

# Lista de exercícios sobre filtros de software

Cristiano Silva Júnior

23 de Junho de 2017

## 1 Filtro de média

### 1.1 Mostrar a derivação da forma recursiva.

A partir da definição da média

$$\bar{x}_k = \frac{1}{k} \sum_{i=1}^k x_i$$

Podemos fazer a seguinte manipulação

$$\begin{aligned}\bar{x}_k &= \frac{1}{k-1} \sum_{i=1}^{k-1} x_i + \frac{x_k}{k} \\ \bar{x}_k &= \left( \frac{1}{k-1} \sum_{i=1}^{k-1} x_i \right) \frac{k-1}{k} + \frac{x_k}{k}\end{aligned}$$

Como

$$x_{k-1}^- = \frac{1}{k-1} \sum_{i=1}^{k-1} x_i$$

Então

$$\bar{x}_k = x_{k-1}^- \frac{k-1}{k} + \frac{x_k}{k} \square$$

### 1.2 Implementação em MATLAB

Para implementar o filtro dado em MATLAB, foi escrito o seguinte script:

```
function [mediaAtual] = media1(a)
% Calcula a média dos valores dados.
% 'a' é o valor atual.
%
persistent k; % número de amostras
persistent m; % média

if isempty(k)
    m = 0;
    k = 0;
end

k = k+1;
m = (k-1)*m/k + a/k;
mediaAtual = m;
```

Para gerar os dados aleatórios de altitude, a seguinte função foi escrita:

```
function [tempo altitude] = gerarEntrada()  
% Gera os dados para serem usados nos filtros. Tempo em segundos e altitude em  
% metros.  
%  
tempo = 1:5:10000;  
altitude = [ ];  
for t = 1:length(tempo)  
    altitude(t) = 11000 + random('norm', 0, 4);  
end
```

Essas duas funções foram consolidadas em um script principal:

```
% Gerando dados de entrada  
[tempo altitude] = gerarEntrada();  
limite = length(tempo);  
mediaAtual = [ ];  
for n = 1:limite  
    mediaAtual(end+1) = media1(altitude(n));  
end  
% Plotando os gráficos  
figure;  
hold on;  
plot(tempo, altitude, 'r');  
plot(tempo, mediaAtual, 'b');  
hold off;
```

A partir deste script, foi gerado o gráfico 1, onde a onda vermelha são os dados gerados aleatoriamente e em azul, temos a média calculada.

## 2 Filtro de média móvel

### 2.1 Derivação da fórmula recursiva

Seguindo da definição de média móvel com janela de tamanho  $n$ ,

$$\bar{x}_k = \frac{x_{k-n+1} + x_{k-n+2} + \dots + x_{k-1} + x_k}{n}$$

É possível afirmar que

$$\bar{x}_k = \frac{x_{k-n}}{n} + \frac{x_{k-n+1} + x_{k-n+2} + \dots + x_{k-1} + x_k}{n} - \frac{x_{k-n}}{n}$$

$$\bar{x}_k = \bar{x}_{k-1} + \frac{x_k}{n} - \frac{x_{k-n}}{n} \square$$

### 2.2 Implementação em MATLAB

Para implementar o filtro dado em MATLAB, foi escrito o seguinte script:

```
function [mediaAtual] = media2(a)  
% Calcula a média móvel dos valores dados.  
% 'a' é o valor atual.
```

```

%
persistent m; % média
persistent n; % tamanho da janela
persistent q; % fila de dados

if isempty(q)
    m = 0;
    n = 100;
    q = java.util.LinkedList;
end

q.add(a);
if q.size < n
    m = media1(a);
else
    m = m + (a - q.pop())/n;
end

mediaAtual = m;

```

Reaproveitando a entrada gerada na função *gerarEntrada()*, foi possível gerar o gráfico na figura 2.

## 3 Filtro de média móvel ponderada

### 3.1 Derivação da fórmula recursiva

Uma média móvel ponderada é definida por

$$\bar{x}_k = \alpha x_{k-1}^- + (1 - \alpha)x_k$$

onde  $\alpha$  é um número real entre 0 e 1. Note que, se calcularmos  $x_{k-1}^-$ , temos somente uma mudança de índices para realizar:

$$x_{k-1}^- = \alpha x_{k-2}^- + (1 - \alpha)x_{k-1}$$

O mesmo acontece com  $x_{k-2}^-$ :

$$x_{k-2}^- = \alpha x_{k-3}^- + (1 - \alpha)x_{k-2}$$

Substituindo as médias anteriores na primeira equação:

$$x_{k-1}^- = \alpha(\alpha x_{k-3}^- + (1 - \alpha)x_{k-2}) + (1 - \alpha)x_{k-1}$$

$$\bar{x}_k = \alpha(\alpha(\alpha x_{k-3}^- + (1 - \alpha)x_{k-2}) + (1 - \alpha)x_{k-1}) + (1 - \alpha)x_k$$

$$\bar{x}_k = \alpha^3 x_{k-3}^- + \alpha^2(1 - \alpha)x_{k-2} + \alpha(1 - \alpha)x_{k-1} + (1 - \alpha)x_k \square$$

### 3.2 Implementação em MATLAB

Para implementar o filtro dado em MATLAB, foi escrito o seguinte script:

```

function [mediaAtual] = media3(x)
% Calcula a média dos valores dados.
% 'x' é o valor atual.
%

```

```
persistent m; % média
persistent a; % parâmetro alfa
```

```
if isempty(m)
    m = 0;
    a = 0.7;
end
```

```
m = a*m + (1-a)*x;
mediaAtual = m;
```

Reaproveitando a entrada gerada na função *gerarEntrada()*, foi possível gerar o gráfico na figura 3.

### 3.3 Discussão

Nota-se que o filtro de média regular manteve sua média sempre no valor verdadeiro, haja visto que o ruído foi simulado para que ele obedecesse um comportamento gaussiano ao redor da altitude de 11000 metros. Isso pode parecer que este é o comportamento ideal de um filtro, mas não suporta bem as características de um sistema real.

Os outros dois filtros (de média móvel e média ponderada) funcionaram melhor neste sentido. Embora ambos estejam dependentes da determinação de um parâmetro (o tamanho da janela no caso do móvel; e o parâmetro  $\alpha$  no caso do ponderado), eles atuam como filtros passa-baixa, acompanhando bem as oscilações do modelo real.

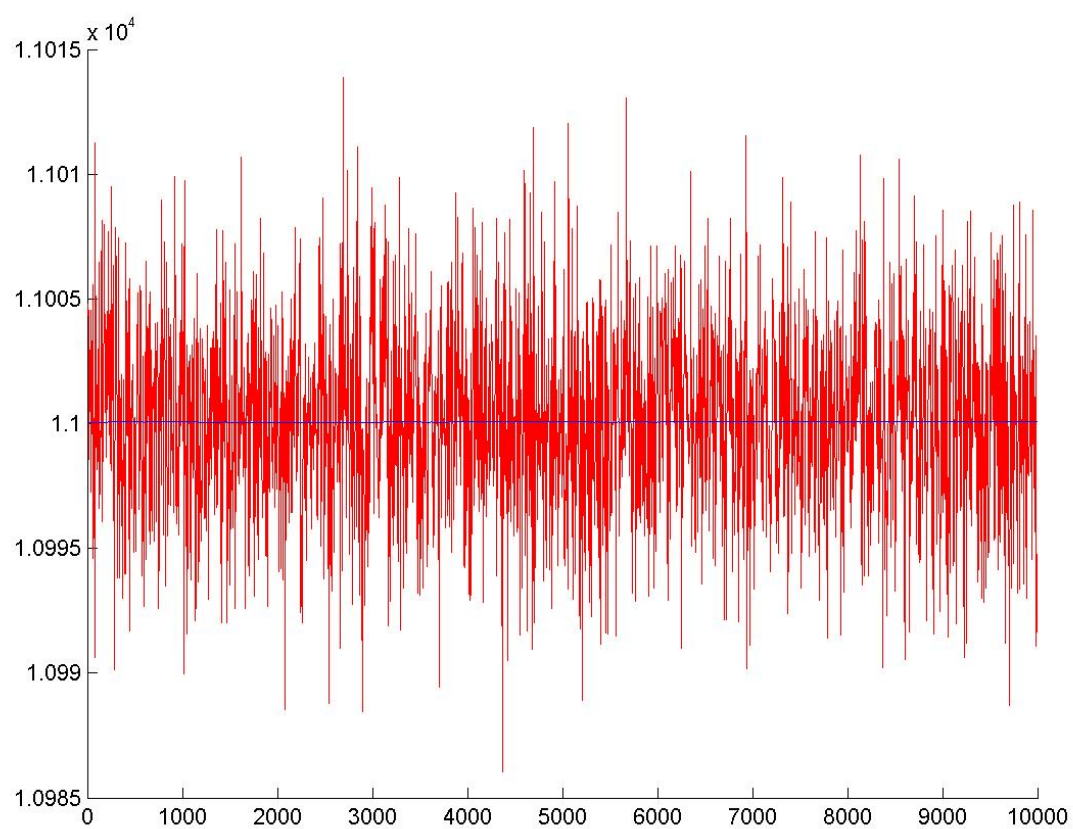


Figure 1: Figura 1. Gráfico da média regular

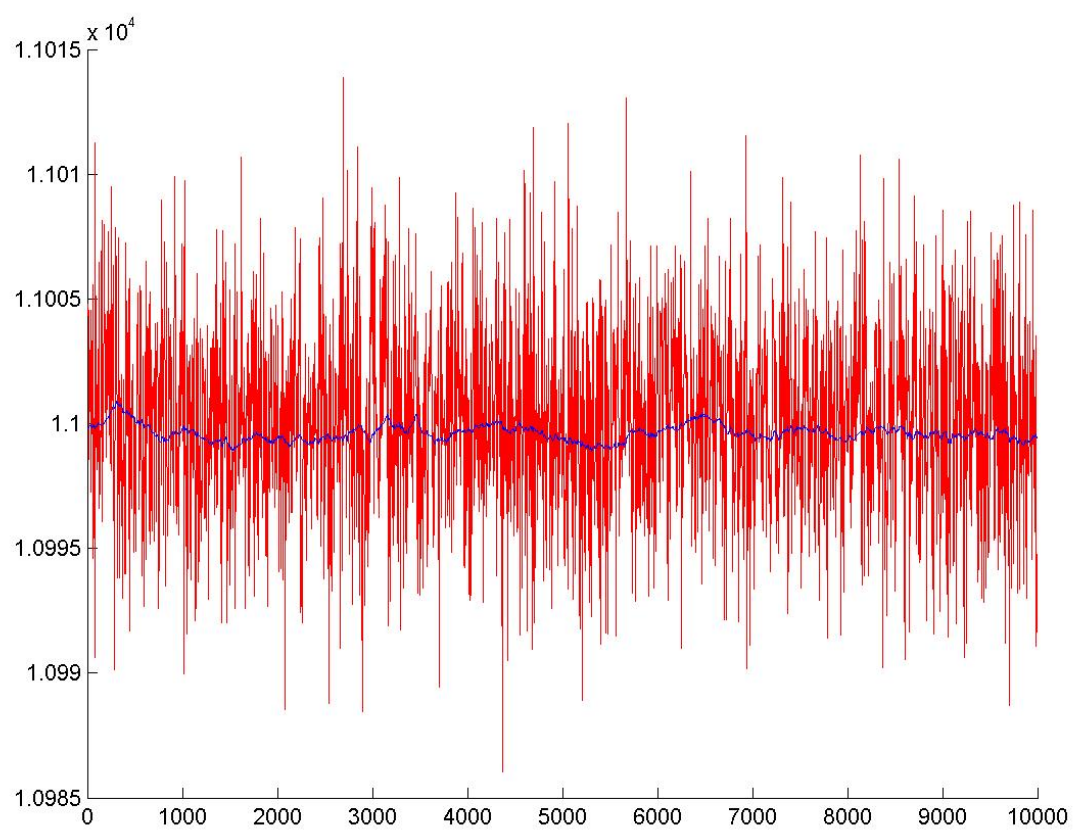


Figure 2: Figura 2. Gráfico da média móvel

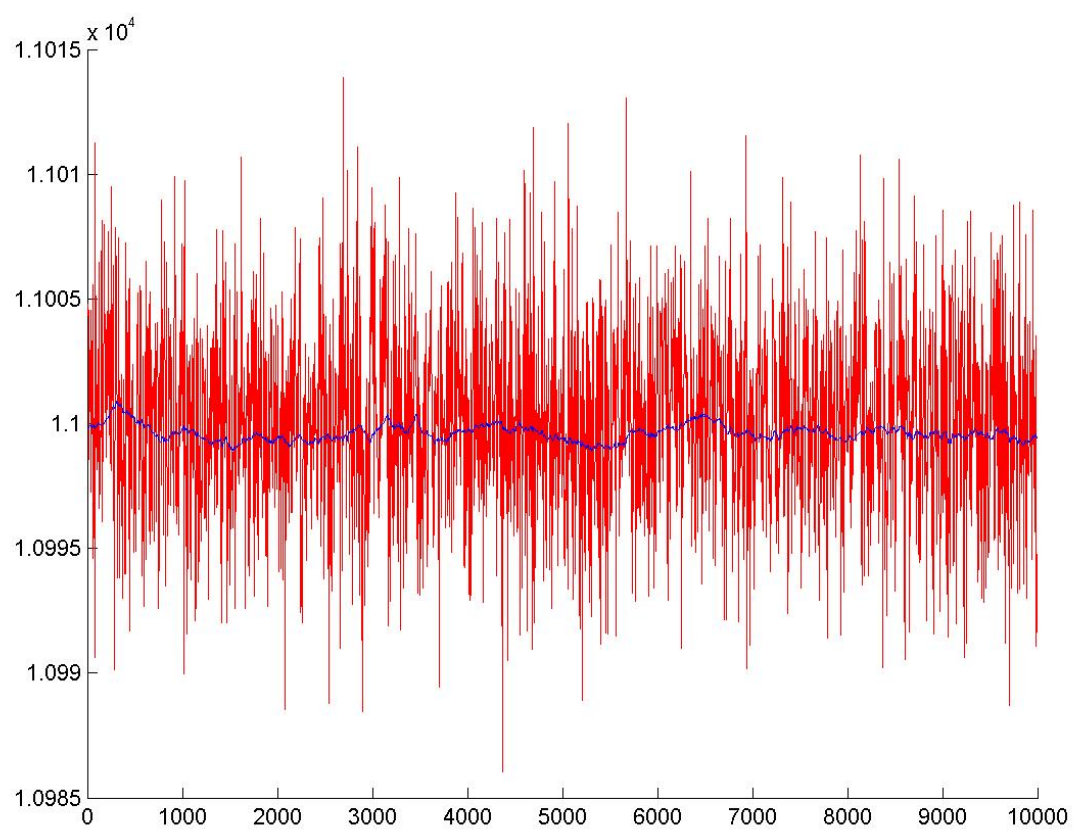


Figure 3: Figura 3. Gráfico da média ponderada