Adaptive Filtering Exercise 2: Frequency Domain Adaptive Filtering (FDAF) Algorithm

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

13.07.2022 & 18.07.2022

The objective in this session is to implement the Frequency Domain Adaptive Filtering (FDAF) and perform a system identification of a real acoustical impulse response. Measurements of real acoustical signals have been performed in the lab. The sound signal is coming from a single loudspeaker and is picked up by one microphone located at a distance of approximately 2 m. Different sound signals have been used as excitation signals. These include a **white noise signal**, a **speech signal** and a **music signal**. For all signals a sampling frequency of $f_s = 16000$ Hz was used.

All signals are stored in Matlab the file `FDAF_system_identification.mat` which was provided to you using the following names:

- the transmitted white noise (u_{Wnoise})
- the recorded white noise (d_{Wnoise})
- the transmitted speech signal (u_{speech})
- the recorded speech signal (d_{speech})
- the transmitted music signal (u_{music})
- the recorded music signal (d_{music}).

1. Write a Matlab function to implement the FDAF algorithm. The function inputs should be the desired response signal, d, the adaptive filter input u, and scalars for the adaptation gain, μ, and the length of the adaptive filter, N. The output should be the error signal $e(k)$, and the adaptive filter coefficients $\mathbf{w}(k)$ in the time domain at each time step k.

   To remind you, we will first repeat the frequency-independent FDAF update rule here, see also Figure 5 and Table I in [1]. This closely follows the flow diagram covered in the lecture. *Hint:* Implementing the FDAF algorithm using the below stated matrix notation is usually not computationally efficient. By using the Matlab function `fft` instead of the DFT matrix $\mathbf{F}$ and by recognizing that the matrix $\mathbf{g}$ performs selection of specific parts of the signals as well as using element-wise multiplication of vectors instead of diagonal matrix and vector multiplications the computational efficient can be significantly improved.

   Assuming a filter length of N samples, a block length of $L = N$ samples and an FFT size of $2N$ the FDAF update rule is:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \Delta(k)\mathbf{F}\mathbf{g}\mathbf{F}^{-1}\mathbf{U}^H(k)\mathbf{E}(k) = \mathbf{W}(k) + \Delta(k)\mathbf{G}\mathbf{U}^H(k)\mathbf{E}(k) \tag{1}$$

   where $\mathbf{F}$ is the $2N \times 2N$-dimensional DFT matrix,

$$\mathbf{G} = \mathbf{F}\mathbf{g}\mathbf{F}^{-1}; \quad \text{with } \mathbf{g} = \begin{bmatrix} \mathbf{I}_N & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{0}_N \end{bmatrix}$$

the gradient constraint matrix with $\mathbf{I}_N$ the $N \times N$-dimensional identity matrix and $\mathbf{0}_N$ the $N \times N$-dimensional matrix containing only zeros. When multiplying from the left side this matrix effectively sets the second half to zero. Furthermore

$$\begin{aligned} \mathbf{E}(k) &= \mathbf{F} \begin{bmatrix} \mathbf{0} \\ \mathbf{e}(k) \end{bmatrix}, \\ \mathbf{e}(k) &= \mathbf{d}(k) - \mathbf{y}(k) \end{aligned}$$

the frequency domain representation of the error signal,

$$\mathbf{d}(k) = [d(kN) \ \cdots \ d(kN+N-1)]^T$$

the desired response signal vector,

$$\begin{aligned} \mathbf{y}(k) &= \underbrace{\begin{bmatrix} \mathbf{0}_N & \mathbf{I}_N \end{bmatrix}}_{\substack{\text{select elements} \\ kN \text{ to } kN+N-1}} \mathbf{F}^{-1}\mathbf{Y}(k), \\ \mathbf{Y}(k) &= \mathbf{U}(k)\mathbf{W}(k) \end{aligned}$$

the filter output vector,

$$\Delta(k) = \mu\big(\alpha + \frac{1}{2N} \sum_{i=0}^{2N-1} P_i(k)\big)^{-1}$$

the time-varying frequency-independent step-size with α a (usually small) positive constant and

$$P_i(k) = \lambda P_i(k-1) + (1-\lambda)\|\mathbf{U}_i(k)\|^2 \qquad i = 0, \cdots, 2N-1$$

the power in the i-th frequency bin, where λ is a smoothing constant and $\mathbf{U}_i(k)$ is the i-th element of the main diagonal of the frequency domain representation $\mathbf{U}(k)$ of the filter input signal $\mathbf{u}(k)$, i.e.,

$$\mathbf{U}(k) = \text{diag}\,\{\mathbf{F}\,\underbrace{[u(kN-N), \ u(kN-N+1), \cdots \ u(kN+N-1)]^T}_{\mathbf{u}^T(k)}\}$$

2. Use the recorded **white noise** signal as desired signal and the transmitted white noise as input signal.
   - Experiment with different values of μ. Plot $e^2(k)$ and the EMSE for different values of μ as well as the estimated acoustical impulse response. Which μ leads to the best identification of the impulse response.
3. Extend the FDAF algorithm to allow for frequency-dependent step step-size parameter $\tilde{\Delta}(k)$ defined as

$$\tilde{\mathbf{\Delta}}(k) = \mu \cdot \text{diag}\,\{\big[(P_0(k)+\alpha)^{-1}, \cdots, (P_{2N-1}(k)+\alpha)^{-1}\big]\}$$

   This also requires to rearrange the update rule given in (1) to

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mathbf{G}\tilde{\mathbf{\Delta}}(k)\mathbf{U}^H(k)\mathbf{E}(k) \tag{2}$$

   - Explain why the rearrangement of the update rule is necessary to allow for the correct implementation of the algorithm.
   - Use the frequency-dependent stepsize $\tilde{\Delta}(k)$. What is the impact of $\tilde{\Delta}(k)$ on the convergence of the algorithm? Compare convergence and EMSE to the case of the frequency-independent stepsize $\Delta(k)$. Explain the observed differences.
   - Experiment with the step-size μ and the smoothing factor λ ($0 \le \lambda \le 1$) for updating the power $P_i(k)$ in each frequency bin. The equivalent smoothing time T_λ in seconds can be calculated as

$$T_\lambda = \frac{1}{1-\lambda}\frac{L}{f_s} \tag{3}$$

     For $\lambda \to 1$ the equivalent smoothing time corresponds to $T_\lambda = \infty$.

- Experiment with the filter length. What is the impact on convergence and EMSE when the filter length is changed? Also plot the estimated acoustical impulse response for different filter lengths.

4. As desired signal, use the recorded **speech** and **music signal** and their respective transmitted signal as input signal. What happens with the convergence performance of the algorithm compared to white noise?
5. What is the difference between the NLMS algorithm implemented in Exercise 8 and the FDAF? Compare the convergence of both algorithms for the recorded white noise, speech and music signals.

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

References

[1] J. J. Shynk, "Frequency-Domain and Multirate Adaptive Filtering," *IEEE Signal Processing Magazine*, vol. 9, no. 1, pp. 14–37, Jan. 1992.