

Course material

Course:

Digital Signal Processing II

Video:

2.3.2.a Optimal Least Squares

Concepts (extracted from automatically generated subtitles):

Error measure. Adaptive signal processing. Input signal. Partial derivatives. Filter h. Error signal. Desired signal. Optimal lee squares. First term. Filter coefficients. Quadratic cost function. Communication system scenario. Transmitted signal. Optimal least squares. Simple case.



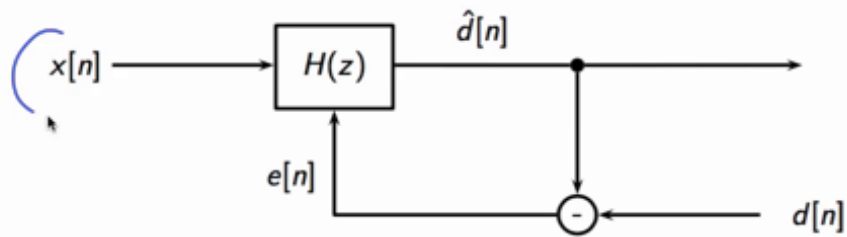
[to video sequence search](#)
(within Digital Signal Processing II.)



[to video](#)

Center for Digital Education. More educational support material here:

<https://exoset.epfl.ch/booc/>



- ▶ $d[n]$: desired signal
- ▶ $\hat{d}[n]$: adaptive approximation
- ▶ $e[n]$: error signal

73

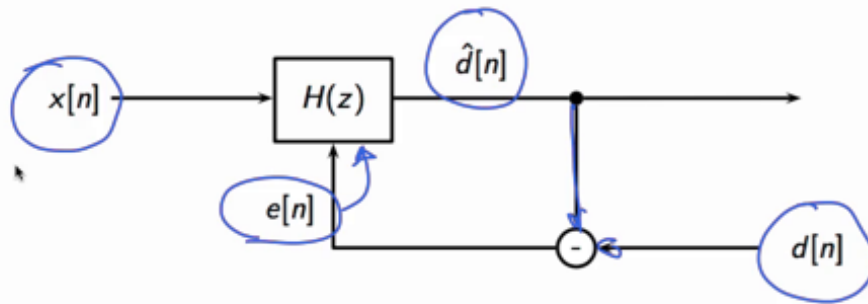
The classic setup for adaptive signal processing is the following.

notes

summary

0m 0s





- ▶ $d[n]$: desired signal
- ▶ $\hat{d}[n]$: adaptive approximation
- ▶ $e[n]$: error signal

73

We have an input signal, a stochastic process, and a desired signal here, again, a stochastic process. We want to find a filter h of z , that transforms, our input, into our desired, signal. A typical situation is a communication system scenario, where x of n is the transmitted signal that has been corrupted, by the transmission channel. If we have access to a copy of the unmodified signal, we can use this adaptive signal processing scheme, to build an equalizer. So the filter h of z will be an equalizer that undoes, the disruption introduced by the channel. So we filtered input with h of z . We obtain what we hope to be a good estimate of our desired signal. We can compute an error measure as the difference between our estimate, and the desired signal, and use this error signal to drive an adaptation inside a filter. So we adapt the filter coefficients, to bring the estimate as close as possible to the desired signal. Now, we could try and design a system, that brings the error to 0, but that's not very realistic because there's no guarantee that we can exactly, replicate the desired signal starting from the input.

notes

summary

0m 7s



optimal filter $H(z)$ *minimizes* the Mean Square Error

$$H(z) = \arg \min_{H(z)} \{E[|e[n]|^2]\}$$

74

A better strategy is to optimize the filter so that it minimizes the so-called Mean Square Error. So we find that the filter h of z that minimizes the expected value of the square magnitude of the error signal.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

summary

.....

.....

.....

.....

1m 18s



$$H(z) = \arg \min_{H(z)} \{E [|e[n]|^2]\}$$

Advantages of a squared error measure:

- ▶ minimum always exist
- ▶ error easily differentiable
- ▶ output will be orthogonal to error
- ▶ only need second moments!

75

There are several advantages to using the quadratic cost function for the error measure, and the first one is that because the surface is quadratic, the minimum will always exist and it will be a global minimum. Also, a quadratic function is easily differentiable, and this will lead to a theoretical formulation and an algorithmic implementation that are simple. We will see that a quadratic error measure,

notes

summary

1m 32s



$$H(z) = \arg \min_{H(z)} \{E[|e[n]|^2]\}$$

Advantages of a squared error measure:

- ▶ minimum always exist
- ▶ error easily differentiable
- ▶ output will be orthogonal to error
- ▶ only need second moments!

75

guarantees orthogonality between the error and the input, and this is very useful in linear prediction, as we will see later. As we pointed out in the previous lectures, to compute a squared error measure, we only need the second moments of the stochastic processes that are

notes

summary

1m 56s



$$H(z) = \arg \min_{H(z)} \{E[|e[n]|^2]\}$$

Advantages of a squared error measure:

- ▶ minimum always exist
- ▶ error easily differentiable
- ▶ output will be orthogonal to error
- ▶ only need second moments!

involved in the computation.

notes

summary

2m 12s



Will only consider FIR adaptive filters:

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

76

We will only consider FIR adaptive filters, in this lecture. There are more complicated, systems that we could implement, but this captures the essence

notes

summary

2m 13s



Will only consider FIR adaptive filters:

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

76

of adaptive signal processing. So in general, our approximation of the desired signal,

notes

summary

2m 25s



Will only consider FIR adaptive filters:

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

~

76

will be the convolution of an n tab impulse response, with past input samples. All the quantities, input and output, are stochastic processes.

notes

summary

2m 35s



Two cases:

- ▶ for WSS signals, one-shot solution: Optimal Least Squares
- ▶ for “almost” WSS signals, iterative solutions: stochastic gradient descent or LMS

77

There are now two cases, for wide-sense stationary signals, the solution is simple, it's a one shot solution called optimal lee squares.

notes

summary

2m 42s



Two cases:

- ▶ for WSS signals, one-shot solution: Optimal Least Squares
- ▶ for “almost” WSS signals, iterative solutions: stochastic gradient descent or LMS

77

For almost stationary signals, we will introduce an iterative solution called stochastic gradient descent. What we mean by almost stationary signals, is signals that are locally stationary, but in the long run, they exhibit some change that can now be classified as stationary. Typical example, is the human voice, that on a small-scale, resembles a stationary signal, but on the long run, of course, exhibits large statistical variations.

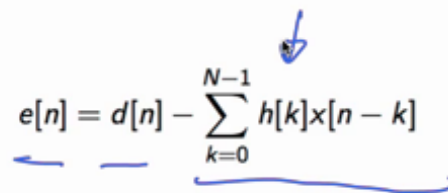
notes

summary

2m 52s



$$e[n] = d[n] - \sum_{k=0}^{N-1} h[k]x[n-k]$$



Minimum is found by setting all partial derivatives to zero

$$\begin{aligned} \frac{1}{2} \frac{\partial E[e^2[n]]}{\partial h[i]} &= E \left[e[n] \frac{\partial e[n]}{\partial h[i]} \right] \\ &= -E[e[n]x[n-i]] = 0 \end{aligned}$$

Let's start with the simple case, the optimal least squares. So we have an input signal and a desired signal that are assumed to be wide-sense, stationary. The error signal is given by the desired signal minus our approximation that we saw we obtain via FIR filtering of the input. We want to find the optimal coefficients, of the filter, of the optimal h of k ,

notes

summary

3m 22s



$$e[n] = d[n] - \sum_{k=0}^{N-1} h[k]x[n-k]$$

Minimum is found by setting all partial derivatives to zero

$$\frac{1}{2} \frac{\partial E[e^2[n]]}{\partial h[i]} = E \left[e[n] \frac{\partial e[n]}{\partial h[i]} \right]$$

$$\uparrow \quad \quad \quad \leftarrow \quad \quad \quad = -E[e[n]x[n-i]] = 0$$

78

for k that goes from 0 to n minus one, so that the mean square error is minimized. To find the optimal coefficients, we take the expected value of the squared error, and we set all of its partial derivatives with respect to the filter coefficients to 0. So just a technicality, we prepend 1.5 to the partial derivatives so that we don't have a leading factor that is carried over in the computation. So if we take the partial derivatives with respect to the filter coefficients, we recognize that the expectation operator is linear, so we can interchange expectation and differentiation, and we have the expected value of the partial derivatives of the squared error. Now of course, this is two times, the error times, derivative of the error, and that's where the 1.5 factor comes in, canceling the two, and we obtain that the partial derivative with respect to the coefficient number i here is equal to minus, the expected value of the product, between the error, at time n , and the input,

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

summary

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3m 49s



$$E[e[n]x[n-i]] = 0$$

error is orthogonal to input: all useful information has been extracted!

79

at time n minus i . So we want to set this quantities to 0. Before we proceed, let's just examined this expression for a second. The fact that the expected value of the product between

notes

summary

4m 49s



$$0 \leq i \leq N-1$$



$$E[e[n]x[n-i]] = 0$$

error is orthogonal to input: all useful information has been extracted!

79

the error and the input for all values of i , because remember we're computing the derivatives for all coefficients. So i goes from 0 to n minus one. So the fact that this product has an expected value of 0 means that the error is orthogonal to the input. If you remember the discussion for the previous lecture, that means that all useful information has been extracted from the signal. The error and the signal are independent. This is called the orthogonality principle, and we will use this shortly.

notes

summary

5m 1s



$$e[n] = d[n] - \sum_{k=0}^{N-1} h[k]x[n-k]$$

$$\begin{aligned} \frac{1}{2} \frac{\partial E[e^2[n]]}{\partial h[i]} &= -E[e[n]x[n-i]] \\ &= E\left[\sum_{k=0}^{N-1} h[k]x[n-k]x[n-i]\right] - E[d[n]x[n-i]] \\ &= \sum_{k=0}^{N-1} h[k]r_x[i-k] - r_{dx}[i] \quad (\text{WSS signals}) \end{aligned}$$

Let's go back now to the derivation of the partial derivatives. So we have found that each partial derivative is the expected value of the product, between the error, and the input,

notes

summary

5m 32s



$$e[n] = d[n] - \sum_{k=0}^{N-1} h[k]x[n-k]$$

$$\frac{1}{2} \frac{\partial E[e^2[n]]}{\partial h[i]} = -E[e[n]x[n-i]]$$

$$= E \left[\sum_{k=0}^{N-1} h[k]x[n-k]x[n-i] \right] - E[d[n]x[n-i]]$$

$$= \sum_{k=0}^{N-1} h[k]r_x[i-k] - r_{dx}[i] \quad (\text{WSS signals})$$

delayed by eye samples. We now replace the value of the error by the expression here.

notes

summary

5m 42s



$$e[n] = d[n] - \sum_{k=0}^{N-1} h[k]x[n-k]$$

$$\begin{aligned} \frac{1}{2} \frac{\partial E[e^2[n]]}{\partial h[i]} &= -E[e[n]x[n-i]] \\ &= E\left[\sum_{k=0}^{N-1} h[k]x[n-k]x[n-i]\right] - E[d[n]x[n-i]] \\ &= \sum_{k=0}^{N-1} h[k]r_x[i-k] - r_{dx}[i] \quad (\text{WSS signals}) \end{aligned}$$


80

So desired signal minus FIR approximation based on the input, we have this first term that depends on the filter, and this term that depends only on the product between the desired signal and the input signal. Now here we use the hypothesis that all signals are wide-sense, stationary. So we bring the expectation operator inside the sum. So this is the convolution between the filter and the auto-correlation of the input signal. The second term is the cross-correlation between input and desired signal. So now let's set all partial derivatives to 0,

notes

summary

5m 49s



setting all partial derivatives to zero:

$$\sum_{k=0}^{N-1} h[k] r_x[i-k] = r_{dx}[i]$$

in matrix form:

$$\mathbf{R}\mathbf{h} = \mathbf{g}$$

81

and what we obtain is the following formula,

notes

summary

6m 24s



setting all partial derivatives to zero:

$$\sum_{k=0}^{N-1} h[k] r_x[i-k] = r_{dx}[i]$$

in matrix form:

$$\mathbf{R} \mathbf{h} = \mathbf{g}$$

81

that relates, the filter coefficients, to the auto-correlation of the input, and the cross-correlation between input and desired signal. We have to solve this system of equation,

notes

summary

6m 26s



setting all partial derivatives to zero:

$$\sum_{k=0}^{N-1} h[k] r_x[i-k] = r_{dx}[i]$$

$$0 \leq i \leq N-1$$

in matrix form:

$$\mathbf{R} \mathbf{h} = \mathbf{g}$$

81

for i goes from 0 to n minus one, and we can express the system of equations in compact form using matrix notation, as $\mathbf{R} \mathbf{h} = \mathbf{g}$. Now,

notes

summary

6m 37s



$$\mathbf{h} = [h[0] \ h[1] \ h[2] \ \dots \ h[N-1]]^T$$

$$\mathbf{h} = \mathbf{R}^{-1} \mathbf{g}$$

$$\mathbf{R} = \begin{bmatrix} r_x[0] & r_x[1] & r_x[2] & \dots & r_x[N-1] \\ r_x[1] & r_x[0] & r_x[1] & \dots & r_x[N-2] \\ r_x[2] & r_x[1] & r_x[0] & \dots & r_x[N-3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_x[N-1] & r_x[N-2] & \dots & \dots & r_x[0] \end{bmatrix}$$

$$\mathbf{g} = [r_{dx}[0] \ r_{dx}[1] \ r_{dx}[2] \ \dots \ r_{dx}[N-1]]^T$$

R is the auto-correlation matrix that you see here. It's a Toeplitz matrix. In other words, all the values on the diagonals are the same. $\mathbf{g}$ is the vector that contains the filter coefficients, and $\mathbf{g}$ is a vector that contains n values of the cross-correlation between input and desired signal. So you can solve the system in one shot, and get the optimal coefficients, as, $\mathbf{R}^{-1} \mathbf{g}$. The optimal least squares method, finds the minimum, of the mean square error function. This function is a multivariate function of the filter coefficients. So in general, it's a very high-dimensional function that it's hard to visualize. We can however, again, some intuition about the property of the optimal least-squares method,

notes

summary

6m 50s



$$\begin{aligned}
 J &= E[e^2[n]] \\
 &= E[(d[n] - \hat{d}[n])^2] \\
 &= E[(d[n] - (h_0 x[n] + h_1 x[n-1]))^2] \\
 &= \sigma_d^2 + r_x[0]h_0^2 + r_x[0]h_1^2 + 2r_x[1]h_0h_1 - 2r_{dx}[0]h_0 - 2r_{dx}[1]h_1 \\
 &= \sigma_d^2 + [h_0 \ h_1] \begin{bmatrix} r_x[0] & r_x[1] \\ r_x[1] & r_x[0] \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} - 2[h_0 \ h_1] \begin{bmatrix} r_{dx}[0] \\ r_{dx}[1] \end{bmatrix}
 \end{aligned}$$

if we consider a simple example, where the filter only has two taps. In this case, the mean square error, is a function of two variables, and we can represent it as a three-dimensional block. So let's define this error surface as J , which is equal to the expected value of the squared error, so the mean square error. We now expand the expression for the error as

notes

summary

7m 37s



$$\begin{aligned}
 J &= E[e^2[n]] \\
 &= E\left[\left(\underline{d[n]} - \underline{\hat{d}[n]}\right)^2\right] \\
 &= E\left[\left(d[n] - (h_0x[n] + h_1x[n-1])\right)^2\right] \\
 &= \sigma_d^2 + r_x[0]h_0^2 + r_x[0]h_1^2 + 2r_x[1]h_0h_1 - 2r_{dx}[0]h_0 - 2r_{dx}[1]h_1 \\
 &= \sigma_d^2 + \begin{bmatrix} h_0 & h_1 \end{bmatrix} \begin{bmatrix} r_x[0] & r_x[1] \\ r_x[1] & r_x[0] \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} - 2 \begin{bmatrix} h_0 & h_1 \end{bmatrix} \begin{bmatrix} r_{dx}[0] \\ r_{dx}[1] \end{bmatrix}
 \end{aligned}$$

the difference between the desired signal and our estimate.

notes

summary

7m 58s



$$\begin{aligned}
 J &= E[e^2[n]] \\
 &= E[(d[n] - \hat{d}[n])^2] \\
 &= E[(d[n] - (h_0 x[n] + h_1 x[n-1]))^2] \\
 &= \sigma_d^2 + r_x[0]h_0^2 + r_x[0]h_1^2 + 2r_x[1]h_0h_1 - 2r_{dx}[0]h_0 - 2r_{dx}[1]h_1 \\
 &= \sigma_d^2 + [h_0 \ h_1] \begin{bmatrix} r_x[0] & r_x[1] \\ r_x[1] & r_x[0] \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} - 2[h_0 \ h_1] \begin{bmatrix} r_{dx}[0] \\ r_{dx}[1] \end{bmatrix}
 \end{aligned}$$

We have replaced the estimate, by the explicit convolution between the impulse response and the input. The two taps in the impulse response will be denoted as h_0 and h_1 for convenience,

notes

summary

8m 2s



$$\begin{aligned}
 J &= E[e^2[n]] \\
 &= E[(d[n] - \hat{d}[n])^2] \\
 &= E[(d[n] - (h_0 x[n] + h_1 x[n-1]))^2] \\
 &= \sigma_d^2 + r_x[0]h_0^2 + r_x[0]h_1^2 + 2r_x[1]h_0h_1 - 2r_{dx}[0]h_0 - 2r_{dx}[1]h_1 \\
 &= \sigma_d^2 + \underbrace{[h_0 \ h_1] \begin{bmatrix} r_x[0] & r_x[1] \\ r_x[1] & r_x[0] \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix}}_{\text{quadratic form}} - 2 \underbrace{[h_0 \ h_1] \begin{bmatrix} r_{dx}[0] \\ r_{dx}[1] \end{bmatrix}}_{\text{cross-product}}
 \end{aligned}$$

and if we work out the products, we end up with this expression in the end, that, comprises, three terms. We have a scalar here. We have a vector matrix, vector multiplication, which is a quadratic form, and therefore defines a paraboloid, in 3D space. Then an extra term here, a two-by-two matrix, which is the cross-product between the filter coefficients and

notes

summary

8m 14s



minimum achievable MSE

$$J = \sigma_d^2 + \mathbf{h}^T \mathbf{R} \mathbf{h} - 2 \mathbf{h}^T \mathbf{g}$$

the cross-correlation between input and desired signal. If we look at the roles of these three terms, we have that the first term is equal to the variance of the desired signal, or I should say the second moment, if the desired signal is non-zero mean, and it's the minimum achievable mean-square error. So this is the minimum error that we can have with the least squares solution.

notes

summary

8m 37s



translation term (minimum is not in origin)

$$J = \sigma_d^2 + \mathbf{h}^T \mathbf{R} \mathbf{h} - \boxed{2\mathbf{h}^T \mathbf{g}}$$

The last term, is just a translation term that displaces the paraboloid, away from the origin. So we can neglect this term, and just look at the first two.

notes

summary

8m 56s



$$J = \sigma_d^2 + \mathbf{h}^T \mathbf{R} \mathbf{h}$$

↑

error surface is an elliptic paraboloid:

- ▶ major and minor axes are proportional to $1/\sqrt{\lambda_{0,1}}$
- ▶ signal's autocorrelation determines the shape of the error surface

$$\mathbf{R} = \begin{bmatrix} \sigma & \sigma \\ \sigma & \sigma \end{bmatrix}$$

We have that, the error surface is an elliptic paraboloid, and the major and minor axis of this paraboloid are going to be proportional to one over the square root of the eigenvalues of the matrix $\mathbf{R}$. So the signal is auto-correlation determines the shape, of the error surface. If we have, for instance, a wide signal, and remember for a wide signal, the matrix $\mathbf{R}$, will be equal to,

notes

summary

9m 5s



$$J = \sigma_d^2 + \mathbf{h}^T \mathbf{R} \mathbf{h}$$

↑

error surface is an elliptic paraboloid:

- ▶ major and minor axes are proportional to $1/\sqrt{\lambda_{0,1}}$
- ▶ signal's autocorrelation determines the shape of the error surface

$$\mathbf{R} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

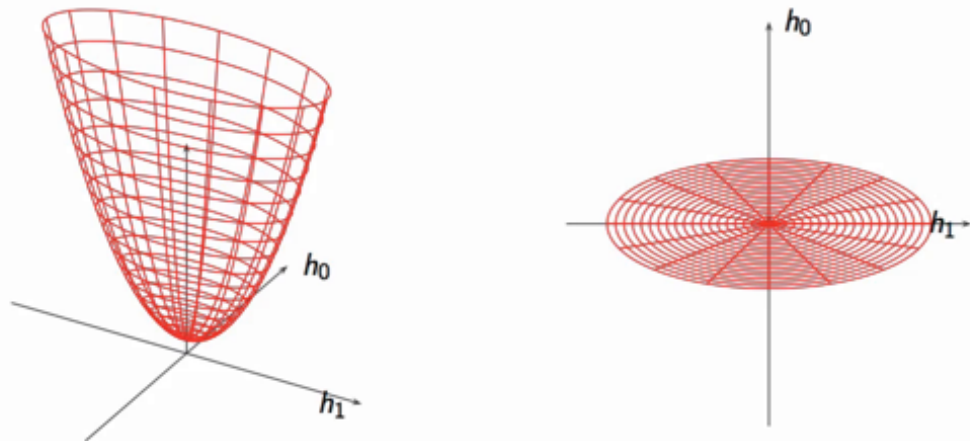
Sigma square, on the main diagonal, and 0 everywhere else, and the two eigenvalues of the matrix are going to be the same, and equal to Sigma square.

notes

summary

9m 32s





So the error surface will look like, it's a paraboloid with circular cross section, and the least square, algorithm, will find, the minimum that is here. For auto-correlation matrix that is not diagonal, so for an input that is not wide, the eigenvalues will be different. So the error paraboloid will have an elliptical cross-section like so. cross-section like so.

notes

summary

9m 40s

