Toolbox de Filtragem Adaptativa para MATLAB

Breno Espindola, Markus Lima, Paulo Diniz

***Resumo*— O objetivo desse trabalho de iniciação científica (TIC) é apresentar a *toolbox* de filtragem adaptativa implementada para MATLAB. Essa *toolbox* contém todos os algoritmos apresentados em [1], bem como *scripts* de teste das funções mais básicas, isto é, aquelas que implementam os algoritmos das famílias mais conhecidas: *Least Mean Square*, *Recursive Least Squares* e *Set-Membership*.**

***Palavras-Chave*— Filtragem Adaptativa, Processamento Digital de Sinais, *Toolbox*, MATLAB.**

***Abstract*— This scientific initiation work aims to present an adaptive filtering toolbox implemented for MATLAB. This toolbox contains all the algorithms presented in [1], as well as test scripts for the most basic functions, i.e., the ones that implement the algorithms of the most well-known families: Least Mean Square, Recursive Least Squares and Set-Membership.**

***Keywords*— Adaptive Filtering, Digital Signal Processing, Toolbox, MATLAB.**

I. Introdução

Filtragem adaptativa é uma técnica muito utilizada em processamento digital de sinais para diversos fins, entre eles equalização de canal, identificação de sistema, eliminação de ruído, dentre outros. Embora filtragem adaptativa seja uma área bem consolidada, os autores desconhecem uma ferramenta de fácil utilização que implemente uma grande variedade de algoritmos adaptativos.

Dessa forma, com a finalidade de suprir essa carência, decidimos criar uma *toolbox* para MATLAB que implemente os algoritmos de filtragem adaptativa apresentados em [1]. Essa *toolbox* é disponibilizada para o público em [2].

II. A *Toolbox*

A *toolbox* desenvolvida é composta de funções que implementam algoritmos adaptativos, *scripts* de teste para as funções mais básicas, e um arquivo "*readme_toolbox*". A seguir descreveremos cada um deles:

A. Funções

Os algoritmos foram implementados em funções para MATLAB (acessadas por linha de comando) que possuem:

1) Protótipo Padronizado: O protótipo padronizado das funções simplifica a utilização da *toolbox*. Pois tendo aprendido a utilizar uma delas a utilização das outras é análoga.

Para efeito de ilustração, consideraremos o protótipo da função NLMS:

```
[outputVector,...
 errorVector,...
 coefficientVector] = NLMS(desired,input,S)
```

Todas as funções possuem essas 3 entradas: os vetores linha *desired* e *input* representando o sinal desejado e o de entrada do filtro adaptativo, respectivamente, e a estrutura *S* composta por parâmetros específicos de cada algoritmo. Além disso, praticamente todas as funções possuem os mesmos 3 parâmetros de saída: *outputVector* e *errorVector* são vetores coluna que representam, respectivamente, a saída do filtro adaptativo e o sinal de erro, ambos ao longo das iterações, e a matriz *coefficientVector* que guarda os coeficientes do filtro adaptativo ao longo das iterações (cada coluna representa os coeficientes do filtro em uma dada iteração).

2) Help: A existência de uma documentação, acessada através do commando *help*, que explica a utilização da função e cada um dos seus parâmetros de entrada e saída, torna a utilização da *toolbox* ainda mais simples.

B. Scripts de Teste

Os *scripts* de teste são programas, feitos em MATLAB, que ilustram a utilização das funções mais básicas dessa *toolbox* em um cenário prático. Com isso, além de ser possível comparar o desempenho dos algoritmos, o usuário tem em mãos um *script* que pode auxiliá-lo nos primeiros trabalhos.

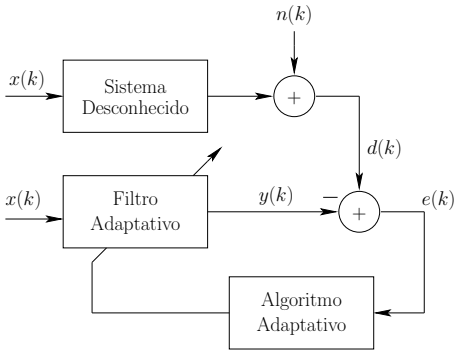

Fig. 1. Identificação de sistema.

O cenário utilizado nos *scripts* é o de identificação de sistema, representado na figura 1, cujo objetivo é estimar a resposta ao impulso do sistema desconhecido.

Com essa finalidade, um sinal de entrada $x(k)$ é introduzido simultaneamente no sistema desconhecido e no filtro adaptativo[1] e, em seguida, deseja-se comparar a saída dos dois

Breno Espindola, Markus Lima, Paulo Diniz, Departamento de Engenharia Eletrônica e de Computação, Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil. E-mail: brenoespindola@lps.ufrj.br, markus@lps.ufrj.br, diniz@lps.ufrj.br. Este trabalho foi financiado pelo CNPq.

[1]Assumimos que o sistema desconhecido e o filtro adaptativo são filtros FIR (*Finite Impulse Response*) de mesma ordem. Dessa forma, eliminamos o erro de modelagem do sistema.

sistemas. Na prática, o sinal de saída do sistema desconhecido estará sempre associado a algum tipo de ruído (de quantização, imprecisão de medida, dentre outros) aditivo $n(k)$ de forma que o sinal desejado $d(k)$ será dado pela soma de ambos. Por fim, calcula-se um sinal de erro $e(k) = d(k) - y(k)$, onde $y(k)$ é a saída do filtro adaptativo. Esse erro representa o quão distante os coeficientes do filtro adaptativo estão, comparados com os do sistema desconhecido. Portanto, para que ao final do processo os coeficientes do filtro adaptativo sejam uma boa aproximação dos coeficientes do sistema desconhecido, o algoritmo adaptativo deve minimizar esse erro (ou uma função do erro)[2].

O algoritmo adaptativo, então, nada mais é do que uma regra, baseada em conceitos de otimização, que governa a atualização dos coeficientes do filtro adaptativo.

Exibimos os resultados dos *scripts* em gráficos que mostram a evolução do primeiro coeficiente do filtro adaptativo e do erro quadrático médio (*Mean Square Error* - MSE) ao longo das iterações. Para o caso do *Set-Membership* também disponibilizamos o número médio de atualizações realizadas.

C. O Arquivo "readme_toolbox"

Esse arquivo contém orientações para usuários que estão utilizando a *toolbox* pela primeira vez, facilitando assim a familiarização com a ferramenta.

III. Simulações e Resultados

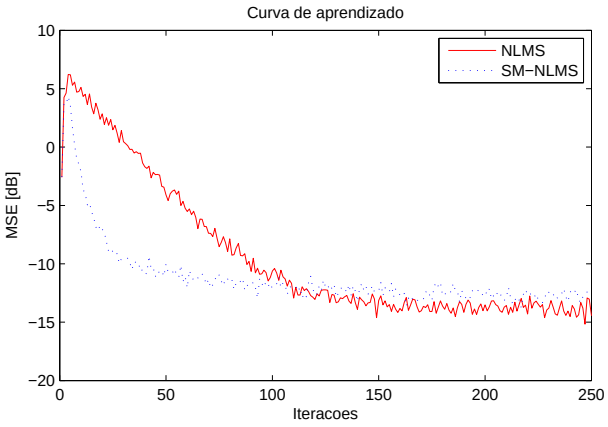

Fig. 2. Evolução do Erro Quadrático Médio ao Longo das Iterações

Os gráficos presentes nessa seção são relativos aos *scripts* utilizando os algoritmos *Normalized Least Mean Square* (NLMS) e *Set-Membership* NMLS (SM-NLMS), respectivamente. Nesses *scripts* foram utilizados parâmetros iguais (sempre que possível) para termos uma comparação coerente. Esses parâmetros são apresentados a seguir:

- Número de realizações: 100
- Número de iterações: 250
- Resposta ao impulso do sistema desconhecido: $[\,0.32+0.21j, -0.30+0.70j, 0.50-0.80j, 0.20+0.50j\,]^T$

[2]Assumimos que o sinal de entrada é descorrelacionado com o ruído.

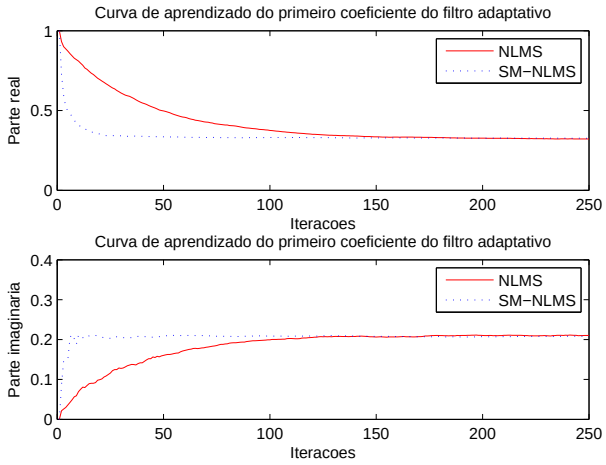

Fig. 3. Evolução do Primeiro Coeficiente do Filtro Adptativo ao Longo das Iterações

- Variância do ruído: 0.04
- Número de coeficientes do filtro adaptativo: 4
- Fator de regularização: 10^{-12}
- Vetor de coeficientes iniciais do filtro adaptativo: $[1, 1, 1, 1]^T$

Além dos parâmetros acima, escolhemos o passo de atualização do NLMS igual a 0.1. Já para o SM-NLMS, esse passo é calculado pelo algoritmo a cada iteração e depende do parâmetro $\bar{\gamma}$ (limite superior do erro).

Observando os gráficos podemos notar que o SM-NLMS converge mais rapidamente, porém para um valor mais elevado de MSE. O número médio de atualizações realizadas pelo algoritmo SM-NLMS foi aproximadamente 25 (num total de 250 iterações) o que corresponde a 10% do tempo.

Podemos notar também que o valor de convergência do MSE dos dois algoritmos ficou próximo a -14 dB. Isto se dá porque para um ruído de variância 0.04 o menor MSE possível é de -13.98 dB.

IV. Conclusões

Filtragem adaptativa é uma área bastante consolidada. Por isso é importante termos uma ferramenta para auxiliar no momento de sua utilização. O desenvolvimento de uma *toolbox* para MATLAB ajudará bastante alunos e pesquisadores na utilização dessa técnica.

Agradecimentos

Agradecemos aos pioneiros deste projeto, listados em [2], ao Laboratório de Processamento de Sinais (LPS) da Universidade Federal do Rio de Janeiro, local onde essa *toolbox* foi desenvolvida, ao CNPq que possibilitou o andamento do projeto, e ao SBrT e organizadores do Simpósio que tornaram possível uma maior divulgação desse projeto.

Referências

[1] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation* , 3rd ed. Boston, MA: Springer, 2008.

[2] G. O. Pinto, M. V. S. Lima, W. A. Martins, L. W. P. Biscainho e P. S. R. Diniz, "Adaptive Filtering Toolbox", http://www.lps.ufrj.br/∼markus.