Adaptive Systems

Problems for the Classroom

Christian Feldbauer

with significant revisions and extensions from

Bernhard C. Geiger

geiger@ieee.org

Signal Processing and Speech Communication Laboratory, Inffeldgasse 16c/EG

last modified on February 7, 2014 for Winter term 2014/15

Organizational Information

Course Webpage

`http://www.spsc.tugraz.at/courses/adaptive/`
You can possibly find a newer version of this document there.

Newsgroup

There is a newsgroup for the discussion of all course-relevant topics at:
`news:tu-graz.lv.adaptive`

Schedule

Eight or nine meetings ($\approx$ 90 minutes each) on Tuesdays from 12:15 to 13:45 in lecture hall i11. Please refer to TUGraz.online or our website to get the actual schedule.

Grading

Three homework assignments consisting of analytical problems as well as MATLAB simulations (30 to 35 points each, 100 points in total without bonus problems). Solving bonus problems gives additional points. Work should be done in pairs.

achieved points	grade
≥ 88	1
$75 \ldots 87$	2
$62 \ldots 74$	3
$49 \ldots 61$	4
≤ 48	5

A delayed submission results in a penalty of 10 points per day. Submitting your work as a LaTeX-document can earn you up to 3 (additional) points.

Prerequisites

- (Discrete-time) Signal Processing (FIR/IIR Filters, z-Transform, DTFT, ...)
- Stochastic Signal Processing (Expectation Operator, Correlation, ...)
- Linear Algebra (Matrix Calculus, Eigenvector/-value Decomposition, Gradient, ...)
- MATLAB

Contents

1. The Optimum Linear Filtering Problem—Least-Squares and Wiener Filters

1.1. Transversal Filter

We write the convolution sum as an inner vector product

$$y[n] = \sum_{k=0}^{N-1} c_k^*[n]x[n-k] = \mathbf{c}^{\mathsf{H}}[n]\mathbf{x}[n].$$

where

n	...	time index, $n \in \mathbb{Z}$
$x[n]$	...	input sample at time n
$y[n]$	...	output sample at time n
$(\cdot)^{\mathsf{H}}$	...	Hermitian transpose
$\mathbf{x}[n] = \Big[x[n], x[n-1], \ldots, x[n-N+1]\Big]^{\mathsf{T}}$	...	tap-input vector at time n
$\mathbf{c}^{\mathsf{H}}[n] = \Big[c_0^*[n], c_1^*[n], \ldots, c_{N-1}^*[n]\Big]$	...	Hermitian transpose of coefficient vector at time n (time-varying system)
N	...	number of coefficients, length of $\mathbf{x}[n]$
$N-1$	...	number of delay elements, filter order

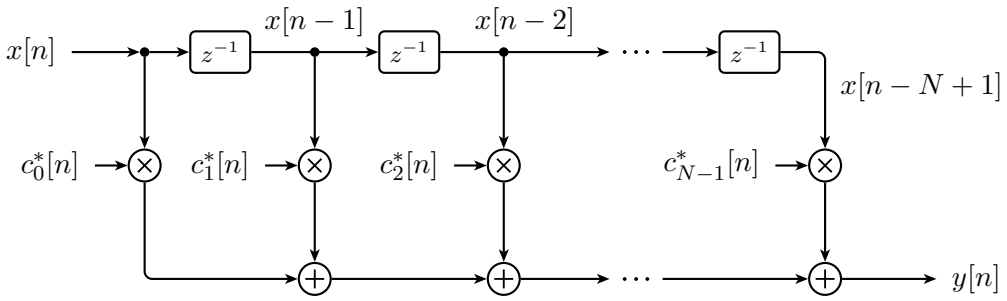


Figure 1: Transversal filter structure.

Special case: $\mathbf{c}[n] = \mathbf{c} \Rightarrow$ time-invariant FIR filter of order $N-1$

1.2. The Linear Filtering Problem

The problem is to approximate a desired signal $d[n]$ by filtering the input signal $x[n]$. For simplicity, we first consider a fixed (i.e., non-adaptive) filter $\mathbf{c}[n] = \mathbf{c}$.

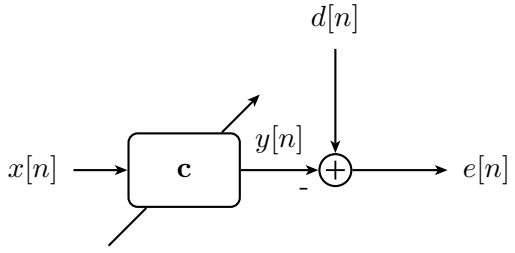


Figure 2: The linear filtering problem.

The goal is to find the optimum filter coefficients $\mathbf{c}$. But what does *optimum* mean?

1.3. Least-Squares Filters

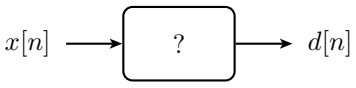
Consider a (finite) set of observations of $\{x[n]\}$ and of $\{d[n]\}$ is given (e.g., all past samples from $n = 0$ to now). We define a *deterministic cost function* as

$$J_{LS}(\mathbf{c}) = \sum_{k=0}^{n} |e[k]|^2,$$

and the problem is to find those filter coefficients that minimize this cost function:

$$\mathbf{c}_{LS} = \underset{\mathbf{c}}{\operatorname{argmin}}\, J_{LS}(\mathbf{c}).$$

Problem 1.1. The following input/output measurements performed on a black box are given:



n	$x[n]$	$d[n]$
0	-1	-3
1	-1	-5
2	1	0

Find the optimum Least-Squares Filter with $N = 2$ coefficients. Use matrix/vector notation for the general solution. Note that the input signal $x[n]$ is applied to the system at time $n = 0$, i.e., $x[-1] = 0$.

Problem 1.2. The previous problem has demonstrated that gradient calculus is important. To practice this calculus, determine $\nabla_{\mathbf{c}} J(\mathbf{c})$ for the following cost functions:

(*i*) $J(\mathbf{c}) = K$

(*ii*) $J(\mathbf{c}) = \mathbf{c}^{\mathsf{T}}\mathbf{v} = \mathbf{v}^{\mathsf{T}}\mathbf{c} = \langle \mathbf{c}, \mathbf{v} \rangle$

(*iii*) $J(\mathbf{c}) = \mathbf{c}^{\mathsf{T}}\mathbf{c} = ||\mathbf{c}||^2 = \langle \mathbf{c}, \mathbf{c} \rangle$

(*iv*) $J(\mathbf{c}) = \mathbf{c}^{\mathsf{T}}\mathbf{A}\mathbf{c}$, where $\mathbf{A}^{\mathsf{T}} = \mathbf{A}$.

MATLAB/Octave Exercise 1.1: Exponentially-Weighted Least Squares

(*i*) For the linear filtering problem shown before, derive the optimum filter coefficients $\mathbf{c}$ in the sense of exponentially weighted least squares, i.e., find $\mathbf{c}[n] = \operatorname{argmin}_{\mathbf{c}} J(\mathbf{c}, n)$, where the cost function is

$$J(\mathbf{c}, n) = \sum_{k=n-M+1}^{n} \lambda^{n-k} \cdot |e[k]|^2$$

with the so-called 'forgetting factor' $0 < \lambda \leq 1$. Use vector/matrix notation. Hint: a diagonal weighting matrix may be useful. Explain the effect of the weighting and answer for what scenario(s) such an exponential weighting may be meaningful.

(*ii*) Write a MATLAB function that computes the optimum filter coefficients in the sense of exponentially weighted least squares according to the following specifications:

```
function c = ls_filter( x, d, N, lambda)
% x ... input signal
% d ... desired output signal (of same length as x)
% N ... number of filter coefficients
% lambda ... optional "forgetting factor" 0<lambda<=1 (default =1)
```

(*iii*) We now identify a time-varying system. To this end, implement a filter with the following time-varying 3-sample impulse response:

$$\mathbf{h}[n] = \begin{bmatrix} 1 \\ -1 + 0.002 \cdot n \\ 1 - 0.002 \cdot n \end{bmatrix}.$$

Generate 1000 input/output sample pairs ($x[n]$ and $d[n]$ for $n = 0 \ldots 999$) using stationary white noise with zero mean and variance $\sigma_x^2 = 1$ as the input signal $x[n]$. All delay elements are initialized with zero (i.e., $x[n] = 0$ for $n < 0$). The adaptive filter has also 3 coefficients ($N = 3$). By calling the MATLAB function `ls_filter` with length-M segments of both $x[n]$ and $d[n]$, the coefficients of the adaptive filter $\mathbf{c}[n]$ for $n = 0 \ldots 999$ can be computed. Visualize and compare the obtained coefficients with the true impulse response. Try different segment lengths M and different forgetting factors λ. Compare and discuss your results and explain the effects of M and λ (e.g., $M \in \{10, 50\}$, $\lambda \in \{1, 0.1\}$).

1.4. The Wiener Filter

We consider $x[n]$ and $d[n]$ as (jointly) stationary stochastic processes[1]. The cost function is now stochastic:

$$J_{MSE}(\mathbf{c}) = \mathrm{E}\left\{|e[n]|^2\right\} \ \ldots \text{ Mean Squared Error (MSE)}$$

and the optimum solution in the MSE sense is obtained as:

$$\mathbf{c}_{MSE} = \underset{\mathbf{c}}{\operatorname{argmin}}\, J_{MSE}(\mathbf{c}).$$

Problem 1.3. The autocorrelation sequence of a stochastic process $x[n]$ is defined as

$$r_{xx}[n, k] := \mathrm{E}\left\{x[n + k]x^*[n]\right\}.$$

If $x[n]$ is *stationary*, then the autocorrelation sequence does not depend on time n, i.e.,

$$r_{xx}[k] = \mathrm{E}\left\{x[n + k]x^*[n]\right\}.$$

Calculate the autocorrelation sequence for the following signals (A and θ are constant and φ is uniformly distributed over $(-\pi, \pi]$):

(*i*) $x[n] = A\sin(\theta n)$

(*ii*) $x[n] = A\sin(\theta n + \varphi)$

(*iii*) $x[n] = Ae^{\jmath(\theta n + \varphi)}$

Problem 1.4. For the optimum linear filtering problem, find $\mathbf{c}_{MSE}$ (i.e., the *Wiener-Hopf equation*). What statistical measurements must be known to get the solution?

Problem 1.5. Assume that $x[n]$ and $d[n]$ are a jointly wide-sense stationary, zero-mean processes.

(*i*) Specify the autocorrelation matrix $\mathbf{R}_{xx} = \mathrm{E}\left\{\mathbf{x}[n]\mathbf{x}^T[n]\right\}$.

(*ii*) Specify the cross-correlation vector $\mathbf{p} = \mathrm{E}\left\{d[n]\mathbf{x}[n]\right\}$.

[1] Note that if two processes are jointly WSS, they are WSS. The converse, however, is not necessarily true (i.e., two WSS processes need not be jointly WSS).

(*iii*) Assume that $d[n]$ is the output of a linear FIR filter to the input $x[n]$, i.e., $d[n] = \mathbf{h}^T\mathbf{x}[n]$. Furthermore, $\dim(\mathbf{h}) = \dim(\mathbf{c})$. What is the optimal solution in the MSE sense?

Problem 1.6. In order to get the MSE-optimal coefficients, the first N samples of $r_{xx}[k]$, the auto-correlation sequence of $x[n]$, and the cross-correlation between the tap-input vector $\mathbf{x}[n]$ and $d[n]$ need to be known. This and the next problem are to practice the computation of correlations.

Let the input signal be $x[n] = A\sin(\theta n + \varphi)$, where φ is a random variable, uniformly distributed over $(-\pi, \pi]$.

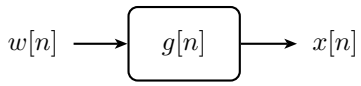
(*i*) Calculate the auto-correlation sequence $r_{xx}[k]$.

(*ii*) Write the auto-correlation matrix $\mathbf{R}_{xx}$ for a Wiener-filtering problem with $N = 1$, $N = 2$, and $N = 3$ coefficients.

(*iii*) Answer for these 3 cases, whether the Wiener-Hopf equation can be solved or not?

(*iv*) Repeat the last tasks for the following input signal: $x[n] = Ae^{j(\theta n+\varphi)}$.

Problem 1.7. The input signal $x[n]$ is now zero-mean white noise $w[n]$ filtered by an FIR filter with impulse response $g[n]$.



Find the auto-correlation sequence $r_{xx}[k]$.

Filtering with the Wiener filter allows us to make a few statements about the statistics of the error signal:

Theorem 1 (Principle of Orthogonality)**.** *The estimate $y[n]$ of the desired signal $d[n]$ (stationary process) is optimal in the sense of a minimum mean squared error if, and only if, the error $e[n]$ is orthogonal to the input $x[n-m]$ for $m = 0 \ldots N-1$.*

Proof. Left as an exercise. □

Corollary 1. *When the filter operates in its optimum condition, also the error $e[n]$ and the estimate $y[n]$ are orthogonal to each other.*

Proof. Left as an exercise. □

Problem 1.8. Show that the minimum mean squared error equals

$$J_{MMSE} = J(\mathbf{c}_{MSE}) = \mathrm{E}\left\{|d[n]|^2\right\} - \mathbf{p}^{\mathsf{H}}\mathbf{R}_{xx}^{-1}\mathbf{p}.$$

1.5. System Identification

We now apply the solution of the linear filtering problem to system identification. Let $d[n] = \mathbf{h}^{\mathsf{H}}\mathbf{x}[n]$, where $\mathbf{h}$ is the impulse response of the system to be identified.

Problem 1.9. Let the order of the unknown system be $M-1$, and let the order of the Wiener filter be $N-1$, where $N \geq M$. Determine the MSE-optimal solution under the assumption that the autocorrelation sequence $r_{xx}[k]$ of the input signal is known.

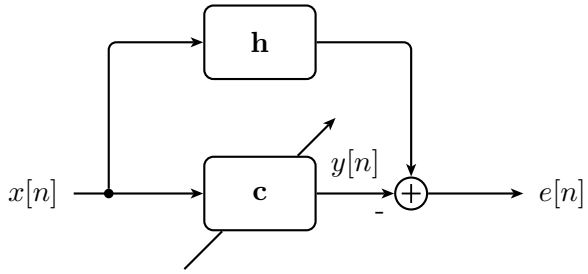


Figure 3: The system identification problem in a noise-free environment.

Problem 1.10. Repeat the previous problem when the order of the unknown system is $M-1$ and the order of the Wiener filter is $N-1$ with $N < M$. Use vector/matrix notation!

Problem 1.11. Let the order of the unknown system be 1 ($d[n] = h_0 x[n] + h_1 x[n-1]$) but the Wiener filter is just a simple gain factor ($y[n] = c_0 x[n]$). Determine the optimum value for this gain factor. The autocorrelation sequence $r_{xx}[k]$ of the input signal is known. Consider the cases when $x[n]$ is white noise and also when $x[n]$ is not white.

1.6. System Identification in a Noisy Environment

In contrary to the previous scenario, we now consider the case where the output signal of the system we want to identify is superimposed by a noise signal $w[n]$, as depicted in Fig. 4.

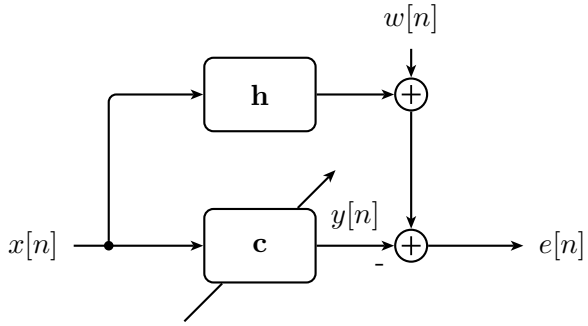


Figure 4: The system identification problem in a noisy environment.

The desired signal is now given as

$$d[n] = \mathbf{h}^{\mathsf{H}}\mathbf{x}[n] + w[n]$$

where $\mathbf{h}$ is the impulse response of the system to be identified and $w[n]$ is stationary, additive noise.

Problem 1.12. Show that the optimal coefficient vector of the Wiener filter equals the impulse response of the system, i.e., $\mathbf{c}_{MSE} = \mathbf{h}$ if, and only if, $w[n]$ is orthogonal to $x[n-m]$ for $m = 0 \ldots N-1$.

Problem 1.13. Under what condition is the minimum mean squared error equal to $J_{MSE}(\mathbf{c}_{MSE}) = \mathrm{E}\left\{|w[n]|^2\right\}$?

1.7. Iterative Solution without Matrix Inversion—Gradient Search

Recall that for the optimal filtering problem the cost function evaluates to

$$J_{MSE}(\mathbf{c}) = \mathrm{E}\left\{|d[n] - \mathbf{c}^{\mathsf{H}}\mathbf{x}[n]|^2\right\} = \mathrm{E}\left\{|d[n]|^2\right\} - 2\mathbf{p}^{\mathsf{H}}\mathbf{c} + \mathbf{c}^{\mathsf{H}}\mathbf{R}_{xx}\mathbf{c}$$

and that the gradient of this cost function with respect to the coefficient vector $\mathbf{c}$ equals

$$\nabla_{\mathbf{c}} J_{MSE}(\mathbf{c}) = 2\left(\mathbf{R}_{xx}\mathbf{c} - \mathbf{p}\right).$$

In Problem 1.4 we used these expressions to derive the Wiener-Hopf solution, which required the inversion of the autocorrelation matrix $\mathbf{R}_{xx}$.

In contrary to that, the *Gradient Search Method* is an interative method which updates the coefficient vector $\mathbf{c}[n]$ depending on the gradient of the cost function in a direction minimizing the MSE. Thus, this iterative algorithm is also called the *Method of Steepest Descent*. Mathematically, the coefficient update rule is given by

$$\mathbf{c}[n] = \mathbf{c}[n-1] + \mu\left(\mathbf{p} - \mathbf{R}_{xx}\mathbf{c}[n-1]\right)$$

where μ is a stepsize parameter and where the term in parentheses is the negative gradient, i.e.,

$$\mathbf{p} - \mathbf{R}_{xx}\mathbf{c}[n-1] = -\nabla_{\mathbf{c}} J_{MSE}(\mathbf{c})\Big|_{\mathbf{c}=\mathbf{c}[n-1]}.$$

Problem 1.14. Assuming convergence, show that this algorithm converges toward the MSE-optimal coefficient vector $\mathbf{c}_{MSE}$. Derive the range for the step size parameter μ for which the algorithm is stable.

Problem 1.15. Calculate the convergence time constant(s) of the decay of the misalignment vector $\mathbf{v}[n] = \mathbf{c}[n] - \mathbf{c}_{MSE}$ (coefficient deviation) when the gradient search method is applied.

Problem 1.16.

(*i*) Express the MSE as a function of the misalignment vector $\mathbf{v}[n] = \mathbf{c}[n] - \mathbf{c}_{MSE}$.

(*ii*) Find an expression for the learning curve $J_{MSE}[n] = J_{MSE}(\mathbf{c}[n])$ when the gradient search is applied.

(*iii*) Determine the time constant(s) of the learning curve.

Problem 1.17. Consider a noise-free system identification problem where both the adaptive and the unknown transversal filter have 2 coefficients. The statistics of the input signal are known as

$$\mathbf{R}_{xx} = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}.$$

The Gradient Method with $\mu = 1/2$ is used to solve for the filter coefficients. The coefficients of the unknown system are $\mathbf{h} = [2,\ 1]^{\mathsf{T}}$.

(*i*) Simplify the adaptation algorithm by substitution for $\mathbf{p} = \mathrm{E}\{\mathbf{x}[n]d[n]\}$ according to the given system identification problem. Additionally, introduce the misalignment vector $\mathbf{v}[n]$ and rewrite the adaptation algorithm such that $\mathbf{v}[n]$ is adapted.

(*ii*) The coefficients of the adaptive filter are initialized with $\mathbf{c}[0] = [0,\ -1]^{\mathsf{T}}$. Find an expression for $\mathbf{v}[n]$ either analytically or by calculation of some (three should be enough) iteration steps. Do the components of $\mathbf{v}[n]$ show an exponential decay? If yes, determine the corresponding time constant.

(*iii*) Repeat the previous task when the coefficients are initialized with $\mathbf{c}[0] = [0,\ 3]^{\mathsf{T}}$. Do the components of $\mathbf{v}[n]$ show an exponential decay? If yes, determine the corresponding time constant.

(*iv*) Repeat the previous task when the coefficients are initialized with $\mathbf{c}[0] = [0,\ 0]^{\mathsf{T}}$. Do the components of $\mathbf{v}[n]$ show an exponential decay? If yes, determine the corresponding time constant.

(*v*) Explain, why an exponential decay of the components of $\mathbf{v}[n]$ can be observed although the input signal $x[n]$ is not white.

2. Adaptive Transversal Filter Using The LMS Algorithm

2.1. The LMS Adaptation Algorithm

We now analyze the LMS adaptation algorithm, whose update rule is given as:

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

where

$$e[n] = d[n] - y[n] = d[n] - \mathbf{c}^{\mathsf{H}}[n-1]\mathbf{x}[n]$$

$\mathbf{c}[n]$	...	new coefficient vector
$\mathbf{c}[n-1]$	...	old coefficient vector
μ	...	step-size parameter
$e[n]$	...	error at time n
$d[n]$	...	desired output at time n
$\mathbf{x}[n]$	...	tap-input vector at time n

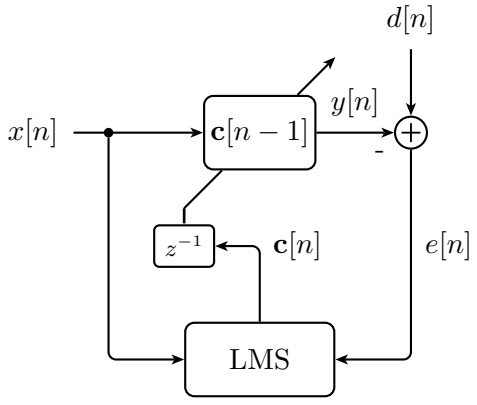


Figure 5: Adaptive transversal filter.

How to choose μ? As it was shown in the lecture course, a sufficient (deterministic) stability condition is given by

$$0 < \mu < \frac{2}{||\mathbf{x}[n]||^2} \qquad \forall n$$

where $||\mathbf{x}[n]||^2$ is the tap-input energy at time n.

Note that the stochastic stability conditions in related literature, i.e.,

$$0 < \mu < \frac{2}{\mathrm{E}\left\{||\mathbf{x}[n]||^2\right\}} \qquad \forall n$$

or

$$0 < \mu < \frac{2}{N\sigma_x^2} \quad \text{for stationary input,}$$

only ensure 'stability on average'.

2.2. Normalized LMS Adaption Algorithm

To make the step size parameter independent of the energy of the input signal, the normalized LMS can be used:

$$\mathbf{c}[n] = \mathbf{c}[n-1] + \frac{\tilde{\mu}}{\alpha + \mathbf{x}^{\mathsf{H}}[n]\,\mathbf{x}[n]}\, e^*[n]\,\mathbf{x}[n]$$

where α is a small positive constant to avoid division by zero.

How to choose $\tilde{\mu}$?

Here the algorithm can be shown to be stable if (sufficient stability condition)

$$0 < \tilde{\mu} < 2.$$

> **MATLAB/Octave Exercise 2.1:** Write a MATLAB function `y=lms1(x,d,N,mu)` which implements an adaptive transversal filter using LMS.
>
> ```
> function y = lms1( x, d, N, mu)
> %LMS1   Adaptive transversal filter using LMS
> % y = lms1( x, d, N, mu)
> % INPUT
> %   x ... vector with the samples of the input signal x[n], length(x) = xlen
> %   d ... vector with the samples of the desired output signal d[n]
> %         length(d) = xlen
> %   N ... number of coefficients
> %   mu .. step-size parameter
> % OUTPUT
> %   y ... vector with the samples of the output signal y[n]
> %         size(y) = [ xlen, 1] ... column vector
> ```
>
> Test your function using a constant input $x[n] = 2$ and a constant desired signal $d[n] = 1$ for $n = 0, \ldots, 999$. The adaptive filter should be zeroth-order, i.e., $N = 1$. Try different values for μ. Plot $x[n]$, $y[n]$, and $d[n]$ into the same figure by executing `plot([x,y,d])` (note: `x`, `y`, and `d` should be column vectors here).

> **MATLAB/Octave Exercise 2.2:** Usually we are interested to see how the adaptation of the coefficients works or how the error behaves over time. Therefore we need the function `[y,e,c]=lms2(x,d,N,mu)` which provides us more output arguments. Extend the function from MATLAB Exercise 2.1 by the additional output arguments.

```
function [ y, e, c] = lms2( x, d, N, mu)
%LMS2   Adaptive transversal filter using LMS (for algorithm analysis)
% [y, e, c] = lms2( x, d, N, mu)
% INPUT
%   x ... vector with the samples of the input signal x[n], length(x) = xlen
%   d ... vector with the samples of the desired output signal d[n]
%         length(d) = xlen
%   N ... number of coefficients
%   mu .. step-size parameter
% OUTPUT
%   y ... vector with the samples of the output signal y[n]
%         size(y) = [ xlen, 1] ... column vector
%   e ... vector with the samples of the error signal e[n]
%         size(y) = [ xlen, 1] ... column vector
%   c ... matrix with the used coefficient vectors c[n]
%         size(c) = [ N, xlen]
```

Test this function by applying the same input as in MATLAB Exercise 2.1 and plot the squared error $e^2[n]$ versus time (*learning curve*).

MATLAB/Octave Exercise 2.3: Write the function `[y,e,c]=nlms2(x,d,N,mu)` which implements the normalized LMS algorithm according to Section 2.2 and has the same arguments as `lms2()`.

2.3. System Identification Using an Adaptive Filter

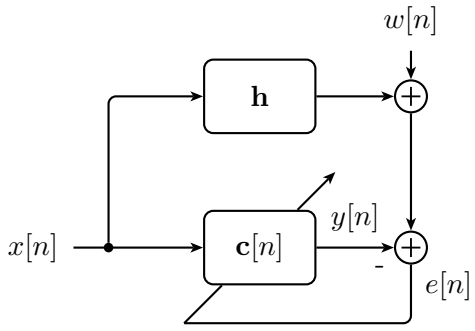


Figure 6: System identification using an adaptive filter and LMS.

Minimum Error, Excess Error, and Misadjustment If we can get only noisy measurements from the unknown system

$$d[n] = \sum_{k=0}^{M-1} h_k^* \, x[n-k] + w[n] \; ,$$

the MSE $J_{MSE}[n] = \mathrm{E}\left\{|e[n]|^2\right\}$ does not vanish completely if the time goes toward infinity.

We assume $x[n]$ and $w[n]$ are jointly stationary, uncorrelated processes. The remaining error can be written as

$$\lim_{n\to\infty} J_{MSE}(\mathbf{c}[n]) = J_{\text{excess}} + J_{MSE}(\mathbf{c}_{MSE})$$

where $J_{MSE}(\mathbf{c}_{MSE}) = \sigma_w^2$ is the minimum MSE (MMSE), which would be achieved by the Wiener-Hopf solution. The *excess error* ($J_{MSE}(\mathbf{c}_{MSE})$) is caused by a remaining misalignment

between the Wiener-Hopf solution and the coefficient vector at time n, i.e., it relates to $\mathbf{v}[n] \neq 0$ (as it happens all the time for the LMS).

Finally, we define the *ratio* between the excess error and the MMSE as the *misadjustment*

$$M = \frac{J_{\text{excess}}}{J_{MSE}(\mathbf{c}_{MSE})} \approx \frac{\mu N \sigma_x^2}{2} .$$

From the stability bounds on μ follows that $0 < M < 1$.

Problem 2.1. Let $\bar{\tau}$ be the average convergence time constant defined by $\bar{\tau} = 1/\mu\bar{\lambda}$, where $\bar{\lambda}$ is the average of all N eigenvalues of $\mathbf{R}_{xx}$. Show that the following trade-off between convergence time, filter order, and misadjustment exists:

$$\bar{\tau} M = \frac{N}{2}$$

As a consequence, a large filter order leads either to slow convergence or to a large misadjustment.

MATLAB/Octave Exercise 2.4: For a noise-free system identification application, write a MATLAB script to **visualize the adaptation process both in the time domain and in the frequency domain**. Take the function `lms2()` from MATLAB Exercise 2.2 and let $x[n]$ be either normally distributed or uniformly distributed random numbers with zero mean and unit variance (use the MATLAB code `sqrt(12)*(rand(...)-0.5)` or `randn(...)`). Choose a proper value for the step-size parameter μ.

For the unknown system, you can take an arbitrary coefficient vector $\mathbf{h} \neq [0, 0, \ldots, 0]^{\mathsf{T}}$ and let the order of the adaptive filter N be the same as of the unknown system M. To calculate $d[n]$, use the MATLAB function `filter()`. To transform the impulse response $\mathbf{h}$ of the unknown system and the instantaneous impulse response $\mathbf{c}[n]$ of the adaptive filter into the frequency domain, use the MATLAB function `freqz()`.

Additionally, modify your script and

- examine the case $N > M$ ('overmodeling').
- examine the case $N < M$ or when the unknown system is an IIR filter ('undermodeling').

For the above cases, try white and also non-white input signals (pass the white $x[n]$ through a non-flat filter to make it non-white; do you need to recompute μ?). Compare your observations with the theoretical results from Problem 1.10 and Problem 1.9.

MATLAB/Octave Exercise 2.5: Persistent Excitation For the two-coefficient case ($N = M = 2$), visualize the adaptation path in the $\mathbf{c}[n]$-plane ($\mathbf{c}[n] = [c_0[n], c_1[n]]^{\mathsf{T}}$). Let the unknown system be $\mathbf{h} = [1, 2]^{\mathsf{T}}$. Use the normalized LMS algorithm (`nlms2()`) with a proper $\tilde{\mu}$ and compare the results for the following different input signals:

(i) $x[n] = \cos[0.5\pi n]$

(ii) $x[n] = \cos[\pi n]$

(iii) $x[n] = \cos[\pi n] + 2$

(iv) $x[n] = \text{randn}[n]$

Describe your observations. Can the unknown system be identified successfully? Explain why (or why not). See also Problem 1.6.

MATLAB/Octave Exercise 2.6: Convergence Time Constant For a system identification task such as in Fig. 6, we want to determine the convergence time constant τ using the ensemble-averaged misalignment vector $\mathrm{E}\{\mathbf{v}[n]\}$.

For the input signal $x[n]$, we take uniformly distributed random numbers with zero mean and variance σ_x^2. Choose a step-size μ according to the stability condition. For the unknown system, let the number of coefficients M be 2 and set h_0 and h_1 to arbitrary non-zero values. The number of coefficients of the adaptive filter N has to be equal to M.

Write a MATLAB script to produce the following plots:

(*i*) *Effect of* μ. For two different values for μ, plot

$$\ln \frac{\mathrm{E}\{v_k[n]\}}{\mathrm{E}\{v_k[0]\}} \quad \text{versus} \quad n.$$

(*ii*) *Effect of* σ_x^2. For two different values for σ_x^2, plot the functions from the previous task again.

(*iii*) *What about non-white input signals?* For example, let $x[n]$ be an MA process (see Appendix A) or a sinusoid (for the sinusoid we can introduce a random phase offset such as in Problem 1.6, such that the ensemble-averaging yields a smooth curve). Note that the signal power σ_x^2 should remain constant and should be a value from the last task to allow comparisons. Transform $\mathbf{v}$ into its eigenvector space to obtain $\widetilde{\mathbf{v}}$ (use either the known auto-correlation matrix or use the MATLAB function `xcorr()`; you also might use `eig()`). Plot the obtained decoupled functions as in the previous tasks.

The convergence time constant τ_k should be measured automatically and printed into the plots. Describe your observations.

MATLAB/Octave Exercise 2.7: Misadjustment Write a MATLAB script to automatically calculate the misadjustment in a noisy system identification problem plotted in Fig. 6. This script should also plot $J_{MSE}[n]$ (in a logarithmic scale) versus n.

Examine the effects of varying μ, σ_x^2, and σ_w^2. Describe your observations and create a table with the following columns:

μ (given)	σ_x^2 (given)	σ_w^2 (given)	$\lim_{n\to\infty} J_{MSE}[n]$	MSE_{EXCESS}	$MISADJ$
...					

MATLAB/Octave Exercise 2.8: Tracking Repeat task (*iii*) of MATLAB Exercise 1.1 and identify the time-varying system using the LMS algorithm. Examine the effect of μ.

Problem 2.2. Convergence analysis of the LMS algorithm.

(*i*) Assuming convergence of the LMS algorithm

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

find the limit $\mathbf{c}_\infty$ of the sequence of coefficient vectors.

(*ii*) Show that the following expression for a sufficient condition for convergence is true in the case of a noise-free system identification task

$$||\mathbf{v}[n]||^2 < ||\mathbf{v}[n-1]||^2 \quad \forall n$$

where $\mathbf{v}[n]$ is the misalignment vector $\mathbf{v}[n] = \mathbf{c}[n] - \mathbf{c}_\infty$. Which requirements on μ can be derived from this expression?

(*iii*) What happens when the environment is noisy?

Problem 2.3. Joint Recursive Optimality. For a wide class of adaptation algorithms (see [3] for more information), the underlying cost function $J(\mathbf{c}, n)$ can be written as

$$J(\mathbf{c}, n) = \big(\mathbf{c} - \mathbf{c}[n-1]\big)^{\mathsf{T}} \mathbf{G}^{-1}[n] \big(\mathbf{c} - \mathbf{c}[n-1]\big) + \gamma^{-1}[n] \big(d[n] - \mathbf{c}^{\mathsf{T}} \mathbf{x}[n]\big)^2$$

where $\mathbf{c}$ is the new coefficient vector of the adaptive transversal filter and $\mathbf{c}[n-1]$ is the previous one, $\mathbf{x}[n]$ is the tap-input vector, and $d[n]$ is the desired output. Note that the expression $d[n] - \mathbf{c}^{\mathsf{T}}\mathbf{x}[n]$ is the *a-posteriori* error $\epsilon[n]$. The weights $\mathbf{G}[n]$ and $\gamma[n]$ are normalized such that

$$\gamma[n] + \mathbf{x}^{\mathsf{T}}[n]\mathbf{G}[n]\mathbf{x}[n] = 1,$$

and $\mathbf{G}[n]$ is symmetric ($\mathbf{G}[n] = \mathbf{G}^{\mathsf{T}}[n]$).

(*i*) Find a recursive expression to adapt $\mathbf{c}[n]$ given $\mathbf{c}[n-1]$ such that the cost function $J(\mathbf{c}, n)$ is minimized:

$$\mathbf{c}[n] = \arg\min_{\mathbf{c}} J(\mathbf{c}, n).$$

Note, this expression should contain the *a-priori* error $e[n]$

$$e[n] = d[n] - \mathbf{c}^{\mathsf{T}}[n-1]\mathbf{x}[n]$$

and not the a-posteriori error (Hint: find the ratio between the a-posteriori and the a-priori error first.).

(*ii*) Determine the weights $\mathbf{G}[n]$ and $\gamma[n]$ for the case of the LMS algorithm and for the case of the Normalized LMS algorithm.

(*iii*) Show that

$$\min_{\mathbf{c}} J(\mathbf{c}, n) = e^2[n]$$

for all $n > 0$.

Problem 2.4. For a noisy system identification problem, the following two measurements could be accomplished at the plant: $\sigma_x^2 = 1$ and $\sigma_d^2 = 2$. $x[n]$ and $w[n]$ can be assumed to be stationary, zero-mean, white noise processes and orthogonal to each other. The unknown system can be assumed to be linear and time-invariant. The adaptive filter has been specified to have $N = 100$ coefficients, and we can assume that no undermodeling occurs. The LMS algorithm is used to adapt the coefficients, and a maximum misadjustment of -10dB should be reached.

(*i*) How should you choose the step size μ and what convergence time constant will be obtained?

(*ii*) After applying the adaptive filter another measurement has been performed and an error variance of $\sigma_e^2 = 0.011$ has been obtained. Calculate σ_w^2 and the 'coefficient-to-deviation ratio' $10 \log_{10} \frac{||\mathbf{h}||^2}{||\mathbf{v}||^2}$ in dB (i.e., a kind of signal-to-noise ratio).

Problem 2.5. The Coefficient Leakage LMS Algorithm. We will investigate the leaky LMS algorithm which is given by

$$\mathbf{c}[n] = (1 - \mu\alpha)\mathbf{c}[n-1] + \mu e^*[n]\mathbf{x}[n]$$

with the leakage parameter $0 < \alpha \ll 1$.

Consider a noisy system identification task (proper number of coefficients) and assume the input signal $x[n]$ and the additive noise $w[n]$ to be orthogonal. The input signal $x[n]$ is zero-mean white noise with variance σ_x^2. Assume μ and α have been chosen to assure convergence. Determine where this algorithm converges to (on average). Compare your solution with the Wiener-Hopf solution. Also, calculate the average bias.

3. Interference Cancelation

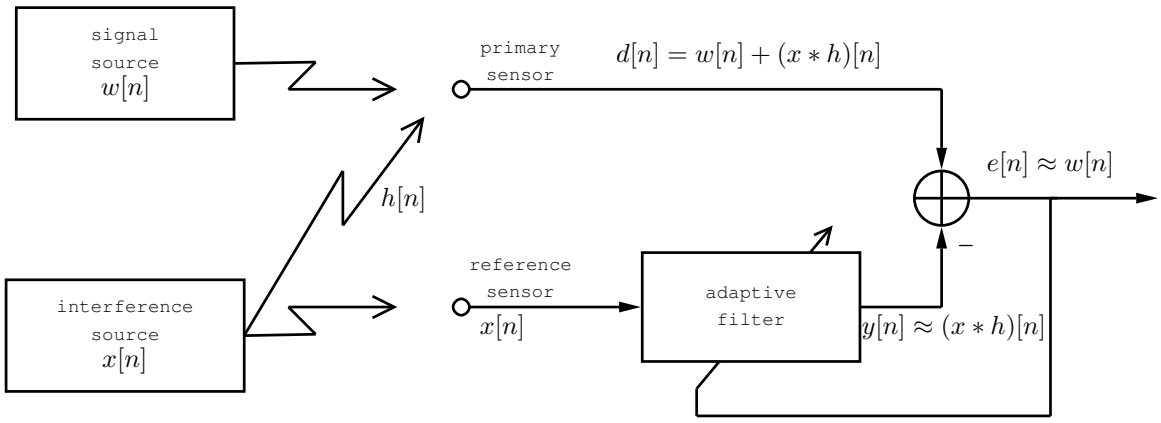


Figure 7: Adaptive noise canceler.

The primary sensor (i.e., the sensor for the desired signal $d[n]$) receives the signal of interest $w[n]$ corrupted by an interference that went through the so-called interference path. When the isolated interference is denoted by $x[n]$ and the impulse response of the interference path by $h[n]$, the sensor receives

$$d[n] = w[n] + \mathbf{h}^T \mathbf{x}[n].$$

For simplification we assume $\mathrm{E}\{w[n]x[n-k]\} = 0$ for all k.

The reference sensor receives the isolated interference $x[n]$.

The error signal is $e[n] = d[n] - y[n] = w[n] + \mathbf{h}^T\mathbf{x}[n] - y[n]$. The adaptive filtering operation is perfect if $y[n] = \mathbf{h}^T\mathbf{x}[n]$. In this case the system output is $e[n] = w[n]$, i.e., the isolated signal of interest.

FIR model for the interference path: If we assume that $\mathbf{h}$ is the impulse response of an FIR system (i.e., $\dim \mathbf{h} = N$) the interference cancelation problem is equal to the system identification problem.

> **MATLAB/Octave Exercise 3.1:** Simulate an interference cancelation problem according to Fig. 7 (e.g. "speaker next to noise source," "50Hz interference in electrocardiography," "baby's heartbeat corrupted by mother's" ...). Additionally, simulate the possibly realistic scenario where a second cross path exists such that the reference sensor receives $x[n] + \mathbf{h}_2^T\mathbf{w}[n]$.

> **Problem 3.1.** We use a first-order transversal filter to eliminate an interference of the $f_{AC} = 50$ Hz AC power supply from an ECG signal. The sampling frequency is f_s ($f_s > 100$ Hz).
>
> (*i*) Using the method of *equating the coefficients*, express the optimum coefficients c_0 and c_1 that fully suppress the interference in terms of $A := |H(e^{j\theta_{AC}})|$ and $\vartheta := \arg H(e^{j\theta_{AC}})$, i.e., in terms of the magnitude and the phase of the frequency response of the interference path at the frequency of the interference.
>
> (*ii*) Calculate the auto-correlation sequence $r_{xx}[k]$ of the reference input signal as a function of the sampling frequency and build the auto-correlation matrix $\mathbf{R}_{xx}$.

(*iii*) Determine the cross-correlation vector $\mathbf{p}$ and solve the Wiener-Hopf equation to obtain the MSE-optimal coefficients. Show that these coefficients are equal to those found in (*i*).

(*iv*) Determine the condition number $\kappa = \frac{\lambda_{max}}{\lambda_{min}}$ of the auto-correlation matrix for the given problem as a function of the sampling frequency. Is the unit delay in the transversal filter a clever choice?

MATLAB/Octave Exercise 3.2: Canceling a Periodic Interference

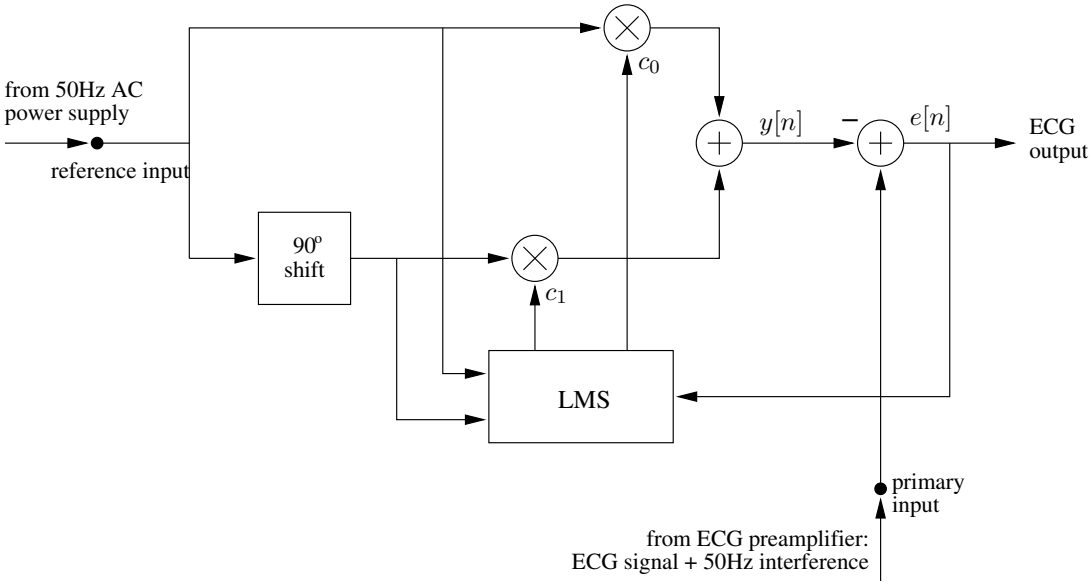
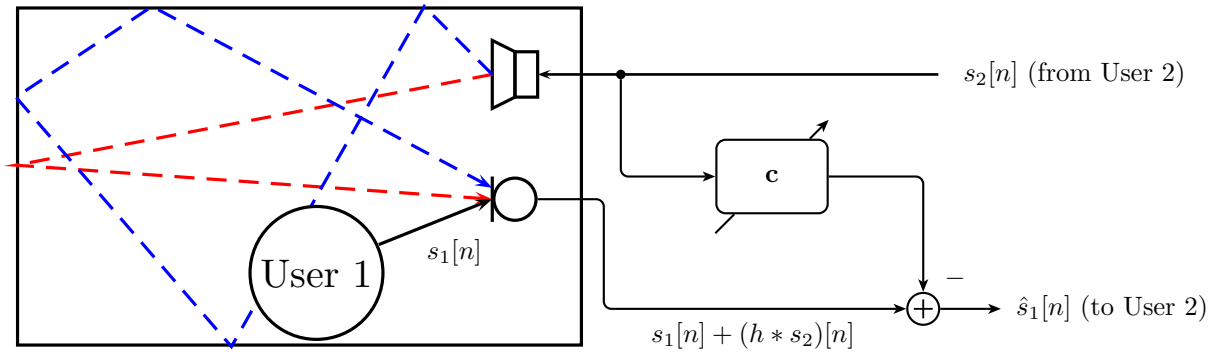


Figure 8: Adaptive notch filter.

To implement the filter structure shown in Fig. 8, you have to modify your MATLAB function of the LMS algorithm. Instead of the transversal filter you need a 90°-shifter. You may use the MATLAB expression `x90=imag(hilbert(x))`. Compare the SNR of the primary input signal and the SNR of the output signal. Measure the convergence time constant and compare your result with the theory. What is the advantage of the 90°-shifter over a unit-delay element in a transversal filter (see also Problem 3.1)?

Problem 3.2. Consider the acoustic echo cancelation scenario below:



(i) Assuming that all s_1 and s_2 are jointly stationary and uncorrelated, derive the coefficient vector $\mathbf{c}$ optimal in the MSE sense.

(ii) Given that the room has a dimension of 3 times 4 meters, and assuming that the speech signals are sampled with $f_s = 8$ kHz, what order should $\mathbf{c}$ have such that at least first-order reflections can be canceled? Note, that *physically* the impulse response of the room is infinite!

(iii) Assume the filter coefficients are updated using the LMS algorithm to track changes in the room impulse response. What problems can occur?

4. Adaptive Linear Prediction

4.1. Autoregressive spectrum analysis

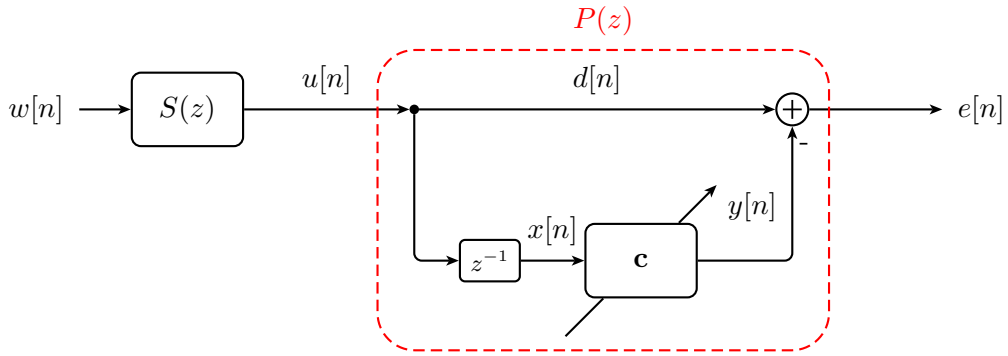


Figure 9: Linear prediction using an adaptive filter.

Let $w[n]$ be a white input sequence, and let $S(z)$ be an all-pole synthesis filter with difference equation

$$u[n] = w[n] - \sum_{k=1}^{L} a_k^* u[n-k].$$

In this case, $u[n]$ is called an autoregressive (AR) process (see appendix B). We can estimate the AR coefficients $a_1, \ldots, a_L$ by finding the MSE-optimal coefficients of a linear predictor. Once the AR coefficients have been obtained, the squared-magnitude frequency response of the recursive process-generator filter can be used as an estimate of the power-spectral density (PSD) of the process $u[n]$ (sometimes called *AR Modeling*).

> **Problem 4.1. Autocorrelation Sequence of an AR Process** Consider the following difference equation
>
> $$u[n] = w[n] + 0.5\, u[n-1],$$
>
> i.e., a purely recursive linear system with input $w[n]$ and output $u[n]$. $w[n]$ are samples of a stationary white noise process with zero mean and $\sigma_w^2 = 1$. Calculate the auto-correlation sequence $r_{uu}[k]$ of the output.

4.2. Linear prediction

A linear predictor tries to predict the present sample $u[n]$ from the N preceding samples $u[n-1], \ldots, u[n-N]$ using a linear combination:

$$\hat{u}[n] = \sum_{k=1}^{N} c_k^* u[n-k].$$

The prediction error is given by $e[n] = u[n] - \hat{u}[n] = w[n] - \sum_{k=1}^{L} a_k^* u[n-k] - \sum_{k=1}^{N} c_k^* u[n-k]$. Minimizing the mean-squared prediction error yields the proper predictor coefficients c_k for $k = 1, \ldots, N$. In the ideal case ($N = L$), the error is a minimum when only the non-predictable white noise excitation $w[n]$ remains as $e[n]$. In this case, we obtain the (negative) AR coefficients: $a_k = -c_k$ for $k = 1, \ldots, L$.

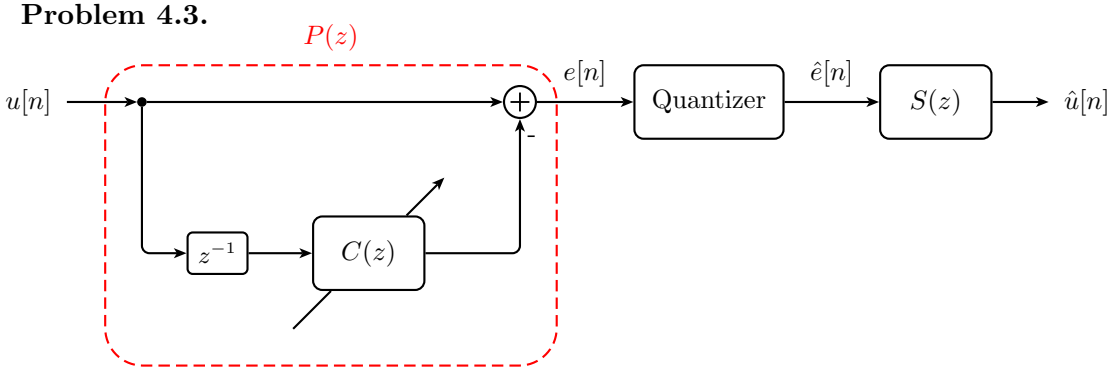
For adaptive linear prediction (see Fig. 9), the adaptive transversal filter is the linear combiner, and an adaptation algorithm (e.g., the LMS algorithm) is used to optimize the coefficients and to minimize the prediction error.

Problem 4.2. AR Modeling. Consider a linear prediction scenario as shown in Fig. 9. The mean squared error should be used as the underlying cost function. The auto-correlation sequence of $u[n]$ is given by

$$r_{uu}[k] = 4/3 \cdot (1/2)^{|k|} \, .$$

Compute the AR coefficients a_1, $a_2, \ldots$ and the variance of the white-noise excitation σ_w^2. Start the calculation for an adaptive filter with 1 coefficient. Then, repeat the calculation for 2 and 3 coefficients.

Problem 4.3.



Consider the scenario depicted above, where $u[n]$ is an AR process. The quantizer depicted shall have a resolution of B bits, and is modeled by an additive source with zero mean and variance $\gamma\sigma_e^2$, where γ is a constant depending on B and where σ_e^2 is the variance of the prediction error $e[n]$. $S(z)$ is the synthesis filter, which is the inverse of the prediction filter.

(i) For a ideal prediction (i.e., the prediction error $e[n]$ is white), compute the output SNR, which is given as

$$\frac{\mathrm{E}\left\{u^2[n]\right\}}{\mathrm{E}\left\{(u[n]-\hat{u}[n])^2\right\}} = \frac{\sigma_u^2}{\sigma_r^2}$$

where $r[n] = u[n] - \hat{u}[n]$.

(ii) Repeat the previous task for the case where no prediction filter was used. What can you observe?

4.3. Yule-Walker Equations

In order to get the Wiener-Hopf solution for the MSE-optimal coefficient vector $\mathbf{c}_{MSE}$

$$\mathbf{R}_{xx}\mathbf{c}_{MSE} = \mathbf{p},$$

we have to substitute $x[n] = u[n-1]$ and $d[n] = u[n]$ and get

$$\mathbf{R}_{uu}\mathbf{c}_{MSE} = \mathbf{r}_{uu_{+1}}.$$

In non-vector notation this reads

$$\begin{bmatrix} r_{uu}[0] & r_{uu}[1] & \ldots & r_{uu}[L-1] \\ r_{uu}^*[1] & r_{uu}[0] & \ldots & r_{uu}[L-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_{uu}^*[L-1] & r_{uu}^*[L-2] & \ldots & r_{uu}[0] \end{bmatrix} \mathbf{c}_{MSE} = \begin{bmatrix} r_{uu}^*[1] \\ r_{uu}^*[2] \\ \vdots \\ r_{uu}^*[L] \end{bmatrix}.$$

These equations are termed the *Yule-Walker equations*. Note that in the ideal case $\mathbf{c}_{MSE} = [-a_1, -a_2, \ldots, -a_L]^\mathsf{T}$ (given that the order of the transversal filter matches the order of the AR process).

MATLAB/Octave Exercise 4.1: Power Spectrum Estimation Generate a finite-length sequence $u[n]$, which represents a snapshot of an arbitrary AR process, and let us denote it as the unknown process. We want to compare different PSD estimation methods:

1. *Direct solution of the Yule-Walker equations.* Calculate an estimate of the auto-correlation sequence of $u[n]$ and solve the Yule-Walker equations to obtain an estimate of the AR coefficients (assuming that the order L is known). Use these coefficients to plot an estimate of the PSD function. You may use the MATLAB functions `xcorr` and `toeplitz`.
2. *LMS-adaptive transversal filter.* Use your MATLAB implementation of the LMS algorithm according to Fig. 9. Take the coefficient vector from the last iteration to plot an estimate of the PSD function. Try different step-sizes μ.
3. *RLS-adaptive transversal filter.* Use `rls.m` (download it from our web page). Try different forgetting factors λ.
4. *Welch's periodogram averaging method.* Use the MATLAB function `pwelch`. Note, this is a non-parametric method, i.e., there is no model assumption.

Plot the different estimates into the same axis and compare them with the original PSD.

4.4. Periodic Interference Cancelation without an External Reference Source

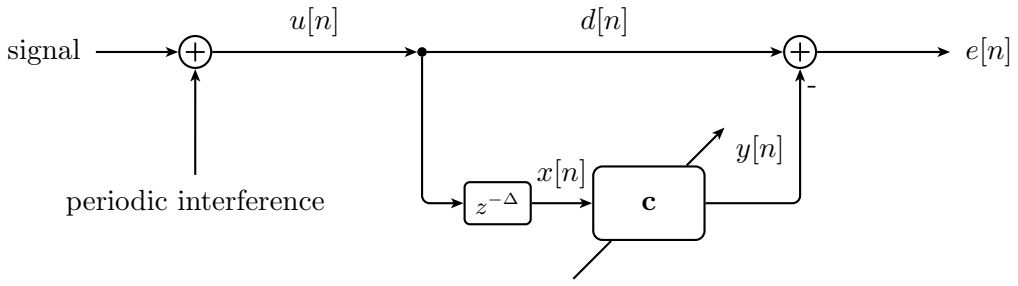


Figure 10: Canceling a periodic interference using a linear predictor.

Fig. 10 shows the usage of an adaptive linear predictor to remove a periodic interference of a broadband signal. The output is simply the whitened prediction error.
Things to be aware of:

- The delay length Δ must be longer than the correlation time of the broadband signal (but not too long to avoid echoes).
- More coefficients yield a sharper filter and therefore less distortion of the broadband signal. But more coefficients increase also the convergence time.

Problem 4.4. Periodic Interference Cancelation without an External Reference Source Consider a measured signal $u[n]$ that is the sum of a white-noise signal $w[n]$ with variance $\sigma_w^2 = 1$ and a sinusoidal: $u[n] = w[n] + \cos(\pi/2 \cdot n + \varphi)$ (with φ a random phase offset).

(*i*) Calculate the auto-correlation sequence $r_{uu}[k]$ of $u[n]$.

(*ii*) Let's attenuate the sinusoidal using the setup of Fig. 10. A delay of 1 should be enough for the white $v[n]$. Compute the optimum coefficients c_0, c_1 of the first-order adaptive filter in the sense of a minimum mean squared error.

(*iii*) Determine the transfer function of the prediction-error filter, compute its poles and zeros, and plot the pole/zero diagram. Sketch its frequency response.

MATLAB/Octave Exercise 4.2: Periodic Interference Cancelation without an External Reference Source Simulate the scenario shown in Fig. 10. Take a speech signal as the broadband signal. Try a delay around 10ms and an order of at least 100.

5. Adaptive Equalization

5.1. Principle

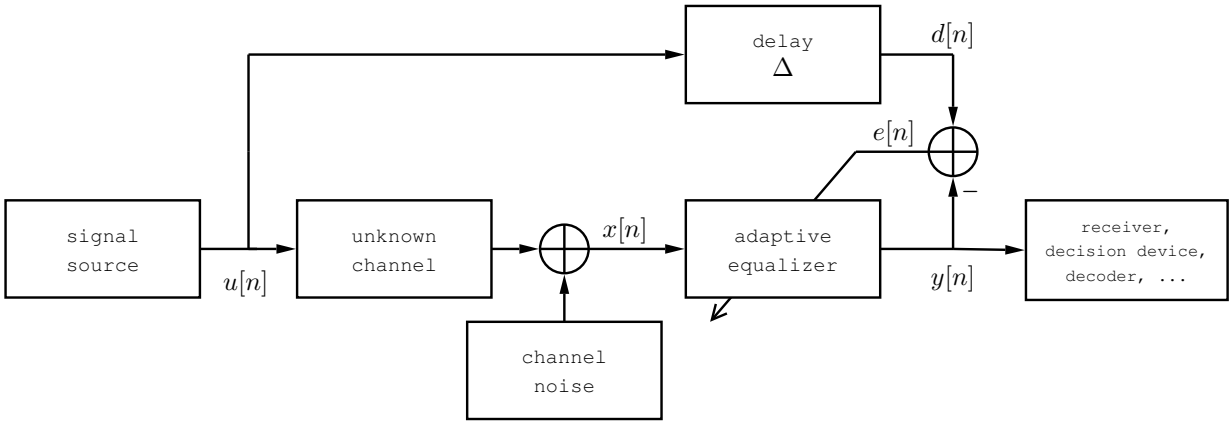


Figure 11: Adaptive equalization (or inverse modeling).

Fig. 11 shows the principle of adaptive equalization. The goal is to adapt the transversal filter to obtain

$$H(z)C(z) = z^{-\Delta} ,$$

i.e., to find the inverse (except for a delay) of the transfer function of the unknown channel. In the case of a communication channel, this eliminates the intersymbol interference (ISI) introduced by the temporal dispersion of the channel.

Difficulties:

1. Assume $H(z)$ has a finite impulse response (FIR) $\Rightarrow$ the inverse system $H^{-1}(z)$ is an IIR filter. Using a finite-length adaptive transversal filter only yields an approximation of the inverse system.
2. Assume $H(z)$ is a non-minimum phase system (FIR or IIR) $\Rightarrow$ the inverse system $H^{-1}(z)$ is not stable.
3. We typically have to introduce the extra delay Δ (i.e., the group delay of the cascade of both the channel and the equalizer).

Situations where a reference, i.e., the original signal, is available to the adaptive filter:

- Audio: adaptive concert hall equalization, car HiFi, airplanes, ... (equalizer = *pre-emphasis* or *pre-distortion*; microphone where optimum quality should be received)
- Modem: transmission of an initial *training sequence* to adapt the filter and/or periodic interruption of the transmission to re-transmit a known sequence to re-adapt.

Often there is no possibility to access the original signal. In this case we have to 'guess' the reference: *Blind Adaptation*. Examples are *Decision-Directed Learning* or the *Constant Modulus Algorithm*, which exploit a-priori knowledge of the source.

> **MATLAB/Octave Exercise 5.1: Inverse Modeling** Setup a simulation according to Fig. 11. Visualize the adaptation process by plotting the magnitude of the frequency response of the channel, the equalizer, and the overall system $H(z)C(z)$.

Problem 5.1. ISI and Open-Eye Condition For the following equivalent discrete-time channel impulse responses

(*i*) $\mathbf{h} = [0.8,\ 1,\ 0.8]^{\mathsf{T}}$

(*ii*) $\mathbf{h} = [0.4,\ 1,\ 0.4]^{\mathsf{T}}$

(*iii*) $\mathbf{h} = [0.5,\ 1,\ 0.5]^{\mathsf{T}}$

calculate the worst-case ISI for binary data $u[n] \in \{+1, -1\}$. Is the channel's eye opened or closed?

Problem 5.2. Least-Squares and MinMSE Equalizer For a noise-free channel with given impulse response $\mathbf{h} = [1,\ 2/3,\ 1/3]^{\mathsf{T}}$, compute the optimum coefficients of the equal-length, zero-delay equalizer in the least-squares sense. Can the equalizer open the channel's eye? Is the least-squares solution equivalent to the minimum-MSE solution for white data $u[n]$?

Problem 5.3. MinMSE Equalizer for a Noisy Channel Consider a channel with impulse response $\mathbf{h} = [1,\ 2/3,\ 1/3]^{\mathsf{T}}$ and additive white noise $\eta[n]$ with zero mean and variance σ_η^2. Compute the optimum coefficients of the equal-length equalizer in the sense of a minimum mean squared error.

5.2. Decision-Directed Learning

Let us now assume that we know the modulation alphabet of the digital transmission system (e.g., binary antipodal modulation, PSK, etc.). The demodulator chooses the output symbol as the element of the modulation alphabet with the minimum distance to the received signal. (For binary antipodal modulation this can be accomplished by a simple threshold device.)

If we now assume that the distortion by the channel is moderate, one can use the distance between the chosen output and the received symbol as the error signal for adapting the equalizer (see Fig. 12).

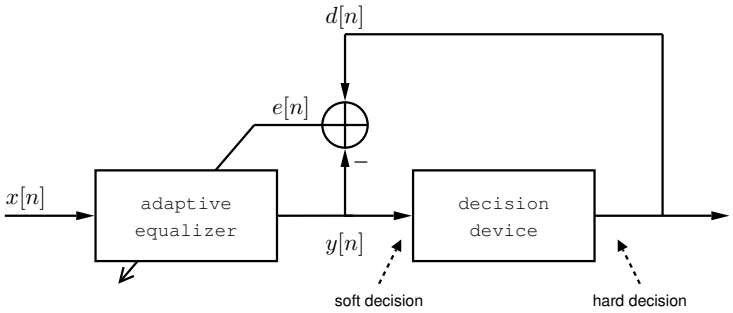


Figure 12: Decision-directed adaptive channel equalizer.

MATLAB/Octave Exercise 5.2: Decision-Directed Channel Equalization Simulate the equalization of a baseband transmission of a binary signal (possible symbols: -1 and $+1$). Plot bit-error graphs for the equalized and unequalized transmission (i.e, a stem plot that indicates for each symbol, whether it has been decoded correctly or not). Extend

your program to add an initialization phase for which a training sequence is available. After the training the equalizer switches to decision-directed mode.

Problem 5.4. Decision-Feedback Equalizer (DFE). Consider the *feedback-only equalizer* in Fig. 13. Assume that the transmitted data $u[n]$ is white and has zero mean.

(*i*) For a general channel impulse response $\mathbf{h}$ and a given delay Δ, calculate the optimum (min. MSE) coefficients $\mathbf{c}_b$ of the feedback equalizer.

(*ii*) What is the resulting impulse response of the overall system when the equalizer operates at its optimum?

(*iii*) Do the MSE-optimum coefficients $\mathbf{c}_b$ of the feedback equalizer change for a noisy channel?

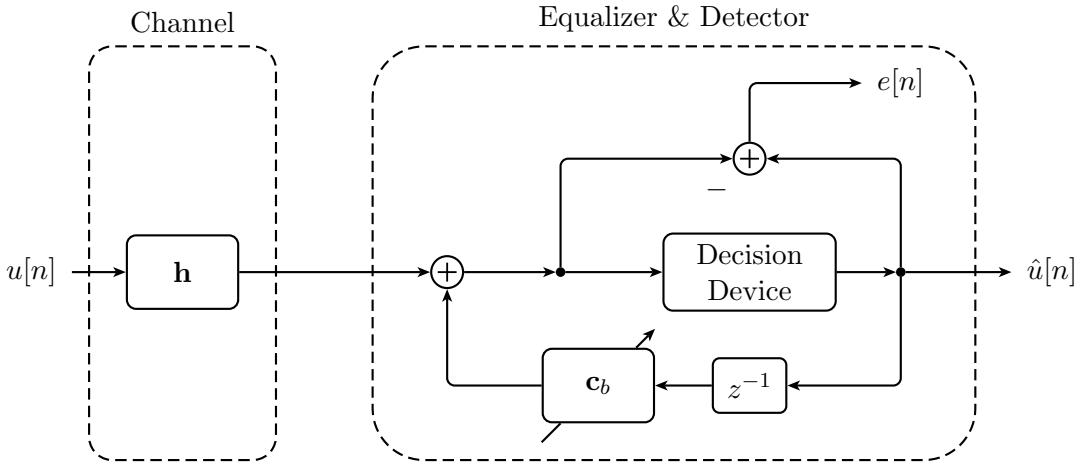


Figure 13: Decision-feedback equalizer.

5.3. Alternative Equalizer Structures

Problem 5.5. Extend the decision-feeback equalizer structure in Fig. 13 by an additional forward (or transversal) equalizer filter with coefficients $\mathbf{c}_f$ right after the channel. Derive the design equation for both MSE-optimum $\mathbf{c}_f$ and $\mathbf{c}_b$ (use an overall coefficient vector $\mathbf{c}^{\mathsf{T}} = [\mathbf{c}_f^{\mathsf{T}}\, \mathbf{c}_b^{\mathsf{T}}]$).

Problem 5.6. Fractionally-Spaced Equalizer. A fractionally-spaced equalizer runs at a sampling rate that is higher than the symbol rate. Consider the $T/2$-fractionally-spaced equalizer (i.e., it runs at the double rate) in Fig. 14 where T is the symbol duration. The decision device is synchronized with the transmitted symbols, which correspond to the even-indexed samples at the double rate.

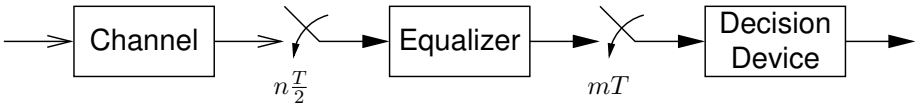


Figure 14: Fractionally-spaced equalizer.

The discrete-time description of the channel for the high sampling rate is

$$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} = 1/2 + z^{-1} + 1/2\, z^{-2} + 1/4\, z^{-3},$$

i.e., the unit delay z^{-1} corresponds to $\frac{T}{2}$.

(*i*) Calculate the coefficients of the equal-length equalizer

$$C(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} + c_3 z^{-3}$$

such that the cascade of the given channel and the equalizer $H(z)C(z) = 1$, i.e., it enables a delay-free and ISI-free transmission.

(*ii*) Calculate the coefficients of the equalizer such that the cascade is a pure delay of 1 symbol, i.e., $H(z)C(z) = z^{-2}$.

(*iii*) Consider the channel to be noisy (additive white noise). Compute the noise gains of the two equalizers of the previous tasks. Which one should be chosen?

(*iv*) Let the channel be

$$H(z) = 1 + 1/2\, z^{-1} + 1/4\, z^{-2} + 1/8\, z^{-3}.$$

Compute again the coefficients of an equal-length equalizer.

A. Moving Average (MA) Process

A stationary MA process $u[n]$ satisfies the difference equation

$$u[n] = v[n] + \sum_{k=1}^{K} g^*[k]v[n-k],$$

where K is the order and $v[n]$ is white noise with zero mean and variance σ_v^2, i.e., $u[n]$ is white noise filtered by an FIR filter with impulse response $g[n]$ where $g[0] = 1$ (as defined in [6, 7]).

The *auto-correlation sequence* of the output $u[n]$ is given by (see Problem 1.7)

$$r_{uu}[k] = \sigma_v^2 \sum_i g[i]g^*[i-k].$$

The variance of $u[n]$ can be obtained by setting $k = 0$:

$$\sigma_u^2 = \sigma_v^2 \sum_i |g[i]|^2.$$

The factor $\sum_i |g[i]|^2$ is termed the *Noise Gain*.

B. Autoregressive (AR) Process

A stationary AR process $u[n]$ satisfies the recursive linear difference equation

$$u[n] = v[n] - \sum_{k=1}^{L} a_k u[n-k],$$

where L is the order, and $v[n]$ is white noise with zero mean and variance σ_v^2, i.e., $u[n]$ is white noise filtered by an all-pole IIR filter. The process is fully specified by the AR coefficients a_k, $k = 1 \dots L$ and the white-noise variance σ_v^2.

The *auto-correlation sequence* $r_{uu}[n]$ can be expressed by a zero-input version of the above recursive difference equation (see Problem 4.1)

$$r_{uu}[n] = -\sum_{k=1}^{L} a_k r_{uu}[n-k] \quad \text{for} \quad n > 0.$$

For instance, knowing the first L samples of the auto-correlation sequence $r_{uu}[0] \dots r_{uu}[L-1]$ is sufficient to calculate $r_{uu}[n]$ $\forall\, n \in \mathbb{Z}$ by recursion (when the AR coefficients a_k, $k = 1 \dots L$ are known). Considering the symmetry of $r_{uu}[n]$ and evaluating the difference equation for $n = 1 \dots L$ yields the Yule-Walker equations (see Problem 4.2) that allow the computation of the AR coefficients from the first $L+1$ samples of the auto-correlation sequence $r_{uu}[0] \dots r_{uu}[L]$. For $n = 0$, the following equation is obtained

$$r_{uu}[0] + \sum_{k=1}^{L} a_k r_{uu}[k] = \sigma_v^2,$$

which shows the relation between the variances σ_v^2 and σ_u^2. Using this equation, the *noise gain* of the AR process-generator filter can be calculated as

$$\frac{\sigma_u^2}{\sigma_v^2} = \frac{1}{1 + \sum_{k=1}^{L} a_k \frac{r_{uu}[k]}{r_{uu}[0]}}.$$

Problem B.1. Assume the process generator difference equation is given as

$$u[n] = v[n] + au[n-1]$$

where $v[n]$ is white noise with variance $\sigma_v^2 = r_{vv}[0]$ and $|a| < 1$. We know that for $k > 0$,

$$r_{uu}[k] = ar_{uu}[k-1] = a^k r_{uu}[0].$$

To fully specify the autocorrelation function r_{uu} we therefore only need to determine $r_{uu}[0] = \sigma_u^2$. To this end, observe that the impulse response of above system is given as $h[n] = (-a)^n u[n]$. To a white noise input, the variance of the output can be computed using the noise gain of the system, i.e.,

$$r_{uu}[0] = \sigma_u^2 = \sigma_v^2 \sum_{n=-\infty}^{\infty} |h[n]|^2 = \sigma_v^2 \frac{1}{1-a^2}.$$

Thus, and with the symmetry of r_{uu},

$$r_{uu}[k] = \sigma_v^2 \frac{a^{|k|}}{1-a^2}.$$

References

[1] Simon Haykin: "Adaptive Filter Theory," Fourth Edition, Prentice-Hall, Inc., Upper Saddle River, NJ, 2002.

[2] George Moschytz and Markus Hofbauer: "Adaptive Filter," Springer-Verlag, Berlin Heidelberg, 2000.

[3] Gernot Kubin: "Joint Recursive Optimality—A Non-Probabilistic Approach to Joint Recursive Optimality—A Non-Probabilistic Approach to," Journal *Computers and Electrical Engineering*, Vol. 18, No. 3/4, pp. 277–289, 1992.

[4] Bernard Widrow and Samuel D. Stearns: "Adaptive Signal Processing," Prentice-Hall, Inc., Upper Saddle River, NJ, 1985.

[5] Edward A. Lee and David G. Messerschmitt: "Digital Communication," Third Edition, Kluwer Academic Publishers, 2004.

[6] Steven M. Kay: "Fundamentals of Statistical Signal Processing—Estimation Theory," Volume 1, Prentice-Hall, Inc., 1993.

[7] Steven M. Kay: "Fundamentals of Statistical Signal Processing—Detection Theory," Volume 2, Prentice-Hall, Inc., 1998.