|  |  |
| --- | --- |
| ­  CEFET-logo | **CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA**  **DE MINAS GERAIS CAMPUS III - LEOPOLDINA**  **Disciplina:** Tópicos Especiais em PDS aplicados ao Controle **Prática:** 01  **Professor:** Janison Rodrigues de Carvalho  **Aluno:** Lucas Guimarães da Rocha |
| **1. Tópico:** Operações básicas com sequências e filtro de média móvel.  **2. Objetivo:** Realizar operações básicas com sinais de tempo discreto (sequências) e implementar o primeiro filtro digital estudado, o filtro de média móvel.    **3. Desenvolvimento:**  **Parte 1.** O filtro ou sistema de média móvel (MAF – *Moving-Average Filter*) é recorrentemente utilizado em aplicações de processamento digital de sinais, com o intuito de reduzir ruídos em sinais discretos e/ou digitais. Um exemplo desta aplicação é apresentado na Figura 1.    **Figura 1.** Redução de ruído em sinal utilizando o MAF  Esse sistema calcula a média aritmética de amostras,  de um sinal para produzir a -ésima saída do sinal . Matematicamente,  **Parte 2.** Implementar no SCILAB um filtro de média móvel de ordem *M* para filtrar os dados disponíveis na WEB referentes à pandemia COVID-19 no Brasil: sinal de número de casos e sinal de número de óbitos.   1. Apresentar o resultado em forma de gráficos (curva original e curva filtrada) para *M*=7, ou seja, uma média tomada para o período de uma semana.      1. Aumentar a ordem do filtro (parâmetro *M*), apresentar os resultados em formato gráfico e realizar uma análise comparativa com o resultado do item anterior.   Comparando as duas Figuras a seguir, utilizando M = 14, comparadas com as suas respectivas apresentadas na questão 3 com M = 7 pode-se observar que o resultado gerado para M =14 foi ainda mais suavizado, mantendo uma maior linearidade e convergência de valores ao longo dos dias.     1. Coloque seu programa aqui:   clear  close  clc  calcAndPlotConfirmedCasesApplyingMovingAverage(7);  calcAndPlotDeathsCasesApplyingMovingAverage(7);  calcAndPlotConfirmedCasesApplyingMovingAverage(14);  calcAndPlotDeathsCasesApplyingMovingAverage(14);  function [data] = getDataIntegersFromText(url, delimiter)  data = textread(url, "%d", "delimiter", delimiter);  end  function [dataFiltered] = applyMovingAverageFilter(data, slidingWindowLength)  M = slidingWindowLength;  slidingWindow = zeros(1, M);  N = length(data);  dataFiltered = zeros(1, N);  for i=1:N  slidingWindow(M) = data(i);  dataFiltered(i) = mean(slidingWindow);  slidingWindow(1:(M-1)) = slidingWindow(2:M);  end  end  function [] = calcAndPlotConfirmedCasesApplyingMovingAverage(slidingWindowLength)  data = getDataIntegersFromText("casos\_confirmados.txt", "\n");  dataFiltered = applyMovingAverageFilter(data, slidingWindowLength);  figure();  hold on  plot(data, "b", "linewidth", 2);  plot(dataFiltered, "k", "linewidth", 3);  legend("Confirmados", strcat("Óbitos [Média Móvel M = ", int2str(slidingWindowLength), "]"));  xlabel("Dias");  ylabel("Número de casos diários");  hold off  end  function [] = calcAndPlotDeathsCasesApplyingMovingAverage(slidingWindowLength)  data = getDataIntegersFromText("obitos.txt", "\n");  dataFiltered = applyMovingAverageFilter(data, slidingWindowLength);  figure();  hold on  plot(data, "b", "linewidth", 2);  plot(dataFiltered, "k", "linewidth", 3);  legend("Confirmados", strcat("Óbitos [Média Móvel M = ", int2str(slidingWindowLength), "]"));  xlabel("Dias");  ylabel("Óbitos diários");  hold off  end  **Parte 3.** Deduzir a forma recursiva do filtro de média móvel e implementar essa equação no programa desenvolvido.   1. Apresentar o resultado da forma recursiva considerando M=7 e comparar com o resultado do item 1 da Parte 2.   Comparando os resultados do item 1 da Parte 2 com os resultados mostrados nas figuras abaixo pode-se notar que foi alcançado o mesmo resultado aplicando a média móvel na forma recursiva.  Para comprovar que a forma recursiva possibilita de fato um desempenho melhor foi mensurado o tempo de execução dos cálculos no método sem recursividade e com recursividade para M = 50 para os dados de casos confirmados.  - Sem recursividade, M=50, dados de casos confirmados: 0,039390 seg.  - Com recursividade, M=50, dados de casos confirmados: 0,004253seg.     1. Coloque seu programa aqui:   OBS: Só mudei a função que faz o cálculo  function [dataFiltered] = applyMovingAverageFilterRecursively(data, slidingWindowLength)  M = slidingWindowLength;    N = length(data);  dataFiltered = zeros(1, N);    for i=1:N  if (i == 1)  dataFiltered(i) = data(i)\*(1/M);  elseif (i <= M)  dataFiltered(i) = dataFiltered(i - 1) + (data(i)\*(1/M));  else  dataFiltered(i) = dataFiltered(i - 1) + (data(i) - data(i - M))\*(1/M);  end  end  end  **Considerações Finais:** Discorras as conclusões sobre a atividade prática 01.  A prática possibilitou o aprendizado teórico e prático sobre o filtro de média móvel, que age atenuando os sinais considerando um determinado tamanho da média que deve ser definido. No final o resultado permite visualizar um sinal mais suavizado.  Foi aprendido também como deduzir e implementar o filtro de média móvel na forma recursiva e como essa redução no número de operações matemáticas influenciam no desempenho do processamento do sinal.  Bom trabalho! | |