

Filtro Savitzky-Golay

João Felipe Amaral Santiago

Pós Graduação em Engenharia Elétrica, Universidade Federal de Santa Maria
Santa-Maria, RS

joaofelipe.santiago@ecom.ufsm.br

Resumo—No contexto da espectroscopia e na química analítica em geral, o filtro de Savitzky-Golay é largamente utilizado. Ele foi concebido com o intuito de suavizar curvas as quais foram previamente registradas em algum elemento de memória e expostas em ambiente com ruído significativo. Seu princípio de funcionamento consiste na divisão de todos os dados em janelas de análise com um conjunto finito de pontos. Para cada janela aplica-se o método numérico de mínimos quadrados de ordem arbitrária. Após conhecer os coeficientes da janela, atribui-se à imagem do elemento do domínio central de acordo com a equação geral dada pelo método numérico. Para atribuir o valor do próximo elemento repete-se o procedimento, porém agora para um novo conjunto, criando assim uma janela móvel.

Keywords— janela, interpolação, filtro, suavização

I. INTRODUÇÃO

Matematicamente o filtro Savitzky-Golay pode ser interpretado como interpolações numéricas sucessivas. Originalmente [1], os autores exploraram o método de mínimos quadrados para isso. Uma interpolação é tal que um conjunto de pontos pode ser descrito por uma equação de curva, como por exemplo $y = a_3x^3 + a_2x^2 + a_1x + a_0$

de modo que os coeficientes a_n precisam se aproximar o máximo possível de todas as ordenadas. Vale ressaltar que o filtro S-G pode ser classificado como não causal, uma vez que antes de atribuir um valor à imagem da função, as etapas de interpolação dependem dos dados passados e futuros em relação ao elemento central do subconjunto.

II. MATERIAIS E MÉTODOS

Para demonstrar a aplicação do filtro, criou-se um *script* na ferramenta Octave. Primeiramente, gerou-se uma função espectral ideal, a qual variava de 400 nm até 2500 nm, possuindo picos de energia em 500, 100, 1500 e 2000 nm, os quais foram gerados com o auxílio de uma função gaussiana somada à uma senoide. Em seguida, adicionou-se ruído à função por meio da função *radn()*.

Do ponto de vista computacional, o filtro pode ser visualizado como uma função cuja a quantidade de argumentos é dois (tamanho da janela móvel e ordem da interpolação). Portanto, uma possível abordagem é tal que de início calcula-se a quantidade de janelas que serão necessárias, considerando apenas a quantidade de pontos existentes nela e a quantidade total de elementos no conjunto. Conhecendo a quantidade de subconjuntos necessários pode-se, a partir de uma varredura completa de todos os elementos da função

(domínio e imagem); atribuir em duas matrizes os pontos de cada janela associados aos seus respectivos domínios.

O próximo passo consiste em aplicar interpolações nas janelas de tal modo que seja possível conhecer a equação que descreve a curva do subconjunto. Uma possível alternativa é a aplicação de regressão polinomial. Sabe-se que tal regressão sempre culmina na necessidade de se resolver um sistema linear. Com o auxílio da função nativa do Octave chamada *polifyfit()*, pode-se facilmente apenas passar a janela e a ordem do polinômio como parâmetro. A saída da função são os coeficientes resultantes do sistema linear.

Por fim, com os coeficientes estimam-se os valores que compoem a função espectral suavizada. Para isso utiliza-se, em cada ponto central, de cada janela; a equação de ordem n gerada no passo anterior.

O filtro Savitzky Golay, além de ser um suavizador de curvas permite o cálculo da derivada espectral a partir de um conjunto de procedimentos relativamente simples. Segundo [1], a n -ésima derivada de um ponto central de uma janela é dado pela equação 1, onde n é a ordem da derivada, f_c o ponto central da janela e d_n o coeficiente que corresponde à posição n .

$$\frac{d^n f_c}{dx^n} = n! d_n \quad (1)$$

A implementação da equação 1 consistiu-se na criação de uma função chamada *DerivadaSavitzkyGolay()*, cujos argumentos eram coeficientes e ordem de derivação.

III. RESULTADOS E DISCUSSÕES

O *script* Octave implementado inicia-se com a definição dos comprimentos de onda, em seguida define-se os picos, suas magnitudes máximas e magnitudes máximas dos demais comprimentos de onda. Na sequência, a função espectral é tal que respeita a função gaussiana dada pela equação 2, onde a representa a amplitude de cada pico, c o desvio padrão, b o pico da abscissa e x , o comprimento de onda associado ao ponto máximo.

$$f(x) = a \cdot e^{(-0.5 \cdot c^{-2} \cdot (x-b)^2)} \quad (2)$$

O próximo passo consistiu em criar uma função senoidal de amplitude 0.2 e frequência 5 Hz (considerando, idealmente que de 400 nm até 2500 nm há uma duração total de 1 segundo). Por fim, para criar um função espectral arbitrária,

somou-se a senoide com as curvas gaussianas e adicionou-se ruído aos dados.

Após a geração da função espectral ruidosa definiu-se os parâmetros de tamanho do janelamento, ordem de interpolação, ordem da derivada, suavização. Um outro parâmetro adicional, cuja menção é feita em [1] diz respeito à resuavização dos dados. Eles destacam que esse procedimento enriquece a relação sinal ruído, uma vez que o ruído é reduzido aproximadamente pela raiz quadrada do número de pontos usados. Nos resultados obtidos o sinal foi resuavizado 100 vezes.

A figura 1 apresenta a função espectral ideal. Já a figura 2 ilustra o efeito do ruído no sinal. A figura 3 demonstra a capacidade da função *FiltroSavitzkyGolay()* em reconstruir o sinal. A título de comparação, verifica-se na figura 4 o resultado obtido pela função *sgolayfilt()*, a qual é a solução interna fornecida pelo Octave. Percebe-se que ambos os resultados são bastante similares, com exceção das extremidades das funções suavizadas da função construída, que nesse caso utilizou de janelas com tamanho fixo para toda a extensão de dados, que por sua vez prejudicou as estimativas criadas, uma vez que as interpolações dessas posições levaram em consideração dados que não estavam presentes no conjunto. Talvez a função interna do Octave trabalhe com janelas com tamanhos dinâmicos nas extremidades com o intuito de retardar tal efeito.

As figuras 5, 6 e 7 relacionam-se com a derivada do sinal. As figuras indicam que a função *DerivadaSavitzkyGolay()* apresentou erro significativo quando comparada com o resultado obtido pela função *diff()* (função nativa).

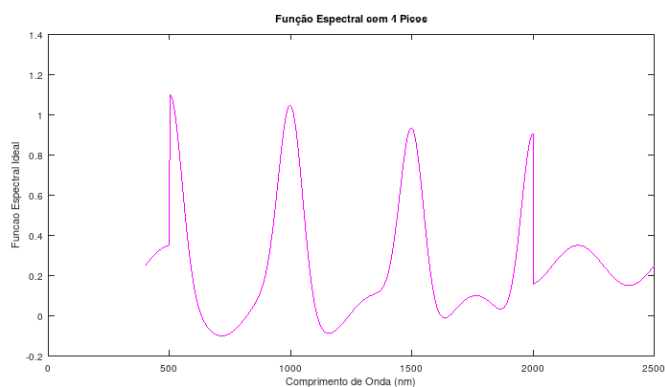


Figura 1. Função Espectral Ideal.

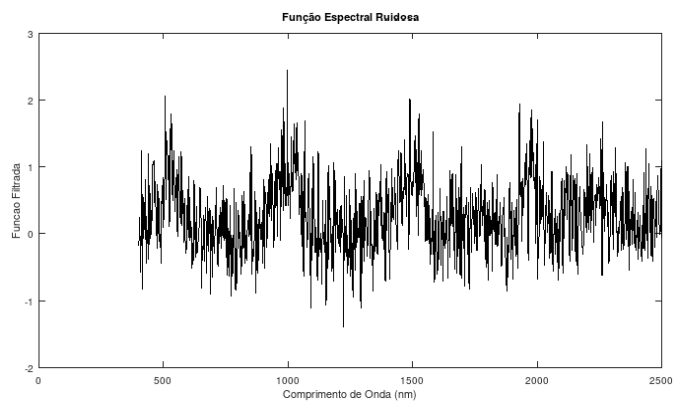


Figura 2. Função Espectral Ruidosa.

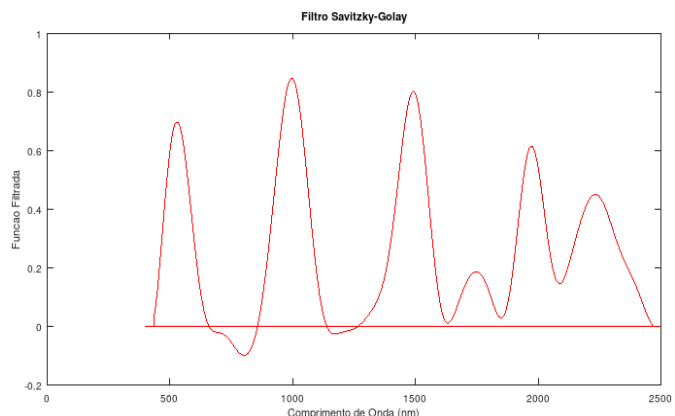


Figura 3. Reconstrução do Sinal com Função Própria.

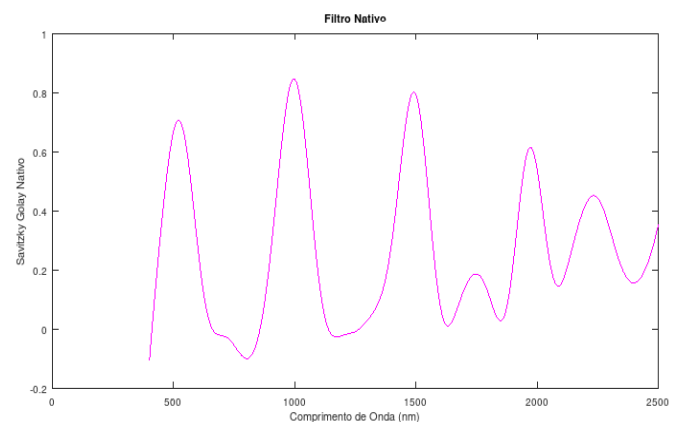


Figura 4. Suavização Via *sgolayfilt()*

Todos os scripts desenvolvidos neste trabalho podem ser encontrados e reproduzidos em [2].

REFERENCES

- [1] Savitzky, A., & Golay, M. J. E. (1964). *Smoothing and Differentiation of Data by Simplified Least Squares Procedures*. *Analytical Chemistry*, 36(8), 1627–1639. doi:10.1021/ac60214a047
- [2] Santiago, J (2023). *Repositório Filtro Savitzky Golay*. <https://github.com/JaoIndio/SavGolayFilter>

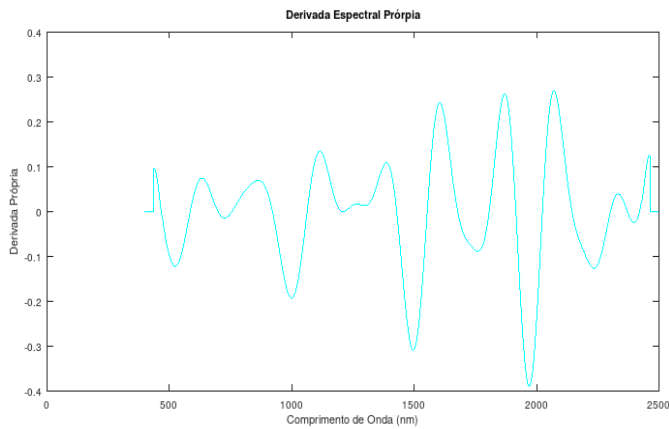


Figura 5. Derivada Própria

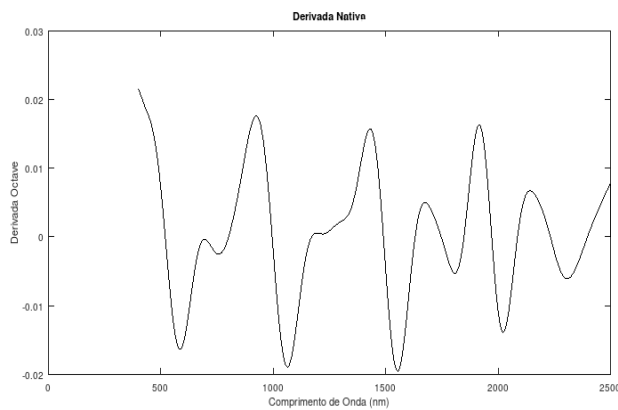


Figura 6. Derivação com *diff()*.

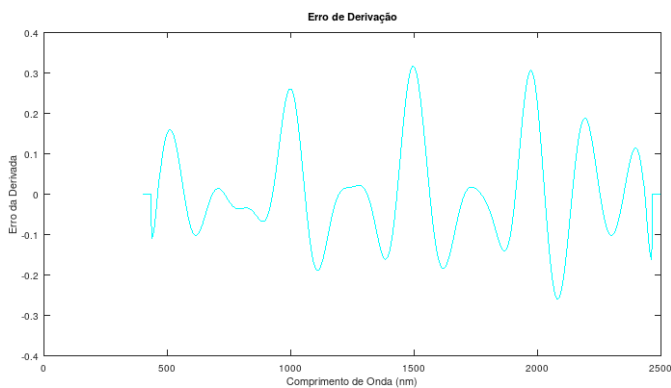


Figura 7. Erro de Derivação.

IV. CONCLUSÃO

A partir das funções *DerivadaSavitzkyGolay()* e *FiltroSavitzkyGolay()* pode-se com sucesso reproduzir os resultados obtidos por [1]. Tais sub-rotinas permitem observar empiricamente os efeitos da modificação do tamanho da janela, ordem de interpolação e suavização no que tange a redução dos efeitos do ruído no sinal original, atenuação dos picos e riqueza de curvas. Isso significa dizer que o emprego adequado do filtro depende fortemente de análises prévias do sinal que pretende-se estudar, uma vez que tais parâmetros podem enriquecer ou empobrecer o resultado final.