Adaptive Filtering Exercise 1: Normalized Least-Mean Square (NLMS) Algorithm

Digital Signal Processing
Sommersemester 2022
S. Doclo, D. Fejgin

06.07.2022 & 13.07.2022

Adaptive filters are widely used in signal processing application, for example, system identification, echo cancellation and noise cancellation. The goal of this exercise session is to implement the normalized least mean squares (NLMS) algorithm and evaluate its performance in a system identification task.

Write a MATLAB function to implement the NLMS algorithm. The function inputs should be the desired response signal, d, the adaptive filter input signal u, the adaptation stepsize, μ, and the length of the adaptive filter, N. The output should be the error signal $e(k)$, and the adaptive filter coefficients $\mathbf{w}(k)$ at each sample k. Note that, therefore the adaptive filter coefficients $\mathbf{w}(k)$ need to be stored in a $N \times L$-dimensional matrix with L the length of the input signal.

To remind you, the NLMS update rule is defined as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mu}{\alpha + \mathbf{u}^T(k+1)\mathbf{u}(k+1)} \cdot \mathbf{u}(k+1) \underbrace{\left(d(k+1) - \mathbf{u}^T(k+1)\mathbf{w}(k)\right)}_{e(k+1)}, \tag{1}$$

where $\mathbf{w}(k) = [w_0(k) \;\; w_1(k) \;\; \cdots \;\; w_{N-1}(k)]^T$ is the adaptive filter coefficient vector, $\mathbf{u}(k) = [u(k) \;\; u(k-1) \;\; \cdots \;\; u(k-N+1)]^T$ is the corresponding input signal sequence and α is a (usually small) positive constant.

While in a realistic scenario the length of, e.g., the room impulse response, and hence the required adaptive filter length may be over several 1000 coefficients (and in general is not known exactly), in this exercise we will restrict ourselves to the identification of artificial unknown finite impulse response (FIR) systems h of length 50. To create the artificial FIR systems h, a white gaussian noise input signal u and the desired response d use the matlab commands "randn" and "filter", i.e., generate

`h = randn(50,1), u = randn(L,1) and d = filter(h,1,u).`

with an adequately chosen L (usually $L = 10000$ is adequate, however, carefully choose L to make sure the adaptive filter is indeed converged when evaluating its performance using the EMSE as stated below).

To analyze the performance of an adaptive filter the squared error signal $e^2(k)$ and the normalized filter coefficient error, i.e.,

$$\Delta\mathbf{h}(k) = \frac{\|\mathbf{w}(k) - \mathbf{h}\|^2}{\|\mathbf{h}\|^2} \tag{2}$$

can be used. In addition, to analyze the performance *after convergence* ($L \to \infty$) of the filter the so-called excess mean squared error (EMSE) at steady state can be used. The EMSE is the difference between the mean square error (MSE) of the adaptive filter algorithm *after convergence* ($L \to \infty$) and the the MSE of the optimal Wiener filter solution, i.e., for a stationary input $\mathbf{u}(k)$

$$EMSE = \underbrace{\mathcal{E}\{\left(d(k) - \mathbf{u}^T(k)\mathbf{w}(\infty)\right)^2\}}_{J_{MSE}(\mathbf{w}(\infty))} - \underbrace{\mathcal{E}\{\left(d(k) - \mathbf{u}^T(k)\mathbf{w}_{WF}\right)^2\}}_{J_{MSE}(\mathbf{w}_{WF})} \tag{3}$$

with $\mathcal{E}$ denoting expectation and $\mathbf{w}_{WF}$ the Wiener filter solution. The term $J_{MSE}(\mathbf{w}_{WF})$ is also known to be the irreducible error. For a white noise input sequence $u(k)$ the Wiener solution $\mathbf{w}_{WF}$ depends on the length of the adaptive filter and corresponds to a) $\mathbf{h}$ for the sufficient order case of $N = 50$, b) a zero-padded version of $\mathbf{h}$ for the sufficient order case $N > 50$ and c) a truncated version of $\mathbf{h}$ for the insufficient order case $N < 50$.

For the special case of a sufficient filter order $N = 50$ (considered in part 1 of this exercise) and no near-end disturbance signal (considered throughout this exercise) the EMSE can thus be approximated as

$$EMSE = J_{MSE}(\mathbf{w}(\infty)) \approx \frac{1}{L}\sum_{k=0}^{L-1}(d(k) - \mathbf{u}^T(k)\mathbf{w}(L-1))^2 \tag{4}$$

while for the sufficient filter order $N > 50$ and insufficient filter order $N < 50$ also the irreducible error needs to be computed using the corresponding Wiener filter solution (see above). Using these measures analyze the convergence, EMSE properties and $J_{MSE}(\mathbf{w}_{WF})$ of the NLMS algorithm in the following settings:

1. Sufficient order case: Set $N = 50$
   - Run the NLMS algorithm for different values of μ and observe their impact on the convergence properties and (excess) mean squared error.
   - Run the NLMS algorithm for 4 independent realizations of the input sequence u, i.e. call `u=randn(L,1)` and `d=filter(h,1,u)` to generate new input and desired response vectors, using the same value of μ. Plot $e^2(k)$ and $\Delta\mathbf{h}(k)$ for all 4 realizations. Explain the observed differences in convergence and EMSE.
   - Run the NLMS algorithm for 40 independent realizations of the input sequence u using the same value of μ. Plot the time course of the average squared error and average coefficients error vector norm of these 40 independent realizations. Explain why this corresponds to an approximate ensemble average. Describe convergence behavior and EMSE.
     - What are the effects of increasing and decreasing μ on the ensemble average?
     - Which value for μ would you choose (and why)?
     - What is the maximum value for μ?
2. Sufficient order case: $N > 50$
   - Run the NLMS algorithm for different values of μ and $N > 50$ and compare by means of the approximate ensemble average of 40 independent realizations. How does performance (EMSE and convergence) of the algorithm change for longer N? Compare with the sufficient order case $N = 50$.
3. Insufficient order case: $N < 50$
   - Run the NLMS algorithm for different values of N using a fixed value for μ, e.g., the best μ determined for the sufficient order case. What is the impact of the insufficient order on the convergence and EMSE of the filter? Use the approximate ensemble average of 40 independent realizations to compare your results.
     Hint: to achieve good results you may need to change the length of the input sequence u.
   - Run the NLMS algorithm for different values of μ using a fixed choice of $N < 50$ and compare with the sufficient order case of $N = 50$. Use the approximate ensemble average of 40 independent realizations to compare your results. What are the effects of the insufficient filter order on different choices of μ?
4. Use the **audio signal** (roland.wav) which was provided to you as input signal u. Set $N = 50$ (sufficient filter order). Use different choices of μ. How long does it take for the filter coefficients to converge? Compare to the white noise case and comment on reasons for observed differences.

In the report, please thoroughly describe all performed experiments and include the most important plots