recursive: e.g., solve exact least square problem on recorded data in a given time window, or
  recursive: update after every sample, e.g, gradient methods

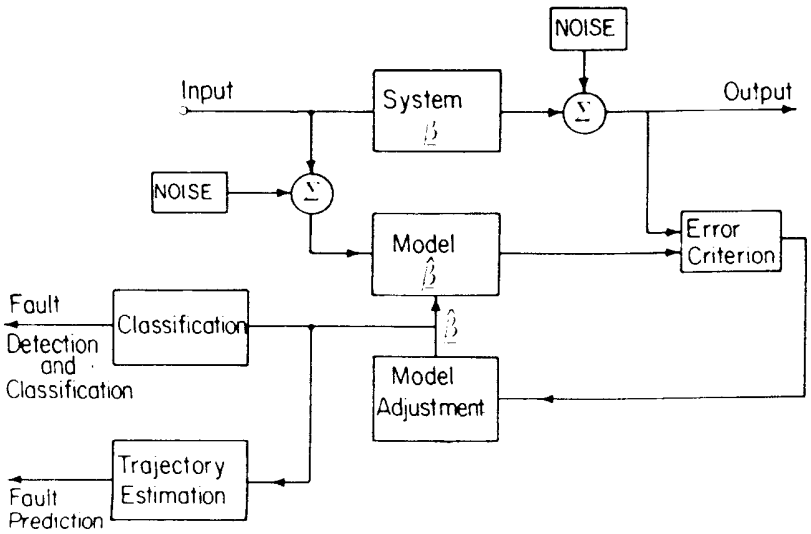
  3. structure of the filter itself



Application example

- **Adaptive System Parameter Identification**
  a typical problem in control systems: one knows the dynamic behaviour of a system (in terms of differential equations), but the parameters need to be estimated.

- Biomedical apps: (neuro)physiological system identification,control of drug delivery systems, control of movement



Applications

- **Adaptive Signal Estimation**
  Problem: given a noisy or distorted recorded output signal, try to estimate the actual output signal. We may or may not have access to (noisy) inputs.

- Here, we do not require information on the system nor do we wish to model the system, just estimate the correct output

Least Mean Squares (LMS) adaptive filter

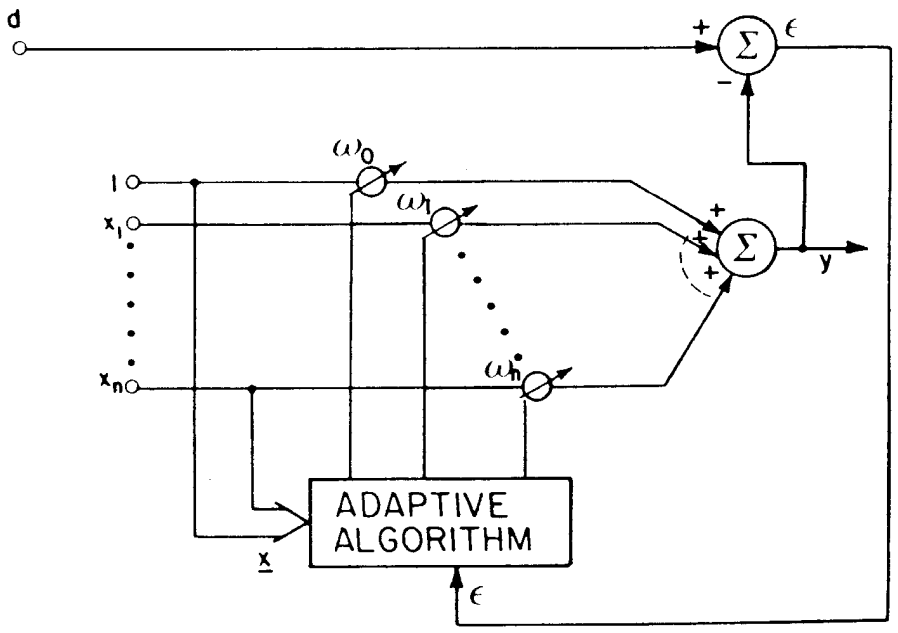
developed by Widrow and Hoff (1960)

it has reference inputs, variable gains multipliers (weights), an adaptation algorithm, an additional input denoted 'primary input' (or 'bias') and has access to a 'desired' output

its main component is the adaptive linear combiner

(a.k.a. adaptive linear element: ADALINE)

note: these are the same building blocks many artificial neural networks have, the difference is in the linearity/non-linearity - indeed, a 'backpropagation neural network' can be viewed upon as a non-linear version of the LMS adaptive filter



We consider:
An input vector dealing with n input values:

$$\mathbf{x}^T = \left(x_0, x_1, x_2, \ldots, x_n\right)$$

x_0 is typically set to 1 (it is used to take care of biases in the inputs)

The vector of variable gains (weights):

$$\mathbf{w}^T = \left(w_0, w_1, w_2, \ldots, w_n\right)$$

The actual output:

$$y = \mathbf{w}^T\mathbf{x} = \sum_{i=0}^{n} w_i \cdot x_i = w_0 \cdot 1 + w_1 \cdot x_1 \ldots + w_n \cdot x_n$$

The desired output: d

consider a number of input/ desired output pairs: $(\mathbf{x}_k, d_k)$

for the performance criterion we introduce a cost function, G:

$$G(\mathbf{w}) = E\left[(d_k - y_k)^2\right] \quad \text{or}$$

$$G(\mathbf{w}) = \lim_{N\to\infty}\left(\frac{1}{N}\right)\sum_{k=1}^{N}(d_k - y_k)^2$$

the goal is to minimize this cost function

This cost function represents a parabolic surface, we can 'slide' towards the minimum of this function by moving the weights in the opposite direction of its gradient (steepest descent): $-\nabla_w G$

using $\nabla_w(-y_k) = -\mathbf{x}_k$, we get

$$\nabla_w G(\mathbf{w}) = \lim_{N\to\infty}\left(\frac{1}{N}\right)\sum_{k=1}^{N} 2\cdot(d_k - y_k)\cdot(-\mathbf{x}_k)$$

$$\nabla_w G(\mathbf{w}) = -2E\left[d_k\mathbf{x}_k\right]$$

with d_k the error made with processing input k

- this implies: average a large number of vectors $d_k \mathbf{x}_k$, multiply the result by 2 and move the weights in that direction

- Widrow & Hoff: update the weights after every presentation of a new input/ desired output pair → **delta rule**

$$\mathbf{w}_{k+1} = \mathbf{w}_k + a d_k \mathbf{x}_k$$

α is usually referred to as the learning rate

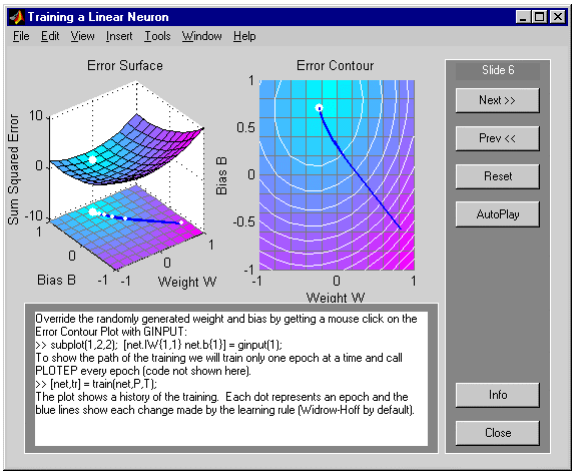
It can be shown that the weight vector converges to one obeying the Wiener-Hopf equation if the inputs are uncorrelated over time

some variations:

- update the weight vector only after a number of inputs have been presented, this adheres more to the 'limit' version of updating (this is often called 'batched updating')
- use the most recent weight update also in the current weight update (momentum term, *m*)

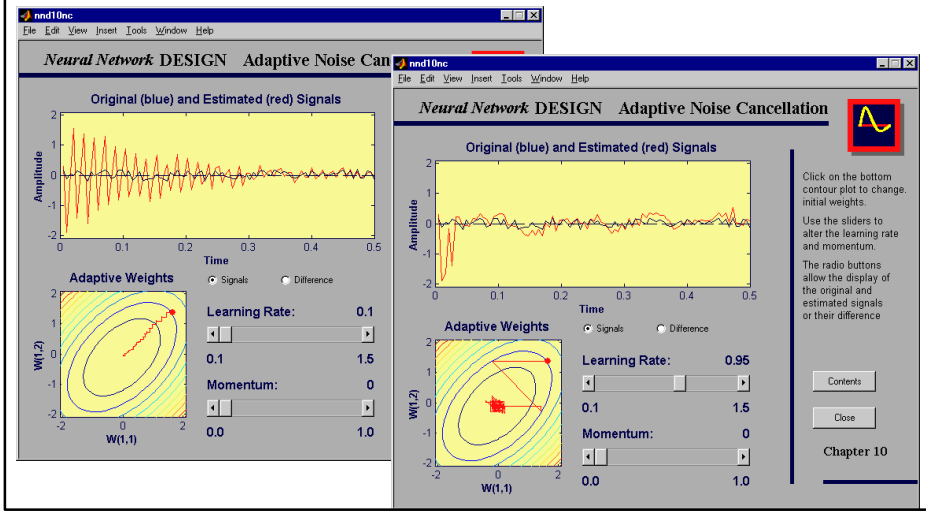
$$\mathbf{w}_{k+1} = \mathbf{w}_k + a d_k \mathbf{x}_k + m \cdot \left(\mathbf{w}_k - \mathbf{w}_{k-1} \right)$$

Example parabolic surface

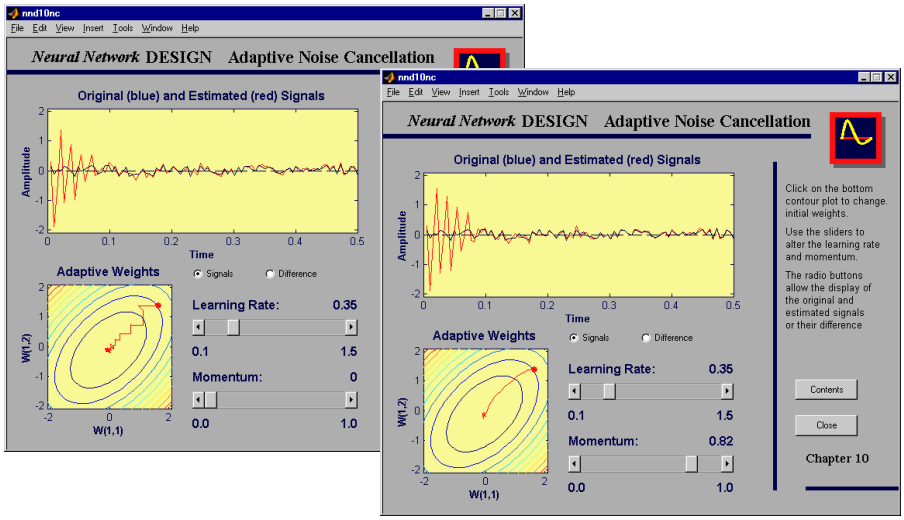


(from Matlab's NNToolbox demo)

Choice of learning rates (small = safe but slow large = fast but can be 'risky')



Use of momentum term to make process smoother



This algorithm is very easy to implement (it does not require differentations or matrix inversions) and can be quite powerful and fast.

For really complex problems however, the fact that the thing is linear makes its usefulness somewhat limited $\rightarrow$ solution: change the output function:

$$y = \sum_{i=0}^{n} w_i \cdot x_i \quad \rightarrow \quad y = f\left(\sum_{i=0}^{n} w_i \cdot x_i \right)$$

with f some non-linear function and we 'suddenly' have the basic element of a neural network. Combining lots of those elements and using, e.g., the *generalized* delta rule we are (in principle) able to solve arbitrarily complex problems

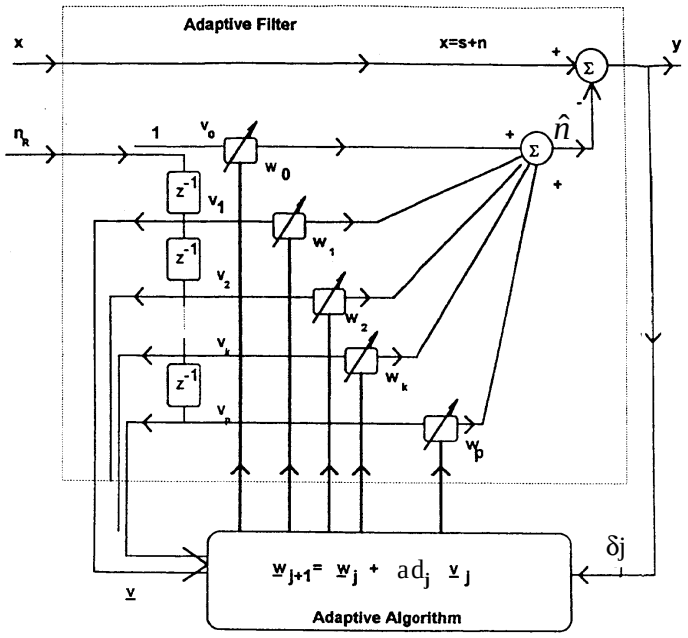
Adaptive noise canceller with reference input

The recorded signal noise, $x(t)$, contains a noise component, $n(t)$. Try to find an estimation of it, $\hat{n}(t)$, subtract it from $x(t)$ and thus obtain an estimate, $\hat{s}(t)$, for the 'pure' signal, $s(t)$

use a reference noise input that is somehow correlated with noise in the recorded signal: $n_R(t)$

output of system:

$$y(t) = x(t) - \hat{n}(t) = s(t) + \left[n(t) - \hat{n}(t)\right] = \hat{s}(t)$$



Since there is a correlation between the noise in the recorded signal and the reference noise, we can write:

$$N_R(w) = H(w) \cdot N(w)$$

i.e., the reference noise can be represented as the output of an unknown filter $H(w)$.
An adaptive filter estimates the inverse of this unknown filter and from it estimates the noise:

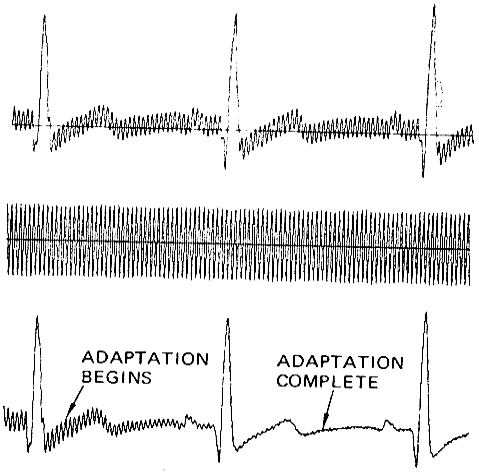
$$\hat{n}(t) = FT^{-1}\left\{\hat{H}^{-1}(w) \cdot N_R(w)\right\}$$

minimize the mean squared error (using that signal and noise are uncorrelated):

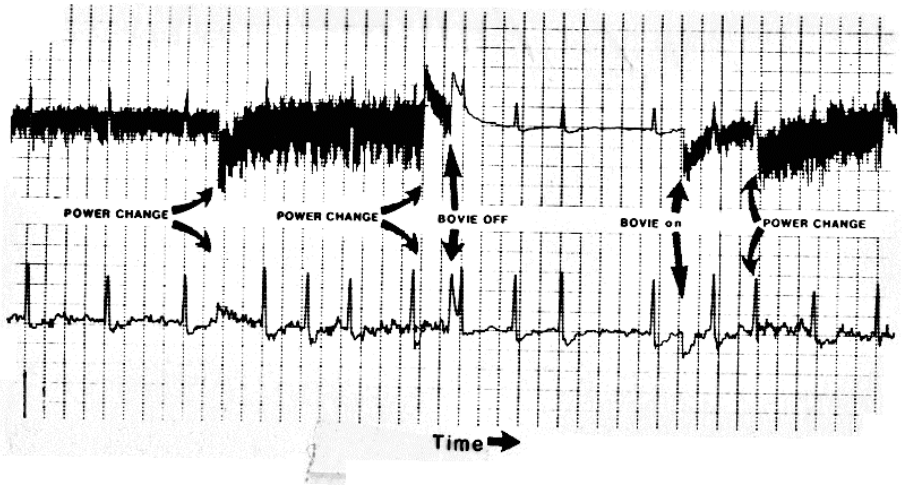
$$E\left\{e^2\right\} = E\left\{y^2(t)\right\} = E\left\{\left(s(t) + \left[n(t) - \hat{n}(t)\right]\right)^2\right\}$$
$$= E\left\{s^2(t)\right\} + E\left\{\left[n(t) - \hat{n}(t)\right]^2\right\}$$

The filter only affects the estimated noise → minimization of noise term minimizes the whole equation

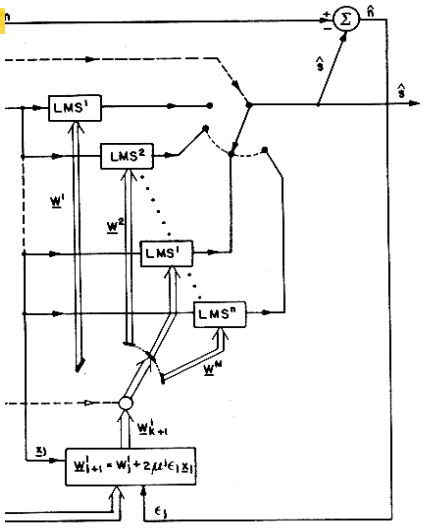
adaptive cancellation of power line interference



cancellation of electrosurgical noise



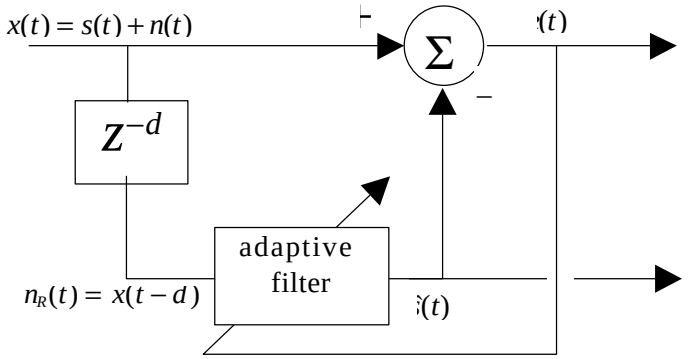
Time-sequenced adaptive filtering



adaptive line enhancer (ALE)

- In practice it is not always possible to obtain a separate channel that provides reference noise.
- In such a situation we can create a reference signal ourselves from the main signal and then use the earlier described adaptive noise canceller method

adaptive line enhancer



input signal:
$x(t) = s(t) + n(t)$

reference input is a delayed replica of the input signal
$n_R(t) = s(t-d) + n(t-d)$

estimatedsignal:

$$\hat{s}(t) = \sum_{m=0}^{M} w_m(t) \cdot n_R(t-m)$$

$$\hat{s}(t) = \sum_{m=0}^{M} w_m(t) \cdot x(t-m-d)$$

the error: $e(t) = x(t) - \hat{s}(t)$

1.calculate estimated signal
2.estimate error at the output
3.update
weights:
$w_m(t+1) = w_m(t) + 2\mu \cdot e(t) \cdot x(t-m-d)$
4.repeat

- For example, if we have a signal that is a sinusoid buried in white noise.
- By choosing an appropriate delay the ALE decorrelates the noise in the reference input from the noise in the main input. However, the signal (the sinusoid) in both inputs is still correlated. Eventually ALE will tune itself so that it produces produce a sharp peak at the frequency of the sinusoid.

Adaptive line enhancement of diastolic heart sounds

- noninvasive assessment of coronary occlusions by recording heart sounds. (detection of coronary artery disease [CAD])
- these measurements are heavily corrupted by background noise.
- ALE filtering effectively eliminated the background noise

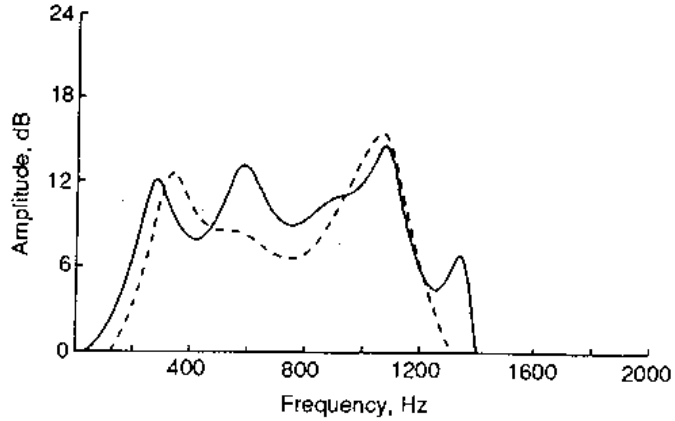


Fig. 8.10. PSD function obtained from the ARMA model applied to the diastolic heart sounds of a CAD patient. Solid line, after ALE; broken line, before ALE. [From Akay *et al.* [18]].

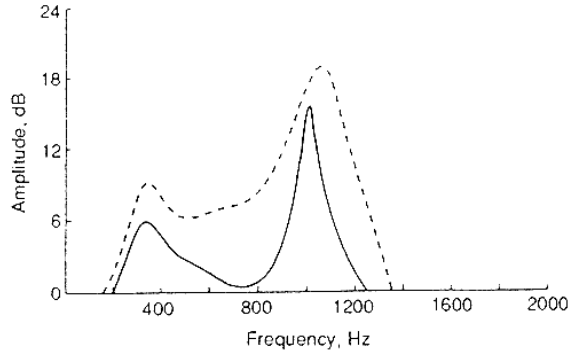


Fig. 8.11. PSD function obtained from the ARMA model applied to the isolated heart sounds of a normal patient. Solid line, after ALE; broken line, before ALE. [From Akay *et al.* [18]].