

EXERCÍCIO 2 - IDENTIFICAÇÃO DE SISTEMAS LINEARES COM RNA

BRUNO ANDREGHETTI DANTAS*

Email: brunoandreggetti@gmail.com

Abstract— The purpose of this exercise is to study and evaluate the performance of a FIR adaptive filter with an ADALINE network used to approximate a discrete time linear system.

Keywords— ADALINE, Adaptive, Signal Processing, Neural Networks.

Resumo— Este exercício visa estudar e avaliar o desempenho da aproximação de um sistema linear em tempo discreto utilizando um algoritmo adaptativo FIR com uma rede ADALINE.

Palavras-chave— ADALINE, Adaptativo, Processamento de Sinais, Redes Neurais.

1 Introdução

O algoritmo ADALINE (ADAPtive LInear NEuron) é um identificador adaptativo de resposta finita ao impulso (em inglês, FIR), ou seja, é um algoritmo capaz de identificar em tempo discreto sistemas lineares variantes no tempo.

Por ser simples e de fácil implementação, é a base para diversos algoritmos de identificação e filtragem adaptativos utilizados atualmente.

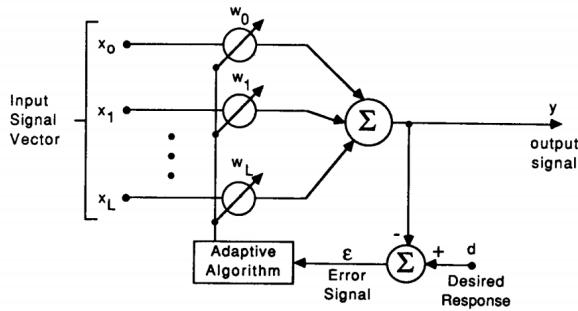


Figura 1: Diagrama de funcionamento do ADALINE.(Widrow and Winter, 1988)

Seu funcionamento é facilmente compreendido através da Figura 1. O sinal de entrada $X_0, X_1, X_2, \dots, X_L$ consiste nas respostas ao impulso do sinal de entrada ao longo de L períodos de amostragem do sistema a ser identificado. Tais entradas são respectivamente multiplicadas pelos pesos $w_0, w_1, w_2, \dots, w_L$ e então somadas para se obter a resposta y_L da rede.

O próximo passo é de cálculo do erro ϵ da identificação, subtraindo a saída da rede pela saída do sistema a ser identificado. Este erro é realimentado para ajuste dos pesos através do seguinte cálculo:

$$w_k = w_k + \mu \epsilon x_k$$

Em que μ é uma taxa de aprendizagem tal que $0 < \mu < 1$.

Deste modo, a saída da rede converge para a saída do sistema identificado adaptativamente.

2 Metodologia e Desenvolvimento

Para desenvolvimento do exercício foi utilizada a linguagem Python 3.6 com o módulo NumPy para computação algébrica, o módulo SciPy para simulação dos sistemas lineares e o módulo Matplotlib para plotagem dos dados obtidos.

O sistema a ser identificado foi gerado através da concatenação de dois sistemas lineares G_1 e G_2 , simulados através do módulo SciPy, de modo que o comportamento do sistema demonstra uma variância no tempo, ocorrida em $t = 80s$.

$$G_1 = \frac{1}{s+0.2s+1}$$

$$G_2 = \frac{3}{s+2s+1}$$

O algoritmo ADALINE foi executado com as combinações dos seguintes parâmetros:

$$\mu \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3\}$$

$$L \in \{8, 16, 32, 64, 128, 256, 512\}$$

Em que μ é a taxa de aprendizagem e L é o número de atrasos do sinal de entrada utilizados, ou seja, o número de pesos a serem aprendidos pela rede.

O algoritmo foi executado utilizando como sinal de entrada dois *Pseudo-Random Binary Signal* contidos em 1% e 5% da banda de Nyquist, respectivamente.

Para comparação, também foram obtidos os resultados da execução do ADALINE com pesos fixos na última configuração obtida na execução adaptativa e com pesos calculados algebricamente para minimização do erro, e para todas as execuções foi calculado o erro médio quadrático dos algoritmos ao longo de toda a amostragem do sinal.

3 Resultados

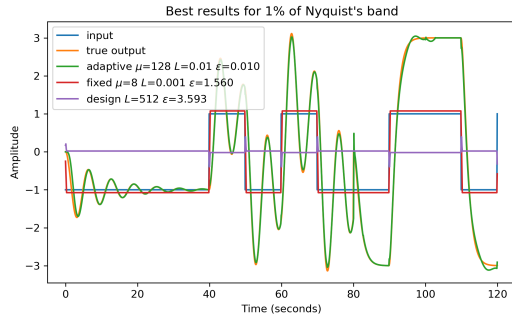


Figura 2: Melhores resultados para entrada PRBS com 1% da banda de Nyquist.

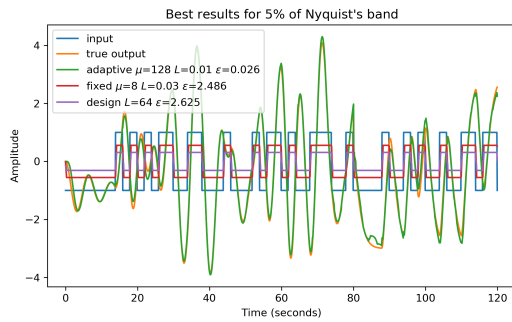


Figura 3: Melhores resultados para entrada PRBS com 5% da banda de Nyquist.

Os melhores resultados para cada sinal de entrada podem ser vistos nos gráficos das Figuras 2 e 3.

Como é possível perceber, para nenhum dos casos estudados os algoritmos que utilizam pesos fixos obtiveram desempenho sequer próximo do aceitável.

Fixando L em 128, o que obteve melhor desempenho de acordo com as Figuras 2 e 3, foi variado μ a fim de observar o efeito deste parâmetro no comportamento do algoritmo. O resultado é visto na Figura 4. É importante notar que para valores de μ iguais ou maiores a 0.03, o algoritmo diverge e portanto não estão presentes no gráfico.

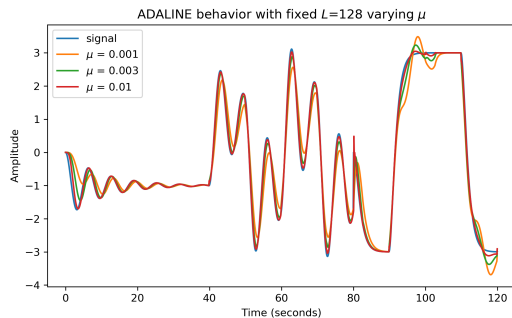


Figura 4: Comportamento do algoritmo com L fixo variando-se μ

Analogamente, fixando μ em 0.01 e variando L , foi obtido o gráfico da Figura 5. Para maior clareza da imagem, foram omitidos os gráficos de alguns dos valores de L . Novamente, para valores iguais ou maiores que 256, ocorreu divergência do algoritmo.

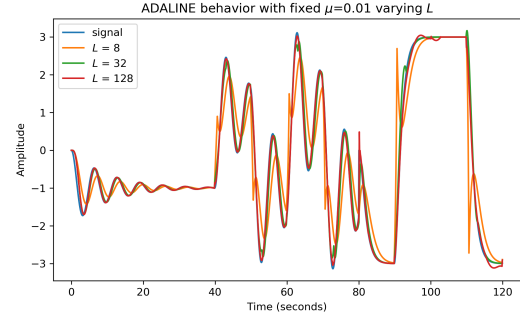


Figura 5: Comportamento do algoritmo com μ fixo variando-se L

4 Análise e Conclusões

Com base nos resultados mostrados nas Figuras 2 e 3, é fácil concluir que entre os algoritmos testados o ADALINE é o melhor para lidar com sistemas variantes no tempo. Porém, cabe observar que em muitos casos de teste ocorreu divergência do ADALINE, portanto ao utilizá-lo deve-se tomar os devidos cuidados para garantir a convergência do algoritmo.

Os efeitos dos parâmetros puderam ser observados nas Figuras 4 e 5. É importante destacar que valores altos para ambos os parâmetros tendem a ocasionar instabilidade no algoritmo.

O aumento da taxa e aprendizagem μ mostrou ter um efeito acelerador na convergência do algoritmo. Por isso, deve-se evitar valores muito pequenos de μ uma vez que o algoritmo reagirá muito lentamente a mudanças de comportamento do sistema.

De maneira similar, o aumento do número de atrasos L também acelerou a convergência do algoritmo. Porém também há um aumento na precisão da identificação, o que é desejável uma vez que para valores muito baixos de L o sinal aproximado mostra-se muito ruidoso.

Por fim, é importante destacar que embora o algoritmo estudado tenha se mostrado efetivo nos testes realizados, o ADALINE possui diversas variantes na literatura propostas a fim de solucionar muitos dos problemas operacionais observados neste exercício.

Referências

Widrow, B. and Winter, R. (1988). Neural nets for adaptive filtering and adaptive.