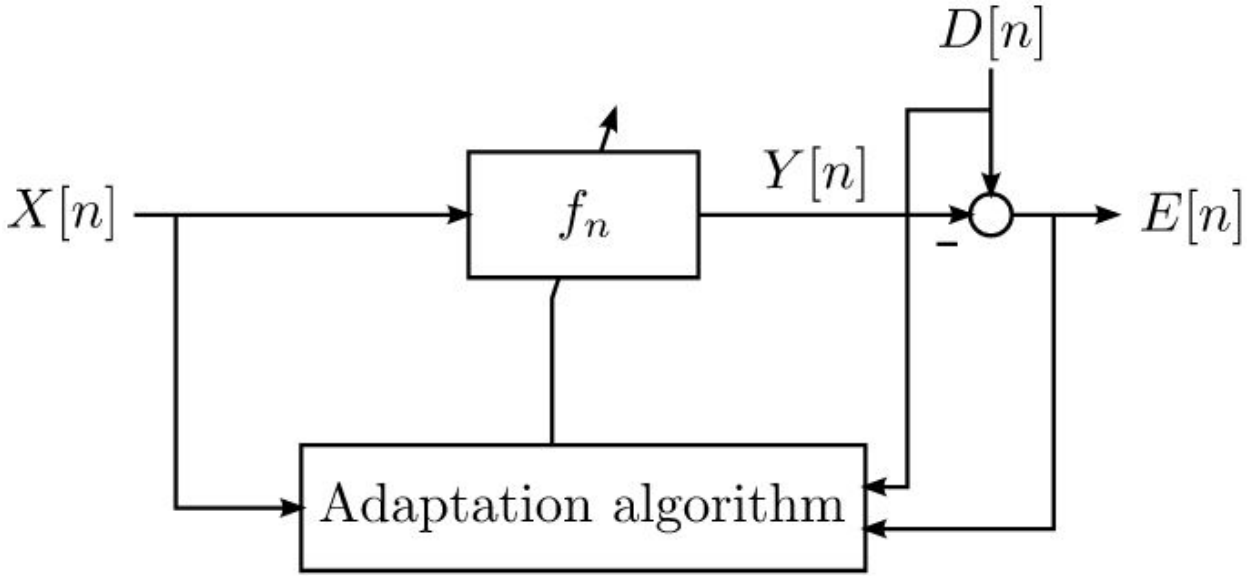
Adaptive filtering

Statistical signal and data processing through applications
COM-500
Spring 2022
Kaan Okumus, Clémence Altmeyerhenzien

The theory behind adaptive filtering



Basic tools

- LMS: standard algorithm $\mathbf{f}_{n+1} = \mathbf{f}_n + \mu \mathbf{X}_n E[n]$
- Normalized LMS : improves **convergence speed** $\mathbf{f}_{n+1} = \mathbf{f}_n + \mu \frac{\mathbf{X}_n}{||\mathbf{X}_n||^2} E[n]$
- Momentum LMS : **buffers two past versions** of the filter $\mathbf{f}_{n+1} = \mathbf{f}_n + \mu \mathbf{X}_n E[n] + \beta(\mathbf{f}_n - \mathbf{f}_{n-1})$
- Signed LMS: decrease **computation time** $\mathbf{f}_{n+1} = \mathbf{f}_n + \mu \cdot sign(\mathbf{X}_n) sign(E[n])$

Advanced tools

- Recursive Least Squares (RLS): takes into account the **previous samples**

$$\mathbf{g}_n = \frac{\mathbf{\Omega}_n \mathbf{x}_n}{\mu + \mathbf{x}_n^T \mathbf{\Omega}_n \mathbf{x}_n} \qquad \mathbf{f}_{n+1} = \mathbf{f}_n + \mathbf{g}_n e[n]$$

- Fast Block LMS: process filtration by blocks and process convolutions using the Fast Fourier Transform algorithm

- Affine Project (AP): uses multiple input vectors → effective for **correlated data**

$$\mathbf{X}_n = (\mathbf{x}_n, \ldots, \mathbf{x}_{n-L}) \qquad \mathbf{f}_{n+1} = \mathbf{f}_n + \mu \mathbf{I}_{L,n} \mathbf{X}_n^T (\epsilon \mathbf{I} + \mathbf{X}_n \mathbf{X}^T)^{-1} \mathbf{E}[n]$$

On to the demo!