VIII. Statistical signal Processing

In this chapter we describe principles of statistical signal processing and tools that can be used to perform filtering, estimation, prediction and detection of random signals. We start with a review of discrete-time random processes and describe how signals and systems that occur in a variety of applications can be modelled by probabilistic tools. We then introduce signal modelling techniques which show how physical phenomena and systems can be represented by transfer functions, or equivalently, by filters or parameter vectors.

We then introduce the discrete-time Wiener filter which is a very powerful tool to estimate parameters, extract information and design signal processing systems. Linear prediction is then described along with the problem of performing prediction of signals and general parameters.

The chapter concludes with an introduction of adaptive signal processing and how a designer can employ adaptive algorithms to perform filtering, estimation, prediction and detection in the absence of statistical knowledge about signals and systems. We show that adaptive algorithms can learn the statistics about the environment and track slow variations.

A. Review of Random Processes

A random signal or a stochastic signal needs to be modelled using statistical information about it. Examples of such signals include speech, music and seismic signals. Discrete-time random signals are models of random processes, which are collections of random variables $X[n]$ characterized by a set of probability distribution functions. where n is the time index. In practice, each sample value $x[n]$ of a random signal is assumed to have resulted from a mechanism that is governed by a probability law.

Random variables:

A random variable $X[n]$ is described by the probability distribution function given by

$$P_{X[n]}(\alpha\_) = Pr\left[X[n] < \alpha\right],$$

where $\alpha\_$ is a particular value of $X[n]$.

If $X[n]$ takes on a continuous range of values, it can be specified by the probability density function of $X[n]$:

$$p_{X[n]}(\alpha) = \frac{\partial P_{X[n]}(\alpha)}{\partial \alpha}$$

or equivalently

$$P_{X[n]}(\alpha) = \int_{-\infty}^{\alpha} p_{X[n]}(u)\, du$$

Properties of the probability density function:

$$p_{X[n]}(\alpha) \geq 0$$

$$\int_{-\infty}^{\infty} p_{X[n]}(\alpha)\, d\alpha = 1$$

Properties of the probability distribution function:

$$0 \leq P_{X(n)}(\alpha) \leq 1$$

$$P_{X(n)}(\alpha_1) \leq P_{X(n)}(\alpha_2) \quad , \quad \text{for all } \alpha_2 > \alpha_1$$

$$P_{X(n)}(-\infty) = 0 \quad , \quad P_{X(n)}(+\infty) = 1$$

$$Pr\left[\alpha_1 < X(n) \leq \alpha_2\right] = P_{X(n)}(\alpha_2) - P_{X(n)}(\alpha_1)$$

A random variable is characterized by a number of statistical properties. The rth moment of a random variable is described by

$$\mu_r = E\left[X^r_{(n)}\right] = \int_{-\infty}^{\infty} \alpha^r \, p_{X(n)}(\alpha) \, d\alpha ,$$

where r is any nonnegative integer and $E[\cdot]$ denotes the expectation operator. A random variable is completely characterized by all its moments.

The most commonly used statistical properties of a random variable are the mean or expected value m_x; the mean-square value $E[X^2_{(n)}]$ and the variance σ_x^2 defined by:

$$m_x = E[X(n)] = \int_{-\infty}^{\infty} \alpha \, p_{X(n)}(\alpha) \, . \, d\alpha$$

$$E[X^2(n)] = \int_{-\infty}^{\infty} \alpha^2 \, p_{X(n)}(\alpha) \, . \, d\alpha$$

$$\sigma^2_x = E\left[(X(n) - m_x)^2\right] = \int_{-\infty}^{\infty} (\alpha - m_x)^2 \, p_{X(n)}(\alpha) \, . \, d\alpha .$$

There is also a relation between the quantities that is useful and is described by

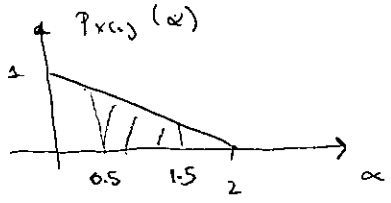
$$\sigma_x^2 = E\left[X^2_{(n)}\right] - (m_x)^2 .$$

where it turns out that $\sigma_x^2 = E[X^2(n)]$ for a random variable with zero mean.

Ex: Let a random variable be characterized by a probability density function

$$p_{x(n)}(\alpha) = \begin{cases} 1 - \frac{\alpha}{2}, & \text{for } 0 \le \alpha \le 2 \\ 0, & \text{otherwise} \end{cases}$$

as shown in the figure below



a) Compute its probability distribution function

$$P_{x(n)}(\alpha) = \int_{-\infty}^{\alpha} p_{x(n)}(u)\, du$$

$$= \int_{0}^{\alpha} \left(1 - \frac{u}{2}\right) du = \alpha - \frac{\alpha^2}{4}, \quad 0 < \alpha \le 2$$

b) Compute the probability that $x(n)$ is in the range $0.5 < x(n) \le 1.5$.

$$P_r\left[0.5 < x(n) \le 1.5\right] = P_{x(n)}(1.5) - P_{x(n)}(0.5)$$

$$= \frac{15}{16} - \frac{7}{16} = \frac{1}{2}$$

c) Compute the mean, the mean-square value and the variance

$$m_x = \int_0^2 \alpha\left(1 - \frac{\alpha}{2}\right) d\alpha = \int_0^2 \alpha - \frac{\alpha^2}{2}\, d\alpha$$

$$= \frac{\alpha^2}{2} - \frac{\alpha^3}{6}\Big|_0^2 = \frac{4}{2} - \frac{8}{6} = \frac{4}{6} = \frac{2}{3}$$

$$E[x^2(n)] = \int_0^2 \alpha^2\left(1 - \frac{\alpha}{2}\right) d\alpha = \frac{2}{3}$$

$$\sigma_x^2 = E[x^2(n)] - (m_x)^2 = \frac{2}{3} - \left(\frac{2}{3}\right)^2 = \frac{2}{9}$$

Amongst the most commonly used probability density functions in digital signal processing applications are the uniform density function and the Gaussian density function. The uniform density function is defined by

$$p_{x[n]}(\alpha) = \begin{cases} \frac{1}{b-a}, & \text{for } a \le \alpha \le b \\ 0, & \text{otherwise} \end{cases}$$

and the Gaussian density function is defined by

$$p_{x[n]}(\alpha) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-(\alpha - m_x)^2 / 2\sigma_x^2},$$

where the parameters m_x and σ_x are, respectively, the mean value and the standard deviation of $x[n]$ which lie in the range $-\infty < m_x < \infty$ and $\sigma_x > 0$. These density functions are illustrated in Fig. 1 below.

(a) (b)

Fig. 1. (a) Uniform and (b) Gaussian probability density functions.

Ex: Determine the mean and the variance of a uniformly distributed random variable.

$$m_x = \int_a^b \alpha \cdot p_{x[n]}(\alpha)\, d\alpha = \int_a^b \alpha \cdot \frac{1}{b-a}\, d\alpha$$

$$= \frac{1}{b-a} \left. \frac{\alpha^2}{2} \right|_a^b = \frac{1}{b-a} \frac{(b^2 - a^2)}{2} = \frac{b+a}{2}$$

$$E[x^2[n]] = \frac{1}{b-a} \int_a^b \alpha^2\, d\alpha = \frac{1}{b-a} \left. \frac{\alpha^3}{3} \right|_a^b = \frac{b^2 + a^2 + ab}{3}$$

$$\sigma_x^2 = E[x^2[n]] - (m_x)^2 = \frac{b^2 + a^2 + ab}{3} - \frac{b^2 + 2ab + a^2}{4} = \frac{(b-a)^2}{12}$$

When we deal with two random variables, say X and Y, it is also of interest to know their joint statistical properties as well as their individual statistical properties. The probability that $X(n)$ takes a value in a specified range from $-\infty$ to α and that $Y(n)$ takes a value in a specified range from $-\infty$ to β is given by their joint probability distribution function described by

$$P_{X(n)Y(n)}(\alpha, \beta) = \Pr\left[X(n) \leq \alpha,\ Y(n) \leq \beta\right]$$

or equivalently, by their joint probability density function

$$p_{X(n),Y(n)}(\alpha, \beta) = \frac{\partial^2 P_{X(n)Y(n)}(\alpha, \beta)}{\partial \alpha\, \partial \beta}$$

The joint probability distribution function is thus given by

$$P_{X(n)Y(n)}(\alpha, \beta) = \int_{-\infty}^{\alpha} \int_{-\infty}^{\beta} p_{X(n)Y(n)}(u, v)\, du\, dv$$

Properties of the joint probability density function:

$$p_{X(n)Y(n)}(\alpha, \beta) \geq 0$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_{X(n)Y(n)}(\alpha, \beta)\, d\alpha\, d\beta = 1$$

Properties of the joint probability distribution function:

$$0 \leq P_{X(n)Y(n)}(\alpha, \beta) \leq 1$$

$$P_{X(n)Y(n)}(\alpha_1, \beta_1) \leq P_{X(n)Y(n)}(\alpha_2, \beta_2) \quad \text{for} \quad \alpha_2 \geq \alpha_1 \text{ and } \beta_2 \geq \beta_1$$

$$P_{X(n)Y(n)}(-\infty, -\infty) = 0, \quad P_{X(n)Y(n)}(+\infty, +\infty) = 1$$

The joint statistical properties of two random variables $X(n)$ and $Y(n)$ are described by their cross-correlation and cross-covariance given by

$$r_{xy} = E\left[X(n)\, Y(n)\right] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \alpha\, \beta\, p_{X(n)Y(n)}(\alpha, \beta)\, d\alpha\, d\beta$$

$$c_{xy} = E\left[(X(n) - m_x)(Y(n) - m_y)\right] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (\alpha - m_x)(\beta - m_y)\, p_{X(n)Y(n)}(\alpha, \beta)\, d\alpha\, d\beta$$

$$= r_{xy} - m_x m_y$$

where m_x and m_y are the mean of the random variables $X(n)$ and $Y(n)$.

Two random variables $x[n]$ and $y[n]$ are said to be linearly independent or uncorrelated if

$$E[x[n]\,y[n]] = E[x[n]]\,E[y[n]]$$

and statistically independent if

$$P_{x[n],y[n]}(\alpha,\beta) = P_{x[n]}(\alpha)\cdot P_{y[n]}(\beta).$$

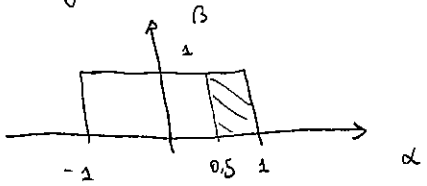
It can be shown that if two random variables are statistically independent then they are also uncorrelated. However, if these variables are uncorrelated they may not be statistically independent.

The statistical independence property makes it easier to compute the statistical properties of a random variable that is a function of several independent random variables

Ex: Consider the two random variables $x[n]$ and $y[n]$ described by a uniformly distributed joint probability density function given by

$$P_{x[n],y[n]}(\alpha,\beta) = \begin{cases} A, & -1 \le \alpha \le 1,\ 0 \le \beta \le 1, \\ 0, & \text{otherwise} \end{cases}$$

Determine the value of the constant A and then compute the probability that $x[n]$ and $y[n]$ lie in the range $0.5 \le \alpha \le 1$, $0 \le \beta \le 1$ shown by



From the properties of joint probability density functions, we have

$$A\int_{-1}^{1}\int_{0}^{1} d\alpha\, d\beta = A\left[\int_{-1}^{1} d\alpha\right]\left[\int_{0}^{1} d\beta\right] = 2A = 1$$

Hence, $\boxed{A = 1/2}$

$$Pr\left[0.5 \le x[n] \le 1,\ 0 \le y[n] \le 1\right] = \frac{1}{2}\int_{0}^{1}\int_{0.5}^{1} d\alpha\, d\beta = \frac{1}{4}.$$

Statistical properties of random signals

In general, the mean, mean-square value and variance of a random discrete-time signal are functions of the time index n and can be considered as sequences.

So far we have considered real-valued random variables and random signal. A generalization to complex-valued random variables and signals is straight forward. The nth sample of a complex-valued random signal $x[n]$ is described by

$$x[n] = x_r[n] + j\, x_i[n],$$

where $x_r[n]$ and $x_i[n]$ are real-valued sequences called the real and imaginary parts of $x[n]$, respectively.

The mean value of a complex random signal is given by

$$m_x = E[x[n]] = E[x_r[n]] + j\, E[x_i[n]]$$

$$= m_{x_r} + m_{x_i}$$

Likewise, the variance σ_x^2 of $x[n]$ is described by

$$\sigma_x^2 = E\left[|x[n] - m_x|^2\right] = E\left[|x[n]|^2\right] - (m_x)^2$$

The statistical relation of the samples of a random discrete-time signal at two different time indices m and n can be of great interest. One such relation is the autocorrelation, which for a complex random discrete-time signal $x[n]$ is given by

$$r_{xx}[m,n] = E\left[x[m]\, x^*[n]\right],$$

where $*$ denotes complex conjugation.

Another important relation is the autocovariance of $x[n]$ given by

$$c_{xx}[m,n] = E\left[(x[m] - m_{x_m})(x[n] - m_{x_n})^*\right]$$

$$= r_{xx}[m,n] - m_{x_m} m_{x_n}^*$$

The correlation between two different random discrete-time signals $x[n]$ and $y[n]$ is described by the cross-correlation function

$$r_{xy}[m,n] = E[x[m]y^*[n]]$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \alpha\beta^* p_{x[m],y[n]}(\alpha, m, \beta, n)\, d\alpha\, d\beta$$

and the cross-covariance function

$$c_{xy}[m,n] = E[(x[m]-m_{xm})(y[n]-m_{yn})^*]$$
$$= r_{xy}[m,n] - m_{xm}m_{yn}^*$$

where $p_{x[m],y[n]}(m,\alpha,n,\beta)$ is the joint probability density function of $x[n]$ and $y[n]$. Both the cross-correlation and the cross-covariance functions can also be considered as two-dimensional sequences. The two random signals $x[n]$ and $y[n]$ are uncorrelated if $c_{xy}[m,n]=0$ for all values of m and n.

Ex: Consider the random signal $X[n] = A\cos(\omega_0 n + \phi)$ and the prob. density functions of A and ϕ described by

$$p_A(\alpha) = \begin{cases} \frac{1}{4} & , 0 \le \alpha \le 4 \\ 0 & , \text{otherwise} \end{cases}$$

$$p_\phi(\phi) = \begin{cases} \frac{1}{2\pi} & , 0 \le \phi \le 2\pi \\ 0 & , \text{otherwise} \end{cases}$$

Since A and ϕ are statistically independent, their joint prob. density function is

$$p_{A\phi}(\alpha,\phi) = \begin{cases} \frac{1}{8\pi} & , 0\le\alpha\le 4,\ 0\le\phi\le 2\pi \\ 0 & , \text{otherwise} \end{cases}$$

The mean of $X[n]$ is

$$m_{xn} = \frac{1}{8\pi}\int_0^4\int_0^{2\pi} \alpha\cos(\omega_0 n+\phi)\, d\alpha\, d\phi$$
$$= \frac{1}{8\pi}\left(\int_0^4 \alpha\, d\alpha\right)\left(\int_0^{2\pi}\cos(\omega_0 n+\phi)\, d\phi\right) = 0$$

The mean-square value is

$$E[x^2[n]] = \frac{1}{8\pi}\int_0^4\int_0^{2\pi} \alpha^2\cos^2(\omega_0 n+\phi)\, d\alpha\, d\phi$$
$$= \frac{1}{8\pi}\left(\int_0^4 \alpha^2 d\alpha\right)\left(\int_0^{2\pi}\cos^2(\omega_0 n+\phi)\, d\phi\right) = \frac{8}{3}$$

The autocorrelation function is

$$r_{xx}[m,n] = E[X[m]X[n]]$$
$$= \frac{1}{8\pi}\int_0^4 \alpha^2 d\alpha \int_0^{2\pi}\cos(\omega_0 m+\phi)\cos(\omega_0 n+\phi)\, d\phi$$
$$= \frac{8}{3}\cos(\omega_0(m-n)).$$

Wide-Sense Stationary Random Signals:

A class of random signals often encountered in digital signal processing applications are the so-called wide-sense stationary (WSS) random processes for which some of the key statistical properties are all independent of time. Specifically, for a WSS random process $x[n]$, the mean $E[x[n]]$ has the same constant value m_x for all n, and the autocorrelation and the autocovariance functions depend only on the difference of l and n, i.e.,

$$m_x = E[x[n]], \quad \text{for all } n$$

$$r_{xx}[l] = r_{xx}[n+l, n] = E[x[n+l]x^*[n]], \quad \text{for all } n \text{ and } l,$$

$$c_{xx}[l] = c_{xx}[n+l, n] = E[(x[n+l]-m_x)(x[n]-m_x)^*]$$

$$= r_{xx}[l] - |m_x|^2, \quad \text{for all } n \text{ and } l.$$

The mean-square value of a WSS random process $x[n]$ is given by

$$E[|x[n]|^2] = r_{xx}[0],$$

and the variance is given by

$$\sigma_x^2 = c_{xx}[0] = r_{xx}[0] - |m_x|^2$$

The cross-correlation and cross-covariance functions between two WSS random processes $x[n]$ and $y[n]$ are given by

$$r_{xy}[l] = E[x[n+l]y^*[n]],$$

$$c_{xy}[l] = E[(x[n+l]-m_x)(y[n]-m_y)^*]$$

$$= r_{xy}[l] - m_x m_y^*.$$

Concept of Power in a Random Signal :

In order to compute the power associated with a random signal $x[n]$ we employ the following definition:

$$P_x = E\left[\lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2 \right]$$

In most practical cases, the expectation and the summation can be interchanged, resulting in a simpler expression given by

$$P_x = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} E\left[|x[n]|^2 \right]$$

In addition, if the random signal has a constant mean-square value, then the above expression reduces to

$$P_x = E\left[|x[n]|^2 \right] = r_{xx}[0] = \sigma_x^2 + |m_x|^2.$$

Ergodic Signals

In many practical situations, the random signal of interest cannot be described in terms of a simple analytical expression and some of its statistical properties must be estimated from a finite set of samples. Such an approach can lead to meaningful results if the ergodicity condition is satisfied. In particular, a stationary random signal is defined to be an ergodic signal if all its statistical properties can be estimated from a single realization of sufficiently large finite length.

For an ergodic signal, time averages equal ensemble averages via the expectation operator in the limit as the length of the realization goes to infinity. For example, for a complex-valued ergodic signal we can compute the mean value, variance and auto covariance as :

$$m_x = \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} x[n].$$

$$\sigma_x^2 = \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} (x[n]-m_x)(x[n]-m_x)^*$$

$$r_{xx}[k] = \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} (x[n]-m_x)(x[n+k]-m_x)^*$$

In practice, we replace the limit with a finite sum to provide an estimate of the desired statistical properties, as described by

$$\hat{m}_x = \frac{1}{M+1} \sum_{n=0}^{M} x[n],$$

$$\hat{\sigma}_x^2 = \frac{1}{M+1} \sum_{n=0}^{M} (x[n]-m_x)(x[n]-m_x)^*,$$

$$\hat{r}_{xx}[l] = \frac{1}{M+1} \sum_{n=0}^{M} (x[n]-m_x)(x[n+l]-m_x)^*.$$

Discrete-Time processing of random signals:

In many situations, we need to study the effect of processing a random discrete-time signal by an LTI discrete-time system. Specifically, we need to determine the statistical properties of the output signal $y[n]$ generated by a stable LTI system with an impulse response $h[n]$ when its input $x[n]$ is a realization of a WSS random process $X[n]$.

Consider the linear convolution of $x[n]$ and $h[n]$ as described by

$$y[n] = \sum_{k=-\infty}^{\infty} h^*[k] \cdot x[n-k]$$

Since the function of a random variable is also a random variable, it follows that the output $y[n]$ is also a sample sequence of an output random process $Y[n]$.

Since the input $x[n]$ is a sample sequence of $X[n]$, its mean m_x is a constant independent of the time index n. The mean $E[y[n]]$ of $Y[n]$ is given by

$$m_y = E[y[n]] = E\left[\sum_{k=-\infty}^{\infty} h^*[k] \cdot x[n-k]\right]$$

$$= \sum_{k=-\infty}^{\infty} h^*[k]\, E[x[n-k]] = m_x \sum_{k=-\infty}^{\infty} h^*[k] = m_x H(e^{j0})$$

which is a constant.

The autocorrelation function of $y[n]$ is given by

$$r_{yy}[n+\ell, n] = E[y[n+\ell]\, y^*[n]]$$

$$= E\left[\left(\sum_{i=-\infty}^{\infty} h^*[i]\, x[n+\ell-i]\right)\left(\sum_{k=-\infty}^{\infty} h^*[k]\cdot x[n-k]\right)^*\right]$$

$$= \sum_{i=-\infty}^{\infty} h^*[i] \sum_{k=-\infty}^{\infty} h[k]\, E[x[n+\ell-i]\, x^*[n-k]]$$

$$= \sum_{i=-\infty}^{\infty} h^*[i] \sum_{k=-\infty}^{\infty} h[k]\, r_{xx}[n+\ell-i, n-k]$$

which depends only on the difference $\ell+k-i$ of the time indices $n+\ell-i$ and $n-k$, i.e.,

$$r_{xx}[n+\ell-i, n-k] = r_{xx}[\ell+k-i].$$

Substituting the above into $r_{yy}[n+\ell, n]$, we arrive at

$$r_{yy}[n+\ell, n] = \sum_{i=-\infty}^{\infty} h^*[i] \sum_{k=-\infty}^{\infty} h[k]\, \phi_{xx}[\ell+k-i] = r_{yy}[\ell],$$

where the output auto correlation sequence depends on ℓ. Thus, the output $y[n]$ is a sample sequence of a WSS random process.

Substituting $m = i-k$ in the previous equation we arrive at

$$r_{yy}[\ell] = \sum_{m=-\infty}^{\infty} r_{xx}[\ell-m] \sum_{k=-\infty}^{\infty} h^*[k]\, h[m+k]$$

$$= \sum_{m=-\infty}^{\infty} r_{xx}[\ell-m]\, r_{hh}[m];$$

where $r_{hh}[m]$ is the autocorrelation sequence of $h[n]$.

The cross-correlation function between the output and the input sequences of the LTI sequences of the LTI system for a real-valued input is given by

$$r_{yx}[n+\ell, n] = E[y[n+\ell]\, x^*[n]]$$

$$= E\left[\sum_{i=-\infty}^{\infty} h^*[i]\, x[n+\ell-i]\, x^*[n]\right]$$

$$= \sum_{i=-\infty}^{\infty} h^*[i]\, E[x[n+\ell-i]\, x^*[n]]$$

$$= \sum_{i=-\infty}^{\infty} h^*[i]\, r_{xx}[\ell-i] = r_{yx}[\ell],$$

which indicates that the cross-correlation sequence depends on ℓ.

B. Statistical Models

In statistical signal processing we employ models to represent or describe systems or physical phenomena, such systems are often excited by random signals and can be modelled by digital filters, which were described in the previous chapters. In general, we employ the following types of linear models:

i) Autoregressive (AR) models

ii) Moving average (MA) models

iii) Auto-regressive moving average (ARMA) models

which can be depicted by the figure below.

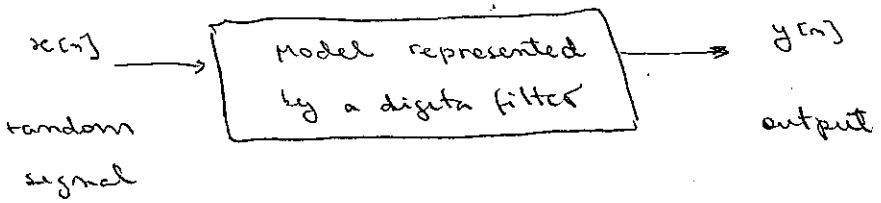


Fig. 2. Statistical model

Let us first consider an ARMA model which has an input x[n] that corresponds to white noise and a rational transfer function given by

$$H[z] = \frac{B[z]}{A[z]} = \frac{\sum_{k=0}^{q} b[k]\, z^{-k}}{1 + \sum_{k=1}^{p} a[k]\, z^{-k}}$$

where H[z] has p poles and q zeros, b[k] and a[k] are the parameters of the filter and it is assumed that the output of the filter y[n] is WSS with $r_{xx} = \sigma_x^2$. This filter corresponds to an IIR digital filter.

Since $x[n]$ and $y[n]$ are related by a difference equation given by

$$y[n] + \sum_{l=1}^{p} a[l]\, y[n-l] = \sum_{l=0}^{q} b[l]\, x[n-l]$$

it follows that the autocorrelation of $x[n]$ and the cross-correlation between $x[n]$ and $y[n]$ satisfy the same difference equation. Therefore, if we multiply both sides of the above equation by $y^*[n-k]$ and take the expected value, we have

$$r_{yy}[k] + \sum_{l=1}^{p} a[l]\, r_{yy}[k-l] = \sum_{l=0}^{q} b[l]\, \underbrace{E\left[x[n-l]\, y^*[n-k]\right]}_{r_{xy}[k-l]}$$

Therefore, the above eq. becomes

$$r_{yy}[k] + \sum_{l=1}^{p} a[l]\, r_{yy}[k-l] = \sum_{l=0}^{q} b[l]\, r_{xy}[k-l]$$

By writing $r_{xy}[k]$ in terms of $r_{yy}[k]$ and impulse response of the IIR filter $y[n] = h[n] * x[n] = \sum_{m=-\infty}^{\infty} h[m]\, x[n-m]$ the cross-correlation $r_{xy}[k-l]$ may be written as

$$E\left[x[n-l]\, y^*[n-k]\right] = E\left[\sum_{m=-\infty}^{\infty} x[n-l]\, h^*[m]\, x[n-k-m]\right]$$

$$= \sigma_x^2\, h^*[l-k],$$

where the last equality follows from $E[x[n-l]\, x^*[m]] = \sigma_x^2\, \delta[n-l-m]$, i.e., $x[n]$ is white noise. Substituting the above eq into the previous one, we obtain

$$r_{yy}[k] + \sum_{l=1}^{p} a[l]\, r_{yy}[k-l] = \sigma_x^2 \sum_{l=0}^{q} b[l]\, h^*[l-k]$$

Assuming that $h[n]$ is causal, $\sigma_x^2 \sum_{l=0}^{q} b[k]\, h^*[l-k] = 0$ for $k > q$, we may write the above eq. as.

$$r_{yy}[k] + \sum_{l=1}^{p} a[l]\, r_{yy}[k-l] = \begin{cases} \sigma_x^2 \sum_{l=0}^{q} b[l]\, h^*[l-k], & 0 \le k \le q \\ 0, & k > q \end{cases}$$

which are the Yule-Walker equations

Writing the Yule-Walker equations for $k = 0, 1, \ldots, p+q$ in matrix form, we have

$$\begin{bmatrix} r_{yy}[0] & r_{yy}[-1] & \cdots & r_{yy}[-p] \\ r_{yy}[1] & r_{yy}[0] & \cdots & r_{yy}[-p+1] \\ \vdots & \vdots & & \vdots \\ r_{yy}[q] & r_{yy}[q-1] & \cdots & r_{yy}[q-p] \\ r_{yy}[q+1] & r_{yy}[q] & \cdots & r_{yy}[q-p+1] \\ \vdots & \vdots & & \vdots \\ r_{yy}[q+p] & r_{yy}[q+p-1] & \cdots & r_{yy}[q] \end{bmatrix} \begin{bmatrix} 1 \\ a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = \sigma_x^2 \begin{bmatrix} \sum_{l=0}^{q} b[l]\, h^*[l] \\ \sum_{l=0}^{q} b[l]\, h^*[l-1] \\ \sum_{l=0}^{q} b[l]\, h^*[l-q] \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

or

$$\underline{R}\ \underline{a} = \underline{r}$$

The above eq. defines a recursion for $r_{yy}[k]$ in terms of the filter coefficients $a[k]$ and $b[k]$. Therefore, the Yule-Walker equations may be used to compute $r_{yy}[k]$ from a finite set of $r_{yy}[k]$. For example, if $p \geq q$ and if $r_{yy}[0], \ldots, r_{yy}[p-1]$ are known, then $r_{yy}[k]$ for $k \geq p$ may be computed as follows

$$r_{yy}[k] = -\sum_{l=1}^{p} a[l] \cdot r_{yy}[k-l]$$

The Yule-Walker equations may also be used to estimate the filter coefficients $a[k]$ and $b[l]$ from $r_{yy}[k]$. However, due to the product $h^*[l]\, b[k+l]$, the Yule-Walker equations are nonlinear in the filter coefficients and solving them for the filter coefficients is, in general, difficult.

The Yule-Walker equations are important in problems such as signal modelling and spectrum estimation. In what follows, we look at AR and MA models which are special cases of ARMA models.

In the AR model, $y[n]$ is generated by filtering $x[n]$ with an all-pole FIR filter of the form

$$H[z] = \frac{b[0]}{1 + \sum_{k=1}^{p} a[k] \cdot z^{k}}$$

The Yule-Walker equations for an AR model can be found by setting $q=0$, which results in

$$r_{yy}[k] + \sum_{l=1}^{p} a[l]\, r_{yy}[k-l] = \sigma_x^2 \, |b[0]|^2 \, \delta[k], \quad k \geq 0$$

which in matrix form yields

$$\begin{bmatrix} r_{yy}[0] & r_{yy}[-1] & \cdots & r_{yy}[-p] \\ r_{yy}[1] & r_{yy}[0] & \cdots & r_{yy}[-p+1] \\ \vdots & \vdots & \ddots & \\ r_{yy}[p] & r_{yy}[p-1] & & r_{yy}[0] \end{bmatrix} \begin{bmatrix} 1 \\ a[1] \\ \vdots \\ a[p] \end{bmatrix} = \sigma_x^2 \, |b[0]|^2 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Note that since the Yule-Walker equations are linear in $a[k]$, it is simple to find $a[k]$ from $r_{yy}[k]$

In the MA model, $y[n]$ is obtained by filtering $x[n]$ with an FIR filter that has a transfer function given by

$$H[z] = \sum_{k=0}^{q} b[k]\, z^{-k}$$

The Yule-Walker equations for an MA model can be found by setting $a[k] = 0$ and noting that $h[n] = b[n]$, which yields

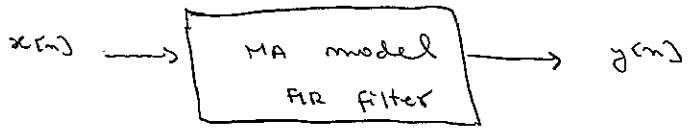
$$r_{yy}[k] = \sigma_x^2 \sum_{l=0}^{q-|k|} b[l+|k|]\, b^{*}[l]$$

Therefore, unlike the AR model the computation of an MA model to describe / process random signal is a difficult problem.

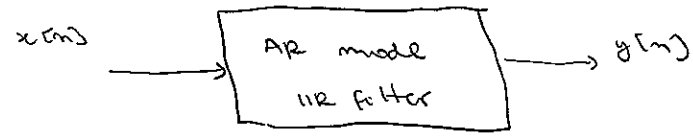
The use of AR models is widely advocated because of a result known as Wold's decomposition theorem. Such result states that a general linear model can be represented by an MA model which is equivalent to an AR model, i.e., whose coefficients obey the following relation

$$b[k] = \overset{*}{a}[k].$$

The basic difference between the MA and AR models is that $b[k]$ operates on the input $x[n]$ of the MA model, whereas $a[k]$ operates on the output of the AR model, as illustrated below.



(a)



(b)

Fig. 3 System models.

C. The Wiener filter

In this section, we study an optimum filter called the Wiener filter that can be used to produce an optimum estimate of a signal from a noisy observation given by

$$x[n] = d[n] + w[n],$$

where $x[n]$ is a random signal that is observed at the input of a discrete-time LTI system, $d[n]$ is a desired signal and $w[n]$ represents noise. Both $d[n]$ and $w[n]$ are random signals taken from wss random processes.

The problem we are interested in solving consists of designing an FIR filter $h[n]$ with order N that can result in the minimum mean-square error (MMSE) estimate of $d[n]$ by observing $x[n]$, as illustrated in the figure below.

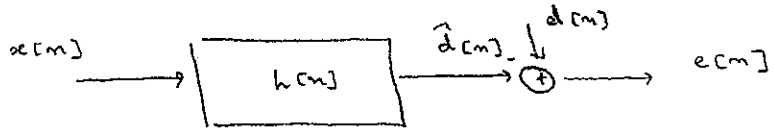


Fig. 4 The MMSE filtering problem.

The output $\hat{d}[n]$ of the filter $h[n]$ is described by

$$\hat{d}[n] = \sum_{k=0}^{N} h^*[k]\, x[n-k] = \underline{h}^H \cdot \underline{x}[n],$$

where the filter $h[n]$ and the input signal $x[n]$ can be represented using matrix notation: $\underline{h} = [h[0]\ h[1] \dots h[N]]^T$

$$\underline{x}[n] = [x[n]\ x[n-1] \dots x[n-N]]^T$$

The design problem corresponds to the following optimisation :

$$\underline{h}_o = \arg \min C(\underline{h}) = E\left[|e[n]|^2 \right]$$

$$= E\left[| d[n] - \underbrace{\underline{h}^H x[n]}_{\hat{d}[n]} |^2 \right]$$

In order to compute the filter coefficients, we need to solve the minimisation of $C(\underline{h})$ by computing the derivative of $C(\underline{h})$ with respect to $\underline{h}^*$ or $h^*[k]$ and equating the result to zero as follows :

$$\frac{\partial C(\underline{h})}{\partial h^*[k]} = \frac{\partial}{\partial h^*[k]} E\left[e[n]\, e^*[n] \right]$$

$$= E\left[e[n] \frac{\partial}{\partial h^*[k]} e^*[n] \right] = 0 \quad , \quad k = 0, \dots, N$$

where $e[n] = d[n] - \sum_{l=0}^{N} h^*[l]\, x[n-l]$ and $\frac{\partial}{\partial h^*[k]} e[n] = -x[n-k]$, leading to the so-called orthogonality principle :

$$E\left[e^*[n]\, x[n-k] \right] = 0 \quad , \quad k = 0, 1, \dots, N$$

If we substitute the error $e[n]$ into the orthogonality principle, we can obtain the parameters for the wiener filter as given by

$$E\left[\left(d[n] - \sum_{l=0}^{N} h^*[l]\, x[n-l] \right)^* x[n-k] \right] = 0$$

$$\underbrace{E\left[d^*[n]\, x[n-k] \right]}_{r_{dx}[k]} - \sum_{l=0}^{N} h_o[l] \underbrace{E\left[x[n-k]\, x^*[n-l] \right]}_{r_{xx}[k-l]} = 0$$

$$r_{dx}[k] - \sum_{l=0}^{N} h_o[l]\, r_{xx}[k-l] = 0 \quad , \quad k = 0, 1, \dots, N,$$

where $r_{dx}[k]$ is the cross-correlation between $d[n]$ and $x[n-k]$ and $r_{xx}[k-l]$ is the autocorrelation of $x[n]$.

The Wiener filter $h[n]$ can also be employed and derived in matrix form. By substituting the previous expressions for different values of k, we obtain

$$\sum_{l=0}^{N} h_o[l]\, r_{xx}[k-l] = r_{dx}[k], \quad k = 0, 1, \dots, N$$

$$\underbrace{\begin{bmatrix} r_{xx}[0] & r_{xx}[1] & \cdots & r_{xx}[N] \\ r_{xx}[1] & r_{xx}[0] & \cdots & r_{xx}[N-1] \\ \vdots & \vdots & \ddots & \\ r_{xx}[N] & r_{xx}[N] & & r_{xx}[0] \end{bmatrix}}_{\underline{\underline{R}}_{xx}} \underbrace{\begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[N] \end{bmatrix}}_{\underline{h}_o} = \underbrace{\begin{bmatrix} r_{dx}[0] \\ r_{dx}[1] \\ \vdots \\ r_{dx}[N] \end{bmatrix}}_{\underline{r}_{dx}},$$

where $\underline{\underline{R}}_{xx}$ is the autocorrelation matrix, $\underline{r}_{dx}$ is the cross-correlation vector and $\underline{h}_o$ is the Wiener filter.

The derivation of the Wiener filter can also be performed using matrix notation, which is more compact for obtaining the expression $\underline{\underline{R}}_{xx} \cdot \underline{h}_o = \underline{r}_{dx}$. Let us consider the optimisation problem given by

$$\underline{h}_o = \arg\min \; C(\underline{h}) = E\left[|d[i] - \underline{h}^H \underline{x}[n]|^2 \right]$$

This problem corresponds to computing the point of minimum of $C(\underline{h})$. In other words, it corresponds to calculating the derivative of $C(\underline{h})$ with respect to $\underline{h}$, equating the terms to zero and solving the resulting equations. To this end, it is convenient to define the partial derivatives of $C(h)$ with respect to $\underline{h}^*$ using Wirtinger calculus:

$$\frac{\partial \underline{h}}{\partial \underline{h}^*} = 0 \qquad \text{and} \qquad \frac{\partial \underline{h}^*}{\partial \underline{h}^*} = \underline{\underline{I}}$$

$$\frac{\partial\, C(\underline{h})}{\partial\, \underline{h}^*} = \frac{\partial}{\partial\, \underline{h}^*} E\left[\, |\, e[n]\,|^2\right] = \underline{0}$$

$$= \frac{\partial}{\partial\, \underline{h}^*} E\left[\, |\, d[n] - \underline{h}^H x[n]\,|^2\right] = \underline{0}$$

$$= \frac{\partial}{\partial\, \underline{h}^*} E\left[\, (d[n] - \underline{h}^H x[n])\,(d[n] - \underline{h}^H x[n])^*\right] = \underline{0}$$

$$= E\left[\, -\underline{x}[n]\,(d[n] - \underline{h}^H x[n])^*\right] = \underline{0}$$

$$\underbrace{E\left[-x[n]\cdot d^*[n]\right]}_{-\underline{r}_{dx}} + \underbrace{E\left[\underline{x}[n]\,\underline{x}^H[n]\right]}_{\underline{R}_{xx}}\,\underline{h}_o = \underline{0}$$

$$\underline{R}_{xx}\,\underline{h}_o = \underline{r}_{dx} \;\rightarrow\; \boxed{\underline{h}_o = \underline{R}_{xx}^{-1}\,\underline{r}_{dx}}$$

The minimum mean-square error (MMSE) can be evaluated by substituting the Wiener filter $\underline{h}_o$ into the cost function as follows:

$$\text{MMSE} = C(\underline{h}_o) = E\left[\, |\, e[n]\,|^2\right] = E\left[\, e[n]\,(d[n] - \sum_{l=0}^{N-1} \underline{h}^*[l]\, x[n-l])^*\right]$$

$$= E\left[\, e[n]\, d^*[n]\right] - \sum_{l=0}^{N} h_o[l]\, E\left[\, e[n]\, x^*[n-l]\right]$$

($e[n]$ is orthogonal to $x^*[n-l]$ — orthogonality principle.)

Taking the expected values, we obtain

$$\text{MMSE} = E\left[\, d[n]\, d^*[n]\right] - \sum_{l=0}^{N} h_o^*[l]\, r_{dx}[l]$$

$$= \sigma_d^2 - \underline{h}_o^H \cdot \underline{r}_{dx}$$

$$= \sigma_d^2 - \underline{r}_{dx}^H\, \underline{R}^{-1}\, \underline{r}_{dx},$$

where the MMSE describes the accuracy of the Wiener filter to estimate $d[n]$.

Ex: Let $d[n]$ be an AR process with an autocorrelation sequence given by

$$r_{dx}[k] = \alpha^{|k|}$$

with $0 < \alpha < 1$ and suppose that $d[n]$ is observed in the presence of uncorrelated white noise that has a variance of σ_n^2 in the form:

$$x[n] = d[n] + w[n]$$

a) Design a Wiener filter to reduce the noise in $x[n]$

With $H[z] = \sum_{l=0}^{N} h[l] z^{-l}$ and $N=1$, we have

$$H[z] = h_0 + h_1 z^{-1}$$

The Wiener-Hopf equations are

$$\begin{bmatrix} r_{xx}[0] & r_{xx}[1] \\ r_{xx}[1] & r_{xx}[0] \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} = \begin{bmatrix} r_{dx}[0] \\ r_{dx}[1] \end{bmatrix}$$

Since $d[n]$ and $w[n]$ are assumed uncorrelated, we have

$$r_{dx}[k] = \alpha^{|k|}$$

$$r_{xx}[k] = r_{dx}[k] + \underbrace{E[w[n]w^*[n-k]]}_{\sigma_w^2 = r_{ww}[k]}$$

$$= r_{dx}[k] + \sigma_w^2$$

The Wiener-Hopf equations become

$$\begin{bmatrix} 1+\sigma_w^2 & \alpha \\ \alpha & 1+\sigma_w^2 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \alpha \end{bmatrix}$$

Solving for h_0 and h_1, we have

$$\begin{bmatrix} h_0 \\ h_1 \end{bmatrix} = \frac{1}{(1+\sigma_w^2)\cdot\alpha^2} \begin{bmatrix} 1+\sigma_w^2 & -\alpha \\ -\alpha & 1+\sigma_w^2 \end{bmatrix} \begin{bmatrix} 1 \\ \alpha \end{bmatrix}$$

$$= \frac{1}{(1+\sigma_w^2)-\alpha^2} \begin{bmatrix} 1+\sigma_w^2 & -\alpha^2 \\ \alpha\sigma_w^2 & \end{bmatrix}$$

The Wiener filter is given by

$$H[z] = h_0 + h_1 z^{-1} = \frac{1}{(1+\sigma_w^2)^2 - \alpha^2}\left((1+\sigma_w^2 - \alpha^2) + \alpha\sigma_w^2 z^{-1} \right),$$

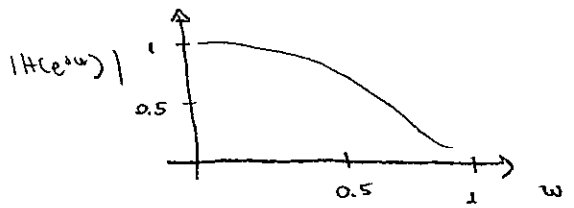
which has a zero at $z = \frac{-\alpha\,\sigma_w^2}{1+\sigma_w^2-\alpha^2}$

b) Compute the power spectrum $H(e^{j\omega})$ for $\alpha = 0.8$ and $\sigma_w^2 = 1$

In this case, the Wiener filter becomes

$$H(e^{j\omega}) = 0.4048 + 0.2381\, e^{-j\omega},$$

which is a lowpass filter with magnitude response.

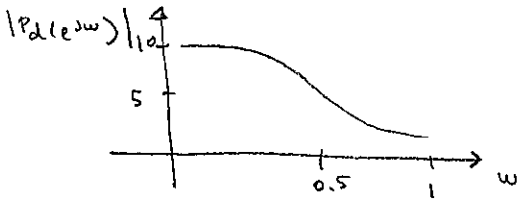


The fact that $H(e^{j\omega})$ is a lowpass filter is not a surprise. The power spectrum of $d[n]$ is

$$P_d(e^{j\omega}) = \frac{1-\alpha^2}{(1+\alpha^2) - 2\alpha\cos\omega}$$

which for $\alpha = 0.8$ becomes

$$P_d(e^{j\omega}) = \frac{0.36}{1.64 - 1.6\cdot\cos\omega}$$



Since $P_d(e^{j\omega})$ decreases with ω and the power spectrum of the noise is constant for all ω, then the signal-to-noise ratio decreases with the increase of ω.

c) Compute the MMSE for $\alpha = 0.8$ and $\sigma_w^2 = 1$

$$\text{MMSE} = E[|e[n]|^2] = \sigma_d^2 - \underline{h}^H \cdot \underline{r}_{dx}$$

$$= r_{dx}^2[0] - h_0^* r_{dx}[0] - h_1^* r_{dx}[1]$$

$$= \sigma_w^2 \cdot \frac{1+\sigma_w^2-\alpha^2}{(1+\sigma_w^2)^2-\alpha^2} = 0.4048$$

d) Evaluate the increase in the signal-to-noise ratio (SNR) obtained by the Wiener filter for $\alpha = 0.8$ and $\sigma_w^2 = 1$.

Prior to filtering, we have

$$SNR = \frac{E[|d[n]|^2]}{E[|w[n]|^2]} = \frac{\sigma_d^2}{\sigma_w^2} = 1$$

$$SNR_{dB} = 10 \log_{10} 1 = 0 \text{ dB}$$

After filtering, we have

$$r_{xx}[k] = r_{dx}[k] + \underline{r}_w[k]$$

The signal power is

$$E[|d'[n]|^2] = \underline{h}^H \cdot \underline{R}_{xx} \underline{h} = [h_0^* \; h_1^*] \begin{bmatrix} 1 & \alpha \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} = 0.3748$$

The noise power is

$$E[|w'[n]|^2] = \underline{h}^H \cdot \underline{R}_w \underline{h} = [h_0^* \; h_1^*] \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} = 0.2206$$

The SNR at the output of the Wiener filter is

$$SNR = \frac{E[|d'[n]|^2]}{E[|w'[n]|^2]}$$

$$SNR_{dB} = 10 \log_{10} \frac{0.3748}{0.2206} = 2.302 \text{ dB}$$

Therefore, the Wiener filter increased the SNR by more than 2 dB.

D. Linear Prediction

Linear prediction is an important signal processing task which is encountered in many applications including communications, time-series analysis, control and biomedical engineering.

The basic problem is to design an FIR filter $h[n]$ to predict a desired signal $d[n+1]$ by observing $x[n] = d[n] + w[n]$. The input signal $x[n]$ contains a noisy version of the desired signal $d[n]$ in a previous time instant, as depicted in the figure below.

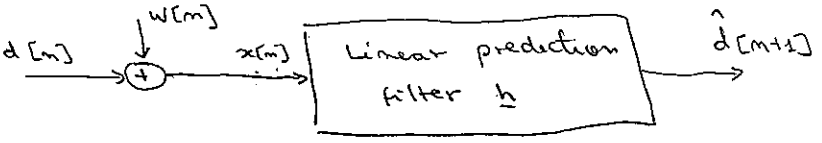


Fig. 5. Linear prediction problem.

Specifically, the linear prediction filter $h[n]$ performs linear prediction of $d[n]$ one step ahead. To this end, we consider the observed signal given by

$$x[n] = d[n] + w[n],$$

where both $d[n]$ and $w[n]$ are taken from WSS random processes.

The linear prediction filter design consists in designing an FIR filter $\underline{h} = [h[0]\ h[1] \ldots h[N]]^T$ which will predict / estimate $d[n+1]$ based on the observation $\underline{x}[n] = [x[n] \ldots x[n-N]]^T$. This is equivalent to the following linear combination:

$$\hat{d}[n+1] = \sum_{k=0}^{N} h^*[k]\, x[n-k],$$

$$= \underline{h}^H \cdot \underline{x}[n]$$

The design of linear prediction filter $\underline{h}$ requires the solution of the following optimisation problem

$$\underline{h}_o = \arg\min C_p(\underline{h}) = E\Big[\underbrace{|d[n+1] - \hat{d}[n+1]|^2}_{e[n]} \Big]$$

$$= E\Big[|d[n+1] - \sum_{k=0}^{N} h^*[k]\, x[n-k]|^2 \Big]$$

In order to compute the parameters of the linear prediction filter $\underline{h}$, we differentiate $C_p(\underline{h})$ with respect to $\underline{h}^*$, equate the terms to a null vector and solve the resulting equations:

$$\frac{\partial C_p(\underline{h})}{\partial \underline{h}^*} = \frac{\partial}{\partial \underline{h}^*} E\Big[\left(d[n+1] - \underline{h}^H \underline{x}[n]\right)\left(d[n+1] - \underline{h}^H \underline{x}[n]\right)^* \Big] = \underline{0}$$

$$= E\Big[-\underline{x}[n] \left(d[n+1] - \underline{h}^H \underline{x}[n] \right)^* \Big] = \underline{0}$$

$$= -\underbrace{E\Big[\underline{x}[n]\, d^*[n+1] \Big]}_{\underline{r}_{dx}} + \underbrace{E\Big[\underline{x}[n]\, \underline{x}^H[n] \Big]}_{\underline{R}_{xx}} \underline{h}_o = \underline{0}$$

$$\underline{R}_{xx}\, \underline{h}_o = \underline{r}_{dx} \Rightarrow \boxed{\underline{h}_o = \underline{R}_{xx}^{-1}\, \underline{r}_{dx}},$$

where $\underline{R}_{xx}$ is the autocorrelation matrix, $\underline{r}_{dx}$ is the cross-correlation vector and $\underline{h}_o$ is the optimal linear predictor. The linear prediction problem is similar to the Wiener filter problem and the main difference lies in the task performed by the filter.

The MMSE associated with the optimal linear predictor h_o is

$$\text{MMSE} = C_p(\underline{h}_o) = E\Big[|d[n+1] - \hat{d}[n+1]|^2 \Big]$$

$$= E\Big[|d[n+1] - \sum_{k=0}^{N} h^*[k]\, x[n-k]|^2 \Big]$$

$$= E\Big[e[n]\, d^*[n+1] \Big] - \sum_{k=0}^{N} h^*[k]\, E\Big[e[n]\, x^*[n-k] \Big]$$

($e[n]$ is orthogonal to $x[n-k]$, so this term $\to 0$)

$$= E\Big[|d[n+1]|^2 \Big] - \sum_{k=0}^{N} h^*[k]\, \underbrace{E\Big[x[n-k]\, d^*[n+1] \Big]}_{r_{dx}[k]}$$

$$= \sigma_d^2 - \underline{h}_o \cdot \underline{r}_{dx},$$

where $r_{dx}[\cdot, k] = E[x[n-k]\, d^*[n+1]]$ is the cross-correlation function which is equivalent to $r_{xx}[k+1]$.

Ex : Consider a linear prediction problem in which the observation is given by

$$x[n] = d[n] + w[n],$$

where $w[n]$ is a zero mean ($E[w[n]]=0$) white noise with a variance of σ_w^2, $d[n]$ and $w[n]$ are uncorrelated and $r_{dx}[k] = \alpha^{|k|}$.

$r_{ww}[0]$

a) Compute $r_{xx}[k]$ and $r_{dx}[k]$

$$r_{xx}[k] = E[x[n]\, x^*[n-k]] = r_{dx}[k] + r_{ww}[k] =$$

$$r_{dx}[k] = E[d[n+1]\, x[n+k]] = \alpha^{|k|}$$

$$r_{xx}[0] = 1 + \sigma_w^2, \quad r_{xx}[1] = \alpha$$

b) Compute the linear prediction filter

Since we know $r_{xx}[k]$ and $r_{dx}[k]$, we have

$$\underline{R}_{xx}\, \underline{h}_o = \underline{r}_{dx} \Rightarrow \begin{bmatrix} 1+\sigma_w^2 & \alpha \\ \alpha & 1+\sigma_w^2 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \alpha^2 \end{bmatrix}$$

Solving for $\underline{h}_o$, we obtain

$$\underline{h}_o = \frac{\alpha}{(1+\sigma_w^2)-\alpha^2} \begin{bmatrix} 1+\sigma_w^2 - \alpha^2 \\ \alpha\sigma_w^2 \end{bmatrix}$$

E. Adaptive Algorithms

In the previous sections, we described the Wiener filter and the linear prediction filter. A key assumption was that the input signal $x[n]$ is WSS. In this case, we have shown how to design time-invariant FIR filters for filtering, estimation and prediction tasks., which result in an MMSE estimate of $d[n]$.

However, in many practical situations the signals that are processed are nonstationary and their statistics change with time. In order to deal with these signals, we may consider adaptive algorithms.

Let us consider the Wiener filtering problem and the input signal given by

$$x[n] = d[n] + w[n],$$

where $d[n]$ is nonstationary.

In this case, the wiener filter is time-varying and produces the following estimate

$$\hat{d}[n] = \underline{h}^H[n] \cdot \underline{x}[n]$$

In order to design the time-varying Wiener filter, we must solve the following optimisation problem

$$\min C_t(\underline{h}) = E\left[|e[n]|^2\right] = E\left[|d[n] - \underline{h}^H[n]\,\underline{x}[n]|^2\right]$$

The solution to this minimisation problem can be found by setting the derivative of $C_t(\underline{h})$ with respect to $\underline{h}^*[n]$ equal to zero.

This results in the following

$$E\left[e^*[n]\, x[n-k]\right] = 0, \qquad k = 0, 1, \ldots, N-1$$

Substituting the error $e[n] = d[n] - \underline{h}^H[n]\,\underline{x}[n]$ into the previous expression, we have

$$E\left[\left(d[n] - \sum_{\ell=0}^{N} h_\ell^*[n]\, x[n-\ell]\right)^* x[n-k]\right] = 0 \quad , \quad k = 0, 1, \ldots, N-1$$

which becomes

$$\sum_{\ell=0}^{N-1} h_\ell^*[n]\, E[x[n-\ell]\, x^*[n-k]] = E\left[d^*[n]\, x[n-k]\right] \quad , \quad k = 0, 1, \ldots, N-1$$

The solution to the above equation is the Wiener filter for nonstationary processes given by

$$\underline{h}[n] = \underline{R}_{xx}^{-1}[n]\; \underline{r}_{dx}[n] \,,$$

where

$$\underline{R}_{xx}[n] = \begin{bmatrix} E[x[n]x^*[n]] & E[x[n-1]x^*[n]] & \cdots & E[x[n-N]x^*[n]] \\ E[x[n]x^*[n-1]] & E[x[n-1]x^*[n-1]] & \cdots & E[x[n-N]x^*[n-1]] \\ \vdots & \vdots & & \vdots \\ E[x[n]x^*[n-N]] & E[x[n-1]x^*[n-N]] & & E[x[n-N]x^*[n-N]] \end{bmatrix}$$

$$\underline{r}_{dx}[n] = \begin{bmatrix} E[d^*[n]\,x[n]] \\ E[d^*[n]\,x[n-1]] \\ \vdots \\ E[d^*[n]\,x[n-N]] \end{bmatrix}$$

are the time-varying autocorrelation matrix and the cross-correlation vector, respectively.

The design of a time-varying filter is more difficult than a time-invariant one because the optimum set of coefficients must be computed every time instant n. This problem can be considerably simplified if we consider an approach that updates the coefficients in the form

$$\underline{h}[n+1] = \underline{h}[n] + \Delta\underline{h}[n],$$

where $\Delta\underline{h}[n]$ is a correction that is applied at time n to obtain the new set of coefficients at time $n+1$, as illustrated by the figure shown next.

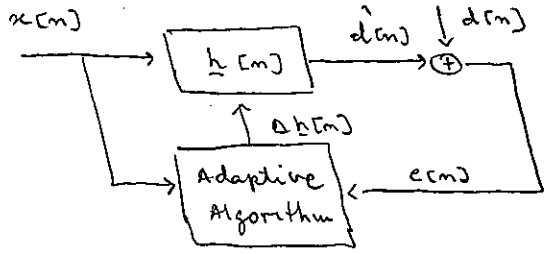


Fig. 6. Block diagram of an adaptive algorithm.

The fundamental approach that allows one to continuously update a set of parameters is at the heart of all adaptive signal processing algorithms. Each algorithm has its own way of obtaining the correction term $\Delta \underline{h}$. Moreover, the adaptive algorithm should have the following properties:

i) In a stationary environment, the algorithm should produce a sequence of corrections $\Delta \underline{h}$ in such a way that it converges to the Wiener solution, i.e.,

$$\lim_{n \to \infty} \underline{h}[n] = \underline{R}_{xx}^{-1} \underline{r}_{dx}$$

ii) The algorithm does not have to know the statistical quantities to compute $\Delta \underline{h}[n]$ as the algorithm should be able to compute these statistics with a built-in mechanism.

iii) For nonstationary signals, the filter should be able to adapt to the changing statistics and track the solution over time.

In what follows, we will describe the least-mean square (LMS) algorithm briefly along with a detailed derivation of it, conditions for its convergence and its requirements in terms of computational complexity.

The LMS algorithm was invented in 1960 by Widrow and Hopf and has become the most popular adaptive algorithm due to its simplicity, robustness and versatility. The LMS algorithm belongs to the class of stochastic gradient algorithms and employs instantaneous values of the gradient of $C_t(\underline{h}[n])$ with respect to $\underline{h}[n]$ in its adaptation.

In the derivation of the LMS algorithm, we consider the random input signal $x[n] = d[n] + w[n]$ and the filter $\underline{h}[n]$ with N+1 coefficients. The LMS solves the following optimisation problem

$$\min \quad C_t(\underline{h}[n]) = E\left[\underbrace{|d[n] - \underline{h}^H[n]\,\underline{x}[n]|}_{e[n]}{}^2 \right]$$

By using a method known in optimisation as steepest descent and the gradient of $C_t(\underline{h}[n])$ with respect to $\underline{h}^*[n]$, we can search for the minimum of the cost function $C_t(\underline{h}[n])$ through the following recursion:

$$\begin{aligned}
\underline{h}[n+1] &= \underline{h}[n] - \mu \underset{\underline{h}^*[n]}{\nabla} C_t(\underline{h}[n]) \\
&= \underline{h}[n] - \mu \left(\frac{\partial}{\partial \underline{h}^*[n]} E\left[(d[n] - \underline{h}^H[n]\underline{x}[n]) \underbrace{(d[n] - \underline{h}^H[n]\underline{x}[n])}_{e[n]}{}^* \right] \right) \\
&= \underline{h}[n] + \mu E\left[-\underline{x}[n] d^*[n] + \underline{x}[n]\underline{x}^H[n] \cdot \underline{h}[n] \right] \\
&= \underline{h}[n] + \mu \left(\underline{r}_{d\underline{x}} - \underline{R}_{\underline{x}\underline{x}}[n]\,\underline{h}[n] \right) \\
&= \underline{h}[n] + \mu E\left[e^*[n]\,\underline{x}[n] \right],
\end{aligned}$$

where μ is the step size.

By employing instantaneous values of the gradient in the above development, we arrive at the LMS algorithm.

$$\begin{aligned}
\underline{h}[n+1] &= \underline{h}[n] - \mu \underset{\underline{h}^*[n]}{\hat{\nabla}} C_t(\underline{h}[n]) \\
&= \underline{h}[n] + \mu\, e^*[n] \cdot \underline{x}[n],
\end{aligned}$$

where $e[n] = d[n] - \underline{h}^H[n]\,\underline{x}[n]$ is the error signal.

The LMS algorithm has three basic steps, namely:

- filtering : $\hat{d}[n] = \underline{h}^H[n]\,\underline{x}[n]$
- Estimation error : $e[n] = d[n] - \underline{h}^H[n]\,\underline{x}[n]$
- Adaptation : $\underline{h}[n+1] = \underline{h}[n] + \mu\, e^*[n].\,\underline{x}[n]$

These steps are illustrated in the block diagram below and in a table.

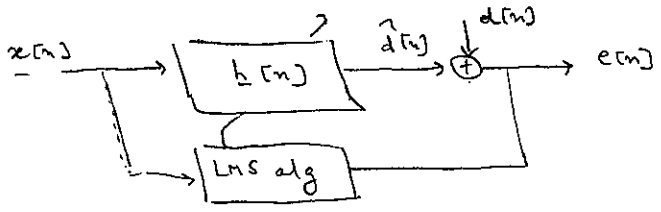


Fig. 7 Block diagram of the LMS algorithm

Table : The LMS algorithm

Initialisation:

- N - filter length
- μ - step size
- $\underline{h}[0] = \underline{0}$ - initial value of the filter

Computations:

for $n = 1, 2, \ldots$

$$\hat{d}[n] = \underline{h}^H[n]\,\underline{x}[n]$$

$$e[n] = d[n] - \hat{d}[n] = \underline{d}[n] - \underline{h}^H[n]\,\underline{x}[n]$$

$$\underline{h}[n+1] = \underline{h}[n] - \mu\, e^*[n]\,\underline{x}[n]$$

In terms of computational complexity, the LMS algorithm has a cost that is linear in the number of filter coefficients, i.e., $O(N)$. It requires $2N+1$ complex multiplications and $2N$ complex additions, as shown below.

Table : Computational complexity of the LMS algorithm

Task	Multiplications	Additions
$\hat{d}[n] = \underline{h}^H[n]\,\underline{x}[n]$	N	$N-1$
$e[n] = d[n] - \hat{d}[n]$	—	1
$\underline{h}[n+1] = \underline{h}[n] + \mu e^*[n]\underline{x}[n]$	$N+1$	N
Total	$2N+1$	$2N$

The LMS algorithm works according to some general rules under which it converges and produces a given level of mean squared error (MSE). The LMS converges for the following conditions

$$0 < \mu < \frac{2}{\lambda_{max}},$$

where λ_{max} is the largest eingenvalue of $\underline{\underline{R}}_{xx} = \sum_{k=0}^{N-1} \lambda_k \Phi_k \Phi_k^H$.

The MSE of the LMS algorithm can be evaluated analytically by the following expression.

$$MSE[n] = MMSE + \sum_{k=0}^{N-1} \frac{\lambda_k \cdot \mu}{2 - \mu \lambda_k} MMSE$$

which is illustrated by the following figure.

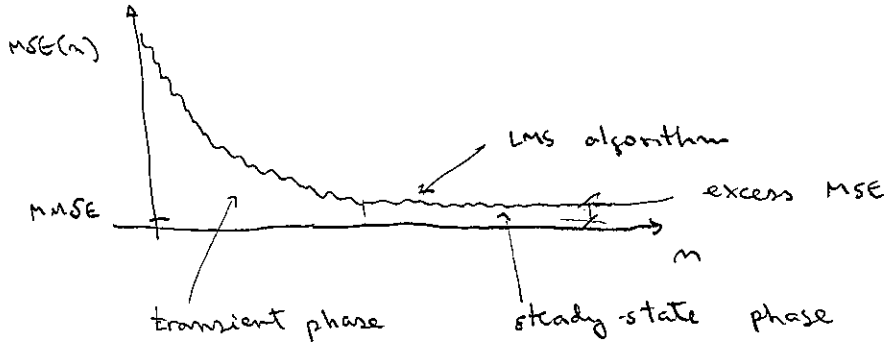


Fig. 8. Behaviour of the learning curve of the LMS algorithm.