

# Análise temporal de sinais digitais

Osmar Tormena Junior, Prof. Me.\*

Uma das justificativas clássicas da introdução da análise de Fourier, ou de outras ainda mais avançadas, são as limitações da simples análise temporal de sinais. Porém, isto não significa que as ferramentas de processamento temporal de sinais não sejam úteis, ou produzam resultados relevantes.

## 1 Segmentação e limiares

Segmentação é o processo no qual um sinal é decomposto em regiões disjuntas no tempo. Cada segmento no tempo é associado à alguma característica que o distingue de segmentos adjacentes.

Um exemplo de segmentação, em aplicações de comunicação por voz, é a análise do sinal no tempo, onde trechos de silêncio são identificados, com o intuito de evitar o envio de pacotes de voz portando apenas ruído.

Isto pode ser feito através da comparação do sinal com um valor limiar pré-definido. Se o sinal for menor que o valor limiar, ele é tomado como nulo.

Obviamente, a escolha do valor limiar será crítica ao desempenho do sistema. Embora existam teorias com rigor matemático para justificar esta escolha, muitas vezes uma abordagem simplesmente experimental (ou técnica) é suficiente.

Uma simples análise do histograma da distribuição de amplitudes de um sinal é um bom ponto de partida para estabelecer um valor limiar.

**Exemplo 1:** O código abaixo demonstra o básico para entrada e saída de áudio no am-

biente GNU Octave/Matlab®.

```
close all
clear
clc

x = audioread('Counting-16-44p1-
mono-15secs.wav');
Fs = 44.1e3;
t = (0:length(x) - 1)/Fs;
```

```
sound(x,Fs)
```

```
plot(t,x), grid on
```

□

**Exercício 1:** Identifique intervalos onde um sinal de voz está silenciado, e elimine o ruído de fundo desses intervalos.

## 2 Filtragem convolucional

No pré-processamento de sinais, dois filtros são largamente utilizados. O *média móvel*

$$h(n) = \frac{1}{2N+1}(u(n+N)-u(n-N-1)) \quad (1)$$

e o *gaussiano*

$$h(n) = K \exp\left(-\frac{9n^2}{2N^2}\right), \quad (2)$$

truncado para um intervalo  $[-N, N]$ , onde  $K$  é escolhido de forma que  $\|h(n)\|_1 = 1$ . Ambos os filtros possuem fase linear.

**Exemplo 2:** O código abaixo insere ruído AWG no sinal de áudio, para uma SNR medida em decibéis.

---

\*tormena@utfpr.edu.br

```

close all
clear
clc

x = audioread('Counting-16-44p1-
    mono-15secs.wav');
Fs = 44.1e3;
t = (0:length(x) - 1)/Fs;

SNR = 20;
x = awgn(x,SNR,'measured');

sound(x,Fs)

plot(t,x), grid on

```

**Exercício 2:** Para um mesmo  $N$  (comprimento  $2N - 1$ ), implemente os filtros média móvel e gaussiano, aplicando-os sobre áudio ruidoso. Subjetivamente, qual filtro obtém melhor resultado?

### 3 Detecção de bordas

A detecção de bordas abruptas em sinais pode ser importante em sua segmentação, ou no reconhecimento da estrutura de um sinal. Uma primeira aproximação, na análise deste problema poderia ser uma comparação limiar de sua derivada. Infelizmente esse caminho apresenta problemas práticos em sinais ruidosos, pois a diferenciação amplifica o ruído.

Alternativas convolucionais são comumente aplicadas. As sequências

$$\text{sgn}(n) = \begin{cases} 1 & \text{se } n > 0, \\ 0 & \text{se } n = 0, \\ -1 & \text{se } n < 0, \end{cases} \quad (3)$$

e a derivada da gaussiana

$$h(n) = -n \exp\left(-\frac{8n^2}{N^2}\right), \quad (4)$$

ambas truncadas para  $[-N, N]$ , produzem um pico de alta magnitude, em sua convolução com um sinal com bordas.

**Exercício 3:** Implemente um detector de bordas e verifique a precisão da detecção de bordas, em sinais com ruído, para rampas abruptas e suaves. Qual a relação entre  $N$ , a SNR do sinal, e a precisão na localização da borda?

### 4 Detecção de padrões

Em muitas aplicações, os valores do sinal em si não são a informação de interesse, mas sim o instante onde eles ocorrem. Isto é comum em aplicações como sonar e radar, onde deseja-se saber apenas qual o atraso (além de outros efeitos) que o canal aplicou sobre o sinal entre sua transmissão e recepção. Ou seja, o “eco” do sinal é de particular interesse.

Já em outras aplicações, o eco é um artefato indesejável. Em telefonia, o eco é particularmente desconcertante. Assim, soluções capazes de mitigar os efeitos do eco são de grande interesse prático.

**Exemplo 3:** O código abaixo insere eco em um sinal de voz, com 250 ms de atraso, e 50 % da amplitude.

```

close all
clear
clc

voice = audioread('speech_dft_
    8kHz.wav');
Fs = 8e3;
alpha = 0.5;
lag = 0.25;
pad = zeros(round(Fs*lag),1);

audio = [voice;pad];
echo = [pad;voice]*alpha;

x = audio + echo;

figure
plot((0:length(audio) - 1)/Fs,audio)
hold on
plot((0:length(audio) - 1)/Fs,echo)
grid on

```

□

O eco pode ser removido através do filtro IIR

$$y(n) + \beta y(n - \Delta) = x(n), \quad (5)$$

onde  $x(n)$  é o sinal com eco,  $y(n)$  é o sinal com eco filtrado,  $\beta$  é uma estimativa da amplitude do eco (dada pela amplitude do pico de correlação) e  $\Delta$  é uma estimativa do atraso do eco (dado pela posição do pico de correla-

ção). A função de transferência do sistema da Eq. (5) é dada por

$$H(z) = \frac{1}{1 + \beta z^{-\Delta}} \quad (6)$$

**Exercício 4:** Implemente o filtro IIR descrito acima para remover o eco de uma gravação de áudio.