

## Experiência E9: Filtragem adaptativa utilizando o algoritmo LMS

### Objetivos

- Utilização do algoritmo LMS para identificação de sistemas e filtragem inversa
- Prática na utilização do MATLAB para análise e projeto de sistemas em tempo discreto.

### Introdução

O algoritmo LMS (*Least Mean Squares*) é um procedimento de filtragem adaptativa que pode ser aplicado na solução de diversos problemas, tais como identificação de sistemas, determinação de sistemas inversos e redução de ruído. A ideia básica, apresentada na Figura 1 e no equacionamento subsequente, é a determinação dos coeficientes de um filtro FIR ( $\mathbf{w}[n]$ ) que conduzem ao mínimo erro médio quadrático entre o sinal de entrada filtrado e o sinal desejado. Para tanto, o algoritmo LMS atualiza os coeficientes do filtro com base na diferença entre o sinal desejado  $d[n]$  e sinal  $y[n]$  obtido na saída do filtro. O passo de adaptação ( $\mu$ ) utilizado para atualizar os coeficientes do filtro, cujo valor ideal está relacionado com a amplitude do sinal de entrada  $x[n]$ , deve ser escolhido para garantir um compromisso adequado entre o tempo de convergência e o erro na determinação dos pesos. Já o fator de esquecimento ( $\alpha$ ) pode ser utilizado para ponderar o peso dos valores passados na atualização dos coeficientes do filtro.

O algoritmo NLMS (LMS normalizado) é uma adaptação do algoritmo LMS que resolve o problema de sensibilidade à amplitude do sinal de entrada  $x[n]$ . Neste caso, o sinal de entrada  $x[n]$  é normalizado em relação à sua potência, fazendo com que a escolha de  $\mu$  independa da amplitude de  $x[n]$ . No caso de ausência de ruído adicionado ao sinal  $d[n]$ , o valor ótimo de  $\mu$  é igual a um para o algoritmo NLMS.

A Figura 2 apresenta a aplicação do algoritmo LMS para identificação de um sistema desconhecido, com resposta à amostra unitária  $h[n]$ . Para minimizar o erro  $e[n]$ , o algoritmo encontrará os coeficientes do filtro FIR que mais aproximam os sinais  $d[n]$  e  $y[n]$  no sentido do mínimo erro médio quadrático. Basicamente, a ordem escolhida para o filtro LMS deve ser igual ou maior do que a ordem do sistema desconhecido. Se  $h[n]$  for um sistema IIR, a ordem do filtro LMS deverá ser muito grande (teoricamente infinita) para garantir uma aproximação adequada.

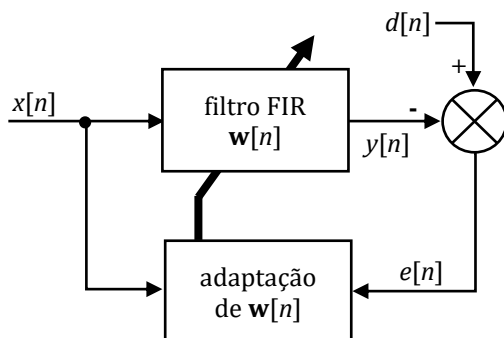


Figura 1 – Algoritmo LMS.

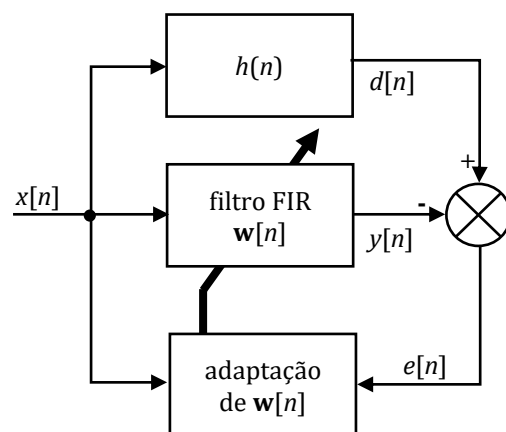


Figura 2 – Identificação de  $h[n]$ .

### Algoritmos LMS e NLMS (LMS normalizado)

$$y[n] = \mathbf{w}^T[n-1] \mathbf{x}[n] \quad (\text{filtragem com o filtro FIR } \mathbf{w}[n])$$

$$e[n] = d[n] - y[n] \quad (\text{cálculo do erro})$$

$$\mathbf{w}[n] = \alpha \mathbf{w}[n-1] + \mu e[n] \mathbf{x}[n] \quad (\text{adaptação de coeficientes - algoritmo LMS})$$

$$\mathbf{w}[n] = \alpha \mathbf{w}[n-1] + \mu e[n] \frac{\mathbf{x}[n]}{\varepsilon + \mathbf{x}^T[n] \mathbf{x}[n]} \quad (\text{adaptação de coeficientes - algoritmo NLMS})$$

$\alpha$  = leakage (fator de esquecimento,  $0 < \alpha \leq 1$ , normalmente é igual a um)

$\mu$  = passo de adaptação

$\varepsilon$  = constante para evitar divisão por zero ( $\varepsilon \ll 1$ )

### Atividade Prática (utilizando o MATLAB)

- O M-file *ExpE9.m* juntamente com a função *Funcao\_NLMS.m* utilizam o algoritmo NLMS para identificação de um sistema desconhecido, cujos coeficientes  $b$  foram pré-determinados. Com a escolha da ordem do filtro LMS e na ausência de ruído, o algoritmo deverá encontrar os mesmos coeficientes para o filtro LMS. Modificar os coeficientes do sistema desconhecido, a ordem do filtro LMS e o ruído e observar os valores dos coeficientes obtidos. **(1,0 ponto)**
- O arquivo *Teste\_xx* possui um sinal de áudio com 100 ms utilizado como referência (ruído branco). O sinal de teste foi adicionado ao início de um sinal de áudio, o qual foi filtrado por um sistema desconhecido, gerando o sinal do arquivo *Sinal\_xx*. Modificar o M-file *ExpE9.m* e utilizá-lo juntamente com a função *Funcao\_NLMS.m* para determinar o filtro inverso do sistema desconhecido e apresentar os coeficientes do filtro. Filtrar o sinal do arquivo *Sinal\_xx* com o filtro inverso e comparar com o sinal de áudio original contido no arquivo *Musica*. Utilizar os arquivos referentes ao seu grupo, especificados na tabela abaixo. **(9,0 pontos)**

Grupo	Sinal de teste	Sinal distorcido
1	Teste_01	Sinal_01
2	Teste_02	Sinal_02
3	Teste_03	Sinal_03
4	Teste_04	Sinal_04
5	Teste_05	Sinal_05
6	Teste_06	Sinal_06
7	Teste_07	Sinal_07
8	Teste_08	Sinal_08
9	Teste_09	Sinal_09