Signal Processing and Speech Communication Lab.
Graz University of Technology

AS

Adaptive Systems—Homework Assignment 2

Name(s) Matr.No(s).

The analytical part of your homework (your calculation sheets) **as well as** the MATLAB simulation protocol have to be delivered as **hard copy** to our mailbox at Inffeldgasse 16c, ground floor, no later than **2011/12/13**. Use a printed version of **this entire document** as the title pages and fill in your **name(s) and matriculation number(s)**. Submitting your homework as a LaTeX document can earn you **up to 3 points!**

Your MATLAB programs (`*.m` files) and the simulation protocol (in pdf format!) have to be submitted via **e-mail** to the address `hw2.spsc@tugraz.at` no later than **2011/12/13**. The subject of the e-mail consists of the assignment number and your matriculation number(s) **"Assignment2, MatrNo1, MatrNo2"**. You have to zip (or tar) all your homework files to one single file with the name `Assignment2_MatrNo1_MatrNo2.zip`, e.g., `Assignment2_9833280_9933281.zip`, which has to be attached to the e-mail.

Please make sure that your approaches, procedures, and results are clearly presented. Justify your answers!

MATLAB Problem 2.1 (6 Points)—(N)LMS Implementation

Write a MATLAB/Octave function that implements both the LMS and the normalized LMS (NLMS), according to the following specification:

```
function [y,e,c] = lms(x,d,N,mu,NORM,c0)
% INPUTS:
% x ...... input signal vector
% d ...... desired output signal (of same length as x)
% N ...... number of filter coefficients
% mu .,... step size parameter
% NORM ... set "1" for NLMS (bias=0), "0" for LMS
% c0 ..... initial coefficient vector (optional; default all zeros)
% OUTPUTS:
% y ...... output signal vector (of same length as x)
% e ...... error signal vector (of same length as x)
% c ...... coefficient matrix (N rows, number of columns = length of x)
```

Test your implementation with the script provided on the course webpage! In addition to that, write down what you observe in the three generated plots.

Analytical Problem 2.2 (6 Points)—Learning Curve of Gradient Method

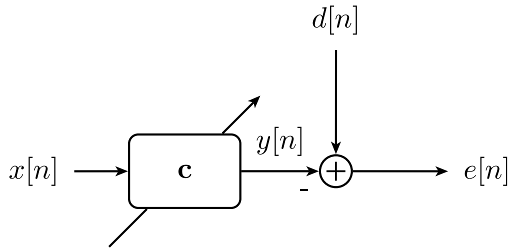

Figure 1: Wiener Filter

In this problem we will use the Gradient Search Method

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

to iteratively solve for the MSE-optimal transveral filter coefficients in a stationary linear filtering problem as depicted in Figure 1.

(a) Express the cost function $J_{MSE}(\mathbf{c}[n])$ as a function of the misalignment vector $\mathbf{v}[n] = \mathbf{c}[n] - \mathbf{c}_{MSE}$ (i.e., coefficient deviation).

(b) Apply a unitary coordinate transform such that the MSE cost function (i.e., its non-constant part) can be expressed as a weighted sum of the squared components (the "decoupled modes") of the transformed misalignment vector at time n.

(c) Use the expression for the decoupled modes to write $J_{MSE}(\mathbf{c}[n])$ as a function of time n and the initial transformed misalignment vector $\tilde{\mathbf{v}}[0]$.

(d) Give an expression for the time constants of these individual exponential terms.

(e) Finally, assume we perform system identification, i.e., $d[n] = \mathbf{c}_{MSE}^T\mathbf{x}[n]$. Let $x[n]$ be a zero-mean, white process with unit variance. Assume further that $\mathbf{c}_{MSE} = [2\ 1]^T$ and that the coefficients of the adaptive system are initialized with a zero vector, i.e., $\mathbf{c}[0] = \mathbf{0}$. Sketch the MSE attributed to each individual decoupled mode and $J_{MSE}(\mathbf{c}[n])$ as a function of time n.

MATLAB Problem 2.3 (15 Points)—(N)LMS Performance Analysis

For the system identification problem shown in Figure 2 we want to determine the convergence time constants τ_i using the *ensemble-averaged*[1] misalignment vector

$$\mathbf{v}[n] = \mathbf{c}[n] - \mathbf{c}_{MSE} = \mathbf{c}[n] - \mathbf{h}.$$

The input signal $x[n]$ is a zero-mean, white Gaussian process with unit variance. The desired signal $d[n]$ is corrupted by additive white Gaussian noise with zero mean and variance $\sigma_\nu^2 = 0.008$. The impulse response of the unknown system is given as

$$\mathbf{h} = [0.6, 0.2, 0.4]^T$$

[1] For every considered time instant n we average over data obtained from independent trials, i.e., using multiple realizations of the input and noise processes.

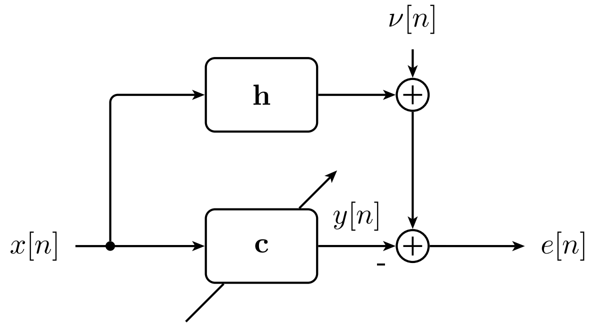

Figure 2: The system identification problem in a noisy environment.

and $N = \dim(\mathbf{c}[n]) = \dim(\mathbf{h})$.

For the following tasks, use the (N)LMS implementation from Problem 2.1. Call your function multiple times and use the coefficient matrices to compute the ensemble averages of the misalignment vector. Initialize you coefficient vector with a zero vector. Write a MATLAB/Octave script to plot

$$\ln \frac{E\{v_k[n]\}}{E\{v_k[0]\}}$$

as a function of time n for all components k of the vector. Your script should automatically determine the time constants τ_k and print them in the legend of the plot. On top of the curves, plot the logarithm of the MSE as a thick line, i.e.,

$$\ln \frac{E\{e^2[n]\}}{E\{e^2[0]\}}.$$

(a) Investigate the effect of the step-size parameter μ for both the LMS and the NLMS algorithms. Use, e.g., $\mu = \{0.0001, 0.001, 0.01, 1\}$. Are the algorithms stable in all these cases?

(b) Now set the step size to $\mu = 0.001$ and vary the variance of the input signal $x[n]$. Use $\sigma_x^2 = \{0.2, 1, 5\}$. What is the difference between the LMS and the NLMS?

(c) Let $x[n]$ be a zero-mean, unit variance, white input process $w[n]$ filtered by a two-tap moving average filter, i.e.,

$$x[n] = \frac{w[n]}{2} + \frac{w[n-1]}{2}.$$

Use again a small step size of $\mu = 0.001$ and plot convergence behavior of the misalignment vector. What can you observe? Can you still determine the time constants? What can you say about the covergence behavior of the MSE as a function of time?

(d) In all previous tasks, how did the MSE behave *after* convergence?

(e) BONUS (4 Points): For all evaluated scenarios compute the *excess MSE*. What can you observe?

Analytical Problem 2.4 (6 Points)—Average Behavior of the LMS

Show that the LMS finds the MSE-optimal solution *on average* in a noisy system identification problem (see Figure 2) with white input $x[n]$.