Processamento Digital de Sinais

Filtros de Média Movente

Prof. Dr. Carlos Alberto Ynoguti

Características

- É o filtro ótimo para a tarefa de remover ruído aleatório de um sinal, e manter uma resposta a degrau aguda.
- Ainda assim, é o filtro mais simples de ser implementado.
- Tem um desempenho péssimo no domínio da freqüência, embora existam filtros relativos a este, com um desempenho não tão ruim.

Implementação por convolução

- O filtro de média movente opera realizando médias de um número de pontos do sinal de entrada para produzir cada amostra no sinal de saída.

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

- Exemplo: para M=5

$$y[80] = \frac{x[80]+x[81]+x[82]+x[83]+x[84]}{5}$$

Observações

- Alternativamente, o grupo de pontos do sinal de entrada pode ser escolhido simetricamente ao redor do ponto de saída:

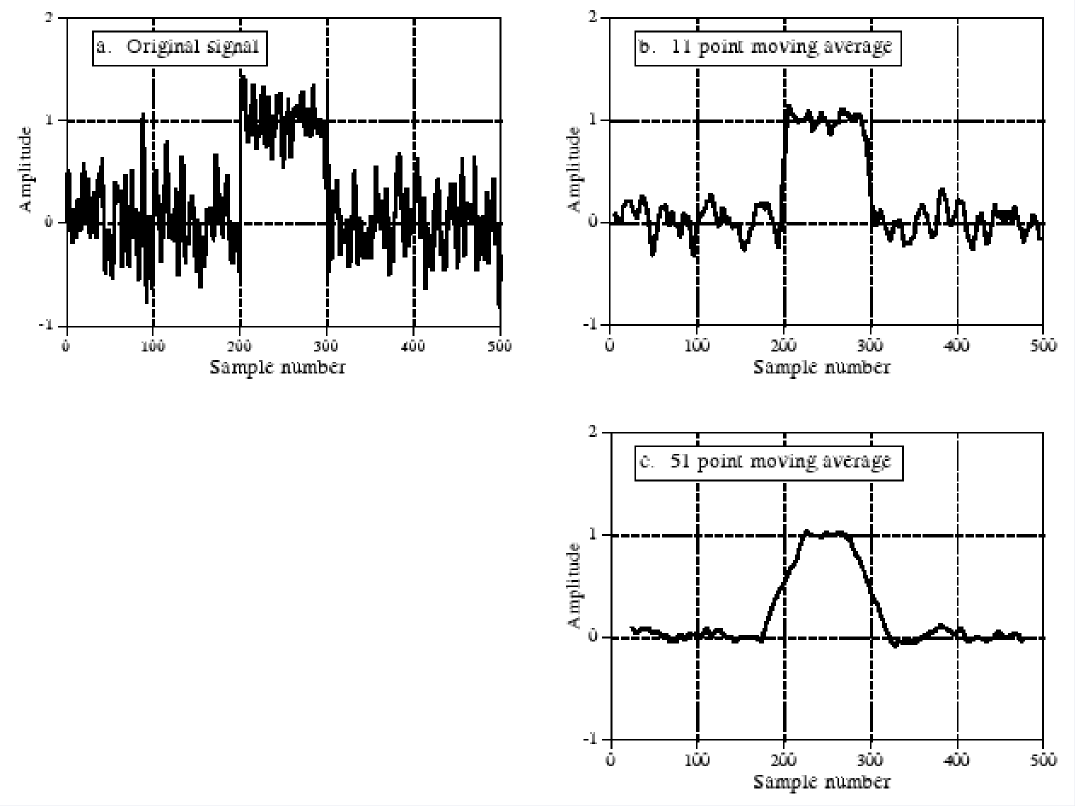
$$y[80]=\frac{x[78]+x[79]+x[80]+x[81]+x[82]}{5}$$

- Isto faz com que o sinal de saída não tenha atraso em relação ao de entrada, mas faz com que o sistema torne-se não causal.
- Você já deve ter percebido que o filtro MA na verdade tem um kerel muito simples:

$$\boxed{h[n]=1/M\,,\, n=0,\ldots,M-1}$$

Redução de ruído vs resposta a degrau

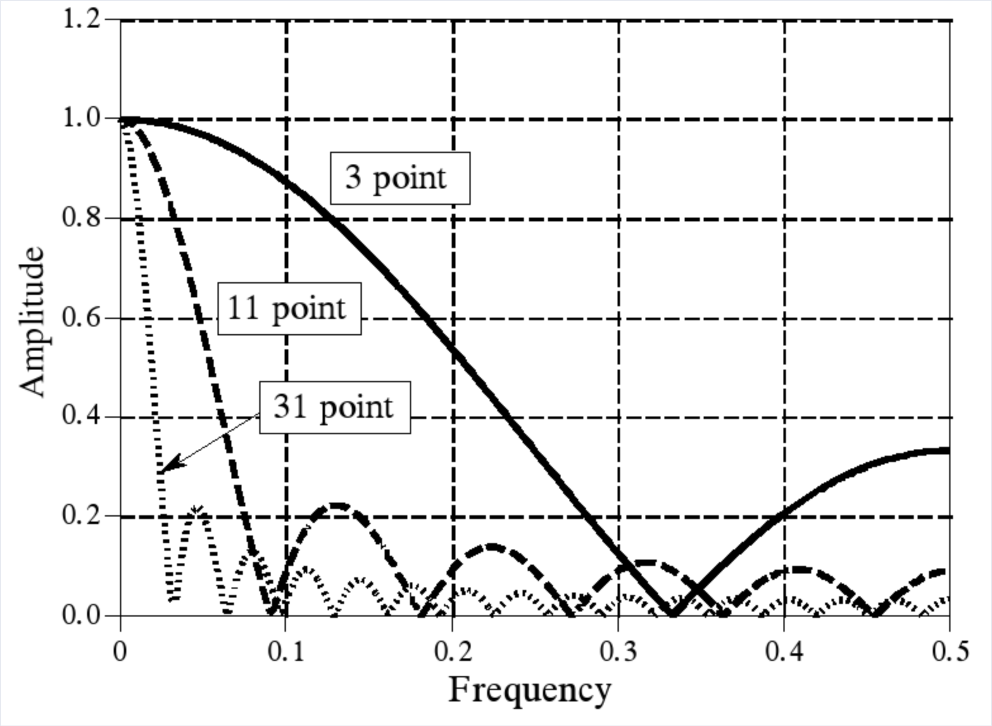
- A redução de ruído é proporcional à raiz quadrada do número de pontos do kernel do filtro.



Porque é o filtro ótimo?

- Imagine que queremos projetar um filtro com um tempo de subida fixo para a resposta a degrau (por exemplo 11 amostras).
- Isto requer que o kernel do filtro tenha 11 pontos.
- Como escolher os valores destes 11 pontos de modo a minimizar o ruído no sinal de saída?
- Desde que o ruído é aleatório, nenhuma amostra de entrada é especial; cada uma é tão ruidosa quanto sua vizinha.
- Assim, é inútil tratar de forma especial qualquer amostra da entrada: o menor ruído é obtido quando todas as amostras são tratadas igualmente.

Resposta em frequência



$$H(f)=\frac{sen(\pi fM)}{M\,sen(\pi f)}$$

- O roll off é bastante lento, e a atenuação na banda de bloqueio é horrível: este filtro não serve para separar uma banda de frequências da outra.

Relativos do filtro MA

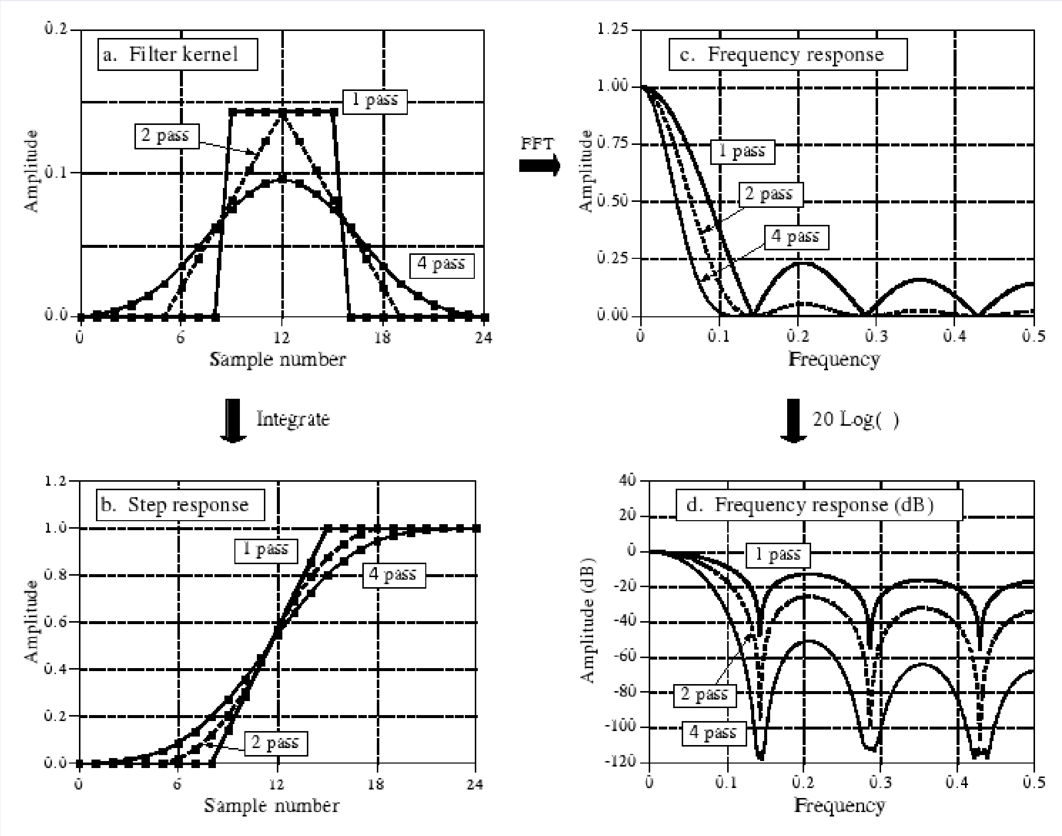
- Vimos anteriormente que filtros que têm um bom desempenho no domínio do tempo, apresentam um desempenho ruim no domínio da freqüência e vice-versa.
- Entretanto, existem casos em que ambos os domínios são simultaneamente importantes:
  - remoção de interferência de sinal de 60Hz (frequência) em um sinal de vídeo (tempo);
  - monitoramento de temperatura ao longo do tempo (tempo), com transdutores sujeitos a contaminação de uma estação de AM local mais um ruído de 60Hz (frequência).

Relativos do filtro MA

- Os relativos do filtro MA apresentam uma resposta em frequência um pouco melhor, e podem ser úteis nestas aplicações de domínios mistos.
- Os filtros mais utilizados para este fim são:
  - Filtros MA de múltiplos passos
  - Filtros Gaussianos
  - Filtros de Blackman

Filtros MA de múltiplos passos

- Envolvem a passagem do sinal de entrada através de um filtro MA duas ou mais vezes.

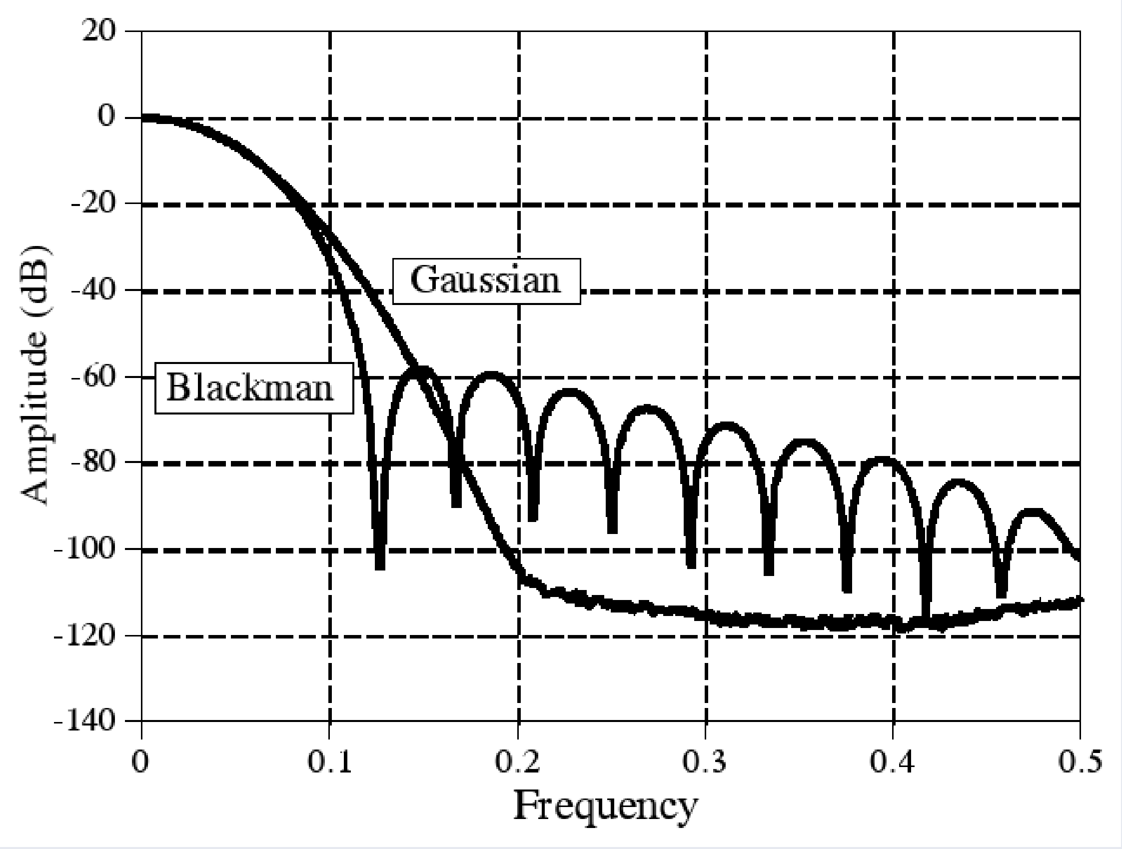


Observações

- Duas passagens equivale a um filtro com kernel triangular.
- Depois de quatro ou mais passagens, o kernel fica cada vez mais parecido com uma gaussiana (lembra do Teorema do Limite Central?)
- A resposta a degrau fica cada vez mais arredondada à medida que passamos mais vezes pelo filtro
- O roll-off melhora um pouquinho, assim como a atenuação na banda de bloqueio, à medida que fazemos mais passagens pelo filtro.

Filtros Gaussiano e de Blackman

- Estes filtros utilizam como kernel uma função gaussiana e uma janela de Blackman, respectivamente.



MA vs relativos

- Os relativos são melhores pois:
  - conseguem maior atenuação na banda de bloqueio
  - os kernels possuem elementos de baixa amplitude em suas extremidades, o que dá menor importância às amostras de entrada nestas regiões
  - as respostas a degrau são curvas suaves (isto pode não ser bom em alguns casos)
- Outras considerações
  - Em termos de redução de ruído, todos são equivalentes
  - Usando um algoritmo recursivo, o filtro MA é muito mais rápido que os relativos (aproximadamente 10x o número de pontos no filtro. Exemplo: um filtro gaussiano de 100 pontos é aproximadamente 1000 vezes mais lento que um MA de mesmo comprimento)

Implementação recursiva

- Vamos imaginar que estamos passando um sinal x[] por um filtro MA de 7 amostras. As amostras y[50] e y[51], por exemplo, são calculadas através de:

$$y[50]=x[47]+\boxed{x[48]+x[49]+x[50]+x[51]+x[52]+x[53]}$$

$$y[51]=\boxed{x[48]+x[49]+x[50]+x[51]+x[52]+x[53]}+x[54]$$

- Note que grande parte do esforço computacional usado para calcular y[50] é repetido no cálculo de y[51] (em destaque)
- Podemos reescrever a segunda equação como:

$$y[51]=y[50]+x[54]-x[47]$$

Implementação recursiva

- Uma vez que temos y[51], podemos calcular y[52] da mesma maneira: subtrai-se o primeiro elemento e adiciona-se o último.
- Desta forma, cada ponto da saída pode ser obtido efetuando-se apenas uma soma e uma subtração, independentemente do comprimento do filtro!
- Este procedimento pode ser generalizado como:

$$y[i]=y[i-1]+x[i+p]-x[i-q]$$

onde: $p=(M-1)/2$

$q=p+1$

Considerações sobre precisão

- Surpreendentemente, a representação em ponto fixo (números inteiros) funciona melhor do que a representação em ponto flutuante (números reais) neste caso, além de ser muito mais rápida. Por que?
  - As operações em ponto fixo são realizadas muito mais rapidamente pelos processadores
  - Em ponto fixo não há propagação de erros de arredondamento, o que certamente ocorre quando usamos ponto flutuante.