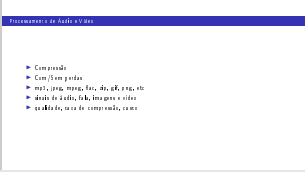
Processamento de Áudio e Vídeo

Prof. Leonardo Araújo

https://sites.google.com/site/leolca/teaching/multimedia-signal-processing

Processamento de Áudio e Vídeo

- Compressão
- Com/Sem perdas
- mp3, jpeg, mpeg, flac, zip, gif, png, etc
- sinais de áudio, fala, imagens e vídeo
- qualidade, taxa de compressão, custo



- O conceito de compressão surge naturalmente quando estamos lidando com comunicação.
- Compressão de dados é o processo de converter dados provenientes de uma fonte em outros dados com menor tamanho.
- Armazenamento e transmissão (no fundo, ambos são formas de comunicação).
  - linha telefônica analógica
  - comunicação digital através desta linha telefônica analógica
  - link de comunicação de rádio entre a sonda espacial Galileu orbitando Júpiter e a Terra
  - armazenamento e reprodução de áudio ou vídeo (ou dados) em um CD, DVD ou disco rígido
  - reprodução celular em que a informação sobre as células é contida no DNA

Métodos de compressão sem perda (alguns são vistos na disciplina Teoria da Informação) possuem como limite a entropia. Reconstrução exata da mensagem produzida pela fonte. Remover redundância.

Métodos de compressão com perda utilizam-se do fato de que muita informação pode ser perdida sem ser percebida ou aceita-se uma distorção do sinal em prol de uma maior compressão.

Processamento de Áudio e Vídeo
Com perdas
Com/Sem perdas
mp3, jpeg, mpeg, flac, dts, gif, png, etc
desafio de áudio, fala, imagens e vídeo
qualidade, razão de compressão, ruído

- Áudio, fala, imagens e vídeo são originalmente sinais analógicos.
- Conversão em sinais digitais: amostragem, quantização, codificação.

A qualidade da compressão pode ser uma medida objetiva ou subjetiva. Na maioria das vezes, iremos realizar medidas objetivas pois realizar testes subjetivos é muito dispendioso. Podemos escolher medidas objetivas que sejam bem correlacionadas com medidas subjetivas.

O custo de compressão e descompressão podem, em geral, serem diferentes. Descompressão deve ser privilegiada pois é realizada diversas vezes e geralmente por terminais com menor poder computacional.

Imagem digital

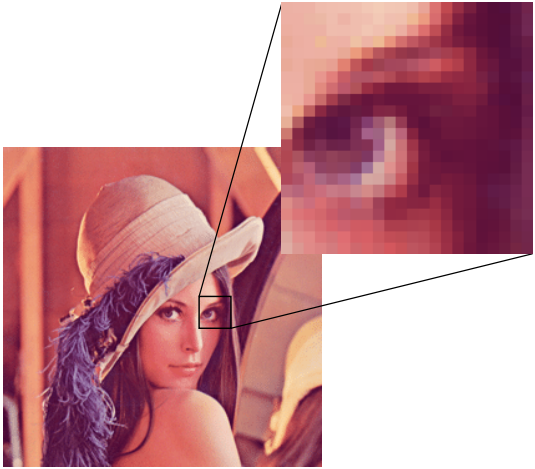


Figura 1: Lena - detalhe.

Espaço necessário para armazenar uma foto

- câmera 10 Mpixel
- 3 bytes por pixel (RGB)
- cada foto requer 30 Mbyte
- um cartão de memória de 2 Gbytes é capaz de armazenar 66 fotos

Espaço necessário para armazenar um vídeo

- 480 x 720, 30 fps
- 345.600 pixels por frame
- RGB 3 bytes por pixel
- 1.036.800 byte, aprox. 1 Mbyte por frame
- 30 frames requerem 31.104.000 bytes, aprox. 31 Mbyte por segundo
- um CD de 650 Mbytes é capaz de armazenar apenas 21 segundos de vídeo e um DVD de 4.7 GB apenas 155 segundos de vídeo.

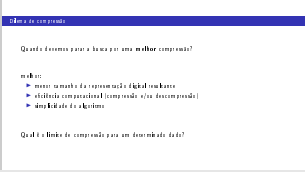
Dilema de compressão

Quando devemos parar a busca por uma **melhor** compressão?

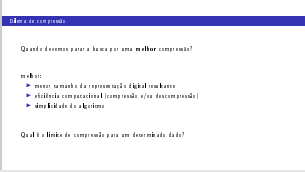
melhor:

- menor tamanho da representação digital resultante
- eficiência computacional (compressão e/ou descompressão)
- simplicidade do algoritmo

Qual é o limite de compressão para um determinado dado?



Modificar um algoritmo para melhorar a taxa de compressão em 1% pode acarretar um aumento de 10% no tempo de execução do algoritmo e ainda mais sobre a complexidade do programa.



Conjecturas[1].

- Compressão de dados pode ser interpretada como o processo de remover complexidades (redundâncias) desnecessárias na informação, e desta forma, maximizando a simplicidade enquanto preserva o máximo possível do poder discricionário dos dados.
- Todo tipo de computação e racionalização formal pode ser compreendida como compressão de informação através do processo de identificar padrões, busca e unificação das instâncias destes padrões.

Termos I

compressor ou codificador é o programa que comprime os dados crus na entrada e cria uma saída de dados comprimida (com baixa redundância).

decompressor ou decodificador converte os dados na direção oposta.

fluxo é o dado a ser comprimido, armazenado como um arquivo ou transmitido.

dado não-codificado, cru, ou original é o fluxo de dados da entrada.

dado codificado ou comprimido é o fluxo de saída.

método de compressão não-adaptativo é rígido e não modifica sua operação ou seus parâmetros em resposta aos dados em particular que estão sendo comprimidos.

método adaptativo analisa os dados crus e modifica sua operação e/ou parâmetros de acordo com os dados em mãos.

método semi-adaptativo utiliza 2 passagens aonde, na primeira, realiza a leitura dos dados e contabiliza estatísticas dos dados a serem comprimidos; na segunda passagem, realiza de fato a compressão utilizados parâmetros determinados na primeira varredura.

Termos II

método localmente adaptativo se adapta às condições locais do fluxo de dados e varia à medida que move ao longo dos dados.

compressão com perdas/sem perdas : Para atingirem maior compressão, os métodos de compressão com perda perdem informação. Os métodos de compressão sem perda não admitem perder informação alguma.

Compressão em cascata ocorre quando diferentes métodos de compressão são utilizados um em seguida do outro.

Compressçao perceptiva ocorre quando apenas a informação imperceptível pelos nosso sentidos é removida.

Compressão simétrica é o caso em que o compressor e descompressor utilizam basicamente o mesmo algoritmo, porém em direções opostas.

Complacente é o codificador/decodificador que gera/lê de forma correta um fluxo de dados (Qualquer pessoa é livre para implementar seu próprio algoritmo).

Universal é o método de compressão de dados que não depende da estatística dos dados.

Termos III

Razão de Compresão = tamanho do dado de saída / tamanho do dado de entrada.

Fator de Compressão = tamanho do dado de entrada / tamanho do dado de saída = (razão de compressão)$^{-1}$.

Ganho de Compressão = $100 \log_e$ (tamanho de referência / tamanho comprimido), aonde o tamanho de referência é o tamanho dos dados de entrada ou o tamanho do dado de saída comprimido por algum algoritmo padrão.

Erro médio quadrático (MSE) e relação sinal ruído de sinal (PSNR) são utilizados para medir a distorção causada por uma compressão com perdas.

Termos

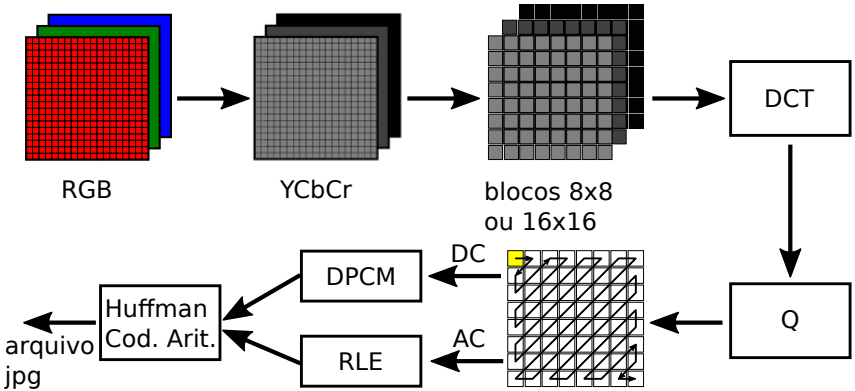


Figura 2: Esquema de compressão JPEG.

Slides - introdução ao GNU Octave

https://drive.google.com/open?id=1ew5fl9v_OIybsy3KdEgIvohLTcuwuru_

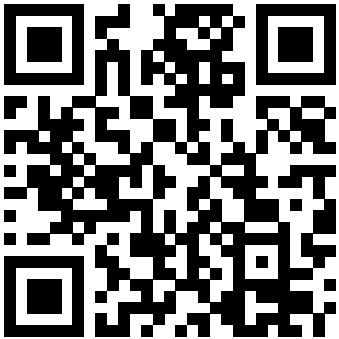
Notebook - introdução

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/introducao.ipynb

Notebook - imagem colorida

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/introdocao_imagem_colorida.ipynb

Leitura

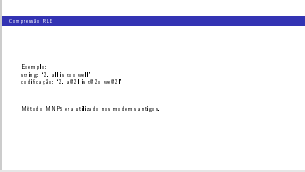


David Salomon, Giovanni Motta - *Handbook of Data Compression*, 2010
https://books.google.com.br/books?id=LHCY4VbiFqAC
Introduction, Basic Techniques (Salomon et al., 2010)

Compressão RLE

Exemplo:
string: '2. all is too well'
codificação: '2. a@2l is t@2o we@2l'

Método MNP5 era utilizado nos modems antigos.



MNP : Microcom Networking Protocol

"The MNP5 method is a two-stage process that starts with run-length encoding, followed by adaptive frequency encoding."(Salomon, 2000)

"With MNP 5, the data received from the computer are first compressed with a simple algorithm, and then passed into the MNP 4 packetizing system for transmission. On best-case data the system offered about 2:1 compression, but in general terms about 1.6:1 was typical, at least on text. As a result a 2400 bit/s modem would appear to transfer text at 4000 bit/s, even though the modem was still running at the same 600 baud * 4 bits per symbol rate.

This dramatic increase in throughput allowed Microcom modems to remain somewhat competitive with models from other companies that were otherwise nominally much faster. For instance, Microcom generally produced 1200 and 2400 bit/s modems using commodity parts, while companies like USRobotics and Telebit offered models with speeds up to 19200 bit/s."(`https://en.wikipedia.org/wiki/Microcom_Networking_Protocol`)

Compressão RLE

Exemplo: uma imagem em tons de cinza com 8-bit de profundidade começa com os seguintes valores
12, 12, 12, 12, 12, 12, 12, 12, 12, 35, 76, 112, 67, 87, 87, 87, 5, 5, 5, 5, 5, 5, 1, ...
será comprimida como 9,12,35,76,112,67,3,87,6,5,1, ...

Se utilizarmos como *flag* o valor 255, então a sequência acima será expressa por
255, 9, 12, 35, 76, 112, 67, 255, 3, 87, 255, 6, 5, 1, ...

grupos de 8
10000010,9,12,35,76,112,67,3,87,100...,6,5,1, ...

Exemplo RLE - GNU Octave

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/rle_mario.ipynb

Move-to-Front Coding

Consideramos o alfabeto de símbolos $\mathcal{A}$ como uma lista onde os símbolos mais frequentes estarão dispostos no início da lista.
O método é localmente adaptativo, já que ele se adapta à frequência dos símbolos em cada região do fluxo de dados.

Move-to-Front Coding - Exemplo (Salomon et al., 2010)

Exemplo: entrada a ser codificada: **abcddcbamnopponm**

C = (0, 1, 2, 3, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3)
– utilizando move-to-front

C' = (0, 1, 2, 3, 3, 2, 1, 0, 4, 5, 6, 7, 7, 6, 5, 4)
– sem utilizar move-to-front

a	abcdmnop	0
b	abcdmnop	1
c	bacdmnop	2
d	cbadmnop	3
d	dcbamnop	0
c	dcbamnop	1
b	cdbamnop	2
a	bcdamnop	3
m	abcdmnop	4
n	mabcdnop	5
o	nmabcdop	6
p	onmabcdp	7
p	ponmabcd	0
o	ponmabcd	1
n	opnmabcd	2
m	nopmabcd	3
	mnopabcd	

a	abcdmnop	0
b	abcdmnop	1
c	abcdmnop	2
d	abcdmnop	3
d	abcdmnop	3
c	abcdmnop	2
b	abcdmnop	1
a	abcdmnop	0
m	abcdmnop	4
n	abcdmnop	5
o	abcdmnop	6
p	abcdmnop	7
p	abcdmnop	7
o	abcdmnop	6
n	abcdmnop	5
m	abcdmnop	4

Move-to-Front Coding - Exemplo (Salomon et al., 2010)

O resultado C obtido pelo move-to-front é tal que, na média, os valores em C são pequenos (os valores no início do dicionário são os mais prováveis). Isto faz com que a saída seja propícia para ser codificada através da codificação de Huffman ou codificação aritmética.

i	Code	Size
1	1	1
2	010	3
3	011	3
4	00100	5
5	00101	5
6	00110	5
7	00111	5
8	0001000	7
9	0001001	7
⋮	⋮	⋮
15	0001111	7
16	000010000	9

Figura 3: Exemplo de código de tamanho variável.

Move-to-Front Coding

Variações:

1) Move-ahead-k: O elemento do alfabeto A que corresponde ao símbolo corrente será deslocado k posições para cima na lista ao invés de ir para o topo da lista.
2) Wait-c-and-move: O elemento do alfabeto A será deslocado para o início da lista apenas após aparecer c vezes durante a codificação. item Wait-c-and-ahead-k: Um combinação das duas variantes anteriores.

Exemplo Move-to-Front - GNU Octave

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/move-to-front.ipynb

Conversão AD/DA

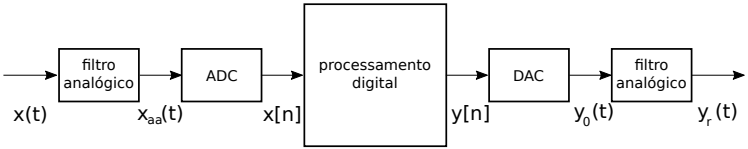


Figura 4: Processamento digital de sinais. Conversão AD e DA.

Quantização

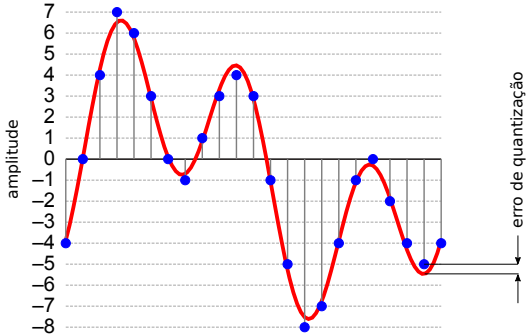


Figura 5: Quantização.

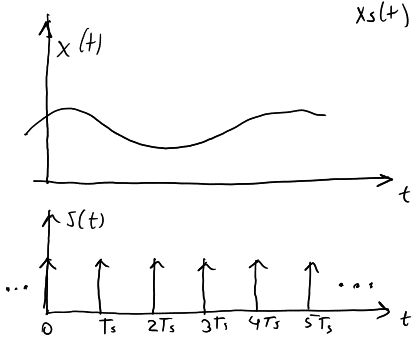
Amostragem

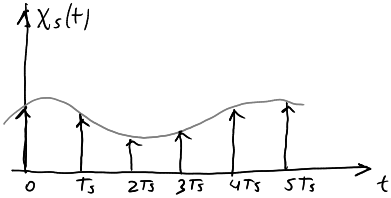
Ao amostrar um sinal $x(t)$ com período de amostragem T_s teremos

$$\begin{aligned} x_s(t) &= x(t)s(t) \\ &= x(t)\sum_{k=-\infty}^{\infty}\delta(t-kT_s) \\ &= \sum_{k=-\infty}^{\infty}x(kT_s)\delta(t-kT_s) \end{aligned} \tag{1}$$

Amostragem

$$x_s(t) = x(t) \cdot s(t)$$





Amostragem I

Como $s(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_s)$ é periódico com período T_s, podemos representá-lo por uma série de Fourier:

$$s(t) = \sum_{k=\infty}^{\infty} \delta(t - kT_s) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi tk/T_s}, \tag{2}$$

onde

$$c_k = \frac{1}{T_s} \int_{-T_s/2}^{T_s/2} \delta(t) e^{-j2\pi tk/T_s} dt = \frac{1}{T_s} \tag{3}$$

Desta forma, $x_s(t) = x(t) \cdot s(t)$ poderá ser expresso por

$$x_s(t) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} x(t) e^{j2\pi tk/T_s}. \tag{4}$$

Amostragem II

A multiplicação por $\exp(j2\pi\alpha t)$ corresponde, na frequência, a um deslocamento de α. Teremos assim

$$X_s(j\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X\left(j(\Omega - n\Omega_s)\right) \tag{5}$$

Amostragem III

Podemos chegar ao mesmo resultado sabendo que, se no domínio do tempo temos $x_s(t) = x(t) \cdot s(t)$, no domínio da frequência temos

$$X_s(j\Omega) = \frac{1}{2\pi} X(j\Omega) * S(j\Omega). \tag{6}$$

Como a transformada de Fourier de $s(t)$ é

$$S(j\Omega) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s), \tag{7}$$

onde $\Omega_s = {}^{2\pi}/_{T}$, então utilizando as Equações (6) e (7) obtemos Equação (5).

Amostragem IV

Se $x(t)$ for um sinal limitado em frequência (Ω_N frequência máxima) e não havendo *aliasing*, $\Omega_s > 2\Omega_N$, podemos reconstruir $x(t)$:

$$X_r(j\Omega) = H_r(j\Omega)X_s(j\Omega) \tag{8}$$

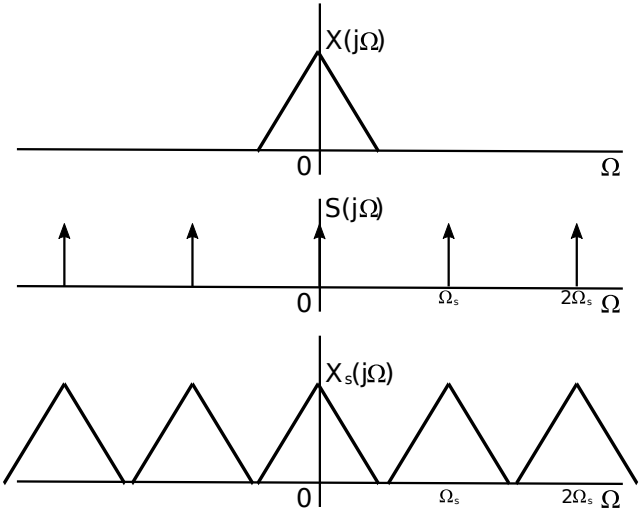
onde H_r é um filtro passa-baixas ideal com $\Omega_N < \Omega_c < \Omega_s$.

$$H_r(j\Omega) = \begin{cases} 1 & \text{, se } \quad |\Omega| \leq \Omega_c, \\ 0 & \text{, caso contrário.} \end{cases} \tag{9}$$

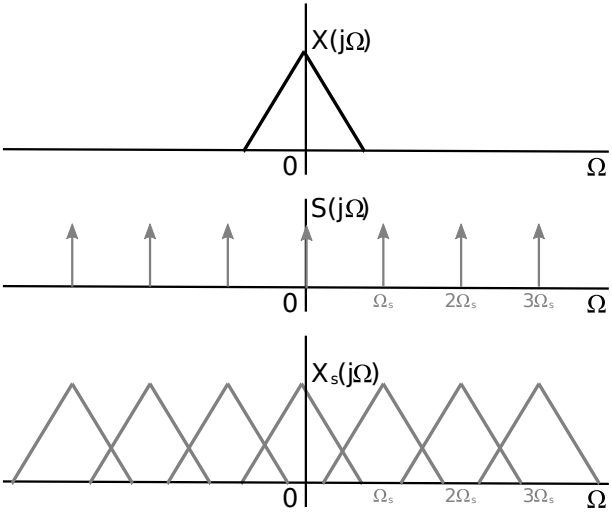
(Teorema da Amostragem)

Leitura: Capítulo 4 Oppenheim, A. V. (2009). *Discrete-Time Signal Processing*. Pearson.

Amostragem



Amostragem



Quantização Escalar I

Quantização escalar é um mapeamento Q de valores reais x de uma variável aleatória contínua X nos valores $y = Q(x)$, mais próximos de x (em termos de uma determinada medida de distorção), de um conjunto discreto e finito $Y = y_1, y_2, \ldots, y_M$. Os valores y_i , $i = 1, 2, \ldots, M$, são chamados níveis de saída, ou valores de representação, ou ainda valores de aproximação. Y é chamado de *codebook* ou conjunto de aproximação.

Quantização Escalar II

O quantizador escalar é determinado pelo conjunto de limiares $\mathcal{T} = \{t_i\}$, $i = 0, 1, \ldots, M$ e pelo conjunto de pontos de representação $\mathcal{Y} = \{y_i\}$, $i = 1, \ldots, M$. Os limiares dividem exaustivamente o domínio R em subintervalos (ou células, regiões de representação) $\Delta_i = (t_{i-1}, t_i]$ disjuntas, ou seja, $\Delta_i \cap \Delta_j = \emptyset$. Diz-se que a divisão é exaustiva pois $\bigcup_{i=1}^{M} \Delta_i = R$. Esta divisão é tal que existe apenas um y_i associado a cada intervalo Δ_i, ou seja, $y_i = Q(x)$ se e somente se $x \in \Delta_i$.

Quantização Escalar III

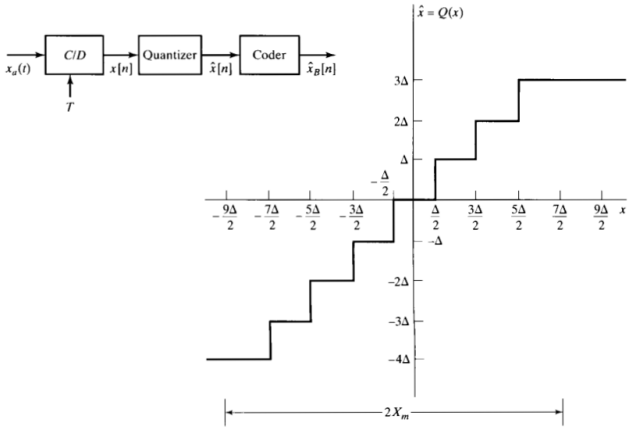


Figura 6: Quantização. Fonte: Oppenheim (2009).

Quantização Escalar IV

É inerente ao processo de quantização a introdução de um erro, chamado *erro de quantização* ou *ruído de quantização*.
O erro de quantização esperado é dado por

$$D(Q) = E\left\{d(x, Q(x))\right\}, \tag{10}$$

onde $d(x, Q(x))$ é uma medida de distorção entre x e $Q(x)$, dada por $d(\cdot)$.
A taxa de quantização é o número de bits R que é utilizado na representação de um valor x.
Ela é dada em bits por amostra.
Para um quantizador com taxa fixa temos $R = \log_2 M$ bits por amostra.

Quantização Escalar V

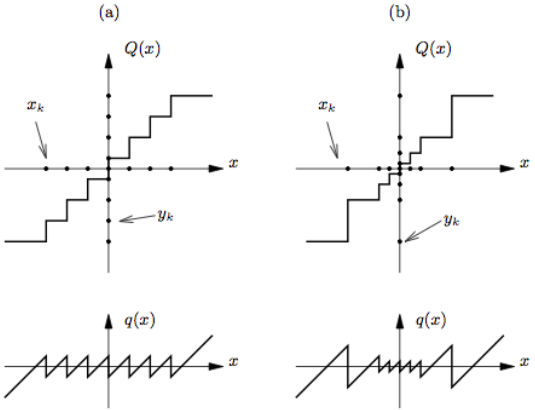


Figura 7: (a) linear (b) logarítmico. Fonte: Oppenheim (2009).

Entropia na saída do quantizador I

As probabilidades dos níveis de representação de um quantizador podem ser determinadas, conhecendo-se a pdf do sinal.
Seja $f(x)$ a pdf (função densidade de probabilidade) de X. Podemos calcular a probabilidade do i-ésimo nível de reprodução (a probabilidade de $x \in \Delta_i$) como

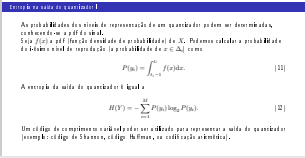
$$P(y_i) = \int_{t_i-1}^{t_i} f(x)\mathrm{d}x. \tag{11}$$

A entropia da saída do quantizador é igual a

$$H(Y) = -\sum_{i=1}^{M} P(y_i)\log_2 P(y_i). \tag{12}$$

Um código de comprimento variável poder ser utilizado para representar a saída do quantizador (exemplo: código de Shannon, código Huffman, ou codificação aritmética).

Entropia na saída do quantizador


Entropia na saída do quantizador I

As probabilidades do nível de representação de um quantizador podem ser determinadas, conhecendo-se a pdf do sinal.
Seja $f(x)$ a pdf (função densidade de probabilidade) de X. Podemos calcular a probabilidade do k-ésimo nível de reprodução (a probabilidade de $x \in \Delta_k$) como

$$P(y_k) = \int_{x_{k-1}}^{x_k} f(x)dx. \quad (11)$$

A entropia da saída do quantizador é igual a

$$H(Y) = -\sum_{k=1}^{N} P(y_k)\log_2 P(y_k). \quad (12)$$

Um código de comprimento variável pode ser utilizado para representar a saída do quantizador (exemplo: código de Shannon, código Huffman, ou codificação aritmética).

Shannon: o limite de representação é a entropia.

O limite para se representar um sinal, sem perdas, será dado pela entropia da fonte.

Quantização escalar uniforme I

A quantização escalar é uniforme quando os limiares estão igualmente espaçados, e desta forma, as células possuem o mesmo tamanho (exceto as extremas, primeira e última), ou seja, $|\Delta_i| = \delta$, e o ponto de representação localiza-se no ponto médio da célula,

$$y_i = \frac{t_{i-1} + t_i}{2} = t_{i-1} + \frac{\delta}{2}, \quad i = 1, 2, \ldots, M. \tag{13}$$

Quantização escalar uniforme II

(*obs.: considerando apenas valores positivos)
Dada a entrada x, a célula associada a x é determinada por

$$i = [x/\delta], \tag{14}$$

onde δ é a largura de cada célula e $[\cdot]$ representa a operação de arredondamento.
O valor de aproximação para a entrada x é dado por

$$y = Q(x) = \delta \left[\frac{x}{\delta}\right] \tag{15}$$

isto é, a i-ésima célula é determinada por $\Delta_i = (i\delta - \delta/2, i\delta + \delta/2]$ e $y_i = i\delta$.

Distorção no quantizador escalar uniforme

Se $f(x)$ é conhecida, então podemos calcular a distorção esperada do quantizador

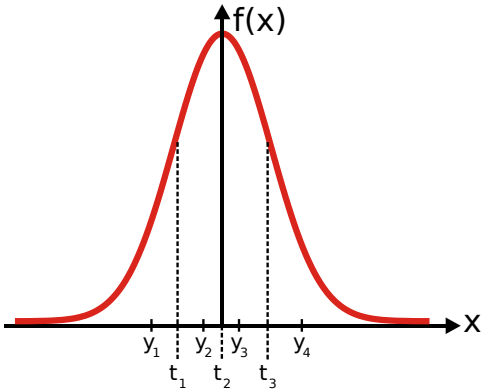
$$D(Q) = \int_{-\infty}^{\infty} f(x) d(x, Q(x)) dx = \sum_i \int_{t_{i-1}}^{t_i} f(x) d(x, y_i) \mathrm{d}x. \tag{16}$$

Se o erro de distorção é medido pelo erro quadrático, então $D(Q)$ fornecerá o erro quadrático médio (MSE, *Mean Squared Error*):

$$D(Q) = \sum_i \int_{t_{i-1}}^{t_i} f(x)(x - y_i)^2 \mathrm{d}x. \tag{17}$$

Quantização escalar não-uniforme I

Se conhecemos as características estatísticas de X, podemos utilizar esta informação para melhorar as características do quantizador.



Algoritmo de Lloyd-Max I

O algoritmo de *Lloyd-Max* é um algoritmo para encontrar os limiares $\{t_i\}$ e os pontos de representação $\{y_i\}$ que minimizam a distorção.
Lloyd e Max criaram um procedimento para construir uma solução para o problema, que satisfaz as condições necessárias (mas não suficientes):

- os limiares devem ficar entre os pontos de representação:

$$t_i = \frac{y_{i+1} + y_i}{2} \quad , 1 \leq i \leq M-1, \tag{18}$$

- os pontos de representação devem ficar no meio (com relação à esperança) de um dado intervalo

$$y_i = E[X(i)] = \frac{\int_{t_{i-1}}^{t_i} x f_X(x) dx}{\int_{t_{i-1}}^{t_i} f_X(x) dx}. \tag{19}$$

Algoritmo de Lloyd-Max II

Algoritmo:

1) Escolher um conjunto inicial arbitrário com M pontos de representação $y_1 < y_2 < \ldots y_M$.
2) Para cada i, $1 \leq j \leq M-1$, fazer $t_i = \frac{1}{2}(y_{i+1} + y_i)$.
3) Para cada i, $1 \leq j \leq M-1$, fazer y_i igual à média condicional de $X \sim f(x)$, dado $X \in (t_{i-1}, t_i]$ (onde t_0 e t_M são respectivamente $-\infty$ e $+\infty$).
4) Repetir os passos (2) e (3) até que a melhoria no MSE seja desprezível; então interromper.

O MSE decresce (ou permanece o mesmo) a cada passo do algoritmo. Como o MSE é não-negativo, ele irá se aproximar de um limite em um número finito de passos, pois o algoritmo será interrompido quando a melhoria no MSE foi menor que um dado $\epsilon > 0$.

Algoritmo de Lloyd-Max III

O exemplo abaixo ilustra que o algoritmo deve chegar a um mínimo local. Considere $M = 2$ pontos de representação e uma pdf $f(x)$ como definida na Figura 8.

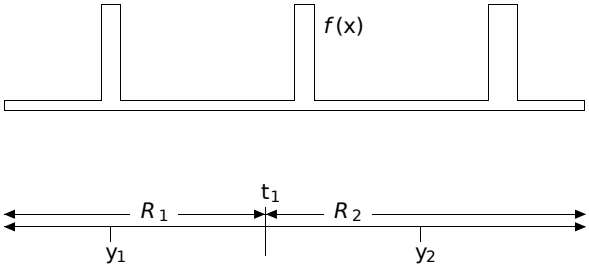


Figura 8: Exemplo Lloyd-Max: regiões e pontos de representação que satisfazem a condição de parada do algoritmo nas não minimizam a distorção média quadrática. Fonte: Gallager (2008)

Algoritmo de Lloyd-Max IV

A configuração apresentada na Figura 8 satisfaz os critérios de parada, entretanto o pico mais a direita é mais provável que os outros dois, desta forma, o MSE poderia ser menor se R_1 cobrisse a regiões dos dois picos à esquerda e R_2 apenas o pico à direita.

Leitura: Capítulo 3 Gallager, R. G. (2008). *Principles of Digital Communication*. Cambridge University Press.

Performance do Quantizador I

Seja $p(x)$ a pdf do sinal de entrada x, então o erro médio quadrático (MSE) devido à quantização será dado por

$$\sigma_q^2 = \sum_{k=1}^{M} \int_{t_{k-1}}^{t_k} (x - y_k)^2 p(x) \mathrm{d}x. \tag{20}$$

Se M for grande e a pdf $p(x)$ for suave, poderemos aproximar $p(x)$ no intervalo $(t_{k-1}, t_k]$ como

$$p(x) \approx p\left(\frac{t_{k-1} + t_k}{2}\right), \quad t_{k-1} < x \leq t_k, \tag{21}$$

e assim, a equação 20 poderá ser reescrita como

$$\sigma_q^2 = \sum_{k=1}^{M} p\left(\frac{t_{k-1} + t_k}{2}\right) \int_{t_{k-1}}^{t_k} (x - y_k)^2 \mathrm{d}x. \tag{22}$$

Performance do Quantizador II

Mostra-se que

$$\int_{t_{k-1}}^{t_k} (x - y_k)^2 \mathrm{d}x = \Delta_k \left[\left(y_k - \frac{t_{k-1} + t_k}{2} \right)^2 + \frac{\Delta_k^2}{12} \right], \tag{23}$$

onde $\Delta_k = t_k - t_{k-1}$ é o tamanho do passo do quantizador.
Para minimizar o MSE devemos escolher $y_k = (t_{k-1} + t_k)/2$, de forma que o primeiro termo em 23 se anule. Ou seja, devemos escolher os pontos de representação como o ponto médio dos limiares dos intervalos. (obs.: Isto ocorre devido à aproximação feita para p suave e M grande. No caso geral, deveremos ter os pontos de representação no valor esperado de cada intervalo)
Vamos definir p_k como a probabilidade de x pertencer ao intervalo $(t_{k-1}, t_k]$. Usando a aproximação feita anteriormente, teremos

$$p_k = \Pr(t_{k-1} < x \leq t_k) \approx p \left(\frac{t_{k-1} + t_k}{2} \right) \Delta_k, \tag{24}$$

Performance do Quantizador III

e assim podemos reescrever a equação 22 como

$$\sigma_q^2 = \frac{1}{12} \sum_{k=1}^{M} p_k \Delta_k^2. \tag{25}$$

Performance do Quantizador Uniforme I

Para o quantizador uniforme o passo é constante ($\Delta_k = \Delta$ para todo k). Teremos assim

$$\sigma_q^2 = \frac{\Delta^2}{12} \underbrace{\sum_{k=1}^{M} p_k}_{=1} = \frac{\Delta^2}{12}. \tag{26}$$

Note que a potência do ruído de quantização é independente da distribuição do sinal.
A performance do quantizador será expressa pela relação sinal-ruído de quantização (SQNR),

$$\text{SQNR} = 10 \log \left(\frac{\sigma_x^2}{\sigma_q^2} \right) = 10 \log \left(\frac{12 \sigma_x^2}{\Delta^2} \right) \text{dB}. \tag{27}$$

Performance do Quantizador Uniforme - Sinal Senoidal I

Vamos supor que o sinal de entrada seja da forma $A \sin \omega t$ e um quantizador uniforme com n bits ($2^n = M$). Podemos escolher Δ para que não ocorra saturação. Faremos então $\Delta = A/2^{n-1}$. A potência do sinal senoidal é $\sigma_x^2 = A^2/2$. Usando agora a Equação 27, teremos

$$\text{SQNR (senoide)} = 6n + 1.76\text{dB}. \tag{28}$$

Performance do Quantizador Uniforme - Sinal Gaussiano I

Iremos supor agora um sinal de entrada com distribuição gaussiana: $p(x) = 1/\sqrt{2\pi\sigma}e^{-(x^2/2\sigma^2)}$. Para que a distorção por saturação seja desprezível, iremos fazer $2^{n-1}\Delta = 4\sigma$, ou seja, teremos $\Delta = \sigma/2^{n-3}$. A potência média quadrática do sinal de entrada é $\sigma_x^2 = \sigma^2$. Usando agora a Equação 27, teremos

$$\text{SQNR (gauss)} = 6n - 7.3\text{dB}. \tag{29}$$

Quantizador Não Uniforme

Os sinais de fala, por exemplo, estão geralmente concentrados em torno da origem. Desta forma, seria interessante propor um quantizador em que os passos de quantização fossem menores na região de menor amplitude do sinal e maiores na região de maior amplitude. Isto levaria a uma redução do ruído de quantização total.

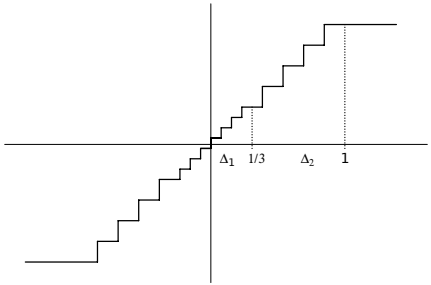


Figura 9: Exemplo de quantizador não uniforme de 4 bits, com $\Delta_1 = \Delta_2/2$ (Ogundunmi and Narasimha, 2010).

Compressor e Expansor

Podemos utilizar compressor e expansor para implementar um quantizador não uniforme.

compressor é feito para amplificar os sinais de baixa amplitude, às custas de atenuar os sinais de alta amplitude;

expansor faz o inverso.

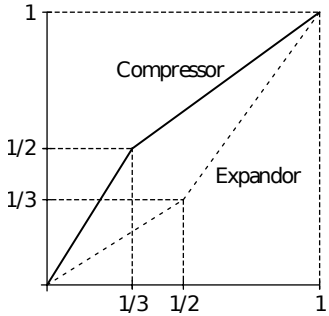


Figura 10: Exemplo de compressor e expansor (Ogundunmi and Narasimha, 2010).

Performance do Quantizador Não Uniforme I

O erro médio quadrático devido à quantização é dado pela Equação 25, repetida a seguir,

$$\sigma_q^2 = \frac{1}{12} \sum_{k=1}^{M} p_k \Delta_k^2,$$

onde $p_k = \Pr(t_{k-1} < x \leq t_k)$ e $\Delta_k = (t_k - t_{k-1})$.
Suponha que o compressor apresentado na Figura 11 seja utilizado.

Performance do Quantizador Não Uniforme II

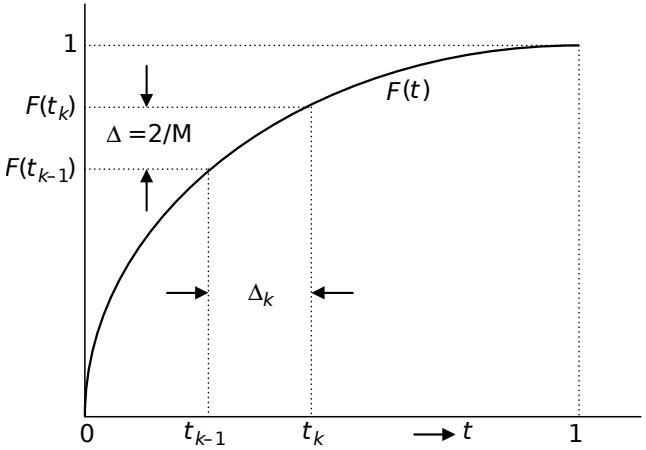


Figura 11: Exemplo de compressor (Ogundunmi and Narasimha, 2010).

Performance do Quantizador Não Uniforme III

Os limiares t_{k-1} e t_k, correspondentes a um codificador não uniforme, são mapeados através da função compressora $F(\cdot)$ nos limiares $F(t_{k-1})$ e $F(t_k)$, uniformemente espaçados. Supondo o sinal no intervalo $[-1, +1]$, teremos $\Delta = 2/M$, o passo do codificador uniforme. Conhecendo Δ e a derivada (inclinação) de $F(t)$ no intervalo $[t_{k-1}, t_k]$, podemos determinar Δ_k,

$$\Delta_k = \frac{\Delta}{F'(t_k^*)} = \frac{2}{MF'(t_k^*)}, \quad t_{k-1} < t_k^* < t_k. \tag{30}$$

Substituindo Δ_k na Equação 25, teremos

$$\sigma_q^2 = \frac{1}{3}\sum_{k=1}^{M} \frac{p_k}{M^2(F'(t_k^*))^2}, \quad t_{k-1} < t_k^* < t_k. \tag{31}$$

Se o número de níveis M for grande, o somatório em 31 poderá ser aproximado por uma integral:

$$\sigma_q^2 = \frac{1}{3M^2}\int_{-1}^{+1} \frac{p(x)}{(F'(x))^2}\mathrm{d}x = \frac{2}{3M^2}\int_{0}^{+1} \frac{p(x)}{(F'(x))^2}\mathrm{d}x, \tag{32}$$

Performance do Quantizador Não Uniforme IV

onde utilizamos a simplificação em que a $p(x)$ é simétrico par.
Para um sinal com excursão entre $-X_m$ e $+X_m$, teremos

$$\sigma_q^2 = \frac{2X_m^2}{3M^2} \int_0^{X_m} \frac{p(x)}{(F'(x))^2} \mathrm{d}x. \tag{33}$$

Compressão Logarítmica I

Em um sistema de telecomunicações, desejamos uma SNR constante, independente da distribuição do sinal de entrada. Desejamos então encontrar o compressor F que alcança este objetivo.

$$\mathsf{SNR} = \frac{\sigma_x^2}{\sigma_q^2} = \frac{2\int_0^1 x^2 p(x)\mathrm{d}x}{\frac{2}{3M^2}\int_0^1 \frac{p(x)}{(F'(x))^2}\mathrm{d}x}. \tag{34}$$

A expressão em 34 pode ser feita constante escolhendo

$$F'(x) = \frac{k^{-1}}{x}, \tag{35}$$

com parâmetro k a ser especificado.
A curva de compressão $F(x)$ é obtida realizando-se a integração e escolhendo a constante de integração para que a condição de contorno $F(1) = 1$ seja satisfeita, obtendo assim

$$F(x) = 1 + k^{-1}\ln x. \tag{36}$$

Compressão Logarítmica II

Para este caso a SNR obtida será

$$\text{SNR} = \frac{3M^2}{k^2}. \tag{37}$$

Para sinal com extensão de $-X_m$ a X_m, teremos

$$F(x) = X_m + k^{-1} \ln\left(\frac{x}{X_m}\right). \tag{38}$$

Compressão Logarítmica III

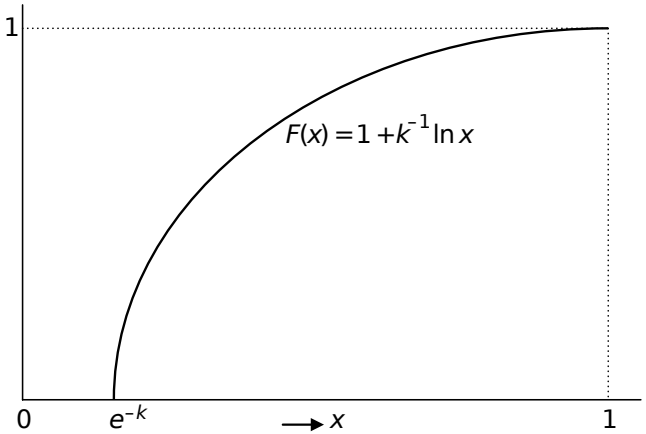


Figura 12: Gráfico de $F(x) = 1 + k^{-1} \ln x$ (Ogundunmi and Narasimha, 2010).

Compressão Logarítmica IV

Através da Equação 30 e da escolha feita em 35, podemos verificar que o tamanho do passo de quantização é proporcional à amplitude do sinal, quando utilizamos a compressão logarítmica dada por $F(x) = 1 + k^{-1} \ln x$.

$$\Delta_k = \frac{2}{MF'(t_k^*)} = \frac{2}{Mk^{-1}} t_k^*. \tag{39}$$

Para valores próximo da origem, esta proporcionalidade não poderá ser mantida, pois $\ln x$ diverge quando $x \to 0$.
Uma lei de compressão prática não pode ter tal descontinuidade, devendo também especificar a compressão dada a sinais de baixa amplitude.

lei μ I

Esta aproximação desloca o cruzamento com zero de $F(x)$, que ocorria em $x = e^{-k}$, para a origem.

$$F(x) = \frac{\log(1+\mu x)}{\log(1+\mu)}, \quad 0 \leq x \leq 1, \tag{40}$$

onde a base do logaritmo é irrelevante.

$$F(x) = \text{sign}(x)\frac{\log(1+\mu|x|)}{\log(1+\mu)}, \quad -1 \leq x \leq 1. \tag{41}$$

lei μ II

Note que, quando $\mu \gg 1$, esta lei aproxima uma curva logaritma para valores grandes.

$$F(x) = \frac{\log(1+\mu x)}{\log(1+\mu)} \approx \frac{\log(\mu x)}{\log(\mu)} = 1 + \frac{\log(x)}{\log(\mu)} = 1 + \frac{\ln x}{\ln \mu}. \tag{42}$$

Teremos então $k = \ln \mu$ e assim a SNR para sinais grandes será aproximadamente $3M^2/(\ln \mu)^2$.

lei μ III

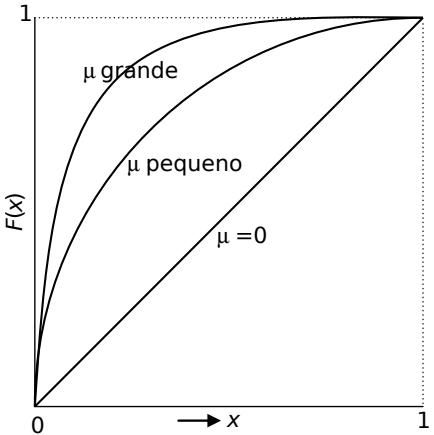


Figura 13: Curva de compressores segundo a lei μ (Ogundunmi and Narasimha, 2010).

lei μ IV

A lei μ é utilizada no padrão ITU G.711 PCM através de uma aproximação discreta usando $\mu = 255$.

lei A I

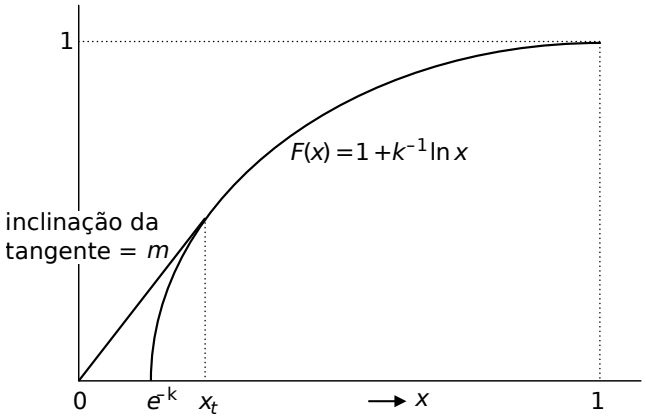


Figura 14: Curva de compressores segundo a lei A (Ogundunmi and Narasimha, 2010).

lei A II

$$k = 1 + \ln A. \tag{43}$$

$$F(x) = \begin{cases} \frac{Ax}{1+\ln A}, & 0 \leq x \leq 1/A, \\ \frac{1+\ln Ax}{1+\ln A}, & 1/A \leq x \leq 1. \end{cases} \tag{44}$$

Ogundunmi, T. and Narasimha, M. (2010). *Principles of Speech Coding*. CRC Press

Quantização Vetorial I

Utilizamos a quantização vetorial em casos em que o sinal de entrada já é um sinal digital e queremos obter uma representação comprimida da informação origial (em geral, representando os dados originais através de um *codebook*).

Considere uma variável aleatória X que assumule valores $x \in \mathcal{X} \subseteq R$, e $\mathcal{X}^n \subseteq R^n$ o conjunto de vetores $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in R^n$.

Quantização Vetorial II

Uma quantização vetorial é um mapeamento Q de vetores de entrada $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, onde $\mathbf{x} \in \mathcal{X}^n$, nos valores $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \mathcal{X}^n$ mais próximos (com relação a alguma medida de distorção), onde $\mathbf{y} \in \mathcal{Y} \subseteq \mathcal{X}^n$, sendo $\mathcal{Y} = \{y_1, y_2, \ldots, y_M\}$, um subconjunto constituído por M elementos em $\mathcal{X}^n$. O parâmetro n indica a dimensionalidade dos dados e do quantizador. O conujunto de aproximação $\mathcal{Y}$ é chamado de *codebook*.

Para se projetar um quantizador, devemos dividir o domínio $\mathcal{X}^n$ em M áreas ou células S_i, $i = 1, 2, \ldots, M$, de forma que $\bigcup_i S_i = \mathcal{X}^n$, $S_i \cap S_j = \emptyset$, $i \neq j$, e $y_i \in S_i$.

Erro de quantização médio I

O erro de quantização esperado de um quantizador é dado por

$$D_n(Q) = E\{d(\mathbf{x}, Q(\mathbf{x}))\} \tag{45}$$

Utilizando a distância Euclideana normalizada como métrica

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &= \frac{1}{n} d_E^2(\mathbf{x},\mathbf{y}) \\ &= \frac{1}{n}(\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^T \\ &= \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2 = \frac{1}{n}\|\mathbf{x}-\mathbf{y}\|^2 \end{aligned} \tag{46}$$

teremos

$$D_n(Q) = \frac{1}{n} E\{\|\mathbf{x} - Q(\mathbf{x})\|^2\} = \frac{1}{n}\sum_{i=1}^{M} E\{\|\mathbf{x} - \mathbf{y_i}\|^2\}. \tag{47}$$

Erro de quantização médio II

Considere que a pdf n-dimensional dos dados, $f(\mathbf{x})$, sobre o conjunto $\mathcal{X}^n$ seja conhecida, então a Equação 47 assume a forma

$$D_n(Q) = \frac{1}{n} \sum_{i=1}^{M} \int_{S_i} f(\mathbf{x}) \|\mathbf{x} - \mathbf{y_i}\|^2 \mathrm{d}\mathbf{x}, \tag{48}$$

e a probabilidade do vetor de representação $\mathbf{y_i}$ é dada por

$$P(\mathbf{y_i}) = \int_{S_i} f(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{49}$$

Taxa de quantização

A taxa de quantização R é o número de bits necessários para representar o vetor $\mathbf{x}$ (utilizando vetores de *codebook* de tamanho M) por dimensão, n. Para um quantizador com taxa fixa (em que cada símbolo é codificado por palavras de mesmo tamanho em um dado *codebook*), a taxa é dada por

$$R = \frac{\log_2 M}{n} \text{ bits/amostra.} \tag{50}$$

Para um quantizador de taxa variável, a taxa estará limitada pela entropia, ou seja,

$$R \geq -\frac{1}{n} \sum_{i=1}^{M} P(\mathbf{y_i}) \log_2 P(\mathbf{y_i}) \text{ bits/amostra.} \tag{51}$$

Quantização vetorial I

As células criadas por um quantizador vetorial em n-dimensões são regiões de Voronoi. O caso especial em que o *codebook* gera uma estrutura regular é chamado de quantizador vetorial em treliça (*lattice vector quantizers*).

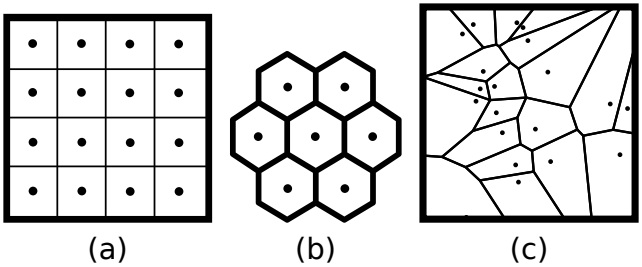


Figura 15: Diagramas de Voronoi. Estruturas em treliça em (a) e (b).

Exemplo Quantização - GNU Octave

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/quantization.ipynb

Exemplos de utilização

A quantização vetorial é utilizada em

- Video codecs: Cinepak, Sorenson codec, Indeo, VQA (utilizada em jogos)
- Audio codecs: CELP, G.729, TwinVQ, Ogg Vorbis, AMR-WB+, DTS

Quantização vetorial de cores e dithering

Araújo, L., Sansão, J., and Fasolo, S. (2018). Quantização vetorial de cores em imagens digitais.
In *Anais de XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*. Sociedade Brasileira de Telecomunicações

`https://biblioteca.sbrt.org.br/articles/807`

Alguns tópicos abordados no texto:

- quantização vetorial;
- espaços de cores;
- difusão de erro (*dithering*);
- dissimilaridade entre imagens.

Dithering

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/dithering.ipynb

Processamento digital de sinais analógicos

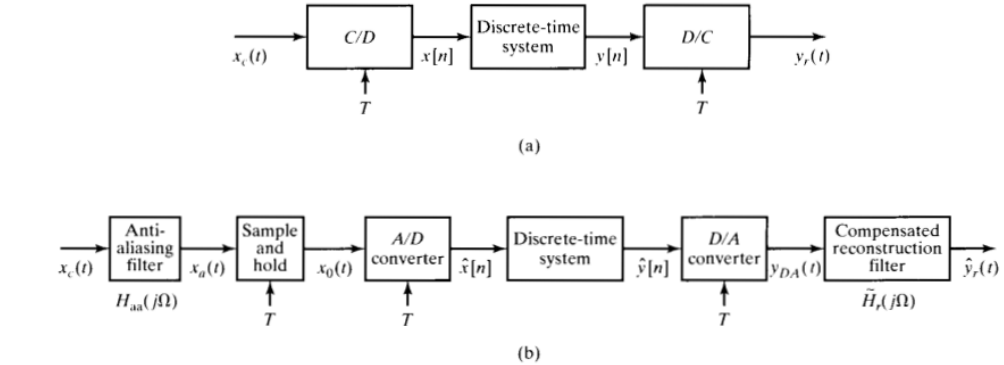


Figura 16: Processamento digital de sinais analógicos (Oppenheim, 2009).

Oppenheim, A. V. (2009). *Discrete-Time Signal Processing*.
Pearson

Figura 26: Processamento digital de sinais analógicos | Oppenheim, 2009).

Oppenheim, A. V. (2009). *Discrete-Time Signal Processing*. Pearson

Na prática temos que,

- os sinais contínuos não são estritamente limitados em frequência;
- filtros ideais não são realizáveis;
- os conversores ideais C/D e D/C são aproximações de conversores A/D (analógico-digital) e D/A (digital-analógico).

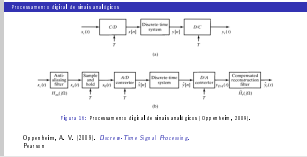


Figura 26: Processamento digital de sinais analógicos (Oppenheim, 2009).

Oppenheim, A. V. (2009). *Discrete-Time Signal Processing*. Pearson.

Devemos utilizar um filtro *anti-aliasing*.

- usualmente é desejável utilizar uma taxa de amostragem baixa;
- o próprio sinal e/ou ruído podem aparecer falseados como informação de baixa frequência;

Filtro *antialiasing* ideal

Resposta em frequência de um filtro *antialiasing* ideal:

$$H_{\text{aa}}(j\Omega) = \begin{cases} 1 & , |\Omega| < \Omega_c < \pi/T, \\ 0 & , |\Omega| > \Omega_c. \end{cases} \tag{52}$$

Processamento digital de sinais analógicos I

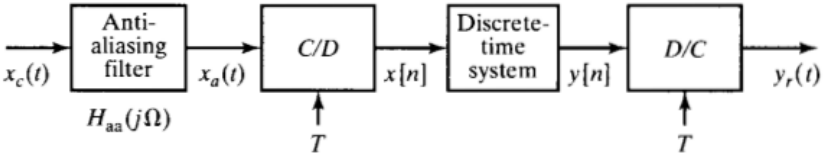


Figura 17: Processamento digital de sinais analógicos (Oppenheim, 2009).

Oppenheim, A. V. (2009). *Discrete-Time Signal Processing*.
Pearson
Considerando a entrada $x_a(t)$ e a saída $y_r(t)$, o sistema todo (compreendido entre entrada e saída) pode ser visto como um sistema linear invariante no tempo com resposta $H(e^{j\Omega T})$. Assim, a resposta total do sistema será

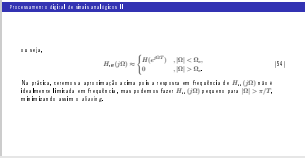
$$H_{\text{eff}}(j\Omega) \approx H_{aa}(j\Omega)H(e^{j\Omega T}), \tag{53}$$

Processamento digital de sinais analógicos II

ou seja,

$$H_{\text{eff}}(j\Omega) \approx \begin{cases} H(e^{j\Omega T}) & ,|\Omega| < \Omega_c, \\ 0 & ,|\Omega| > \Omega_c. \end{cases} \tag{54}$$

Na prática, teremos a aproximação acima pois a resposta em frequência de $H_{\text{aa}}(j\Omega)$ não é idealmente limitada em frequência, mas podemos fazer $H_{\text{aa}}(j\Omega)$ pequeno para $|\Omega| > \pi/T$, minimizando assim o *aliasing*.


Processamento digital de sinais analógicos II

ou seja,

$$H_{ar}(j\Omega) = \begin{cases} H(e^{j\Omega T}) & |\Omega| < \Omega_c \\ 0 & |\Omega| > \Omega_c \end{cases} \quad (54)$$

Na prática, conseguimos aproximações ainda pela resposta em frequência de $H_{ar}(j\Omega)$ não é idealmente limitada em frequência, mas podemos fazer $H_{ar}(j\Omega)$ pequeno para $|\Omega| > \pi/T$, minimizando assim o aliasing.

Filtros abruptos são de difícil implementação e alto custo. Além disso, geralmente possuem resposta em fase altamente não-linear. Para que o sistema opere com diferentes taxas de amostragem, devemos ter filtros ajustáveis.

Utilizando uma conversão A/D com sobre-amostragem para simplificar o filtro analógico *antialiasing* I

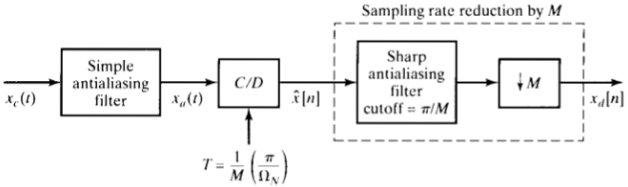


Figura 18: Utilizando uma conversão A/D com sobre-amostragem para simplificar o filtro analógico *antialiasing* (Oppenheim, 2009).

Oppenheim, A. V. (2009). *Discrete-Time Signal Processing*. Pearson

Ω_N : frequência mais alta que desejamos manter

M : fator de sobre-amostragem

Sobre-amostragem e decimação I

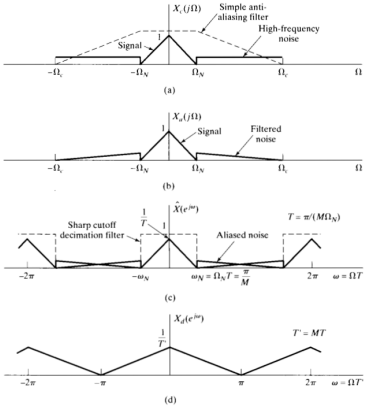
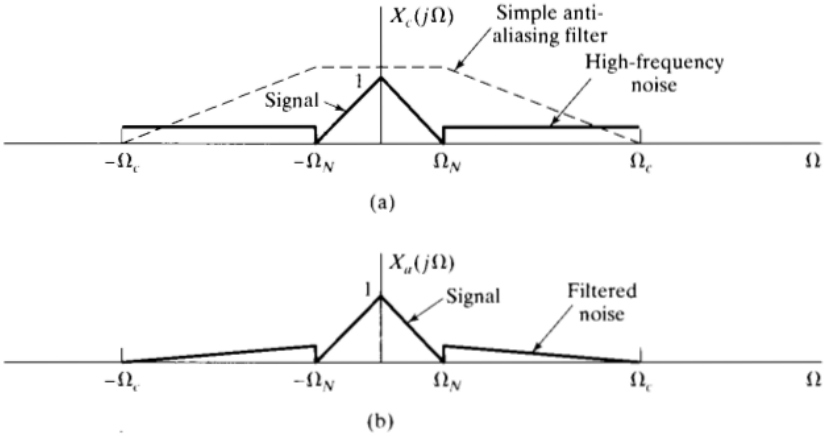
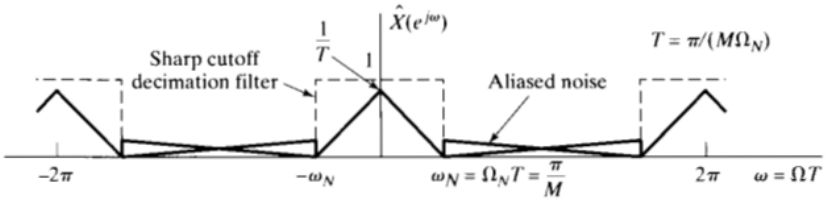


Figura 19: Exemplo (Oppenheim, 2009)

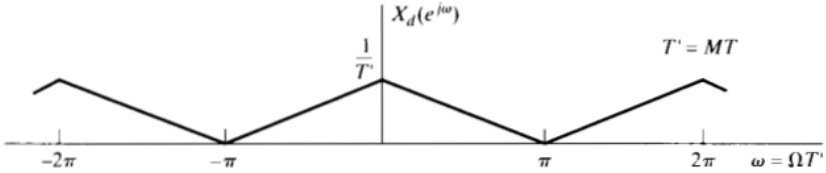
Sobre-amostragem e decimação II



Sobre-amostragem e decimação III



(c)



(d)

Configuração física para conversão analógico-digital

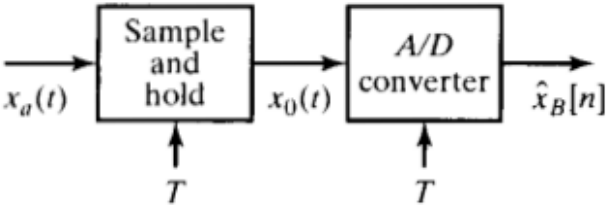


Figura 20: Configuração física para conversão analógico-digital (Oppenheim, 2009).

$x_a(t)$: sinal analógico

$\hat{x}_B[n]$: sinal digital (amostrado e quantizado)

T : período de amostragem

Representação de um *sample-and-hold* ideal I

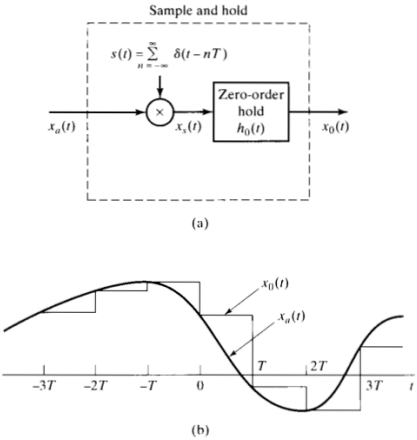


Figura 21: Sample-and-hold (Oppenheim, 2009).

Representação de um *sample-and-hold* ideal II

A saída de um *sample-and-hold* ideal é dada por

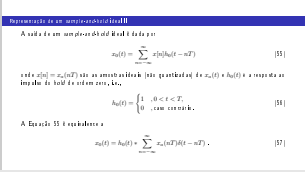
$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n] h_0(t - nT) \tag{55}$$

onde $x[n] = x_a(nT)$ são as amostras ideais (não quantizadas) de $x_a(t)$ e $h_0(t)$ é a resposta ao impulso do *hold* de ordem zero, i.e.,

$$h_0(t) = \begin{cases} 1 & , 0 < t < T, \\ 0 & , \text{caso contrário.} \end{cases} \tag{56}$$

A Equação 55 é equivalente a

$$x_0(t) = h_0(t) * \sum_{n=-\infty}^{\infty} x_a(nT)\delta(t - nT) \ . \tag{57}$$


Representação de um sample-and-hold ideal II

A saída de um sample-and-hold ideal é dada por

$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n] h_0(t - nT) \quad (55)$$

onde $x[n] = x_a(nT)$ são as amostras obtidas pela quantização de $x_a(t)$ e $h_0(t)$ é a resposta ao impulso do hold de ordem zero, i.e.,

$$h_0(t) = \begin{cases} 1 & , 0 < t < T, \\ 0 & , \text{caso contrário.} \end{cases} \quad (56)$$

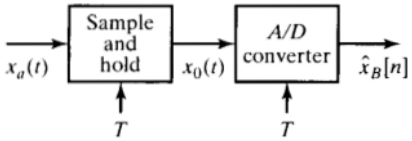
A Equação 55 é equivalente a

$$x_0(t) = h_0(t) * \sum_{n=-\infty}^{\infty} x_a(nT) \delta(t - nT) . \quad (57)$$

O circuito de uma *sample-and-hold* é projetado para amostrar $x_a(t)$ 'instantaneamente' e 'manter' o valor da amostra constante até que a próxima amostra seja tomada. Isto é necessário para fornecer uma tensão de entrada constante no conversor A/D.

Equivalência Conceitual

O sistema composto pelo *sample-and-hold* seguido por um conversor A/D



é equivalente ao seguinte sistema, composto por um conversor C/D ideal seguido por um quantizador

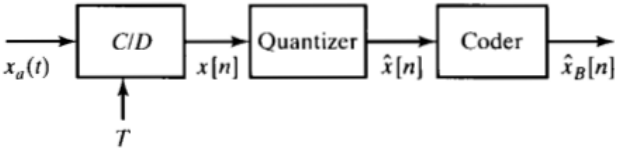


Figura 22: Sistema equivalente (Oppenheim, 2009).

Quantizador Uniforme I

Um quantizador é um sistema não-linear cuja operação é definida por uma função $Q(\cdot)$,

$$\hat{x}[n] = Q(x[n]). \tag{58}$$

Quantizador Uniforme II

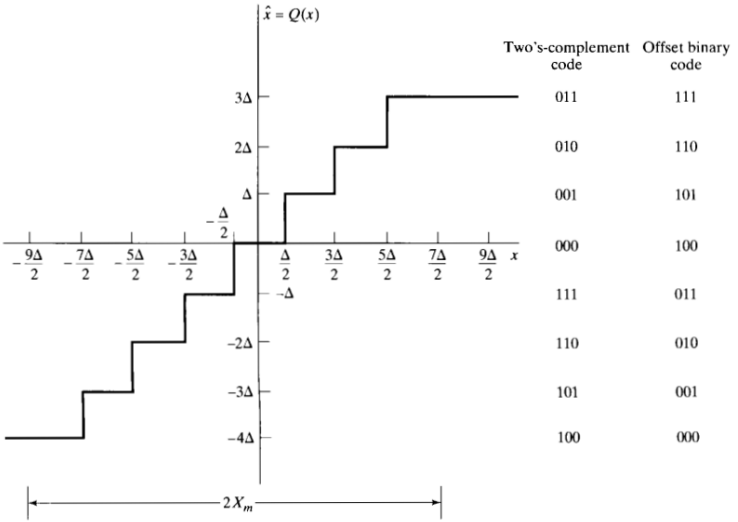


Figura 23: Quantizador uniforme de 3 bits (Oppenheim, 2009).

Erro de quantização I

O erro de quantização é definido por

$$e[n] = \hat{x}[n] - x[n] \ . \tag{62}$$

O erro de quantização satisfaz

$$\Delta/2 < e[n] \leq \Delta/2 \tag{63}$$

sempre que

$$(-X_m - \Delta/2) < x[n] \leq (X_m - \Delta/2) \ . \tag{64}$$

Se $x[n]$ estiver fora desta faixa, o erro de quantização será maior do que $\Delta/2$, e as amostras serão 'grampeadas'.

Modelo aditivo do erro de quantização I

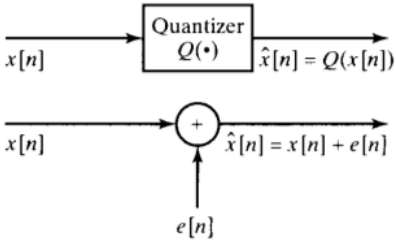


Figura 25: Modelo aditivo do erro de quantização (Oppenheim, 2009).

A representação estatística do erro de quantização é baseada nas seguintes suposições:

- $e[n]$ é um processo estocástico estacionário;
- $e[n]$ é descorrelacionada com $x[n]$;
- o erro é um ruído branco, suas amostras são descorrelacionadas
- a pdf do erro é uniforme

Modelo aditivo do erro de quantização II

Para Δ pequeno, podemos assumir que $e[n]$ é um ruído branco uniforme em $[-\Delta/2, \Delta/2]$.

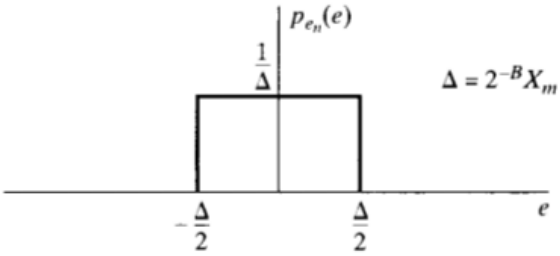


Figura 26: Modelo do ruído (Oppenheim, 2009).

média : $\mu_e = 0$;

Modelo aditivo do erro de quantização III

variância : $\sigma_e^2 = \Delta^2/12$.

$$\sigma_e^2 = \int_{-\Delta/2}^{\Delta/2} e^2 \frac{1}{\Delta} de = \frac{\Delta^2}{12} \ . \tag{65}$$

Para um quantizador de $(B+1)$ bits e fundo de escala X_m, a variância do ruído (ou potência) será dada por

$$\sigma_e^2 = \frac{2^{-2B} X_m^2}{12} \ , \tag{66}$$

onde $\Delta = X_m/2^B$.

Modelo aditivo do erro de quantização IV

A relação sinal-ruído de quantização para um quantizador com $(B+1)$ bits é

$$\begin{aligned} \text{SQNR} &= 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) = 10\log_{10}\left(\frac{12\cdot 2^{2B}\sigma_x^2}{X_m^2}\right) \\ &= 6.02B + 10.8 - 20\log_{10}\left(\frac{X_m}{\sigma_x}\right) . \end{aligned} \tag{67}$$

Aproximadamente 6dB para cada bit.

Conversão discreto-contínuo I

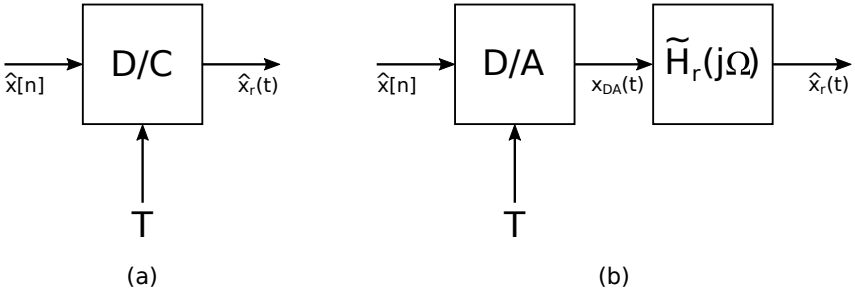


Figura 27: Conversão discreto para contínuo.

Conversão discreto-contínuo II

A reconstrução é representada por

$$X_r(j\Omega) = X(e^{j\Omega T})H_r(j\Omega) \tag{68}$$

onde $X(e^{j\Omega})$ é a transformada discreta de Fourier de $x[n]$ e $X_r(j\Omega)$ é a transformada de Fourier do sinal reconstruído. O filtro de reconstrução ideal é dado por

$$H_r(j\Omega) = \begin{cases} T & ,|\Omega| < \pi/T, \\ 0 & ,|\Omega| > \pi/T \ . \end{cases} \tag{69}$$

A relação entre $x_r(t)$ e $x[n]$ será dada por

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T} \ . \tag{70}$$

Conversão discreto-contínuo III

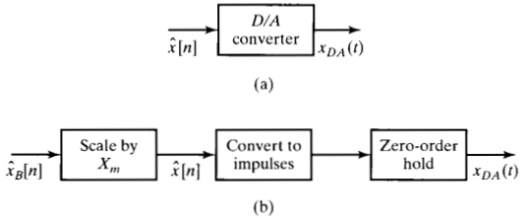


Figura 28: Modelo do conversor D/A (Oppenheim, 2009).

$$\begin{aligned} x_{\text{DA}}(t) &= \sum_{n=-\infty}^{\infty} X_m \hat{x}_B[n] h_0(t - nT) \\ &= \sum_{n=-\infty}^{\infty} \hat{x}[n] h_0(t - nT) \ . \end{aligned} \tag{71}$$

Conversão discreto-contínuo IV

Utilizando o modelo aditivo do ruído de quantização (ver Figura 25), podemos considerar $x_{\mathsf{DA}}(t)$ como composta em duas partes, uma devida ao sinal e outra devida ao ruído de quantização. Assim podemos analisar os efeitos da quantização.

$$x_{\mathsf{DA}}(t) = \sum_{n=-\infty}^{\infty} x[n]h_0(t-nT) + \sum_{n=-\infty}^{\infty} e[n]h_0(t-nT) \ . \tag{72}$$

Definimos

$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n]h_0(t-nT) \ , \text{ e} \tag{73}$$

$$e_0(t) = \sum_{n=-\infty}^{\infty} e[n]h_0(t-nT) \ , \tag{74}$$

de forma que

$$x_{\mathsf{DA}}(t) = x_0(t) + e_0(t) \ . \tag{75}$$

Conversão discreto-contínuo V

A transformada de Fourier da Equação 73 é

$$\begin{aligned} X_0(j\Omega) &= \sum_{n=-\infty}^{\infty} x[n] H_0(j\Omega) e^{-j\Omega nT} \\ &= \left(\sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega nT} \right) H_0(j\Omega) \\ &= X(e^{j\Omega T}) H_0(j\Omega) \ . \end{aligned} \tag{76}$$

Como

$$X(e^{j\Omega T}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a \left(j \left(\Omega - \frac{2\pi k}{T} \right) \right). \tag{77}$$

segue que

$$X_0(j\Omega) = \left[\frac{1}{T} \sum_{k=-\infty}^{\infty} X_a \left(j \left(\Omega - \frac{2\pi k}{T} \right) \right) \right] H_0(j\Omega) \ . \tag{78}$$

Filtro de Reconstrução I

Se $X_a(j\Omega)$ é limitado em frequência abaixo de π/T, as cópias deslocadas de $X_a(j\Omega)$ não se sobrepõem na Equação 78, e se definirmos o filtro de reconstrução compensado como

$$\tilde{H}_r(j\Omega) = \frac{H_r(j\Omega)}{H_0(j\Omega)}, \tag{79}$$

então, a saída do filtro será $x_a(t)$ se a entrada for $x_0(t)$. A resposta em frequência do *hold* de ordem zero é

$$H_0(j\Omega) = \frac{2\sin(\Omega T/2)}{\Omega} e^{-j\Omega T/2}. \tag{80}$$

Desta forma, o filtro de reconstrução compensado é dado por

$$\tilde{H}_r(j\Omega) = \begin{cases} \frac{\Omega T/2}{\sin(\Omega T/2)} e^{j\Omega T/2} & |\Omega| \leq \pi/T, \\ 0 & |\Omega| > \pi/T. \end{cases} \tag{81}$$

Filtro de Reconstrução II

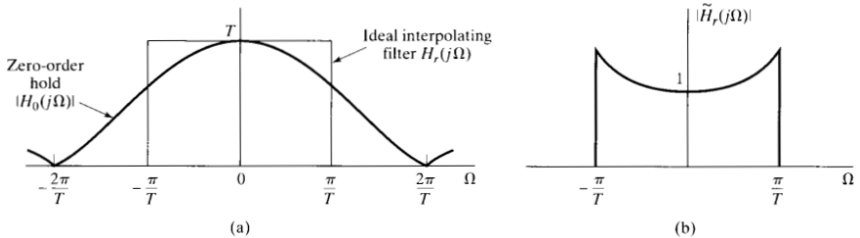


Figura 29: Filtro de reconstrução compensando o efeito do hold (Oppenheim, 2009).

Configuração física da conversão digital analógico I

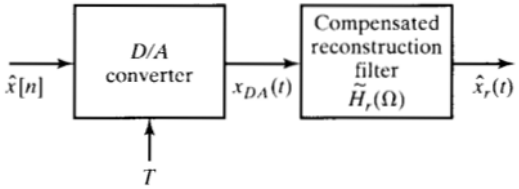


Figura 30: Configuração física para a conversão digital-analógico (Oppenheim, 2009).

Configuração física da conversão digital analógico II

O sinal reconstruído na saída é

$$\begin{aligned} \hat{x}_r(t) &= \sum_{n=-\infty}^{\infty} \hat{x}[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T} \\ &= \sum_{n=-\infty}^{\infty} x[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T} + \sum_{n=-\infty}^{\infty} e[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T}. \end{aligned} \tag{82}$$

Ou seja, a saída é dada por

$$\hat{x}_r(t) = x_a(t) + e_a(t), \tag{83}$$

onde $e_a(t)$ é o ruído branco limitado em frequência.

Sistema para processamento digital de sinais analógicos I

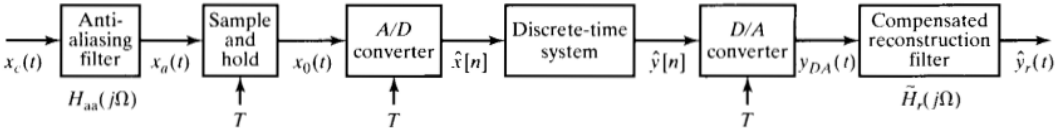


Figura 31: Sistema para processamento digital de sinais analógicos (Oppenheim, 2009).

$$\hat{y}_r(t) = y_a(t) + e_a(t) \tag{84}$$

$$Y_a(j\Omega) = \tilde{H}_r(j\Omega)H_0(j\Omega)H(e^{j\Omega T})H_{\text{aa}}(j\Omega)X_c(j\Omega) \tag{85}$$

onde

- $H_{\text{aa}}(j\Omega)$ filtro *antialiasing*
- $H_0(j\Omega)$ *hold* de ordem zero do conversor D/A

Sistema para processamento digital de sinais analógicos II

- $\tilde{H}_r(j\Omega)$ filtro passa-baixas de reconstrução

Assumindo que o ruído de quantização introduzido pelo conversor A/D é branco com variância $\sigma_e^2 = \Delta^2/12$, podemos mostrar que o espectro de densidade de potência do ruído na saída é

$$P_{e_a}(j\Omega) = |\tilde{H}_r(j\Omega)H_0(j\Omega)H(e^{j\Omega T})|^2\sigma_e^2. \tag{86}$$

A resposta em frequência efetiva total de $x_c(t)$ a $y_r(t)$ é

$$H_{\text{eff}}(j\Omega) = \tilde{H}_r(j\Omega)H_0(j\Omega)H(e^{j\Omega T})H_{\text{aa}}(j\Omega). \tag{87}$$

Oversampling e Noise-Shaping

Realizar uma sobre-amostragem (*oversampling*) e, subsequente, uma filtragem passa-baixas discreta e uma decimação (*down-sampling*) permite uma redução no número de bits do quantizador, para uma mesma relação sinal-ruído-de-quantização (SQNR[2]). Mantendo número de bits do quantizador, é possível reduzir a SQNR.

[2]Signal-to-Quantization-Noise Ratio

Conversão A/D com sobre-amostragem e quantização direta I

Considere o sinal de entrada $x_a(t)$:

- média nula
- estacionário no sentido amplo
- processo estocástico com densidade espectral de potência $\Phi_{x_a x_a}(j\Omega)$
- função de auto-correlação $\phi_{x_a x_a}(\tau)$
- limitado em frequência em Ω_N: $\Phi_{x_a x_a}(j\Omega) = 0$, $\Omega \geq \Omega_N$

Vamos assumir que $2\pi/T = 2M\Omega_N$. A constante inteira M é o fator de sobre-amostragem.

Conversão A/D com sobre-amostragem e quantização direta II

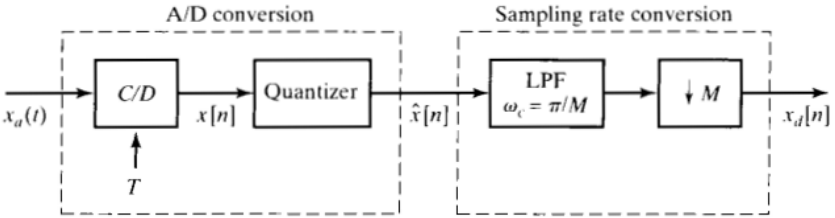


Figura 32: Conversão A/D com sobre-amostragem (Oppenheim, 2009).

Conversão A/D com sobre-amostragem e quantização direta III

Utilizando o modelo do ruído aditivo.

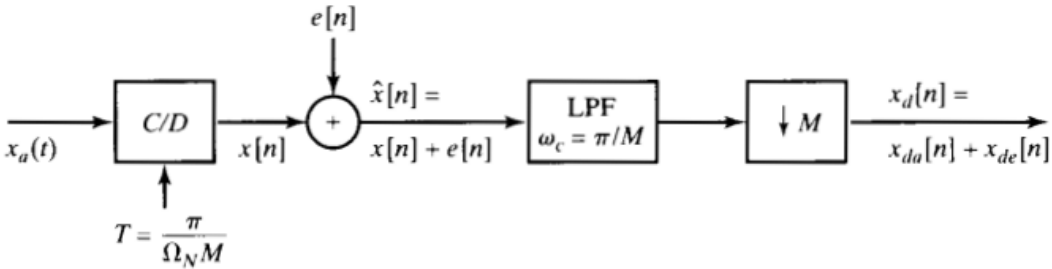


Figura 33: Modelo do ruído aditivo na conversão A/D com sobre-amostragem (Oppenheim, 2009).

A saída $x_d[n]$ possui duas componentes: $x_{da}[n]$ (devido ao sinal $x_a(t)$) e $x_{de}[n]$ (devido ao ruído $e[n]$).
Vamos analisar o efeito de cada componente na saída.

Conversão A/D com sobre-amostragem e quantização direta IV

Primeiramente vamos considerar o efeito da componente sinal.
$\phi_{xx}[m]$ e $\Phi(e^{j\omega})$ são autocorrelação e densidade espectral de potência de $x[n]$, respectivamente. Por definição

$$\phi_{xx}[m] = \varepsilon\{x[n+m]x[n]\}. \tag{88}$$

Como $x[n] = x_a(nT)$ e $x[n+m] = x_a(nT+mT)$

$$\varepsilon\{x[n+m]x[n]\} = \varepsilon\{x_a((n+m)T)x_a(nT)\}. \tag{89}$$

Assim

$$\phi_{xx}[m] = \phi_{x_a x_a}(mT) \tag{90}$$

i.e., a função de autocorrelação da sequência de amostras é a versão amostrada da função de autocorrelação do sinal contínuo correspondente.

Conversão A/D com sobre-amostragem e quantização direta V

Equações (89) e (90), juntamente com a suposição de estacionariedade no sentido amplo, levam a

$$\varepsilon\{x^2[n]\} = \varepsilon\{x_a^2(nT)\} = \varepsilon\{x_a^2(t)\} \quad \text{for all } n \text{ or } t. \tag{91}$$

Como as densidades espectrais de potência são transformadas de Fourier das funções de autocorrelação, como consequência da Equação (90) teremos

$$\Phi_{xx}(e^{j\Omega T}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} \Phi_{x_a x_a}\left(j\left(\Omega - \frac{2\pi k}{T}\right)\right). \tag{92}$$

Conversão A/D com sobre-amostragem e quantização direta VI

Assumindo um fator de sobre-amostragem M, tal que $2\pi/T = 2M\Omega_N$, substituindo $\Omega = \omega/T$ na Equação (92)

$$\Phi_{xx}(e^{j\omega}) = \begin{cases} \frac{1}{T}\Phi_{x_a x_a}\left(j\frac{\omega}{T}\right) & ,|\omega| < \pi/M, \\ 0 & ,\pi/M < \omega \leq \pi. \end{cases} \tag{93}$$

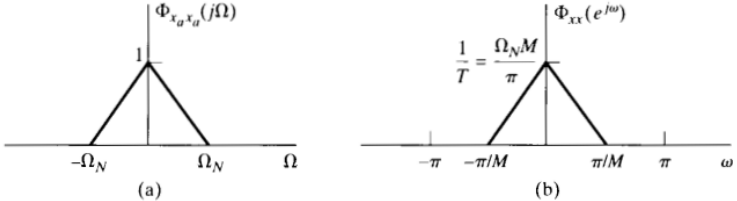


Figura 34: Densidade espectral de potência (Oppenheim, 2009).

Conversão A/D com sobre-amostragem e quantização direta VII

A potência total do sinal analógico é dada por

$$\varepsilon\{x_a^2(t)\} = \frac{2}{2\pi}\int_{-\Omega_N}^{\Omega_N} \Phi_{x_a x_a}(j\Omega)d\Omega. \tag{94}$$

Pela Equação (93), a potência total do sinal amostrado é

$$\begin{aligned}
\varepsilon\{x^2[n]\} &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_{xx}(e^{j\omega})d\omega \\
&= \frac{1}{2\pi}\int_{-\pi/M}^{\pi/M} \frac{1}{T}\Phi_{x_a x_a}\left(j\frac{\omega}{T}\right)d\omega \\
&= \frac{1}{2\pi}\int_{-\Omega_N}^{\Omega_N} \Phi_{x_a x_a}(j\Omega)d\Omega = \varepsilon\{x_a^2(t)\},
\end{aligned} \tag{95}$$

onde utilizamos $\Omega_N T = \pi/M$ e $\Omega = \omega/T$. Assim, a potência total do sinal amostrado é igual à potência total do sinal analógico.

Conversão A/D com sobre-amostragem e quantização direta VIII

Como $\Phi_{xx}(e^{j\omega})$ é limitado em frequência a $|\omega| < \pi/M$,

$$\begin{aligned}\Phi_{x_{da}x_{da}}(e^{j\omega}) &= \frac{1}{M}\sum_{k=0}^{M-1}\Phi_{xx}(e^{j(\omega-2\pi k)/M}) \\ &= \frac{1}{M}\Phi_{xx}(e^{j\omega/M}) \end{aligned} \quad (96)$$

A potência total da saída $x_{da}[n]$ é

$$\begin{aligned}\varepsilon\{x_{da}^2[n]\} &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\Phi_{x_{da}x_{da}}(e^{j\omega})d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{1}{M}\Phi_{xx}(e^{j\omega/M})d\omega \\ &= \frac{1}{2\pi}\int_{-\pi/M}^{\pi/M}\Phi_{xx}(e^{j\omega})d\omega = \varepsilon\{x^2[n]\}. \end{aligned} \quad (97)$$

A potência total da componente de sinal permanece a mesma enquanto atravessa todo o sistema.

Conversão A/D com sobre-amostragem e quantização direta IX

Considere agora a componente de ruído gerada pela quantização. Vamos assumir que $e[n]$ é ruído branco, estacionário no sentido amplo, e com variância

$$\sigma_e^2 = \frac{\Delta^2}{12}. \tag{98}$$

Consequentemente, a função de autocorrelação e densidade espectral de potência de $e[n]$ são dadas, respectivamente, por

$$\phi_{ee}[m] = \sigma_e^2 \delta[m] \tag{99}$$

e

$$\Phi_{ee}(e^{j\omega}) = \sigma_e^2 \quad |\omega| < \pi. \tag{100}$$

Conversão A/D com sobre-amostragem e quantização direta X

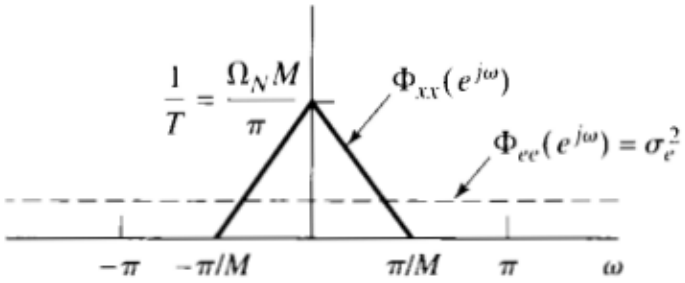


Figura 35: Densidade espectral de potência do sinal e do ruído (Oppenheim, 2009).

À medida que a taxa de sobre amostragem M aumenta, menor será a parte do espectro do ruído de quantização que se sobreporá ao espectro do sinal.

Conversão A/D com sobre-amostragem e quantização direta XI

O filtro passa-baixas ideal remove o ruído de quantização na banda $\pi/M < |\omega| \leq \pi$, enquanto deixa a componente de sinal inalterada. A potência do ruído na saída do filtro será

$$\varepsilon\{e^2[n]\} = \frac{1}{2\pi}\int_{-\pi/M}^{\pi/M} \sigma_e^2 d\omega = \frac{\sigma_e^2}{M}. \tag{101}$$

Conversão A/D com sobre-amostragem e quantização direta XII

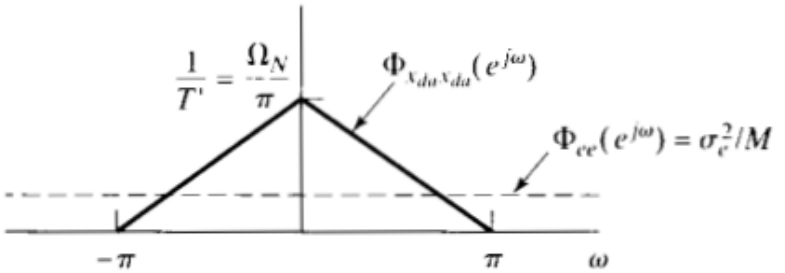


Figura 36: Densidades espectrais de potência do sinal e ruído, após a decimação (Oppenheim, 2009).

Conversão A/D com sobre-amostragem e quantização direta XIII

$$\varepsilon\{x_{de}^2\} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{\sigma_e^2}{M}d\omega = \frac{\sigma_e^2}{M} = \frac{\Delta^2}{12M}. \tag{102}$$

A potência do ruído de quantização $\varepsilon\{x_{de}^2[n]\}$ reduziu por um fator M através do filtro e decimação, enquanto a potência do sinal permaneceu inalterada.

Conversão A/D com sobre-amostragem e quantização direta XIV

Utilizando Equações (102) e (60) ($\Delta = X_m/2^B$), para uma dada potência de ruído de quantização, existe claramente uma relação de compromisso entre o fator de sobre-amostragem M e o passo de quantização Δ.

$$\varepsilon\{x_{de}^2[n]\} = \frac{1}{12M}\left(\frac{X_m}{2^B}\right)^2. \tag{103}$$

Fixando o quantizador, a potência do ruído pode ser diminuída aumentando o fator de sobre-amostragem M.

Conversão A/D com sobre-amostragem e quantização direta XV

Pela Equação (103), fixando a potência do ruído de quantização $P_{de} = \varepsilon\{x_{de}^2[n]\}$,

$$B = -\frac{1}{2}\log_2 M - \frac{1}{2}\log_2 12 - \frac{1}{2}\log_2 P_{de} + \log_2 X_m. \tag{104}$$

Para cada vez que dobrarmos o fator de sobre-amostragem M, precisaremos de $1/2$ bit a menos para obter a mesma relação sinal-ruído-de-quantização (para $M = 4$ podemos utilizar um bit a menos e obter a mesma acurácia na representação do sinal).

Conversão A/D com *Noise Shaping*

O objetivo em da técncia de *noise shaping* é modificar a conversão A/D para que a densidade espectral de potência do ruído de quantização não seja uniforme, de forma que a maior parte de sua potência fique fora da faixa $|\omega| < \pi/M$.

Conversão A/D com *Noise Shaping* I

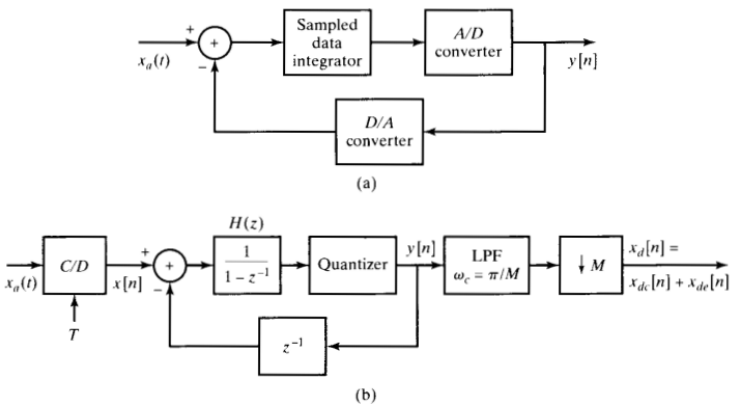


Figura 37: Sistema para conversão A/D com *oversampling* e *Noise Shaping* (Oppenheim, 2009).

Conversão A/D com *Noise Shaping* II

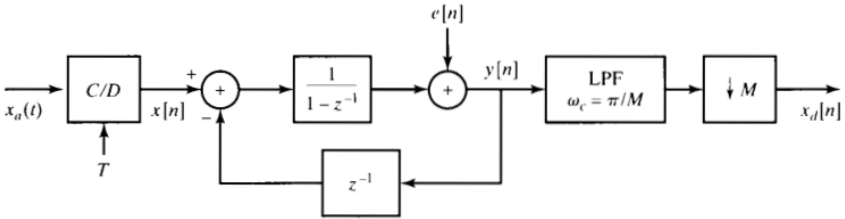


Figura 38: Modelo de ruído aditivo (Oppenheim, 2009).

Conversão A/D com *Noise Shaping* III

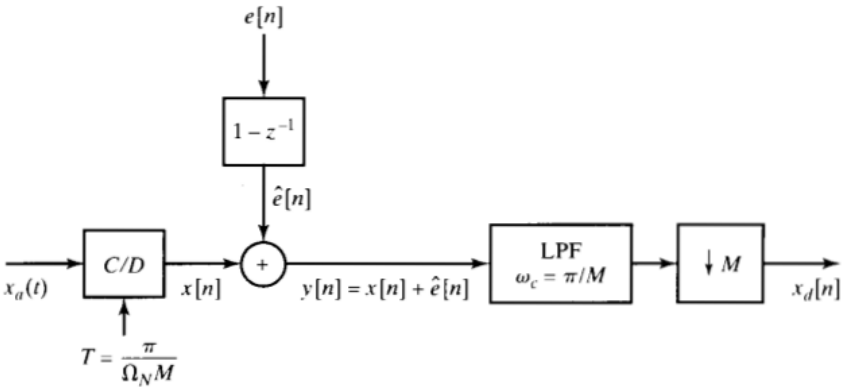


Figura 39: Simplificação (Oppenheim, 2009).

Conversão A/D com *Noise Shaping* IV

Sejam $H_x(z)$ a funcção de transferência de $x[n]$ para $y[n]$ e $H_e(z)$ a função de $e[n]$ para $y[n]$.

$$H_x(z) = 1 \tag{105}$$

$$H_e(z) = (1 - z^{-1}). \tag{106}$$

Consequently

$$y_x[n] = x[n] \tag{107}$$

$$\hat{e}[n] = e[n] - e[n-1] \tag{108}$$

A saída $y[n]$ poderá ser representada por

$$y[n] = x[n] + \hat{e}[n] \tag{109}$$

Conversão A/D com *Noise Shaping* V

A densidade espectral de potência do ruído de quantização $\hat{e}[n]$ presente em $y[n]$ é

$$\begin{aligned} \Phi_{\hat{e}\hat{e}}(e^{j\omega}) &= \sigma_e^2 |H_e(e^{j\omega})|^2 \\ &= \sigma_e^2 [2\sin(\omega/2)]^2. \end{aligned} \tag{110}$$

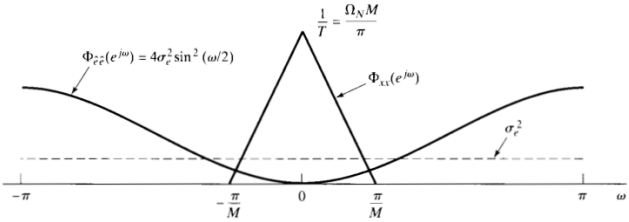


Figura 40: Densidade espectral de potência do ruído de quantização com *noise shaping* (Oppenheim, 2009).

Conversão A/D com *Noise Shaping* VI

Densidade espectral de potência após do *downsampling*.

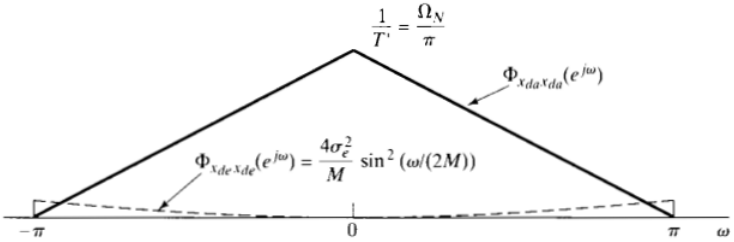


Figura 41: Densidade espectral de potência final (Oppenheim, 2009).

Conversão A/D com *Noise Shaping* VII

A potência do sinal $x_{da}[n]$ é

$$P_{da} = \varepsilon\{x_{da}^2[n]\} = \varepsilon\{x^2[n]\} = \varepsilon\{x_a^2(t)\}. \tag{111}$$

A potência do ruído de quantização final na saída é

$$P_{de} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_{x_{de}x_{de}}(e^{j\omega})d\omega = \frac{1}{2\pi}\frac{\Delta^2}{12M}\int_{-\pi}^{\pi}\left(2\sin\left(\frac{\omega}{2M}\right)\right)^2 d\omega \tag{112}$$

Podemos assumir que M é suficientemente grande e assim

$$\sin\left(\frac{\omega}{2M}\right) \approx \frac{\omega}{2M}, \tag{113}$$

Assim, poderemos considerar

$$P_{de} = \frac{1}{36}\frac{\Delta^2\pi^2}{M^3}. \tag{114}$$

Conversão A/D com *Noise Shaping* VIII

Usando Equação (114), teremos uma relação de compromisso entre o fator de sobre-amostragem M e o tamanho do passo de quantização Δ.

Para obter uma determinada potência P_{de} devemos ter

$$B = -\frac{3}{2}\log_2 M + \log_2(\pi/6) - \frac{1}{2}\log_2 P_{de} + \log_2 X_m. \tag{115}$$

Conversão A/D com *Noise Shaping* IX

TABLE 4.1 EQUIVALENT SAVINGS IN QUANTIZER BITS RELATIVE TO $M = 1$ FOR DIRECT QUANTIZATION AND FIRST-ORDER NOISE SHAPING

M	Direct quantization	Noise shaping
4	1	2.2
8	1.5	3.7
16	2	5.1
32	2.5	6.6
64	3	8.1

Figura 42: Economia de bits (Oppenheim, 2009).

Conversão A/D com *Noise Shaping* X

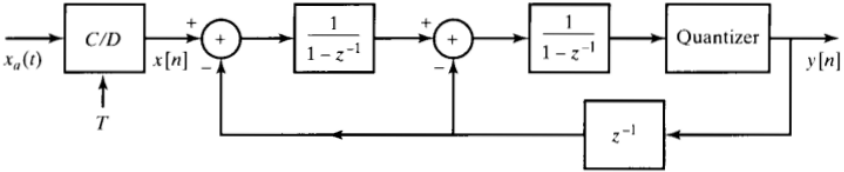


Figura 43: Estratégia incorporando um segundo estágio (Oppenheim, 2009).

Conversão A/D com *Noise Shaping* XI

Utilizando um segundo estágio ($p = 2$), a função de transferência será da forma

$$H_e(z) = (1 - z^{-1})^2, \tag{116}$$

e a densidade espectral de potência na saída em $y[n]$ será

$$\Phi_{\hat{e}\hat{e}}(e^{j\omega}) = \sigma_e^2[2\sin(\omega/2)]^4. \tag{117}$$

De forma geral, para p estágios, teremos

$$\Phi_{\hat{e}\hat{e}}(e^{j\omega}) = \sigma_e^2[2\sin(\omega/2)]^{2p}. \tag{118}$$

Conversão A/D com *Noise Shaping* XII

TABLE 4.2 REDUCTION IN QUANTIZER BITS AS ORDER p OF NOISE SHAPING

Quantizer order p	Oversampling factor M				
	4	8	16	32	64
0	1.0	1.5	2.0	2.5	3.0
1	2.2	3.7	5.1	6.6	8.1
2	2.9	5.4	7.9	10.4	12.9
3	3.5	7.0	10.5	14.0	17.5
4	4.1	8.5	13.0	17.5	22.0
5	4.6	10.0	15.5	21.0	26.5

Figura 44: Redução no número de bits no quantizador para diferentes configurações (Oppenheim, 2009).

Oversampling e *Noise Shaping* na Conversão D/A I

A técnica de *Oversampling* e *Noise Shaping* pode ser utilizada na conversão D/A quando utiliza-se um conversor D/A mais simples, com menos bits.

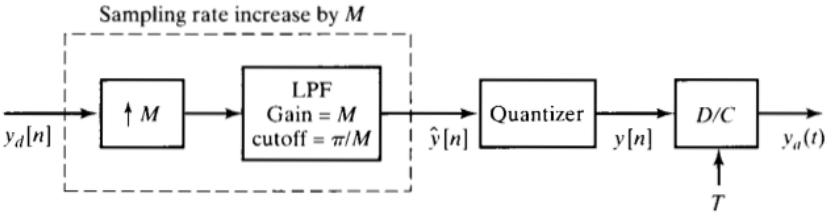


Figura 45: Utilização do *oversampling* com um conversor D/A simples (Oppenheim, 2009).

Oversampling e *Noise Shaping* na Conversão D/A II

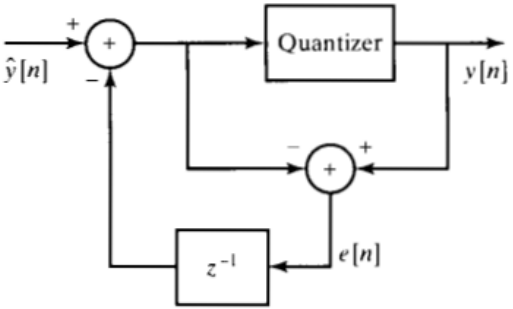


Figura 46: Sistema de *noise shaping* de primeira ordem (Oppenheim, 2009).

Oversampling e *Noise Shaping* na Conversão D/A III

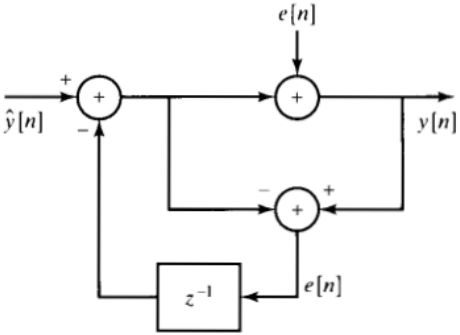


Figura 47: Modelo de ruído aditivo para o quantizador (Oppenheim, 2009).

Oversampling e *Noise Shaping* na Conversão D/A IV

A função de transferência de $\hat{y}[n]$ para $y[n]$ é $H_y(z) = 1$ e a função $H_e(z)$, de $e[n]$ para $y[n]$, é $H_e(z) = 1 - z^{-1}$.
A densidade espectral de potência, na saída, da componente relativa ao ruído de quantização $\hat{e}[n]$ será

$$\Phi_{\hat{e}\hat{e}}(e^{j\omega}) = \sigma_e^2 (2 \sin \omega/2)^2, \tag{119}$$

onde $\sigma_e^2 = \Delta^2/12$.

Oversampling e *Noise Shaping* na Conversão D/A V

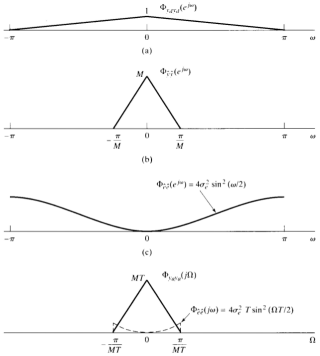


Figura 48: Densidade espectral de potência do sinal e ruído de quantização, na saída (Oppenheim, 2009).

Oversampling e *Noise Shaping* na Conversão D/A VI

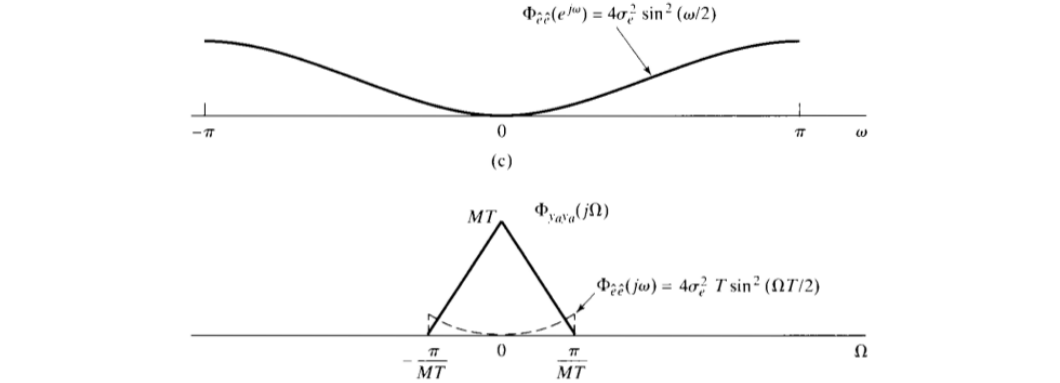


Figura 49: Detalhe: densidade espectral de potência do ruído de quantização, na saída (Oppenheim, 2009).

Oversampling e *Noise Shaping* na Conversão D/A VII

Aplicando a técnica em multi-estágio, podemos obter na saída um espectro da forma

$$\Phi_{\hat{e}\hat{e}}(e^{j\omega}) = \sigma_e^2 (2\sin\omega/2)^{2p}, \tag{120}$$

empurrando ainda mais o ruído para altas frequências, podendo assim relaxar as condições sobre o filtro de reconstrução.

Mudança da Frequência de Amostragem

Muitas vezes é necessário mudar a frequência de amostragem de um sinal

- downsample / decimate
- upsample / zero pad
- reamostragem
- fatores inteiros / racionais
- interpolação

Representação de um sinal contínuo através de um sinal discreto (sequencia de amostras).

$$x[n] = x_c(nT) \tag{121}$$

Muitas vezes é necessário alterar a frequência de amostragem de um sinal.

$$x'[n] = x_c(nT') \tag{122}$$

onde $T' \neq T$

Embora seja sempre possível reconstruir um sinal contínuo limitado em frequência e realizar uma nova amostragem deste sinal, queremos um processamento puramente discreto.

Downsample por um fator inteiro I

Podemos reduzir a frequência de amostragem de uma sequência 'amostrando'-a e assim gerando uma nova sequência.
Reduzir a frequência de amostragem por um fator M

$$x_d[n] = x[nM] = x_c(nMT) \tag{123}$$

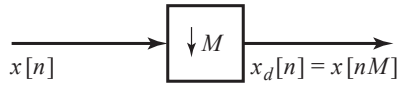


Figura 50: Representação de um compressor ou amostrador discreto. (Oppenheim, 2009).

Downsample por um fator inteiro II

A transformada discreta de Fourier de $x[n] = x_c(nT)$ é

$$X(e^{j\omega}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right). \tag{124}$$

A transformada discreta de Fourier de $x_d[n] = x[nM] = x_c(nT')$, onde $T' = MT$, é

$$X_d(e^{j\omega}) = \frac{1}{T'} \sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T'} - \frac{2\pi k}{T'}\right)\right) \tag{125}$$

$$= \frac{1}{MT} \sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi k}{MT}\right)\right) \tag{126}$$

Downsample por um fator inteiro III

Podemos fazer o índice do somatório da seguinte forma

$$r = i + kM \tag{127}$$

onde k e i são inteiros tais que $-\infty < k < \infty$ e $0 \leq i \leq M-1$. Podemos assim reescrever a equação anterior na seguinte forma:

$$X_d(e^{j\omega}) = \frac{1}{M} \sum_{i=0}^{M-1} \left[\frac{1}{T} \sum_{k=-\infty}^{\infty} X_c \left(j \left(\frac{\omega}{MT} - \frac{2\pi k}{T} - \frac{2\pi i}{MT} \right) \right) \right] \tag{128}$$

$$= \frac{1}{M} \sum_{i=0}^{M-1} X \left(e^{j(\omega - 2\pi i)/M} \right). \tag{129}$$

Teremos M cópias de $X(e^{j\omega})$ dilatadas por um fator M e transladadas por $2\pi i/M$.

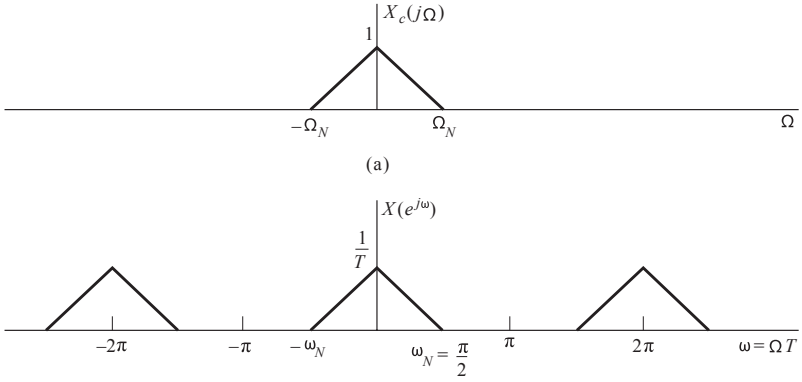
Downsample por um fator inteiro IV

Não haverá *aliasing* se $X(e^{j\omega})$ for limitado em frequência

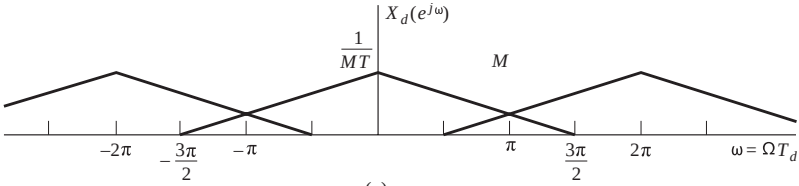
$$X(e^{j\omega}) = 0, \quad \omega_N \leq |\omega| \leq \pi, \tag{130}$$

e $2\pi/M \geq 2\omega_N$.

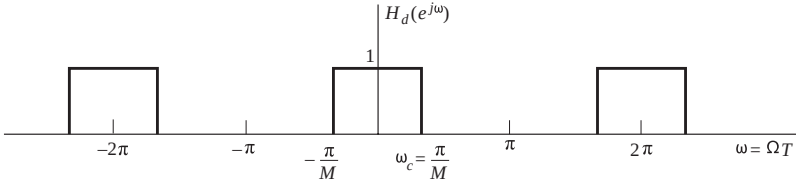
Downsample por um fator inteiro V



Downsample por um fator inteiro VI

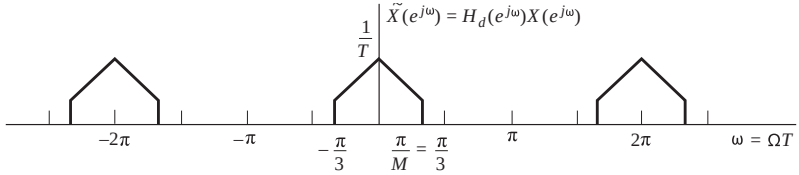


(c)

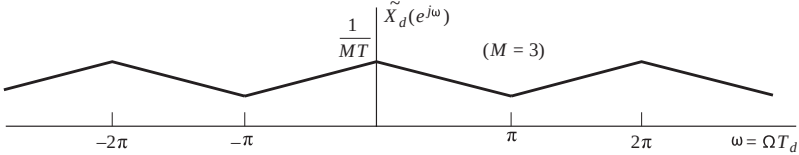


(d)

Downsample por um fator inteiro VII



(e)



(f)

Downsample por um fator inteiro VIII

De forma geral, para evitar *aliasing* ao realizar um *downsample* por um fator M é necessário que

$$\omega_N M \leq \pi \qquad \text{ou} \qquad \omega_N \leq \pi/M. \tag{131}$$

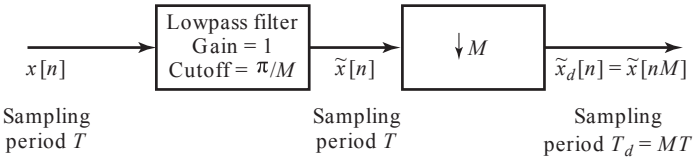


Figura 51: Sistema geral para redução da frequência de amostragem por um fator M. (Oppenheim, 2009).

upsample por um fator inteiro I

Considere o sinal $x[n]$ o qual queremos aumentar a frequência de amostragem por um fator L. Vamos considerar o sinal contínuo subjacente $x_c(t)$. O objetivo é obter as amostras

$$x_i[n] = x_c(nT') \tag{132}$$

onde $T' = T/L$, a partir das amostras

$$x[n] = x_c(nT.) \tag{133}$$

Das equações acima, segue que

$$x_i[n] = x[n/L] = x_c(nT/L), \quad 0, \pm L, \pm 2L, \ldots \tag{134}$$

upsample por um fator inteiro II

Sistema para aumentar a frequência de amostragem por um fator L.

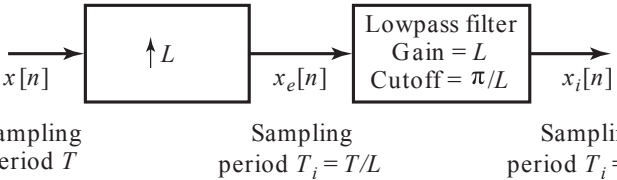


Figura 52: Sistema geral para aumentar a frequência de amostragem por um fator L. (Oppenheim, 2009).

upsample por um fator inteiro III

O sistema da esquerda é chamado expansor. Sua saída será

$$x_e[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \dots \\ 0, & \text{caso contrário,} \end{cases} \tag{135}$$

ou de forma equivalente,

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n - kL]. \tag{136}$$

A transformada de Fourier de $x_e[n]$ pode ser expressa como

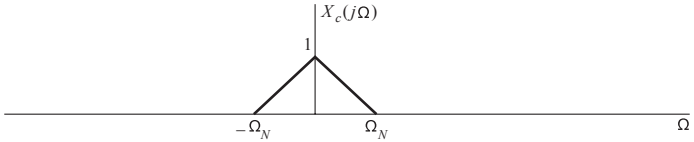
$$X_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} x[k]\delta[n - kL] \right) e^{-j\omega n} \tag{137}$$

$$= \sum_{k=-\infty}^{\infty} x[k]e^{-j\omega Lk} = X(e^{j\omega L}) \tag{138}$$

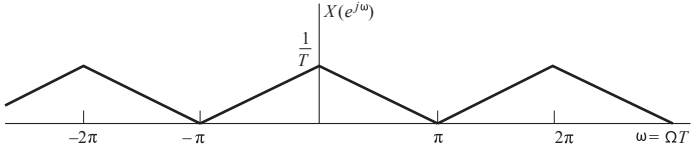
upsample por um fator inteiro IV

Ou seja, $X_e(e^{j\omega})$ é obtido através de $X(e^{j\omega})$ por uma compressão de fator L.
$X_i(e^{j\omega})$ pode ser obtido a partir de $X_e(e^{j\omega})$ realizando a correção da amplitude de $1/T$ para $1/T'$ e removendo todas as réplicas escalonadas de $X_c(j\Omega)$, exceto aquelas nos múltiplos de 2π.

upsample por um fator inteiro V

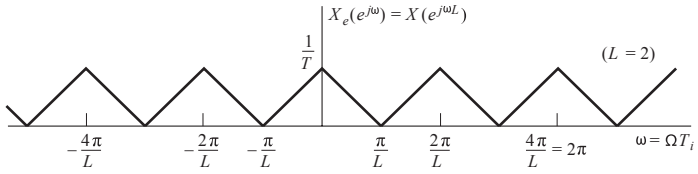


(a)

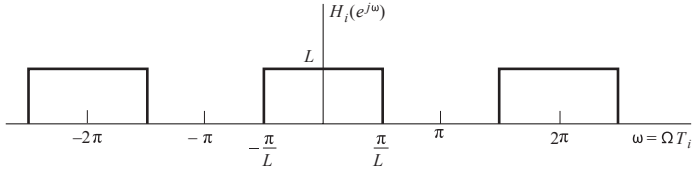


(b)

upsample por um fator inteiro VI

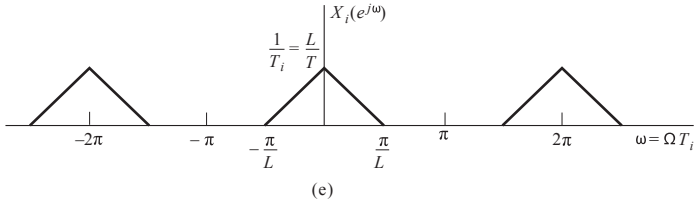


(c)



(d)

upsample por um fator inteiro VII



(e)

O interpolador será um filtro passa-baixas com frequência de corte π/L e ganho L. Sua resposta ao impulso será

$$h_i[n] = \frac{\sin(\pi n/L)}{\pi n/L}. \tag{139}$$

Obteremos então

$$x_i[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{\sin[\pi(n-kL)/L]}{\pi(n-kL)/L} \tag{140}$$

upsample por um fator inteiro VIII

Note que $h_i[n]$ possui as seguintes propriedades

$$h_i[0] = 1, \tag{141}$$
$$h_i[n] = 0, \quad n = \pm L, \pm 2L, \ldots \tag{142}$$

Teremos assim

$$x_i[n] = x[n/L] = x_c(nT/L) = x_c(nT'), \quad n = 0, \pm L, \pm 2L, \ldots \tag{143}$$

Interpolador Linear I

Apenas a título de comparação, vamos analisar um outro interpolador muito comum: o interpolador linear.
Resposta ao impulso de um interpolador linear:

$$h_{\text{lin}} = \begin{cases} 1 - |n|/L, & |n| \leq L, \\ 0, & \text{caso contrário.} \end{cases} \tag{144}$$

A saída do filtro interpolador será

$$x_{\text{lin}}[n] = \sum_{k=n-L+1}^{n+L-1} x_e[k] h_{\text{lin}}[n-k]. \tag{145}$$

onde $x_e[n]$ é a saída de um *upsample* por um fator L. Note que as amostras originais são preservadas pois $h_{\text{lin}}[0] = 1$ e $h_{\text{lin}}[n] = 0$ para $|n| \geq L$.

Interpolador Linear II

A resposta em frequência do interpolador linear é dada por

$$H_{\text{lin}}(e^{j\omega}) = \frac{1}{L}\left[\frac{\sin(\omega L/2)}{\sin(\omega/2)}\right]^2. \tag{146}$$

Interpolador Linear III

Para $L = 5$ teremos o seguinte exemplo.

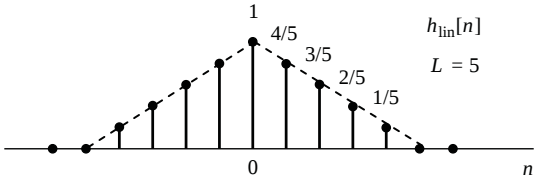


Figura 53: Resposta ao impulso do interpolador linear com $L = 5$. (**?**).

Interpolador Linear IV

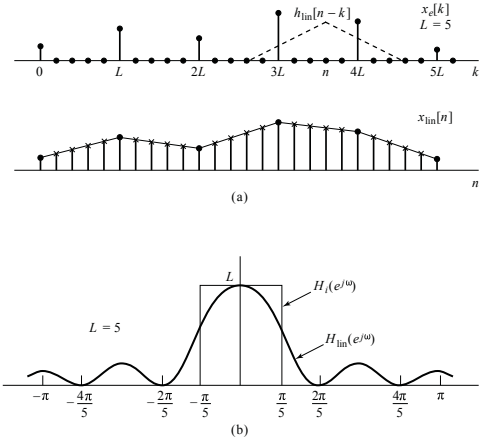


Figura 54: Filtragem com o interpolador linear e sua resposta em frequência. (Oppenheim, 2009).

Mudança da frequência de amostragem por um fator não inteiro I

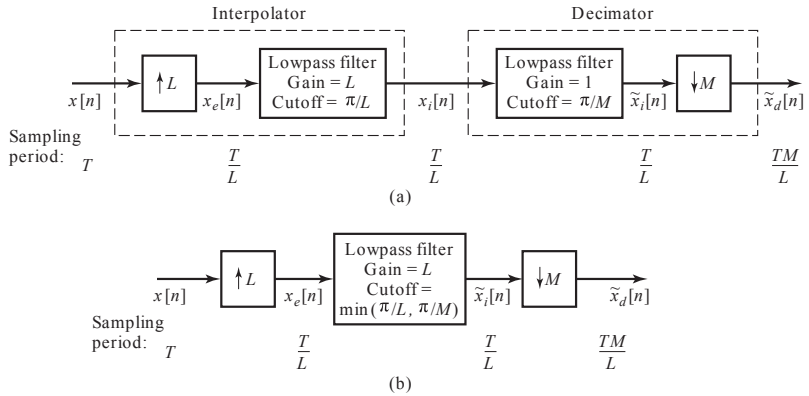


Figura 55: Sistema para mudar a frequência de amostragem por um fator não inteiro. (Oppenheim, 2009).

Mudança da frequência de amostragem por um fator não inteiro II

Exemplo (Mudança da frequência de amostragem por um fator nao inteiro)

Suponha um sinal $X_c(j\Omega)$ limitado em frequência, conforme ilustrado. Este sinal é amostrado à taxa de Nyquist, $2\pi/T = 2\Omega_N$. A DTFT resultante é

$$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right) \tag{147}$$

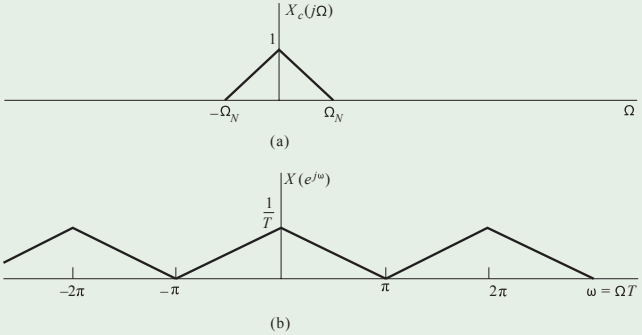
também ilustrado abaixo. Para mudar o período de amostragem para $(3/2)T$, iremos primeiramente interpolar por um fator $L = 2$ e depois decimar por um fator $M = 3$.

...

Mudança da frequência de amostragem por um fator não inteiro III

Exemplo (Mudança da frequência de amostragem por um fator nao inteiro)

continuação...

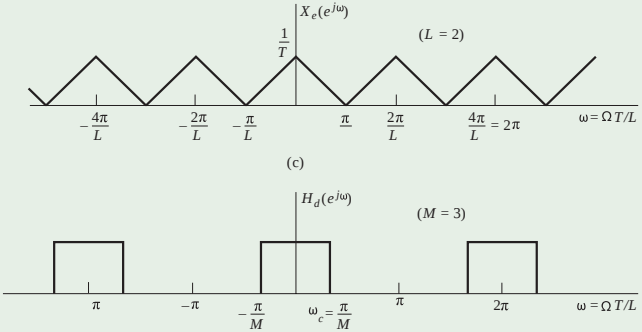


...

Mudança da frequência de amostragem por um fator não inteiro IV

Exemplo (Mudança da frequência de amostragem por um fator nao inteiro)

continuação...

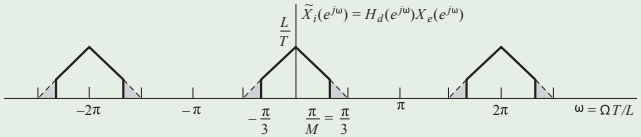


...

Mudança da frequência de amostragem por um fator não inteiro V

Exemplo (Mudança da frequência de amostragem por um fator nao inteiro)

continuação...



Resample

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/resample.ipynb

Processamento Multi-taxa de Sinais I

O processamento de sinais a multi-taxas refere-se à utilização de *upsampling*, *downsampling*, compressores e expansores em uma variedade de maneiras para aumentar a eficiência do sistema de processamento de sinais.

Intercâmbio entre filtro e downsampling/upsampling I

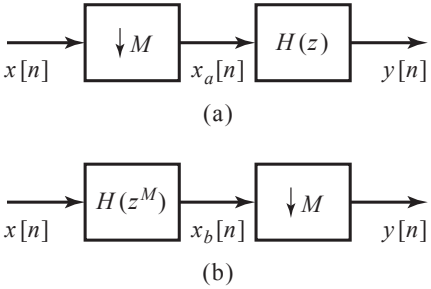


Figura 56: Dois sistemas equivalentes, com base nas identidades do *downsample*. (Oppenheim, 2009).

Intercâmbio entre filtro e downsampling/upsampling II

Analisando o sistema (b) temos:

$$X_b(e^{j\omega}) = H(e^{j\omega M})X(e^{j\omega}) \tag{148}$$

e

$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X_b\left(e^{j(\omega/M - 2\pi i/M)}\right). \tag{149}$$

logo

$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j(\omega/M - 2\pi i/M)}\right) H\left(e^{j(\omega - 2\pi i)}\right). \tag{150}$$

Intercâmbio entre filtro e downsampling/upsampling III

Como $H\left(e^{j(\omega-2\pi i)}\right) = H(e^{j\omega})$, poderemos simplificar, ficando

$$Y(e^{j\omega}) = H(e^{j\omega})\frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j(\omega/M-2\pi i/M)}\right) \tag{151}$$

$$= H(e^{j\omega})X_a(e^{j\omega}), \tag{152}$$

o que corresponde ao sistema (a) da figura.

Intercâmbio entre filtro e downsampling/upsampling IV

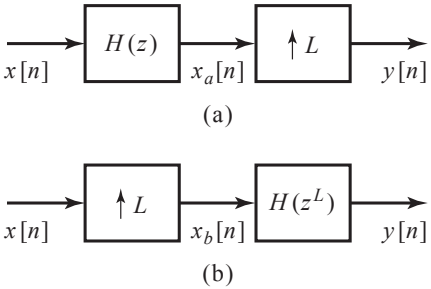


Figura 57: Dois sistemas equivalentes, com base nas identidades do *upsample*. (Oppenheim, 2009).

Intercâmbio entre filtro e downsampling/upsampling V

$$Y(e^{j\omega}) = X_a(e^{j\omega L}) \tag{153}$$
$$= X(e^{j\omega L})H(e^{j\omega L}) \tag{154}$$

$$X_b(e^{j\omega}) = X(e^{j\omega L}) \tag{155}$$

$$Y(e^{j\omega}) = H(e^{j\omega L})X_b(e^{j\omega}) \tag{156}$$

Decimação e Interpolação com múltiplos estágios I

Quando a decimação e interpolação envolvem fatores grandes, utiliza-se filtros com resposta ao impulso longa. É possível reduzir significativamente o custo computacional utilizando múltiplos estágios.

Decimação e Interpolação com múltiplos estágios II

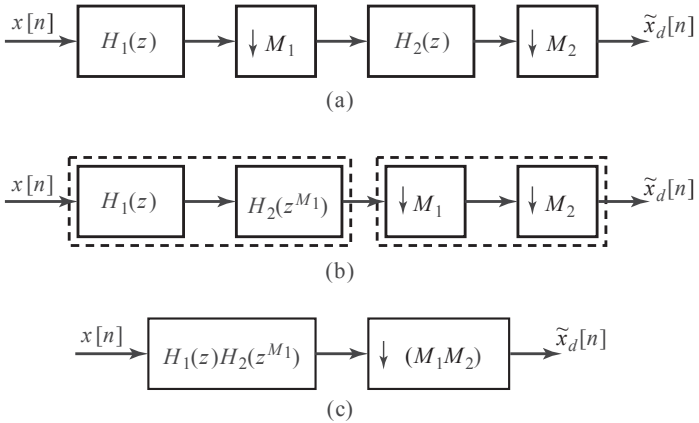


Figura 58: Decimação em múltiplos estágios. (Oppenheim, 2009).

Decimação e Interpolação com múltiplos estágios III

O fator final de decimação é $M = M_1 M_2$. Para realizar a decimação pelo fato M seria necessário um filtro com frequência de corte $\pi/M = \pi/M_1 M_2$. Este filtro é bem mais estreito que os filtros $H_1(z)$ e $H_2(z)$ com frequência de corte π/M_1 e π/M_2, respectivamente. Filtros com banda mais estreita requerem sistemas de ordem maior para obter transição mais abrupta. Desta forma, a implementação em dois estágios será mais eficiente.

Decimação e Interpolação com múltiplos estágios IV

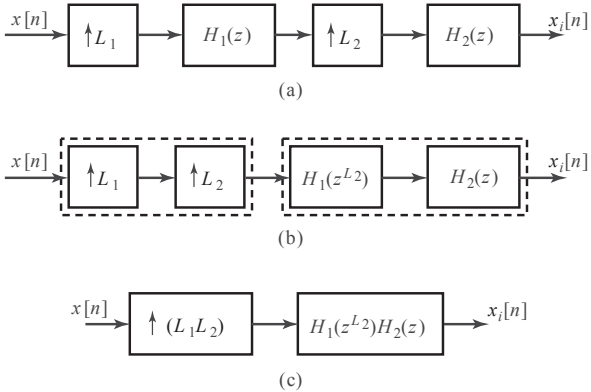


Figura 59: Interpolação em múltiplos estágios. (Oppenheim, 2009).

Decomposição Polifásica I

A resposta ao impulso $h[n]$ pode ser decomposta em M subsequentes $h_k[n]$.

$$h_k[n] = \begin{cases} h[n+k], & n = \text{múltiplo inteiro de } M, \\ 0, & \text{caso contrário.} \end{cases} \tag{157}$$

$$h[n] = \sum_{k=0}^{M-1} h_k[n-k] \tag{158}$$

Decomposição Polifásica II

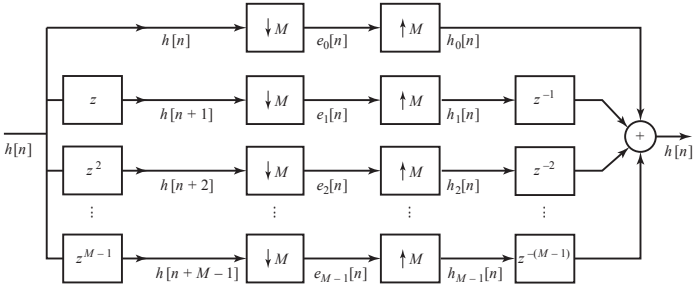


Figura 60: Decomposição polifásica de $h[n]$ usando as componentes $e_k[n]$. (Oppenheim, 2009).

As sequências $e_k[n]$ são dadas por

$$e_k[n] = h[nM + k] = h_k[nM] \tag{159}$$

e são chamadas de componentes polifásicas de $h[n]$.

Decomposição Polifásica III

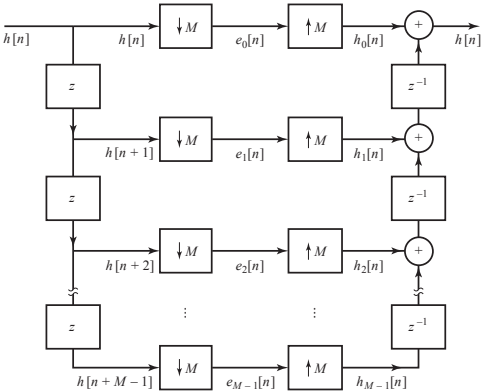


Figura 61: Representação com atrasos em cadeia. (Oppenheim, 2009).

Decomposição Polifásica IV

Representando no domínio da frequência ou no domínio z, teremos

$$Z\{h[n]\} = Z\{\sum_{k=0}^{M-1} h_k[n-k]\} \tag{160}$$

$$H(z) = \sum_{k=0}^{M-1} Z\{h_k[n-k]\} \tag{161}$$

$$= \sum_{k=0}^{M-1} Z\{h_k[n]\}z^{-k} \tag{162}$$

$$= \sum_{k=0}^{M-1} E_k(z^M)z^{-k} \tag{163}$$

onde utilizamos o fato de que $e_k[n] = h_k[nM]$.

Decomposição Polifásica V

Representamos então $H(z)$ como uma soma de filtros polifásico atrasados.

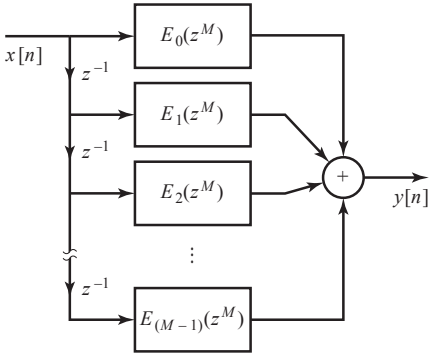


Figura 62: Estrutura da decomposição polifásica de $h[n]$. (Oppenheim, 2009).

Implementação de decimadores usando decomposição polifásica I

Uma aplicação importante da decomposição polifásica é na implementação de filtros cuja saída é seguida por um *downsample* (note que 1 em cada M amostras geradas pelo filtro é mantida e as demais $M-1$ são descartadas).

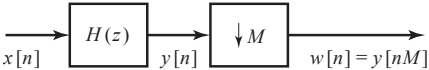


Figura 63: Sistema de decimadão. (Oppenheim, 2009).

Suponha que $H(z)$ é um filtro FIR com N pontos. Serão necessárias N multiplicações e $(N-1)$ adições por unidade de tempo para calcular a saída.
Suponha que $h[n]$ seja expresso em componentes polifásicas

$$e_k[n] = h[nM + k] \tag{164}$$

Implementação de decimadores usando decomposição polifásica II

e assim podemos representar $H(z)$ como

$$H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k} \tag{165}$$

Implementação de decimadores usando decomposição polifásica III

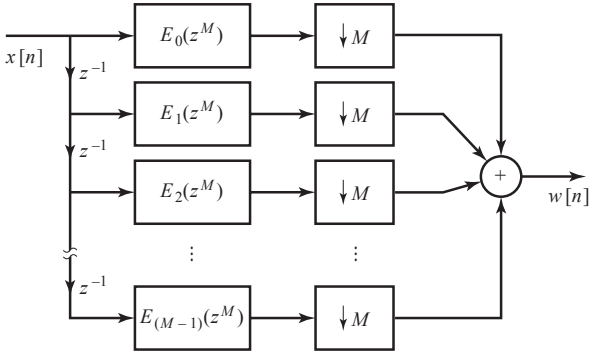


Figura 64: Implementação utilizando decomposição polifásicas. (Oppenheim, 2009).

Implementação de decimadores usando decomposição polifásica IV

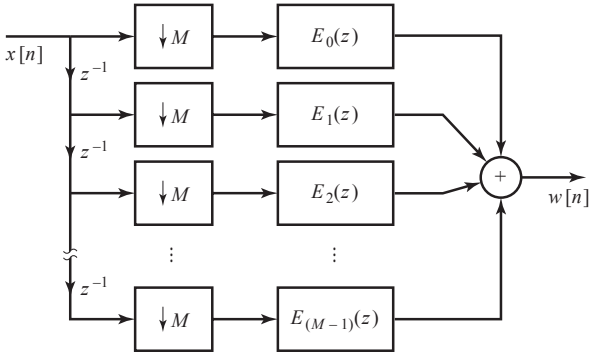


Figura 65: Implementação após utilizar a identidade do *downsapling*. (Oppenheim, 2009).

Implementação de decimadores usando decomposição polifásica V

Os filtros $E_k(z)$ possuem comprimento N/M e estão a uma taxa $1/M$ em relação ao original. Consequentemente, cada filtro fará $\frac{1}{M}\left(\frac{N}{M}\right)$ multiplicações e $\frac{1}{M}\left(\frac{N}{M}-1\right)$ adições por unidade de tempo. Como são M componentes polifásicas o sistema requererá N/M multiplicações e $\left(\frac{N}{M}-1\right)+(M-1)$ adições por unidade de tempo.

Implementação de interpoladores usando decomposição polifásica I

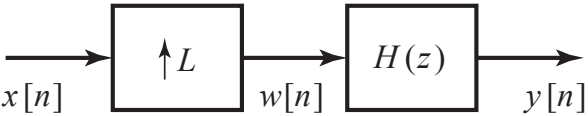


Figura 66: Sistema de interpolação. (Oppenheim, 2009).

Implementação de interpoladores usando decomposição polifásica II

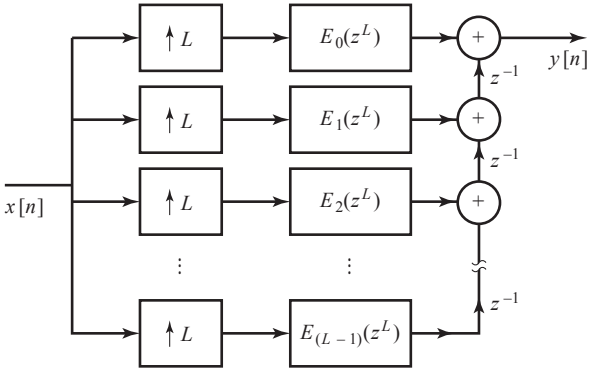


Figura 67: Decomposição polifásica do filtro de interpolação. (Oppenheim, 2009).

Implementação de interpoladores usando decomposição polifásica III

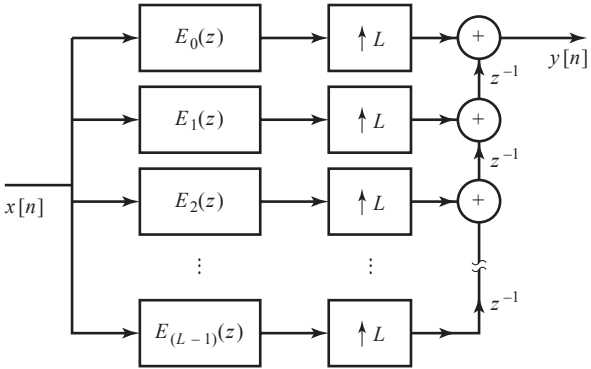


Figura 68: Implementação usando a identidade do *upsample*. (Oppenheim, 2009).

Notebook - implementação polifásica

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/polyphase.ipynb

Banco de Filtros Multi taxa I

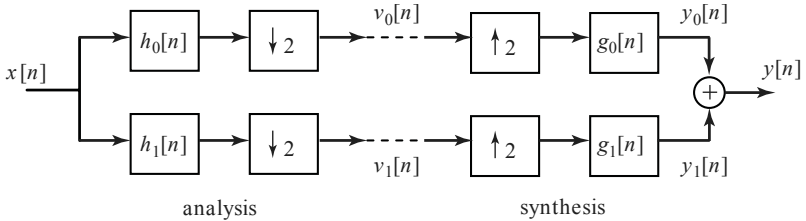


Figura 69: Banco de filtro de análise e síntese com dois canais. (Oppenheim, 2009).

Banco de Filtros Multi taxa II

A decomposição requer que $h_0[n]$ e $h_1[n]$ sejam filtros passa-baixa e passa-alta respectivamente. Uma abordagem comum é obter o filtro passa-alta a partir do filtro passa-baixa fazendo $h_1[n] = e^{j\pi n} h_0[n]$. Isto implica em $H_1(e^{j\omega}) = H_0(e^{j(\omega-\pi)})$. Analisando a figura, queremos

$$Y(e^{j\omega}) = \frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j\omega}) + G_1(e^{j\omega})H_1(e^{j\omega})\right] X(e^{j\omega}) \tag{166}$$

$$+ \frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j(\omega-\pi)}) + G_1(e^{j\omega})H_1(e^{j(\omega-\pi)})\right] X(e^{j(\omega-\pi)}). \tag{167}$$

Note que teremos reconstrução perfeita se os filtros forem ideais, entretanto também é possível obter reconstrução perfeita com filtros não ideias, para tanto o *aliasing* deverá ocorrer. Note que o segundo termo da expressão para $Y(e^{j\omega})$ representa a potencial distorção por *aliasing*. Este termo poderá ser eliminado escolhendo

$$G_0(e^{j\omega})H_0(e^{j(\omega-\pi)}) + G_1(e^{j\omega})H_1(e^{j(\omega-\pi)}) = 0. \tag{168}$$

Banco de Filtros Multi taxa III

Esta condição para cancelamento do *aliasing* é satisfeita, por exemplo, pelo conjunto de condições:

$$h_1[n] = e^{j\pi n} h_0[n] \Longleftrightarrow H_1(e^{j\omega}) = H_0(e^{j(\omega-\pi)}) \tag{169}$$

$$g_0[n] = 2h_0[n] \Longleftrightarrow G_0(e^{j\omega}) = 2H_0(e^{j\omega}) \tag{170}$$

$$g_1[n] = -2h_1[n] \Longleftrightarrow G_1(e^{j\omega}) = -2H_0(e^{j(\omega-\pi)}) \tag{171}$$

Os filtros $h_0[n]$ e $h_1[n]$ são chamados filtros espelhados em quadratura pois eles devem possuir simetria em torno de $\omega = \pi/2$. Usando as condições acima, eq. (166) ficará da forma

$$Y(e^{j\omega}) = \left[H_0^2(e^{j\omega}) - H_0^2(e^{j(\omega-\pi)})\right] X(e^{j\omega}), \tag{172}$$

e assim concluímos que a reconstrução perfeita (com possível atraso de M amostras) requererá

$$H_0^2(e^{j\omega}) - H_0^2(e^{j(\omega-\pi)}) = e^{-j\omega M}. \tag{173}$$

Banco de Filtros Multi taxa IV

Os únicos filtros computacionalmente realizáveis capazes de fornecer uma reconstrução exata são aqueles com resposta ao impulso da forma

$$h_0[n] = c_0\delta[n - 2n_0] + c_1\delta[n - 2n_1 - 1], \tag{174}$$

onde n_0 e n_1 são inteiros arbitrários e $c_0c_1 = 1/4$.
Por exemplo, considere

$$h_0[n] = \frac{1}{2}(\delta[n] + \delta[n - 1]), \tag{175}$$

com resposta em frequência

$$H_0(e^{j\omega}) = \cos(\omega/2)e^{-j\omega}. \tag{176}$$

Para este exemplo, teremos $Y(e^{j\omega}) = e^{-j\omega}X(e^{j\omega})$.
Podemos empregar a decomposição polifásica a este banco de filtros.

Banco de Filtros Multi taxa V

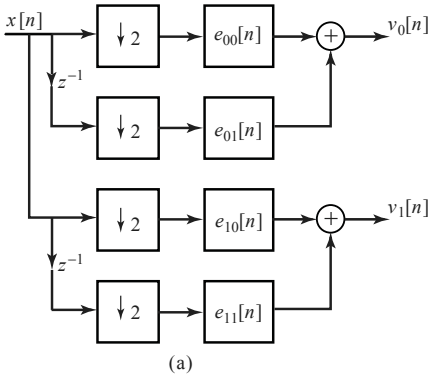


Figura 70: Representação polifásica do banco de filtros de análise. (Oppenheim, 2009).

Banco de Filtros Multi taxa VI

Usando as relações

$$e_{00}[n] = h_0[2n] \tag{177}$$
$$e_{01}[n] = h_0[2n+1] \tag{178}$$
$$e_{10}[n] = h_1[2n] = e^{j2\pi n} h_0[2n] = e_{00}[n] \tag{179}$$
$$e_{11}[n] = h_1[2n+1] = e^{j2\pi n} e^{j\pi} h_0[2n+1] = -e_{01}[n]. \tag{180}$$

Logo poderemos representar o banco de filtros utilizando metade dos cálculos computacionais.

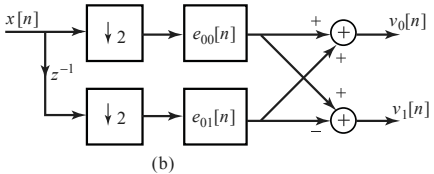


Figura 71: Representação simplificada. (Oppenheim, 2009).

Banco de Filtros Multi taxa VII

O mesmo pode ser feito com o filtro de síntese. Assim teremos o sistema conforme ilustrado.

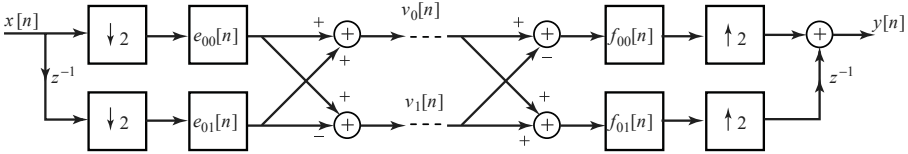


Figura 72: Representação polifásica do banco de filtros de análise e síntese. (Oppenheim, 2009).

Predição Linear e Modelo Autorregressivo

Os problemas de predição linear e modelo autorregressivo são problemas diferentes que compartilham solução numérica. Em ambos os casos, deseja-se encontrar os parâmetros de um filtro linear.

No caso da predição linear (LPC), desejamos encontrar um filtro FIR para prever, a partir de amostras passadas, as amostras futuras de um processo autorregressivo. O erro encontrado é chamado erro de predição (idealmente um ruído branco).

No caso do modelo autorregressivo, busca-se determinar o filtro IIR que, quando excitado por um ruído branco, produza na saída um sinal com as mesmas características estatísticas que o modelo que estamos buscando modelar.

Modelo Autorregressivo

Modelo Autorregressivo (AR), no contexto de processamento de sinais, é visto como um filtro de resposta ao impulso infinita (IIR), ou seja, um filtro com apenas pólos. Na física, é visto como um modelo de máxima entropia. Ele possui 'memória' ou realimentação, e desta forma o sistema é capaz de gerar uma dinâmica interna.

Modelo Autorregressivo de Primeira Ordem

Modelo de primeira ordem:

$$X_t = \varphi X_{t-1} + \mathcal{E}_t, \qquad \mathcal{E}_t \sim iid(0, \sigma^2) \tag{181}$$

Exemplo: variação do preço do petróleo (ΔP_t) em um determinado instante.

$$\Delta P_t = \frac{1}{2}\Delta P_{t-1} + \mathcal{E}_t \tag{182}$$

onde $\mathcal{E}$ é um fator externo.
Observe que o efeito de uma pertubação dura por muito tempo, diferentemente de um processo de média móvel em que o efeito da pertubação passa rapidamente.

Modelo Autoregressivo de Primeira Ordem - Estacionário em Média I

Dizemos que um processo é estacionário em média quando

$$E[X_t] = \text{const.} \tag{183}$$

Para um processo de primeira ordem, dado pela Equação 181, teremos

$$\begin{aligned} X_t &= \varphi X_{t-1} + \mathcal{E}_t \\ &= \varphi\left(\varphi X_{t-2} + \mathcal{E}_{t-1}\right) + \mathcal{E}_t \\ &= \varphi^2 X_{t-2} + \varphi\mathcal{E}_{t-1} + \mathcal{E}_t \\ &\vdots \qquad (184) \\ &= \varphi^t X_0 + \sum_{i=0}^{t-1} \varphi^i \mathcal{E}_{t-i} + \mathcal{E}_t \qquad (185) \end{aligned}$$

Modelo Autoregressivo de Primeira Ordem - Estacionário em Média II

Qual é o valor esperado de X_t?

$$\begin{aligned} E[X_t] &= E\left[\varphi^t X_0 + \sum_{i=0}^{t-1} \varphi^i \mathcal{E}_{t-i} + \mathcal{E}_t\right] \\ &= \varphi^t E\left[X_0\right] + \sum_{i=0}^{t-1} \varphi^i \cancelto{0}{E\left[\mathcal{E}_{t-i}\right]} + \cancelto{0}{E\left[\mathcal{E}_t\right]} \\ &= \varphi^t E\left[X_0\right]. \end{aligned} \tag{186}$$

Para que $E[X_t] = \varphi^t E[X_0]$ seja constante devemos ter

1) $\varphi = 0$ (solução trivial), o que não desejamos; ou
2) $E[X_0] = 0$, e desta forma, valor esperado de X_t nulo, $\forall t$.

Modelo Autorregressivo de Primeira Ordem - Estacionário em Variância I

Dizemos que um processo é estacionário em variância quando

$$\text{var}(X_t) = \text{const.} \tag{187}$$

Utilizando a seguinte propriedade da variância: $\text{var}(aX) = a^2\text{var}(X)$, teremos

$$\begin{aligned} \text{var}(X_t) &= \text{var}(\varphi X_{t-1} + \mathcal{E}_t) \\ &= \text{var}(\varphi X_{t-1}) + \text{var}(\mathcal{E}_t) \\ &= \varphi^2\text{var}(X_{t-1}) + \sigma^2 \end{aligned} \tag{188}$$

Queremos que $\text{var}(X_t) = \text{var}(X_{t-1}) = \text{var}(X_{t-2}) = \ldots$
Teremos então

$$\begin{aligned} \text{var}(X_t) &= \varphi^2\text{var}(X_{t-1}) + \sigma^2 \\ &= \varphi^2\text{var}(X_t) + \sigma^2 \\ &= \frac{\sigma^2}{1-\varphi^2} \end{aligned} \tag{189}$$

Modelo Autorregressivo de Primeira Ordem - Estacionário em Variância II

Teremos que ter $|\varphi| < 1$ por dois motivos

1) se $|\varphi| > 1$, teríamos variância negativa, o que não faz sentido;
2) se $|\varphi| = 1$ teríamos variância infinita, o que também não faz sentido.

Modelo Autorregressivo de Ordem p I

AR(p) : modelo autorregressivo de ordem p

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \mathcal{E}_t \tag{190}$$

$\varphi_1, \varphi_2, \ldots, \varphi_p$ são os parâmetros do modelo

c constante, geralmente omitida

$\mathcal{E}_t$ ruído branco

Um modelo autorregressivo pode ser visto como a saída de um filtro de resposta ao impulso infinita (IIR) com apenas polos quando a entrada é um ruído branco.

Modelo Autorregressivo de Ordem p II

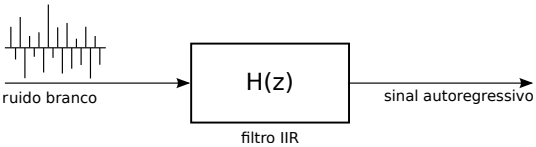


Figura 73: Modelo autorregressivo.

$$\begin{aligned} H(z) &= \frac{1}{1-\varphi_1 z^{-1}-\varphi_2 z^{-2}-\ldots-\varphi_p z^{-p}} \\ &= \frac{1}{1-\sum_{i=1}^{p}\varphi_i z^{-i}} \end{aligned} \tag{191}$$

Para que o modelo AR(p) seja estacionário no sentido amplo (ou seja, suas características estatísticas não mudam com o tempo), as raízes do polinômio $z^p - \sum_{i=1}^{p}\varphi_i z^{p-i}$ devem estar dentro do círculo unitário, i.e. cada raiz z_i deve satisfazer $|z_i| < 1$.

Inversão Direta I

Vamos assumir que uma série temporal com média nula $\{x_t\}_0^{N-1}$ é um processo AR e o modelo é

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \ldots + \varphi_p X_{t-p} + \mathcal{E}_t \tag{192}$$

Inversão Direta - $p = 1$ I

No caso em que $p = 1$, teremos

$$x_t = \varphi_1 x_{t-1} + \mathcal{E}_t \tag{193}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix} = \varphi_1 \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-2} \end{bmatrix} \tag{194}$$

$$\begin{aligned} \mathbf{b} &= \varphi_1 \mathbf{A} \\ &= \mathbf{A}\varphi_1 \end{aligned} \tag{195}$$

Inversão Direta - $p = 1$ II

$$\begin{aligned} \mathbf{A}\hat{\varphi}_1 &= \mathbf{b} \\ \mathbf{A}^T\mathbf{A}\hat{\varphi}_1 &= \mathbf{A}^T\mathbf{b} \\ (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{A}\hat{\varphi}_1 &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \\ \hat{\varphi}_1 &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \end{aligned} \tag{196}$$

$$\begin{aligned} \hat{\varphi}_1 &= \underbrace{(\mathbf{A}^T\mathbf{A})^{-1}}_{\left(\sum_{i=0}^{N-2} x_i^2\right)^{-1}} \quad \underbrace{\mathbf{A}^T\mathbf{b}}_{\sum_{i=0}^{N-2} x_i x_{i+1}} \\ &= \left(\sum_{i=0}^{N-2} x_i^2\right)^{-1} \sum_{i=0}^{N-2} x_i x_{i+1} \\ &= c_0^{-1} c_1 = r_1 \end{aligned} \tag{197}$$

onde c_i é o i-ésimo coeficiente de auto covariância e r_i o i-ésimo coeficiente de autocorrelação.

Inversão Direta - $p = 2$ I

No caso em que $p = 2$, teremos

$$x_t = \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \mathcal{E}_t \tag{198}$$

$$\begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} x_1 & x_0 \\ x_2 & x_1 \\ \vdots & \vdots \\ x_{N-2} & x_{N-3} \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix} \tag{199}$$

$$\mathbf{b} = \mathbf{A}\varphi \tag{200}$$

Inversão Direta - $p = 2$ II

$$\begin{aligned} \mathbf{A}\hat{\varphi} &= \mathbf{b} \\ \mathbf{A}^T\mathbf{A}\hat{\varphi} &= \mathbf{A}^T\mathbf{b} \\ (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{A}\hat{\varphi} &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \\ \hat{\varphi} &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \end{aligned} \tag{201}$$

Inversão Direta - $p = 2$ III

$$\hat{\varphi} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}, \tag{202}$$

onde

$$\begin{aligned}(\mathbf{A}^T\mathbf{A})^{-1} &= \left[\begin{bmatrix} x_1 & x_2 & \dots & x_{N-2} \\ x_0 & x_1 & \dots & x_{N-3} \end{bmatrix} \begin{bmatrix} x_1 & x_0 \\ x_2 & x_1 \\ \vdots & \vdots \\ x_{N-2} & x_{N-3} \end{bmatrix}\right]^{-1}_{2\times 2} \\ &= \begin{bmatrix} \sum_{i=1}^{N-2} x_i^2 & \sum_{i=1}^{N-2} x_i x_{i-1} \\ \sum_{i=1}^{N-2} x_i x_{i-1} & \sum_{i=0}^{N-3} x_i^2 \end{bmatrix}^{-1}_{2\times 2}\end{aligned} \tag{203}$$

Se $\det(\mathbf{A}) \neq 0$, então podemos obter $\mathbf{A}^{-1}$ da seguinte forma:

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})}\text{adj}(\mathbf{A}) \tag{204}$$

Inversão Direta - $p = 2$ IV

onde a matriz adjunta (adj($\mathbf{A}$)) é a transposta da matriz de cofator $\mathbf{C}$ de $\mathbf{A}$.

$$C_{ij} = (-1)^{i+j} M_{ij} \tag{205}$$

sendo M_{ij} o menor ij da matriz $\mathbf{A}$, ou seja, o determinante da submatriz quadrada, obtida a partir de $\mathbf{A}$ pela remoção da sua i-ésima linha e j-éssima coluna.
Teremos assim

$$\begin{aligned}
(\mathbf{A}^T\mathbf{A})^{-1} &= \frac{1}{\det(\mathbf{A}^T\mathbf{A})}\text{adj}(\mathbf{A}^T\mathbf{A}) \\
&= \frac{1}{\sum_{i=1}^{N-2} x_i^2 \sum_{i=0}^{N-3} x_i^2 - \sum_{i=1}^{N-2} x_i x_{i-1} \sum_{i=1}^{N-2} x_i x_{i-1}} \\
&\quad \begin{bmatrix} \sum_{i=0}^{N-3} x_i^2 & -\sum_{i=1}^{N-2} x_i x_{i-1} \\ -\sum_{i=1}^{N-2} x_i x_{i-1} & \sum_{i=1}^{N-2} x_i^2 \end{bmatrix}
\end{aligned} \tag{206}$$

Inversão Direta - $p = 2$ V

Utilizando o fato de que temos uma série temporal estacionária, os elementos de auto-covariância são função apenas do atraso, e não dos limites temporais exatos. Poderemos então escrever

$$\begin{aligned}(\mathbf{A}^T\mathbf{A})^{-1} &= \frac{1}{c_0^2 - c_1^2}\begin{bmatrix} c_0 & -c_1 \\ -c_1 & c_0 \end{bmatrix} \\ &= \frac{1}{c_0^2(1 - r_1^2)}\begin{bmatrix} c_0 & -c_1 \\ -c_1 & c_0 \end{bmatrix} \\ &= \frac{1}{c_0(1 - r_1^2)}\begin{bmatrix} r_0 & -r_1 \\ -r_1 & r_0 \end{bmatrix}\end{aligned}$$

onde utilizamos que $r_i = \frac{c_i}{\sigma}$ e $\sigma = c_0$ (variância é o caso especial da covariância quando as duas variáveis são idênticas).

Inversão Direta - $p = 2$ VI

Analisando agora $\mathbf{A}^T\mathbf{b}$.

$$\begin{aligned}\mathbf{A}^T\mathbf{b} &= \begin{bmatrix} x_1 & x_2 & \dots & x_{N-2} \\ x_0 & x_1 & \dots & x_{N-3} \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_{N-1} \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=2}^{N-1} x_i x_{i-1} \\ \sum_{i=2}^{N-1} x_i x_{i-2} \end{bmatrix} \\ &= \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \end{aligned} \tag{207}$$

onde utilizamos mais uma vez a estacionariedade da série temporal em questão.

Inversão Direta - $p = 2$ VII

Teremos então

$$
\begin{aligned}
(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} &= \frac{1}{c_0(1-r_1^2)}\begin{bmatrix} r_0 & -r_1 \\ -r_1 & r_0 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \\
&= \frac{1}{1-r_1^2}\begin{bmatrix} r_0 & -r_1 \\ -r_1 & r_0 \end{bmatrix}\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \\
&= \frac{1}{1-r_1^2}\begin{bmatrix} 1 & -r_1 \\ -r_1 & 1 \end{bmatrix}\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \\
&= \begin{bmatrix} \frac{r_1(1-r_2)}{1-r_1^2} \\ \frac{r_2-r_1^2}{1-r_1^2} \end{bmatrix} = \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix} = \hat{\varphi}
\end{aligned}
\tag{208}
$$

Inversão Direta - p qualquer

O mesmo procedimento pode ser feito para p qualquer, mas ficará cada vez mais complicado.

Uma maneira mais simples é utilizar as Equações de Yule-Walker.

Equações de Yule-Walker I

Utilizaremos as equações de Yule-Walker para estimar os parâmetros AR a partir dos dados.

- Dada uma série temporal $x[n]$, podemos estimar a sequência de covariância $r_{xx}[k]$ para a série temporal $x[n]$ dada.
- Existe uma relação entre o modelo auto-regressivo e a sequência de covariância. Poderemos então estimar os p parâmetros do modelo AR de ordem p.
- Para tanto, iremos resolver as equações de Yule-Walker e encontrar estes parâmetros a partir de $r_{xx}[k]$.

Equações de Yule-Walker II

O método de Yule-Walker não lida com a questão da escolha da ordem do modelo. Isto é usualmente feito através de um balanceamento do erro contra o número de parâmetros no modelo. Os seguintes métodos são usuais:

1) Critério de informação de Akaike (*Akaike information criterion*, AIC)
2) Critério de informação Bayesiano (*Bayesian information criterion*, BIC)
3) validação cruzada (*cross validation*, CV) - utiliza um subconjunto dos dados para estimar o modelo, outro subconjunto para testar os resultados e um terceiro para validar.
4) balancear o erro quadrático (do ajuste do modelo aos dados) e a ordem do modelo

Ao aumentar a ordem do modelos, podemos diminuir continuamente o erro, obtendo um modelo cada vez mais complexo, mas isto pode não agregar, pois o modelo obtido pode não representar bem os dados.

Equações de Yule-Walker III

Considere o modelo AR(p)

$$X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + \mathcal{E}_t \tag{209}$$

atraso 1: multiplicando ambos os lados por X_{t-1} teremos

$$X_t X_{t-1} = \sum_{i=1}^{p} \varphi_i X_{t-i} X_{t-1} + \mathcal{E}_t X_{t-1} \tag{210}$$

tomando o valor esperado teremos

$$\begin{aligned} E\left[X_t X_{t-1}\right] &= E\left[\sum_{i=1}^{p} \varphi_i X_{t-i} X_{t-1} + \mathcal{E}_t X_{t-1}\right] \\ &= \sum_{i=1}^{p} \varphi_i E\left[X_{t-i} X_{t-1}\right] + \cancelto{0}{E\left[\mathcal{E}_t X_{t-1}\right]} \end{aligned} \tag{211}$$

Equações de Yule-Walker IV

Vamos utilizar $c_l = E\left[X_t X_{t-l}\right]$ e $c_l = c_{-l}$. Poderemos reescrever a equação acima como

$$c_1 = \sum_{i=1}^{p} \varphi_i c_{i-1} \tag{212}$$

dividindo por $c_0 = \sigma$ teremos

$$r_1 = \sum_{i=1}^{p} \varphi_i r_{i-1} \tag{213}$$

Equações de Yule-Walker V

modelo AR(p)

$$X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + \mathcal{E}_t \tag{214}$$

atraso 2: multiplicando ambos os lados por X_{t-2} teremos

$$X_t X_{t-2} = \sum_{i=1}^{p} \varphi_i X_{t-i} X_{t-2} + \mathcal{E}_t X_{t-2} \tag{215}$$

tomando o valor esperado teremos

$$\begin{aligned} E\left[X_t X_{t-2}\right] &= E\left[\sum_{i=1}^{p} \varphi_i X_{t-i} X_{t-2} + \mathcal{E}_t X_{t-2}\right] \\ &= \sum_{i=1}^{p} \varphi_i E\left[X_{t-i} X_{t-2}\right] + \cancelto{0}{E\left[\mathcal{E}_t X_{t-2}\right]} \end{aligned} \tag{216}$$

Equações de Yule-Walker VI

Vamos utilizar $c_l = E\left[X_t X_{t-l}\right]$ e $c_l = c_{-l}$. Poderemos reescrever a equação acima como

$$c_2 = \sum_{i=1}^{p} \varphi_i c_{i-2} \tag{217}$$

dividindo por $c_0 = \sigma$ teremos

$$r_2 = \sum_{i=1}^{p} \varphi_i r_{i-2} \tag{218}$$

Equações de Yule-Walker VII

modelo AR(p)

$$X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + \mathcal{E}_t \tag{219}$$

atraso k: multiplicando ambos os lados por X_{t-k} teremos

$$X_t X_{t-k} = \sum_{i=1}^{p} \varphi_i X_{t-i} X_{t-k} + \mathcal{E}_t X_{t-k} \tag{220}$$

tomando o valor esperado teremos

$$\begin{aligned} E\left[X_t X_{t-k}\right] &= E\left[\sum_{i=1}^{p} \varphi_i X_{t-i} X_{t-k} + \mathcal{E}_t X_{t-k}\right] \\ &= \sum_{i=1}^{p} \varphi_i E\left[X_{t-i} X_{t-k}\right] + \cancelto{0}{E\left[\mathcal{E}_t X_{t-k}\right]} \end{aligned} \tag{221}$$

Equações de Yule-Walker VIII

Vamos utilizar $c_l = E\left[X_t X_{t-l}\right]$ e $c_l = c_{-l}$. Poderemos reescrever a equação acima como

$$c_k = \sum_{i=1}^{p} \varphi_i c_{i-k} \tag{222}$$

dividindo por $c_0 = \sigma$ teremos

$$r_k = \sum_{i=1}^{p} \varphi_i r_{i-k} \tag{223}$$

Equações de Yule-Walker IX

$\vdots$

O mesmo ponde continuar sendo feito até $k = p$.
Desta forma, poderemos montar um sistema com p equações e p incógnitas.

Equações de Yule-Walker X

$$\begin{cases} r_1 = \varphi_1 r_0 + \varphi_2 r_1 + \varphi_3 r_2 + \ldots + \varphi_{p-1} r_{p-2} + \varphi_p r_{p-1} \\ r_2 = \varphi_1 r_1 + \varphi_2 r_0 + \varphi_3 r_1 + \ldots + \varphi_{p-1} r_{p-3} + \varphi_p r_{p-2} \\ \vdots \\ r_{p-1} = \varphi_1 r_{p-2} + \varphi_2 r_{p-3} + \varphi_3 r_{p-4} + \ldots + \varphi_{p-1} r_0 + \varphi_p r_1 \\ r_p = \varphi_1 r_{p-1} + \varphi_2 r_{p-2} + \varphi_3 r_{p-3} + \ldots + \varphi_{p-1} r_1 + \varphi_p r_0 \end{cases} \tag{224}$$

Equações de Yule-Walker XI

Este sistema de equações pode ser escrito na forma matricial.

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{p-1} \\ r_p \end{bmatrix} = \begin{bmatrix} r_0 & r_1 & r_2 & \dots & r_{p-2} & r_{p-1} \\ r_1 & r_0 & r_1 & \dots & r_{p-3} & r_{p-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{p-2} & r_{p-3} & r_{p-4} & \dots & r_0 & r_1 \\ r_{p-1} & r_{p-2} & r_{p-3} & \dots & r_1 & r_0 \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_{p-1} \\ \varphi_p \end{bmatrix} \tag{225}$$

$$\mathbf{r} = \mathbf{R}\varphi \tag{226}$$

Temos um problema bem posto, com uma matriz de coeficientes quadrada $\mathbf{R}$, i.e., o mesmo número de equações e variáveis. $\mathbf{R}$ é de posto cheio e simétrica. Desta forma, a existência de sua inversa é garantida. Assim

$$\varphi = \mathbf{R}^{-1}\mathbf{r}. \tag{227}$$

Este sistema pode ser resolvido de forma eficiente pelo algoritmo de Levinson-Durbin (algoritmo utilizado para resolver sistema de equações com matriz de Toeplitz).

Estabilidade e Localização dos pólos no modelo AR I

Um modelo autorregressivo pode ser expresso em termos de uma sequência de inovações $\mathcal{E}_t$

$$X(z) = z^p \left(\sum_{i=0}^{p} \varphi_i z^{p-i} \right)^{-1} \mathcal{E}(z) \tag{228}$$

no qual temos p zeros em $z = 0$ e os p pólos são determinados pela equação característica do processo autorregressivo

$$\sum_{i=0}^{p} \varphi_i z^{p-i} = 0 \tag{229}$$

As raízes da equação característica devem ficar dentro do círculo unitário para garantir que o processo autorregressivo seja estável.
Se as raízes ficarem sobre o círculo unitário, o processo autorregressivo será estacionário, no caso em que $\mathcal{E}_t = 0$. Neste caso, um processo harmônico será resultante, consistindo em uma soma de funções cosseno.

Estabilidade e Localização dos pólos no modelo AR II

Um processo autorregressivo com pólos próximos ao círculo unitário pode demonstrar um comportamento pseudo-periódico. Neste caso, a função de auto-covariância será descrita como uma soma de funções periódicas levemente amortecidas. Mais adiante, estando o termo de ruído ainda presente, o processo autorregressivo irá apresentar um comportamento quase não-estacionário. Finalmente, os coeficientes de autocorrelação parcial irão ficar próximos de um em valor absoluto. No contexto de filtros lineares, isto significa que a função de transferência relacionando X_t a $\mathcal{E}_t$ estará próxima da instabilidade.
A localização dos pólos também influenciará a confiabilidade de várias técnicas de estimação de parâmetros. Argumenta-se que a técnica de Yule-Walker pode levar a estimação pobre dos parâmetros, mesmo para amostras moderadamente grandes, se o operador autorregressivo possuir um pólo próximo do círculo unitário.

Notebook - Modelo AR

https://github.com/leolca/notebooks/blob/master/aev/ar.ipynb

Predição Linear

Predição linear é uma operação matemática em que se utiliza valores passados de um sinal discreto no tempo para prever valores futuros deste mesmo sinal. Em processamento digital de sinais, a predição linear é conhecida como *linear predictive coding (LPC)* e pode ser entendida no contexto de filtros digitais.

Aplicações da Predição Linear

- Análise de sinais da fala;
- Análise de eletroencefalograma;
- Análise de sinais sísmicos;
- Estimação de Formantes da fala;
- Controle de Ruído;
- Controle;
- etc.

Formulação Matemática

O preditor linear de ordem p é descrito pela equação:

$$\hat{x}(n) = \sum_{i=1}^{p} a_i x(n-i). \tag{230}$$

O erro de predição é dado por

$$e(n) = x(n) - \hat{x}(n). \tag{231}$$

O preditor é determinado pelos coeficientes $a_1, \ldots, a_p$.

Determinação do Preditor

Dado um sinal x queremos encontrar o melhor preditor para este sinal.
Melhor: erro de estimação mínimo (erro médio quadrático)
Objetivo: minimizar o valor esperado do erro médio quadrático, $E\left[|e(n)|^2\right]$.

Minimização do Valor Esperado do Erro Médio Quadrático

Para encontrar os parâmetros que minimizam o erro devemos buscar o conjunto de parâmetros que satisfaz:

1) derivada primeira nula em relação a cada um dos parâmetros a_k, $k = 1, \ldots, p$

$$\frac{\partial E\left[|e(n)|^2\right]}{\partial a_k} = 0 \tag{232}$$

2) derivada segunda positiva em relação a cada um dos parâmetros a_k, $k = 1, \ldots, p$

$$\frac{\partial^2 E\left[|e(n)|^2\right]}{\partial a_k^2} > 0 \tag{233}$$

Garantia de que encontraremos um mínimo qualquer, não necessariamente o mínimo global.

Módulo Quadrado do Erro

$$
\begin{aligned}
|e(n)|^2 &= e(n)e^*(n) \\
&= (x(n) - \hat{x}(n))e^*(n) \\
&= \left(x(n) - \sum_{i=1}^{p} a_i x(n-i)\right) e^*(n) \\
&= (x(n)e^*(n)) - \left(\sum_{i=1}^{p} a_i x(n-i)e^*(n)\right)
\end{aligned} \tag{234}
$$

logo

$$
E\left[|e(n)|^2\right] = E\left[x(n)e^*(n)\right] - \sum_{i=1}^{p} a_i E\left[x(n-i)e^*(n)\right] \tag{235}
$$

Derivada

$$E\left[|e(n)|^2\right] = E\left[x(n)e^*(n)\right] - \sum_{i=1}^{p} a_i E\left[x(n-i)e^*(n)\right]$$

$$\begin{aligned}
\frac{\partial E\left[|e(n)|^2\right]}{\partial a_k} &= 0 - E\left[x(n-k)e^*(n)\right] \\
&= -E\left[x(n-k)\left(x(n)-\hat{x}(n)\right)^*\right] \\
&= -E\left[x(n-k)\left(x(n)-\sum_{i=1}^{p} a_i x(n-i)\right)^*\right] \\
&= -E\left[x(n-k)\left(x^*(n)-\sum_{i=1}^{p} a_i^* x^*(n-i)\right)\right]
\end{aligned}$$

Derivada

Queremos

$$\begin{aligned} \frac{\partial E\left[|e(n)|^2\right]}{\partial a_k} &= 0 \\ -E\left[x(n-k)x^*(n)\right] + & \\ \sum_{i=1}^{p} a_i^* E\left[x^*(n-i)x(n-k)\right] &= 0 \\ \sum_{i=1}^{p} a_i^* E\left[x^*(n-i)x(n-k)\right] &= E\left[x(n-k)x^*(n)\right] \\ \sum_{i=1}^{p} a_i^* r(k-i) &= r(k) \end{aligned} \tag{236}$$

aonde $r(k)$ é a autocorrelação do sinal $x(n)$ avaliada com deslocamento k.

Autocorrelação

A autocorrelação do sinal $x(n)$ avaliada com deslocamento k é dada por

$$r(k) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=-N/2}^{N/2} x(n)x^*(n-k) \tag{237}$$

A autocorrelação de sequências reais é simétrica e par. Desta forma, sendo $x(n)$ um sinal real, teremos $r(k) = r(-k)$.

Sistemas de Equações

Dado que

$$r(k) = \sum_{i=1}^{p} a_i^* r(k-i), \tag{238}$$

vamos criar um sistema com p equações tomando o deslocamento nulo ($k = 0$) como referência.

$$\begin{cases} r(1) = a_1 r(0) + a_2 r(1) + a_3 r(2) + \ldots + a_p r(p-1) \\ r(2) = a_1 r(1) + a_2 r(0) + a_3 r(1) + \ldots + a_p r(p-2) \\ \vdots \\ r(p) = a_1 r(p-1) + a_2 r(p-2) + a_3 r(p-3) + \ldots + a_p r(0) \end{cases}$$

este sistema pode ser escrito na forma matricial a seguir

Sistemas de Equações - Forma Matricial

As equações são conhecidas como Equações de Yule-Walker e podem ser reescritas na forma matricial.

$$\begin{bmatrix} r(0) & r(1) & r(2) & \dots & r(p-1) \\ r(1) & r(0) & r(1) & \dots & r(p-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & r(p-3) & \dots & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(p) \end{bmatrix} \tag{239}$$

Temos um sistema bem posto, i.e., com o mesmo número de equações e incógnitas. A matriz $\mathbf{R}$ é uma matriz de posto cheio e portanto possui inversa. A matriz $\mathbf{R}$ é uma matriz Toeplitz e desta forma o sistema pode ser resolvido de forma rápida pelo algoritmo de Levinson-Durbin.

Recursão de Levinson

Técnicas tradicionais de reslução de sistema de equações possuem complexidade computacional da ordem de $O(N^3)$. Como no sistemas de equações em questão a matriz é do tipo Toeplitz, podemos aplicar a método de recursão de Levinson, resolvendo o sistema de forma mais eficiente com complexidade computacional da ordem de $O(N^2)$.
A recursão de Levinson baseia-se na observação de que a solução de uma problema de ordem m pode ser utilizada para resolver um problema de ordem $m+1$. Inicia-se resolvendo o problema de ordem zero e, incrementalmente, resolve-se os problemas de ordem superior até alguma ordem p desejada. Todas as soluções para preditores com ordem $\leq p$ foram geradas.

Matriz Toeplitz

Uma matriz é dita *matriz Toeplitz* quando cada um de suas diagonais descendentes da esquerda para a direita for constante. A matriz abaixo é um exemplo de matriz Toeplitz:

$$\begin{bmatrix} a & b & c & d & e \\ f & a & b & c & d \\ g & f & a & b & c \\ h & g & f & a & b \\ i & h & g & f & a \end{bmatrix}. \tag{240}$$

Matriz Toeplitz

Qualquer matriz $\mathbf{A}$, $n \times n$, da forma

$$\begin{bmatrix} a_0 & a_{-1} & a_{-2} & \dots & \dots & a_{-n+1} \\ a_1 & a_0 & a_{-1} & \ddots & & \vdots \\ a_2 & a_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{-1} & a_{-2} \\ \vdots & & \ddots & a_1 & a_0 & a_{-1} \\ a_{n-1} & \dots & \dots & a_2 & a_1 & a_0 \end{bmatrix} \tag{241}$$

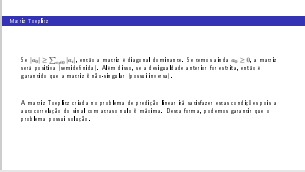
é uma matriz Toeplitz. Teremos que o elemento (i, j) da matriz $\mathbf{A}$, denotado $A_{i,j}$, é da forma

$$A_{i,j} = A_{i+1,j+1} = a_{i-j}. \tag{242}$$

Matriz Toeplitz

Se $|a_0| \geq \sum_{i \neq 0} |a_i|$, então a matriz é diagonal dominante. Se temos ainda $a_0 \geq 0$, a matriz será positiva (semidefinida). Além disso, se a desigualdade anterior for estrita, então é garantido que a matriz é não-singular (possui inversa).

A matriz Toeplitz criada no problema de predição linear irá satisfazer estas condições pois a autocorrelação do sinal com atraso nulo é máxima. Desta forma, podemos garantir que o problema possui solução.



Uma matriz $\mathbf{M}$ é dita **positiva definida** se $z^T M z$ é positivo para qualquer vetor z não nulo. Da mesma forma, definimos como **negativa definida**, **positiva semi-definida** e **negativa semi-definida** as matrizes M que fazem com que a expressão $z^T M z$ ou $z^* M z$ seja sempre negativa, não-negativa ou não-positiva, respectivamente.

Levinson Durbin

O algoritmo de Levinson Durbin é um algoritmo recursivo para resolver sistemas com matrizes Toeplitz.

Complexidade

- resolução convencional de sistemas: $O(N^3)$
- resolução de sistema através de Levinson Durbin: $O(N^2)$

Cálculo Rápido da Autocorrelação

É possível estimar os coeficientes de autocorrelação de forma rápida através da FFT. Desta forma, será possível computar os primeiros M coeficientes de autocorrelação para um vetor de comprimento N através de $O(N \log M)$ operações, ao invés de $O(NM)$ operações.

$$r(k) = ifft(fft(x)fft^*(x)) \tag{243}$$

Obs.: Para M pequeno, o custo $O(N \log M)$, associado à obtenção dos coeficiente de autocorrelação, é dominante. Entretanto, para M suficientemente grande, o custo $O(M^2)$ para resolução das equações de Yule-Walker passa a ser dominante.

Filtro - Preditor Linear

O preditor linear dado pela equação

$$\hat{x}(n) = \sum_{i=1}^{p} a_i x(n-i) \tag{244}$$

pode ser visto como um filtro com resposta ao impulso finita

$$\hat{X}(z) = \sum_{i=1}^{p} a_i X(z) z^{-i} \tag{245}$$

então

$$H(z) = \frac{\hat{X}(z)}{X(z)} = \sum_{i=1}^{p} a_i z^{-i}, \tag{246}$$

Filtro - Preditor Linear

Temos então um filtro FIR, apenas zeros (pólos no infinito) no plano z, que é, por definição, estável. Além disso, o filtro é de fase mínima, isto é, todos os seus zeros estão dentro do círculo unitário, desta forma, o filtro inverso é também estável.

Figura 74: Preditor Linear - filtro digital.

Padrões do ITU-T I

G.7xx: Audio (Voz) Protocolos de Compressão (CODEC)

- G.711 - Modulação por código de pulso (*Pulse code modulation*, PCM) de frequências de voz em um canal de 64 kbps.
- G.721 - Codificação por código de pulso adaptativo (ADPCM) a 32 kbit/s.
- G.722 - Codificação de áudio 7 kHz a 64 kbit/s.
- G.722.1 - Codificação a 24 e 32 kbit/s para operação de sistemas com mãos livres com baixa perda de quadros.
- G.722.2 - Codificação de banda larga de voz a aproximadamente 16 kbit/s utilizando codificação de banda larga com múltiplas taxas adaptativa (*adaptive muti-rate wideband*, AMR-WB).
- G.726 - Codificação por código de pulso adaptativa a 40, 32, 24, 16 kbit/s (ADPCM).
- G.727 - Codificação por código de pulso adaptativa a 5-, 4-, 3- e 2-bit/amostra.
- G.728 - Codificação de voz a 16 kbit/s utilizando código de excitação com predição linear com baixo atraso.

Padrões do ITU-T II

G.729 - Codificação de voz a 8 kbit/s utilizando estrutura conjugada codificação com excitação por códio algébrico e predição linear (*conjugate-structure algebraic-code-excited linear-prediction*, CS-ACELP).

Codificação de Voz

O sinal de fala é um sinal de áudio e é amostrado como qualquer outro sinal de áudio, mas devido à sua natureza particular (sinal de fala), ele possui propriedades que podem ser exploradas para obtermos uma compressão mais eficiente para este tipo de sinal.

Sinal de Voz

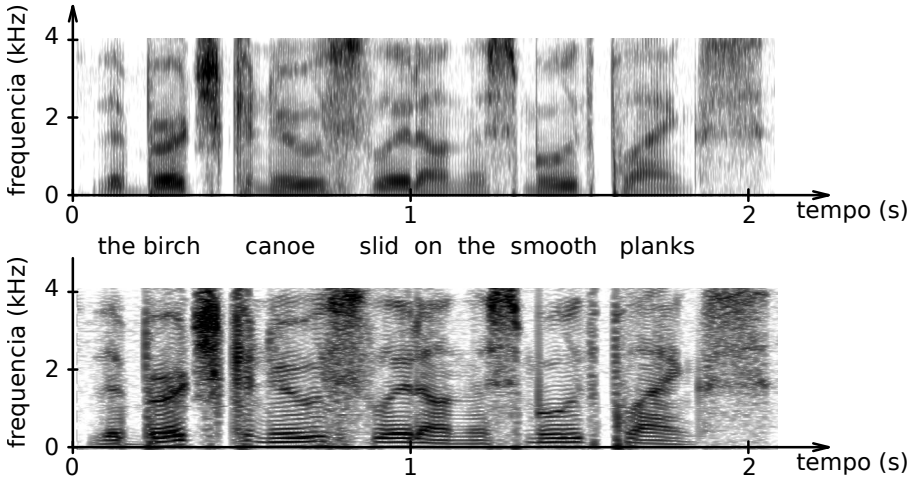


Figura 75: Espectrograma de uma sentença (a) banda-larga (b) banda-estreita.

Trato Vocal

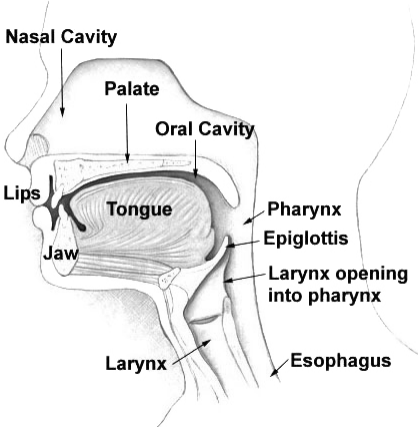


Figura 76: Aparato vocal humano (Wikipedia).

Trato Vocal

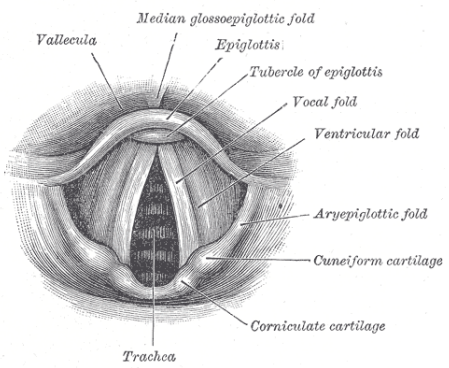


Figura 77: Corte transversal (Gray, 1918).

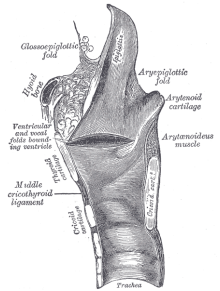


Figura 78: Corte sagital (Gray, 1918).

Sinal de Voz

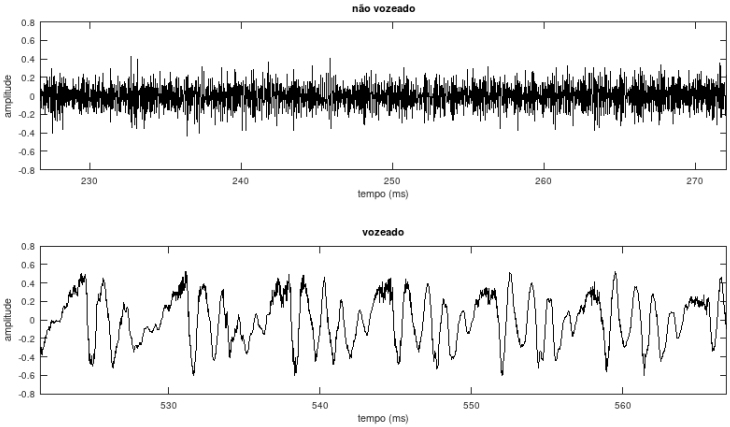


Figura 79: (a) Vozeado e (b) Não-vozeado.

Codificação de Voz

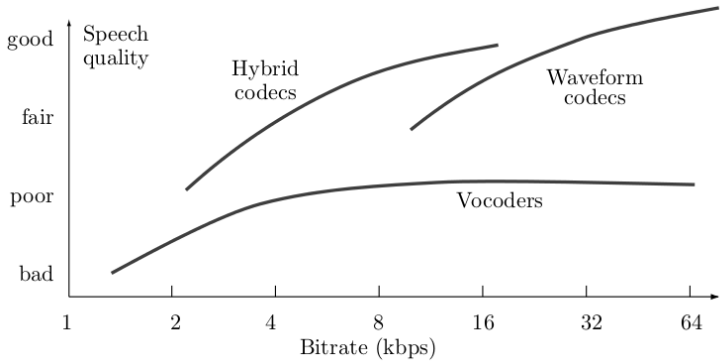


Figura 80: Qualidade da fala versus bitrate para abordagens distintas em codificações de voz.

Codificação de forma de onda

Este tipo de codificação não tenta predizer como o sinal original foi gerado. Apenas busca reproduzir, após a descompressão, as amostras de áudio que são o mais próximas possíveis das amostras originais.

- PCM
- ADPCM
- SBC (*subband coding*) - codificação em sub-banda
- ATC (*adaptive transform coding*) - codificação por transformação adaptativa

ATC - codificação por transformação adaptativa

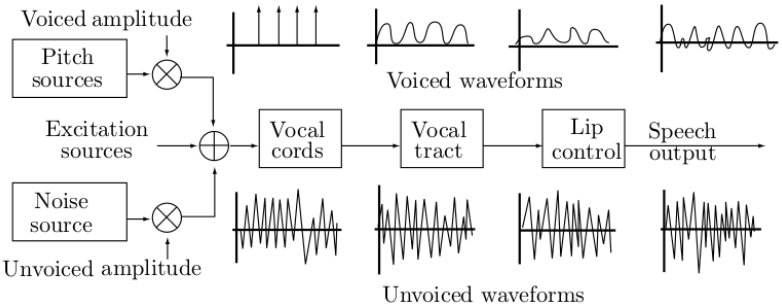
O arquivo de áudio é dividido em blocos de amostras e a DCT é aplicada em cada bloco, resultando em uma certa quantidade de coeficientes de frequências distintas. Cada coeficiente é quantizado de acordo com a sua frequência correspondente. É possível obter um sinal de áudio reconstruído com boa qualidade a taxas tão baixas quanto 16 kbps.

Codificação de Fonte

Um codificador de fonte utiliza um modelo matemático para a fonte produtora dos dados que desejamos codificar/transmitir. Se os dados originais correspondem a um sinal de voz, o codificador de fonte é chamado *vocodes* (*voice coder*). O exemplo mais comum é:

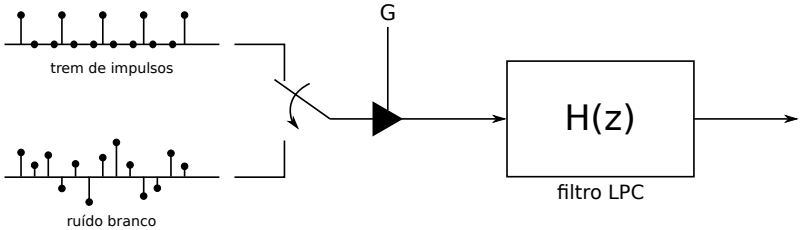
- LPC (linear predictive coder)

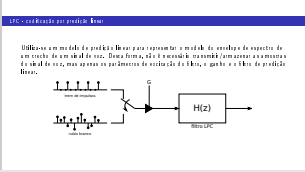
LPC - linear predictive coding



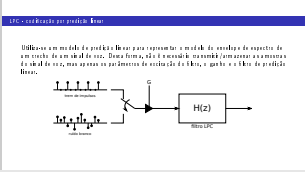
LPC - codificação por predição linear

Utiliza-se um modelo de predição linear para representar o modelo do envelope de espectro de um trecho de um sinal de voz. Desta forma, não é necessário transmitir/armazenar as amostras do sinal de voz, mas apenas os parâmetros de excitação do filtro, o ganho e o filtro de predição linear.





O LPC analisa o sinal de voz realizando um estimativa dos formantes, removendo o efeitos destes no sinal de fala, e estimando a intensidade e frequência do zumbido remanescente. O processo de remover o efeito dos formantes é chamado de filtragem inversa, e o sinal remanescente após a subtração do sinal modelado filtrado é chamado de resíduo. Os parâmetros que descrevem a intensidade e frequência do zumbido, os formantes, e o sinal de resíduo, podem ser armazenados ou transmitidos. A síntese LPC do sinal de fala utiliza o processo inverso: utiliza os parâmetros do zumbido e o resíduo para criar sinal da fonte, utiliza os formantes para criar um filtro (que representa a cavidade do trato vocal), e passa o sinal da fonte pelo filtro, resultando no sinal de fala. Como o sinal de fala varia ao longo do tempo, este processo é feito em pequenos trechos de sinal de fala, chamados quadros; usualmente de 30 a 50 quadros por segundo são utilizados para fornecer um sinal com boa compressão e fala inteligível.



- LPC é usualmente utilizado para análise e ressíntese de sinal de voz.
- Compressão de voz, como por exemplo no padrão GSM.
- Transmissão sem fio, segura e encriptada de voz através um canal estreito; por exemplo: Navajo I, governo EUA.
- Vocoders: aonde instrumentos musicais são utilizados como sinal de excitação para um filtro variante no tempo estimado a partir do sinal de voz de um cantor.
- A predição feita pelo LPC é utilizada em codecs de áudio como por exemplo: Shorten, MPEG-4 ALS, FLAC, e outros.

Codificação LPC

O LPC determina os coeficientes de um preditor linear através da minimização do erro de predição no sentido dos mínimos quadrados (uma das abordagens). Ele encontra os coeficientes de um preditor de ordem p (filtro FIR) que prediz o valor atual de uma série temporal real x a partir de amostras passadas.

$$\hat{x}[n] = -a_2 x[n-1] - a_3 x[n-2] - \cdots - a_{p+1} x[n-p] \tag{247}$$

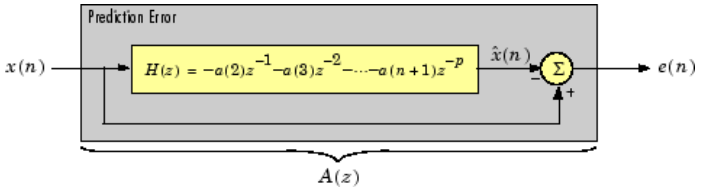


Figura 81: Erro de predição. `https://www.mathworks.com/help/signal/ref/lpc.html` (Mathworks).

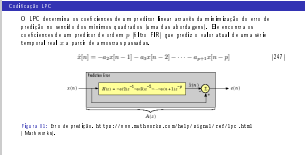


Figura 11: Erro de predição. https://www.mathworks.com/help/signal/ref/lpc.html | MathWorks.

O erro de predição, $e[n]$, pode ser visto como a saída do filtro de predição $A(z)$, onde $H(z)$ é o preditor linear ótimo, $x[n]$ é o sinal de entrada, e $\hat{x}[n]$ é o sinal predito.

LPC utiliza o método da autocorrelação da modelagem auto-regressiva (AR) para encontrar os coeficientes de filtro. O filtro gerado pode não modelar o processo exatamente, mesmo que a sequência de dados seja efetivamente proveniente de um modelo AR com ordem correta p adotada. Isto ocorre porque o método de autocorrelação implicitamente janela o sinal, isto é, assume que as amostras do sinal além do comprimento de x sejam nulas.

Otimização baseada nos mínimos quadrados I

O erro total é definido

$$\varepsilon = \sum_{n=n_0}^{n_1} e^2[n], \tag{248}$$

aonde n_0 e n_1 compreendem o limite da extensão sobre a qual será realizada a otimização. Como temos

$$e[n] = x[n] - \hat{x}[n] = x[n] + \sum_{i=1}^{p} a_i x[n-i] = \sum_{i=0}^{p} a_i x[n-i], \tag{249}$$

onde consideramos $a_0 = 1$. Substituindo 249 em 248 teremos:

$$\begin{aligned} \varepsilon &= \sum_{n=n_0}^{n_1} \left[\sum_{i=0}^{p} a_i x[n-i] \right]^2 = \sum_{n=n_0}^{n_1} \sum_{i=0}^{p} \sum_{j=0}^{p} a_i x[n-i] x[n-j] a_j \\ &= \sum_{i=0}^{p} \sum_{j=0}^{p} a_i c_{ij} a_j, \end{aligned} \tag{250}$$

Otimização baseada nos mínimos quadrados II

onde

$$c_{ij} = \sum_{n=n_0}^{n_1} x[n-i]x[n-j]. \tag{251}$$

Para minimizar ε em relação aos coeficientes do LPC, $a_1, a_2, \ldots, a_p$, devemos fazer a derivada parcial, em relação a cada coeficiente, igual a zero e resolver o sistema de equações remanescente.

$$\frac{\partial \varepsilon}{\partial a_k} = 2 \sum_{i=0}^{p} a_i c_{ik} = 0. \tag{252}$$

Como $a_0 = 1$, teremos então o sistema de equações normais:

$$\sum_{i=1}^{p} a_i c_{ik} = -c_{0k}, \quad k = 1, 2, \ldots, p. \tag{253}$$

Obtemos assim um sistema linear $p \times p$ cuja solução é obtida diretamente pela inversão matricial (complexidade $O(p^3)$, mas existe forma mais eficiente para resolver este sistema).

Método da Autocorrelação I

O método da autocorrelação escolhe os limites $n_0 = \infty$ e $n_1 = \infty$, e ao mesmo tempo, força $x[n] = 0$ para $n < 0$ e $n \geq N$, i.e., limita o sinal a uma janela de tamanho N. Teremos então c_{ij} dado pela função de autocorrelação $r(\tau)$:

$$\begin{aligned} c_{ij} &= \sum_{n=-\infty}^{\infty} x[n-i]x[n-j] \\ &= \sum_{n=0}^{N-1-|i-j|} x[n]x[n+|i-k|] = r(|i-k|). \end{aligned} \tag{254}$$

Devido ao truncamento para zero de $x[n]$ fora de uma janela de N amostras, teremos o equivalente à minimização do erro no intervalo $0 \leq n \leq N + p - 1$.
O sistema

$$\sum_{i=1}^{p} a_i c_{ik} = -c_{0k}, \quad k = 1, 2, \ldots, p. \tag{255}$$

Método da Autocorrelação II

será então reescrito como

$$\sum_{i=1}^{p} a_i r(|i-k|) = -r(k), \quad k = 1, 2, \ldots, p. \tag{256}$$

onde

$$r(\tau) = \sum_{n=0}^{N-1-|i-\tau} x[n]x[n+\tau], \quad \tau \geq 0. \tag{257}$$

O erro de predição do sinal será

$$e[n] = x[n] + \sum_{i=1}^{p} a_i x[n-i], \quad n = 0, 1, 2, \ldots, N+p-1. \tag{258}$$

Método da Autocorrelação III

Podemos também fazer a representação matricial do sistema de equações

$$\mathbf{Xa} = \mathbf{b} \tag{259}$$

onde $\mathbf{X} = \begin{bmatrix} x[0] & 0 & \cdots & 0 \\ x[1] & x[0] & \ddots & \vdots \\ \vdots & x[1] & \ddots & 0 \\ x[N-1] & \vdots & \ddots & x[0] \\ 0 & x[N-1] & \ddots & x[1] \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & x[N-1] \end{bmatrix}$, $\mathbf{a} = \begin{bmatrix} 1 \\ a[1] \\ \vdots \\ a[p] \end{bmatrix}$ $\mathbf{b} = \begin{bmatrix} x[0] \\ 0 \\ \vdots \\ 0 \end{bmatrix}$.

Método da Autocorrelação IV

Fazendo

$$\mathbf{X}^H\mathbf{X}\mathbf{a} = \mathbf{X}^H\mathbf{b} \tag{260}$$

teremos as equações de Yule-Walker

$$\begin{bmatrix} r[0] & r[1]^* & \cdots & r[p-1]^* \\ r[1] & r[0] & \ddots & \vdots \\ \vdots & \ddots & \ddots & r[1]^* \\ r[p-1] & \cdots & r[1] & r[0] \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = \begin{bmatrix} -r[1] \\ -r[2] \\ \vdots \\ -r[p] \end{bmatrix} \tag{261}$$

que podem ser resolvidas pelo algoritmo de Levinson-Durbin com ordem de complexidade $O(p^2)$.

Codificação Híbrida

Os codecs híbridos combinam características de ambos: codificação de forma de onda e codificação de fonte. O mais popular dos codecs híbridos são os algorismos no domínio do tempo de 'Análise através da Síntese', AbS (*Analysis-by-Synthesis*).

Analysis-by-Synthesis (AbS)

Os codificadores AbS começam com um conjunto de amostras de fala (um quadro), codifica-as de forma similar ao LPC, decodifica, e subtrai o resultado da decodificação do sinal original. A diferença sofre um processo de minimização do erro que fornece amostras melhor codificadas. Estas amostras são novamente decodificadas, subtraídas das amostras originais, e novas diferenças são calculadas. Este processo é repetido até que as diferenças satisfaçam uma determinada condição para terminar o processo. O codificado procede então para o próximo quadro.

CELP - Code-excited linear prediction

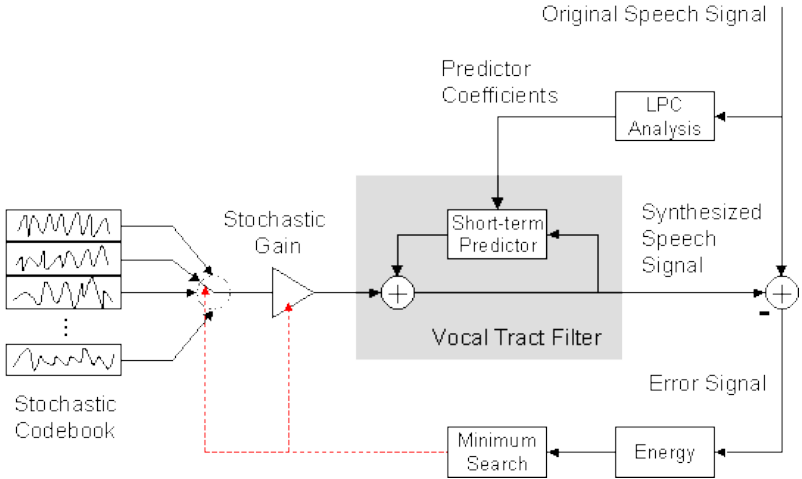
Uma das codificações mais conhecidas do tipo AbS é o CELP, um acronismo que significa predição linear por excitação por código.

- Proposto em 1985 por M.R. Schroeder e B.S. Atal.
- É atualmente o algoritmo de codificação de voz mais utilizado.
- Utilizado no padrão para codificação de voz do MPEG-4 Audio.

CELP - principais ideias

- Utilizar o modelo fonte-filtro da produção da fala através da predição linear (LP).
- Utilizar um codebook adaptativo e fixo como entrada (excitação) do modelo LP.
- Buscar em um loop fechado em um domínio ponderado pelas características perceptivas.
- Quantização vetorial.

Codificador CELP



Decodificador CELP

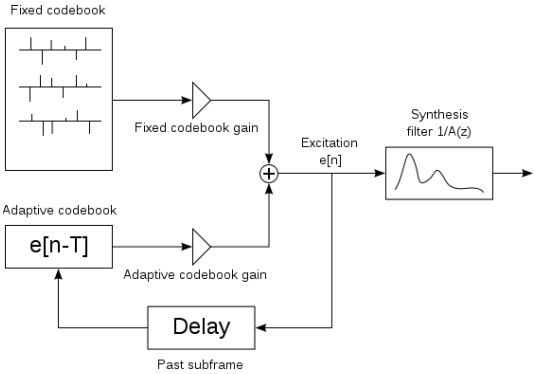


Figura 82: Decodificador CELP (Wikipedia).

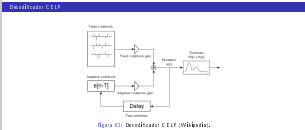


Figura 33: Decodificador CELP (Wikipedia).

A excitação é produzida pela soma das contribuições de um codebook adaptativo (aka pitch) e um codebook estocástico (aka fixo). O codebook fixo é quandizado vetorialmente utilizando um dicionário que é previamente definido no codec. As entradas do codebook adaptativo consistem em versões atrasadas da excitação. Isto torna possível codificar eficientemente sinais periódicos, tais como sinais vozeados. O filtro que molda a excitação é constituído por um modelo com apenas pólos obtido através da predição linear.

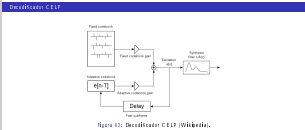


Figura 33: Decodificador CELP (Wikipédia).

A codificação (análise) é realizada através de uma otimização perceptiva do sinal decodificado (síntese) em um loop fechado, utilizando uma função perceptiva simples de ponderação. A codificação é realizada seguindo a seguinte ordem:

1. Os coeficientes LPC são calculados e quantizados.
2. É realizada uma busca no codebook adaptativo (pitch) e sua contribuição é removida.
3. É realizada a busca no codebook fixo.

Notebook - LPC vogais

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/lpc_vowels_formants.ipynb

Notebook - LPC síntese

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/lpc-synthesis.ipynb

Leitura

Sugestão de leitura:

- Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580
- Salomon, D., Bryant, D., and Motta, G. (2010). *Handbook of Data Compression*. Springer London
- Kondoz, A. M. (2004). *Digital Speech: Coding for Low Bit Rate Communication Systems*. Wiley

DPCM

O DPCM (*Differential pulse-code modulation*) foi criado 1950 por Cassius Chapin Cutler. O codificador utiliza um preditor linear para o sinal a ser codificado. O erro de predição é quantizado e transmitido/armazenado. Para a decodificação utiliza-se o mesmo preditor e soma-se o erro de predição recebido na entrada do decodificador.

DPCM - Codificador e Decodificador

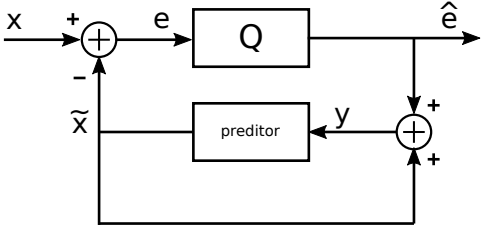


Figura 83: Codificador DPCM.

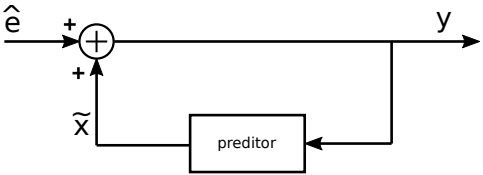
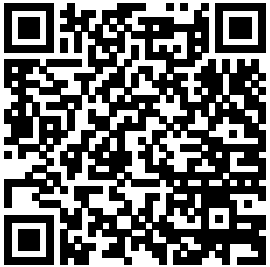


Figura 84: Decodificador DPCM.

DPCM - exemplo



https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/dpcm_example_image.ipynb

Leitura

Sugestão de leitura:

- Salomon, D., Bryant, D., and Motta, G. (2010). *Handbook of Data Compression*. Springer London
- Yehia, H. C. (1993). Análise de funções de erro em sistemas de codificação lpc. Master's thesis, ITA

Transformadas de comprimento finito

Equação de análise:

$$A[k] = \sum_{n=0}^{N-1} x[n]\phi_k^*[n] \tag{262}$$

Equação de síntese:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} A[k]\phi_k[n] \tag{263}$$

As sequências $\phi_k[n]$ são chamadas de **sequências de base** e são ortogonais entre si, i.e.

$$\langle \phi_k, \phi_m \rangle = \frac{1}{N}\sum_{n=0}^{N-1} \phi_k[n]\phi_m^*[n] = \begin{cases} 1, & m = k \\ 0, & m \neq k. \end{cases} = \delta_{m,k}. \tag{264}$$

Discrete Time Fourier Transform (DFT)

Para a DFT temos

$$\phi_k[n] = e^{j\frac{2\pi kn}{N}} \tag{265}$$

Existem N exponencias complexas distintas: $\phi_0, \phi_1, \ldots, \phi_{N-1}$.
$\{\phi_k\}_{k\in[0,N-1]}$ forma uma base para o espaço das sequências de tamanho N.

$$x[n] \underset{\text{iDFT}}{\overset{\text{DFT}}{\rightleftarrows}} X[k] \tag{266}$$

Se $x[n]$ é real, então $X[k]$ é par e complexo.

Outras Transformadas

Existe uma base $\{\phi_k\}$ tal que sendo $x[n]$ real teremos $X[k]$ também real?

- Transformada Haar
- Transformada Hadamard
- Transformada Hartley
- Transformada Discreta em Cossenos

Definição da DCT

- As sequências de base ϕ_k são cossenos.
- Cossenos são funções periódicas com simetria par.
- A extensão de $x[n]$ fora do intervalo $0 \leq n \leq (N-1)$ na equação

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} A[k]\phi_k[n] \tag{267}$$

  será periódica e com simetria par.
- sequência finita $\rightarrow$ sequência periódica
- sequência periódica $\rightarrow$ sequência finita
- Existem 8 formas diferentes de fazer a extensão periódica de uma sequência.
  - borda esquerda/direita; simetria par/ímpar; ponto dos dados/ponto intermediário

Simetrias

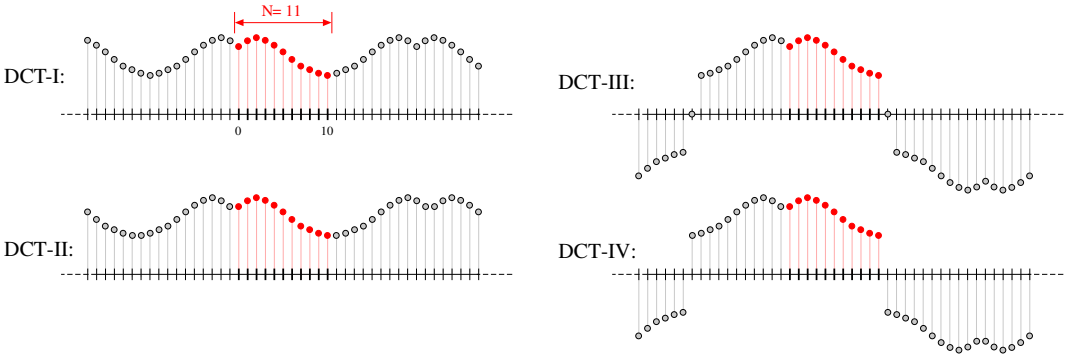
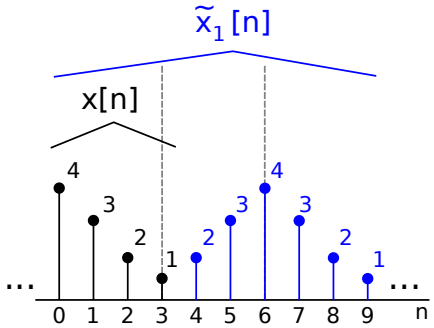


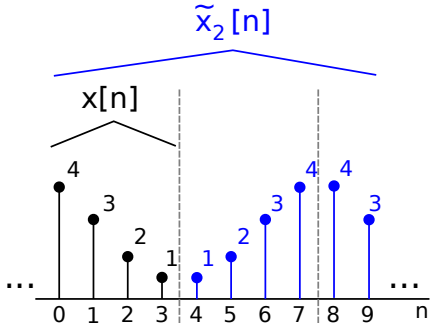
Figura 85: Extensão par e ímpar na definição dos tipos de DCT. (Wikipedia)

DCT-I



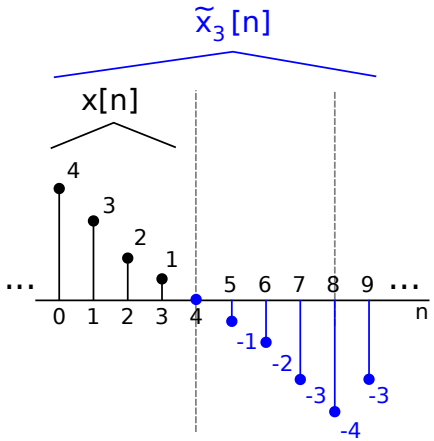
$\tilde{x_1}[n]$ possui período $(2N-2)$ e possui simetria par em torno de $n=0$ e $n=(N-1)$.

DCT-II



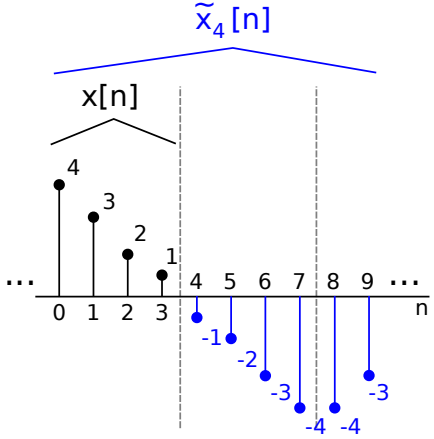
$\tilde{x_2}[n]$ possui período $2N$ e possui simetria par em torno de $n = -1/2$ e $n = N - 1/2$.

DCT-III



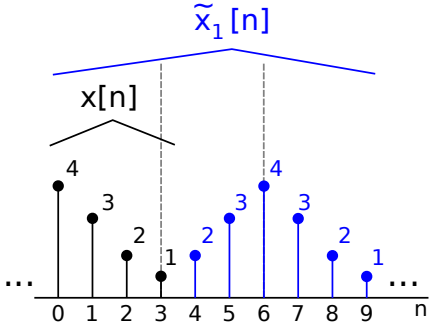
$\tilde{x_3}[n]$ possui período $4N$ e possui simetria par em torno de $n = 0$ e ímpar em torno de $n = N$.

DCT-IV



$\tilde{x_4}[n]$ possui período $4N$ e possui simetria par em torno de $n = -1/2$ e ímpar em torno de $n = N - 1/2$.

DCT-I I



DCT-I II

$\tilde{x_1}[n]$ possui período $(2N-2)$ e possui simetria par em torno de $n=0$ e $n=(N-1)$.

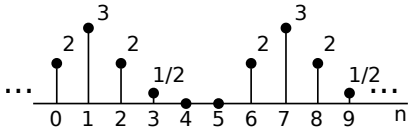
$$\tilde{x_1}[n] = x_\alpha\left[((n))_{2N-2}\right] + x_\alpha\left[((-n))_{2N-2}\right] \tag{268}$$

onde $((n))_N$ significa $n \mod N$, e a sequência modificada é definida por $x_\alpha[n] = \alpha[n]x[n]$, com

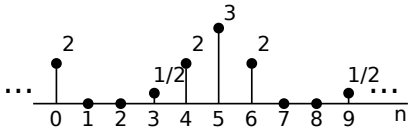
$$\alpha[n] = \begin{cases} \frac{1}{2}, & n = 0 \text{ e } N-1, \\ 1, & 1 \leq n \leq N-2. \end{cases} \tag{269}$$

DCT-I III

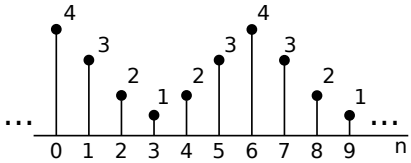
$x_\alpha[((n))_{2N-2}]$



$x_\alpha[((-n))_{2N-2}]$



$\widetilde{x}_1[n]$



DCT-I IV

- $\tilde{x_1}[n]$ possui período igual a $2N-2$.
- base de cossenos harmonicamente relacionados, $\omega_0 = 2\pi/(2N-2)$, teremos: $\cos\left(\frac{2\pi kn}{2N-2}\right) = \cos\left(\frac{\pi kn}{N-1}\right)$.

$$X[k] = \sum_{n=0}^{2N-3} \tilde{x_1}[n] \cos\left(\frac{\pi kn}{N-1}\right) \quad (270)$$

$$= \tilde{x_1}[0]\cos(0) + \sum_{n=1}^{N-2} \tilde{x_1}[n] \cos\left(\frac{\pi kn}{N-1}\right) \quad (271)$$

$$+\tilde{x_1}[N-1]\cos\left(\frac{\pi k(N-1)}{N-1}\right) + \sum_{n=N}^{2N-3} \tilde{x_1}[n] \cos\left(\frac{\pi kn}{N-1}\right)$$

DCT-I V

fazendo $n' = 2N - 2 - n$ no segundo somatório teremos

$$\begin{aligned} X[k] &= \dots \\ &\tilde{x_1}[0]\cos(0) + \sum_{n=1}^{N-2} \tilde{x_1}[n]\cos\left(\frac{\pi k n}{N-1}\right) \\ &+\tilde{x_1}[N-1]\cos\left(\frac{\pi k(N-1)}{N-1}\right) \\ &+ \sum_{n'=N-2}^{1} \underbrace{\tilde{x_1}[2N-2-n']}_{=\tilde{x_1}[n]=\tilde{x_1}[n']} \cos\left(\frac{\pi k(2N-2-n')}{N-1}\right) \end{aligned} \tag{272}$$

$\tilde{x_1}$ é par

DCT-I VI

Iremos utilizar $\cos(A+B) = \cos A \cos B - \sin A \sin B$

$$\begin{aligned}
\cos\left(\frac{\pi k(2N-2-n')}{N-1}\right) &= \underbrace{\cos\left(\frac{\pi k(2N-2)}{N-1}\right)}_{=1}\cos\left(\frac{-n'k\pi}{N-1}\right) \\
&\quad - \underbrace{\sin\left(\frac{\pi k(2N-2)}{N-1}\right)}_{=0}\sin\left(\frac{-n'k\pi}{N-1}\right) \\
&= \cos\left(\frac{n'k\pi}{N-1}\right) \qquad (273)
\end{aligned}$$

DCT-I VII

$$
\begin{aligned}
X[k] &= \dots \\
&\quad \tilde{x_1}[0]\cos(0) + \sum_{n=1}^{N-2} \tilde{x_1}[n]\cos\left(\frac{\pi k n}{N-1}\right) && (274) \\
&\quad + \tilde{x_1}[N-1]\cos\left(\frac{\pi k (N-1)}{N-1}\right) \\
&\quad \sum_{n'=1}^{N-2} \tilde{x_1}[n']\cos\left(\frac{n' k \pi}{N-1}\right) && (275) \\
&= 2\sum_{n=0}^{N-1} \alpha[n]x[n]\cos\left(\frac{\pi k n}{N-1}\right) && (276)
\end{aligned}
$$

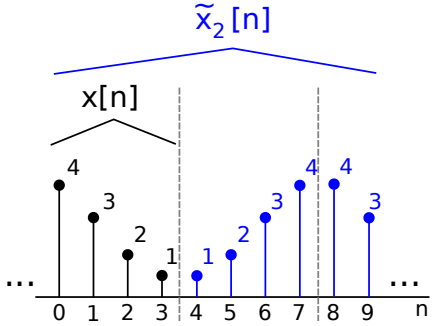
DCT-I VIII

A DCT-I é definida pelo par de transformada:

$$X^{C1}[k] = 2\sum_{n=0}^{N-1} \alpha[n]x[n]\cos\left(\frac{\pi kn}{N-1}\right), \quad 0 \leq k \leq N-1, \tag{277}$$

$$x[n] = \frac{1}{N-1}\sum_{k=0}^{N-1} \alpha[k]X^{C1}[k]\cos\left(\frac{\pi kn}{N-1}\right), \quad 0 \leq n \leq N-1. \tag{278}$$

DCT-II I



$\tilde{x_2}[n]$ possui período $2N$ e possui simetria par em torno de $n = -1/2$ e $n = N - 1/2$. $x[n]$ é estendido para ter período $2N$, sendo a sequência periódica dada por

$$\tilde{x_2}[n] = x\left[((n))_{2N}\right] + x\left[((-n-1))_{2N}\right] \tag{279}$$

DCT-II II

A DCT-II é definida pelo par de transformada:

$$X^{C2}[k] = 2\sum_{n=0}^{N-1} x[n]\cos\left(\frac{\pi k(2n+1)}{2N}\right), \quad 0 \leq k \leq N-1. \tag{280}$$

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} \beta[k]X^{C2}[k]\cos\left(\frac{\pi k(2n+1)}{2N}\right), \quad 0 \leq n \leq N-1. \tag{281}$$

onde a função de peso $\beta[k]$ é dada por

$$\beta[k] = \begin{cases} \frac{1}{2}, & k = 0 \\ 1, & 1 \leq k \leq N-1. \end{cases} \tag{282}$$

DCT-II III

Muitas vezes inclui-se um fator de normalização para tornar a transformada unitária, ou seja, deverá ser ortonormal e terá a propriedade $\sum_{n=0}^{N-1}(x[n])^2 = \sum_{k=0}^{N-1}\left(X^{C2}[k]\right)^2$.

$$\tilde{X}^{C2}[k] = \sqrt{\frac{2}{N}}\tilde{\beta}[k]\sum_{n=0}^{N-1} x[n]\cos\left(\frac{\pi k(2n+1)}{2N}\right), \quad 0 \leq k \leq N-1. \tag{283}$$

$$x[n] = \sqrt{\frac{2}{N}}\sum_{k=0}^{N-1} \tilde{\beta}[k]\tilde{X}^{C2}[k]\cos\left(\frac{\pi k(2n+1)}{2N}\right), \quad 0 \leq n \leq N-1. \tag{284}$$

onde a função de peso $\tilde{\beta}[k]$ é dada por

$$\tilde{\beta}[k] = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & 1 \leq k \leq N-1. \end{cases} \tag{285}$$

Forma Matricial I

Para $0 \leq k \leq N-1$m temos

$$\tilde{X}^{C2}[k] = \sqrt{\frac{2}{N}}\tilde{\beta}[k]\sum_{n=0}^{N-1} x[n]\cos\left(\frac{\pi k(2n+1)}{2N}\right) \quad (286)$$

$$= \sum_{n=0}^{N-1} x[n]\sqrt{\frac{2}{N}}\tilde{\beta}[k]\cos\left(\frac{\pi k(2n+1)}{2N}\right) \quad (287)$$

$$= \sum_{n=0}^{N-1} x[n]c[k,n] \quad (288)$$

onde definimos

$$c[k,n] \triangleq \sqrt{\frac{2}{N}}\tilde{\beta}[k]\cos\left(\frac{\pi k(2n+1)}{2N}\right) \quad k,n = 0,1,\ldots,N-1 \quad (289)$$

Forma Matricial II

Temos então a matriz $N \times N$ da transformada em cossenos

$$\begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & c[n,m] & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0^T \\ \cdots \\ \mathbf{c}_{N-1}^T \end{bmatrix} = \mathbf{C}^T \tag{290}$$

$\mathbf{c}_i^T = [c[i,0], \cdots, c[i,N-1]$ é a i-ésima linha da matriz de DCT $\mathbf{C}$. Estes vetores linha são ortonormais:

$$\langle \mathbf{c}_i, \mathbf{c}_j \rangle = \mathbf{c}_i^T \mathbf{c}_j = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{291}$$

então a matriz $\mathbf{C}$ é ortonormal

$$\mathbf{C}^{-1} = \mathbf{C}^T, \quad \text{i.e.} \quad \mathbf{C}^T \mathbf{C} = \mathbf{I} \tag{292}$$

e real $\mathbf{C} = \mathbf{C}^*$. A DCT sobre $x[n] = \mathbf{x}$ pode ser expressa na forma matricial

$$\mathbf{X} = \mathbf{C}^T \mathbf{x} \tag{293}$$

DCT 2D I

Considere uma imagem $N \times N$, $\mathbf{A}$. A 2D-DCT de $\mathbf{A}$ é dada por $\mathbf{Y} = \mathbf{C}\mathbf{X}\mathbf{C}^T$, e a inversa é $\mathbf{X} = \mathbf{C}^T\mathbf{X}\mathbf{C}$.

DCT 2D II

```
X =

     9    10     8     8     8    11     8     8
     8    10     8    10     8     9    10    10
    10     9    10    10    10     9     9    10
     9    11     9     8    10     8    11     9
     8    10    11    11    10    10     9     8
    11     9     9    11    11     8     8     9
     8    10     8     9    11    10    10    10
    10     8    10     8    10     8    10     9
```

```
octave:12> Xdct = dct2(X)
Xdct =

   74.625000    0.190507   -0.737346    0.225661    0.125000   -0.787713   -1.453469
   -0.931856    0.050711    0.929243   -0.040246   -1.504944   -0.153404    0.206108
   -1.740481   -1.077303    1.359835    1.247607   -0.321833   -0.162454    0.941942
   -0.314447    1.011484   -1.561226    0.883728    0.166465   -0.602120    0.418059
   -0.875000    0.786481    0.028021    0.535813   -2.375000    0.020869    0.394290
   -0.033977   -0.627973    0.988461    1.519200   -0.490116   -1.633728   -0.696998
    0.044436    2.122833   -0.058058    0.486956    2.162793    0.833284    1.890165
    0.684918   -0.372995   -1.813402   -0.169819   -0.809147   -0.302809    1.534776
```

```
octave:13> round(Xdct)
ans =

   75    0   -1    0    0   -1   -1    0
   -1    0    1   -0   -2   -0    0   -2
   -2   -1    1    1   -0   -0    1   -1
   -0    1   -2    1    0   -1    0   -2
   -1    1    0    1   -2    0    0    1
   -0   -1    1    2   -0   -2   -1    0
    0    2   -0    0    2    1    2    1
    1   -0   -2   -0   -1   -0    2   -1
```

```
octave:14> idct2(round(Xdct))
ans =

    8.7493   10.1235    7.8660    8.2482    8.3112   11.1295    8.0026    7.8454
    7.7606    9.9461    8.5292    9.8486    7.8115    8.9943   10.0979    9.9677
    9.9991    8.8032   10.5970    9.9802    9.4472    9.3406    9.0625   10.2427
    9.0575   11.0616    8.4302    7.6851   10.3434    7.9997   11.1072    9.2655
    8.2667   10.2259   11.4582   10.9890   10.1718   10.3699    9.0429    7.7518
   10.4791    9.2223    9.3069   11.0495   11.0876    8.1074    8.0293    9.4101
    8.0251   10.0475    7.7589    9.4060   11.2171   10.6002    9.5565   10.2678
   10.0292    7.7224   10.0122    8.4653    9.6339    7.9689    9.4793    9.1868
```

Figura 86: Exemplo DCT.

Relação entre DCT-I e DFT I

$$\tilde{x_1}[n] = x_\alpha\left[((n))_{2N-2}\right] + x_\alpha\left[((-n))_{2N-2}\right] \tag{294}$$

$\tilde{x_1}[n]$ é construído a partir de $x[n]$ e $\alpha[n]$. Podemos definir uma sequência finita $x_1[n]$ com base na sequência periódica $\tilde{x_1}[n]$:

$$x_1[n] = x_\alpha\left[((n))_{2N-2}\right] + x_\alpha\left[((-n))_{2N-2}\right] = \tilde{x_1}[n], \quad n = 0, 1, \ldots, 2N-3. \tag{295}$$

Relação entre DCT-I e DFT II

A DFT da sequência $x_1[n]$ com $(2N-2)$ pontos é dada por

$$\begin{aligned} X_1[k] &= \text{DFT}\{x_1[n]\} && (296)\\ &= \text{DFT}\{x_\alpha\left[((n))_{2N-2}\right]\} + \text{DFT}\{x_\alpha\left[((-n))_{2N-2}\right]\} && (297)\\ &= X_\alpha[k] + X_\alpha^*[k] && (298)\\ &= 2Re\{X_\alpha[k]\}, \quad k = 0, 1, \ldots, 2N-3 && (299)\\ &= 2\sum_{n=0}^{N-1} \alpha[n]x[n]\cos\left(\frac{2\pi kn}{2N-2}\right) = X^{C1}[k]. && (300) \end{aligned}$$

$X_\alpha[k]$ é a DFT de $(2N-2)$ pontos da sequência $\alpha[n]x[n]$ de N pontos, ou seja, devemos preencher $\alpha[n]x[n]$ com $(N-2)$ zeros.
Então a DCT-I de uma sequência de N pontos é idêntica à DFT de $(2N-2)$ pontos da sequência estendida $x_1[n]$, e também idêntica à duas vezes a parte real dos primeiros N pontos da DFT de $(2N-2)$ pontos da sequência $x_\alpha[n]$.

Relação entre DCT-I e DFT III

A DCT é $O(N^2)$, enquanto a DFT possui um algoritmo rápido (FFT) $O(N \log N)$. Poderemos utilizar a FFT para calcular $X_\alpha[k]$ ou $X_1[k]$ e desta forma teremos uma forma mais conveniente para computar a DCT-I.
Como a DCT-I envolve apenas coeficientes reais, existem também algoritmos eficientes para calcular a DCT-I de sequências reais de forma direta, sem a necessidade de multiplicações e adições de números complexos Ahmed et al. (1974); Chen et al. (1977).

Relação entre DCT-I e DFT IV

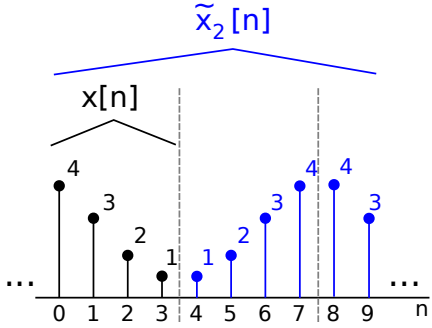
A DCT-I inversa também pode ser calcular através da DFT inversa, para tanto, iremos construir $X_1[k]$ a partir de $X^{C1}[k]$ e então calcular a DFT inversa de $(2N-2)$ pontos.

$$X_1[k] = \begin{cases} X^{C1}[k], & k = 0, \ldots, N-1 \\ X^{C1}[2N-2-k], & k = N, \ldots, 2N-3. \end{cases} \tag{301}$$

Utilizando a DFT inversa de $(2N-2)$ pontos teremos

$$x_1[n] = \frac{1}{2N-2} \sum_{k=0}^{2N-3} X_1[k] e^{j2\pi kn/(2N-2)}, \quad n = 0, 1, \ldots, 2N-3. \tag{302}$$

Relação entre DCT-II e DFT I



Podemos construir $x_2[n]$ a partir de um período de $\tilde{x}_2[n]$.

$$x_2[n] = x\left[((n))_{2N}\right] + x\left[((-n-1))_{2N}\right] = \tilde{x}_2[n], \quad n = 0, 1, 2, \ldots, 2N-1. \tag{303}$$

Relação entre DCT-II e DFT II

$X_2[k]$ é a DFT de $2N$ pontos de $x_2[n]$,

$$X_2[k] = X[k] + X^*[k]e^{j\frac{2\pi k}{2N}}, \quad k = 0, 1, \ldots, 2N-1, \tag{304}$$

onde $X[k]$ é a DFT de $2N$ pontos da sequência $x[n]$ de N pontos, ou seja, $x[n]$ deve ser estendido com N zeros.

Relação entre DCT-II e DFT III

$$X_2[k] = X[k] + X^*[k]e^{j\frac{2\pi k}{2N}} \quad (305)$$

$$= e^{j\frac{\pi k}{2N}}\left(X[k]e^{-j\frac{\pi k}{2N}} + X^*[k]e^{j\frac{\pi k}{2N}}\right) \quad (306)$$

$$= e^{j\frac{\pi k}{2N}} 2\Re\left\{X[k]e^{-j\frac{\pi k}{2N}}\right\} \quad (307)$$

A DFT de $X[k]$ ($2N$ pontos, com N zeros) é dada

$$X[k] = \sum_{n=0}^{2N-1} x[n]e^{-j\frac{2\pi kn}{2N}} \quad (308)$$

como $x[n]$ foi estendido com N zeros (309)

$$= \sum_{n=0}^{N-1} x[n]e^{-j\frac{\pi kn}{N}} \quad (310)$$

Relação entre DCT-II e DFT IV

Substituindo na equação de $X_2[k]$, teremos

$$X_2[k] = e^{j\frac{\pi k}{2N}} 2\Re\left\{X[k]e^{-j\frac{\pi k}{2N}}\right\} \quad (311)$$

$$= e^{j\frac{\pi k}{2N}} 2\Re\left\{\sum_{n=0}^{N-1} x[n]e^{-j\frac{\pi kn}{N}} e^{-j\frac{\pi k}{2N}}\right\} \quad (312)$$

$$= e^{j\frac{\pi k}{2N}} 2\Re\left\{\sum_{n=0}^{N-1} x[n]e^{-j\frac{\pi k(2n+1)}{2N}}\right\} \quad (313)$$

$$= e^{j\frac{\pi k}{2N}} \underbrace{2\sum_{n=0}^{N-1} x[n]\cos\left(\frac{\pi k(2n+1)}{2N}\right)}_{X^{C2}[k]} \quad (314)$$

onde utilizamos o fato de que $x[n]$ é real.

Relação entre DCT-II e DFT V

Concluímos então que

$$X^{C2}[k] = 2\Re\left\{X[k]e^{-j\frac{\pi k}{2N}}\right\}, \quad k = 0, 1, \ldots, N-1. \tag{315}$$

onde $X[k]$ é a DFT de $2N$ pontos de $x[n]$ estendido com N zeros. Ou então, utilizando que $X_2[k] = e^{j\frac{\pi k}{2N}} 2\Re\left\{X[k]e^{-j\frac{\pi k}{2N}}\right\}$, poderemos escrever

$$X^{C2}[k] = e^{-j\frac{\pi k}{2N}} X_2[k], \quad k = 0, 1, \ldots, N-1, \tag{316}$$

ou seja, $X^{C2}[k]$ pode ser obtido através da DFT de $2N$ pontos de $x_2[n]$, a extensão simétrica par de $x[n]$.

- Podemos utilizar a FFT para calcular a DFT, e assim poderemos também calcular a DCT-II através da FFT.
- Também é possível calcular a inversa da DCT-II utilizando a iFFT (FFT inversa).

Relação entre DCT-II e DFT VI

- Embora a complexidade da aplicação direta da fórmula da DCT requereria $O(N^2)$ operações, é possível utilizar a transformada rápida de Fourier (FFT), cuja complexidade é $O(N \log N)$, e calcular a DCT através da FFT com um pré- e pós-processamento de $O(N)$ operações.

Compactação de Energia na DCT-II I

É preferida a utilização da DCT-II em diversas aplicações de compressão de dados em detrimento da DFT pois aquela possui a propriedade de compactação de energia. A DCT-II de uma sequência finita usualmente possui coeficientes mais concentrados do que a DFT. A importância disso segue do teorema de Parseval que, para a DCT-II é

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} \beta[k] |X^{c2}[k]|^2 \tag{317}$$

onde

$$\beta[k] = \begin{cases} \frac{1}{2}, & k = 0 \\ 1, & 1 \leq k \leq N-1 \end{cases} \tag{318}$$

Compactação de Energia na DCT-II II

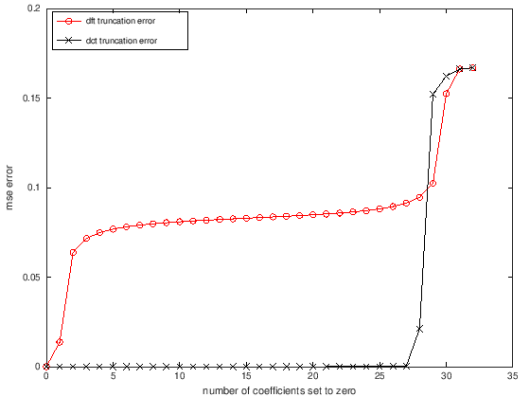


Figura 87: Efeito de compactação de energia da DCT.

Notebook DCT - GNU Octave

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/DCT.ipynb

Compressão JPEG I

O padrão JPEG (*Joint Photographers Expert Group*) é amplamente difundido e utilizando em câmeras digitais e na internet.

Esforço conjunto: ISO/IEC JTC 1 + ITU-T
International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), International Telecommunication Union (ITU)

Ele utiliza a transformada discreta em cossenos (DCT).

Compressão JPEG II

O padrão é dividido em 6 partes:

1) Requisitos e diretrizes,
2) Teste de conformidade,
3) Extensões,
4) Métodos para registrar parâmetros usados pelo JPEG estendido,
5) Formato de intercâmbio de arquivos JPEG (JFIF),
6) Aplicações para sistemas de impressão.

`https://en.wikipedia.org/wiki/JPEG`

Compressão JPEG III

- método de compressão com/sem perdas (*lossy/lossless*) para imagens coloridas/tons de cinza
- não lida bem com imagens de dois níveis (preto e branco)
- melhor desempenho em imagens de tom contínuo, onde pixels adjacentes possuem cores similares
- utiliza diversos parâmetros que podem ser ajustados
- degradação quase imperceptível de qualidade mesmo para fatores de compressão de 10 a 20

Compressão JPEG IV

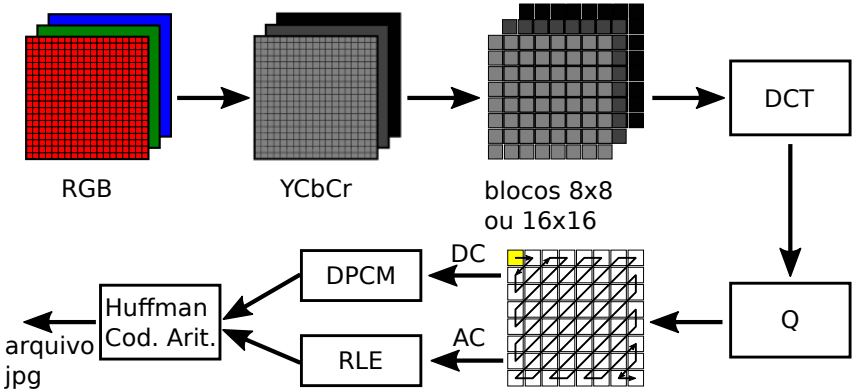


Figura 88: Esquema do padrão de compressão JPEG.

Compressão JPEG V

A compressão JPEG segue os seguintes passos:

1) Separação em intensidade e cor (RGB - YCbCr). Subdivisão em blocos 8×8-pixels.
2) Aplica-se a DCT a cada bloco, armazenados temporariamente com 12 bits (11 bits de precisão e 1 bit de sinal).
3) Quantização dos blocos de 64 coeficientes através da divisão por uma matriz fixa com menor precisão nos termos de alta frequência.
4) O primeiro coeficiente fornece o brilho médio (DC), sendo codificado pela diferença em relação ao mesmo termo no bloco antecessor.
5) Os demais 63 coeficientes são organizados em zigzag de baixas frequências até alta frequência e são codificados por RLE.
6) O fluxo de dados final é codificado por código de Huffman ou codificação aritmética.

Compressão JPEG VI

- Compressão e descompressão para DCT são simétricos (mesma complexidade computacional e tempo de execução)
- Perda de termos de alta frequência acarreta distorções.
- Em geral, qualquer compressão com perdas não devem ser utilizadas em imagens destinadas a medições e análise.
- Usualmente a análise de qualidade por humanos é empregada para comparar diferentes métodos.

Compressão JPEG VII

Figura 89: Artefatos criados pela compressão.

Especificação original do JPEG é de 1992.

A compressão de imagens com perdas baseada na DCT foi originalmente proposta por Nasir Ahmed em 1972. (Ahmed, Nasir; Natarajan, T.; Rao, K. R. (January 1974), 'Discrete Cosine Transform', IEEE Transactions on Computers, C-23 (1): 90-93, doi:10.1109/T-C.1974.223784)

Inicialmente o padrão JPEG utilizava apenas a codificação de Huffman, como codificação de entropia. Existe a possibilidade de utilizar alternativamente a codificação aritmética (levando a arquivos de 5 a 7% menores), porém por causa da proteção de patente, a maior parte dos codificadores e decodificadores implementaram apenas a codificação de Huffman.

Subamostragem de Crominância I

Notação **J:a:b**.
Cada bloco terá J pixels de largura e 2 de altura.
a: número de amostras de crominância presentes na primeira linha
b: número de amostras de crominância presentes na segunda linha
H: resolução relativa na direção horizontal
V: resolução relativa na direção vertical
T: resolução relativa final

Subamostragem de Crominância II

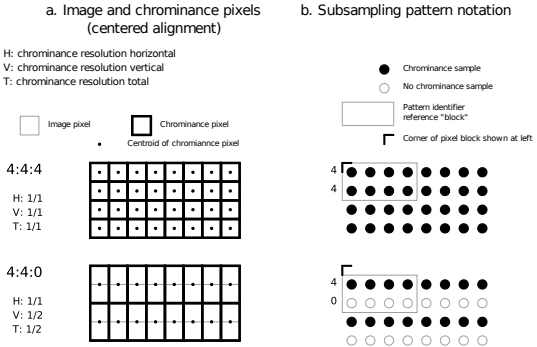


Figura 90: Esquema de subamostragem de crominância (Kerr, 2012).

Subamostragem de Crominância III

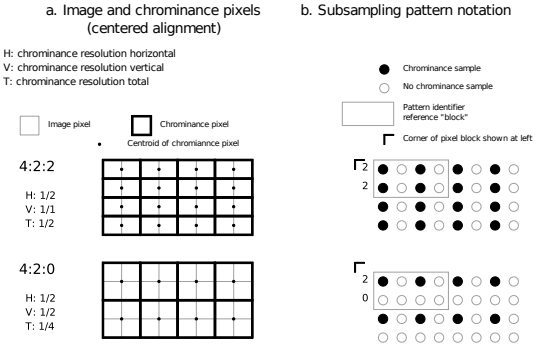


Figura 91: Esquema de subamostragem de crominância (Kerr, 2012).

Subamostragem de Crominância IV

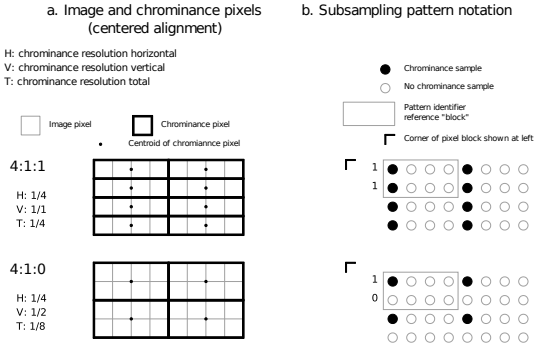


Figura 92: Esquema de subamostragem de crominância (Kerr, 2012).

Quantização dos coeficientes da DCT I

Após o cálculo dos 8×8 coeficientes da DCT, eles serão quantizados. Nesta etapa há efetivamente a perda de informação.

Quantização:

- cada coeficiente da DCT é dividido pelo elemento correspondente na tabela de quantização,
- o resultado é arredondado para o inteiro mais próximo

Cada componente de cor utilizada uma tabela de quantização diferente. O padrão JPEG permite a utilização de até quatro tabelas e o usuário pode selecionar qualquer uma das quatro para quantizar qualquer componente de cor.

Quantização dos coeficientes da DCT II

As tabelas padrão de quantização. Estas tabelas para luminância e crominância são resultado de diversos experimentos realizados pelo comitê do JPEG.

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

luminância

$$\begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

crominância

Quantização dos coeficientes da DCT III

Tabela de quantização simples Q calculada com base em um parâmetro, R, definido pelo usuário. Os elementos da matriz são dados por $Q_{ij} = 1 + (i + j) \times R$.

Para $R = 2$ teremos a seguinte matriz:

$$\begin{bmatrix} 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 \\ 3 & 5 & 7 & 9 & 11 & 13 & 15 & 17 \\ 5 & 7 & 9 & 11 & 13 & 15 & 17 & 19 \\ 7 & 9 & 11 & 13 & 15 & 17 & 19 & 21 \\ 9 & 11 & 13 & 15 & 17 & 19 & 21 & 23 \\ 11 & 13 & 15 & 17 & 19 & 21 & 23 & 25 \\ 13 & 15 & 17 & 19 & 21 & 23 & 25 & 27 \\ 15 & 17 & 19 & 21 & 23 & 25 & 27 & 29 \end{bmatrix}$$

Imagens de Teste I

Testar e comparar diferentes método com relação à eficiência e custo computacional.

- Calgary Corpus
- Canterbury Corpus
- conjunto de documentos do ITU-T para compressão de fax

Imagens de Teste II

Lossless JPEG

- 1993
- codificação preditiva - DPCM

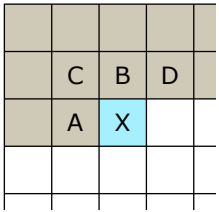


Figura 98: Esquema de predição de pixel no JPEG sem perdas (Wikipedia).

- codificador de entropia - codificação Huffman coding ou codificação aritmética

JPEG2000

- utiliza wavelets
- representação em múltiplas resoluções
- quantizador escalar com zona morta para codificar os coeficientes
- codificação aritmética binária adaptativa guiada por contexto (MQ coder)
- patentes

JPEG-LS

- 2003
- padrão de compressão sem (ou quase sem) perdas
- algoritmo LOCO-I: predição, modelagem do resíduo e codificação do resíduo baseada em contexto
- resíduo segue uma distribuição de Laplace
- códigos Golomb são quase ótimos para este tipo de distribuição
- mais rápido que o JPEG2000 e melhores resultados que o JPEG original

JPEG XT

JPEG XT (ISO/IEC 18477)

- 2015
- Retro compatível com o padrão JPEG (ISO/IEC 10918-1 e ITU Rec. T.81).
- Suporte a maior extensão dinâmica (imagens HDR).
- Suporte a camada alfa (transparência).
- Informações adicionais são salvas como *metadata* do arquivo no padrão JPEG.

Guetzli I

- Projeto do Google (2017)
- `https://github.com/google/guetzli`
- Guetzli: Perceptually Guided JPEG Encoder (arXiv:1703.04421v1)
- utiliza a implementação `libjpeg` do grupo independente JPE
- utiliza a medida de qualidade `Butteraugli` (utiliza modelos de percepção de cor e mascaramento visual)
- loop no processo de codificação para encontrar a 'melhor' matriz de quantização e descartar coeficientes para melhorar o desempenho do RLE
- imagens de alta qualidade com redução de aprox. 35% no tamanho

Guetzli II

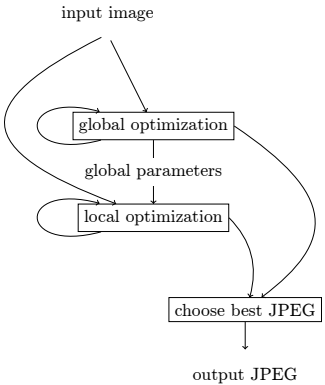


Figura 99: Otimização global: matriz de quantização. Otimização local: descarte de certos coeficientes (Alakuijala et al., 2017).

Guetzli III

Otimização global:

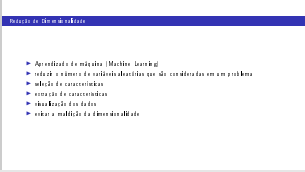
- a tabela de quantização global é uma vetor de tamanho $192 = 3 \times 64$
- realiza-se uma busca em um subconjunto de tabelas de quantização

Notebook JPEG - GNU Octave

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/jpeg.ipynb

Redução de Dimensionalidade

- Aprendizado de máquina (Machine Learning)
- reduzir o número de variáveis aleatórias que são consideradas em um problema
- seleção de características
- extração de características
- visualização dos dados
- evitar a maldição da dimensionalidade



Quando o número de dimensões cresce, o volume do espaço onde estão os dados cresce rapidamente, de forma que os dados tornam-se esparsos. Isto torna-se um problema para muitos métodos que requerem uma significância estatística. Para obter um resultado estatisticamente confiável, a quantidade de dados necessária para embasar o resultado geralmente cresce exponencialmente com o número de dimensões do problema. Além disso, organizar e buscar dados usualmente depende da detecção de áreas onde os objetos formam grupos com propriedades similares; quando os dados possuem um grande número de dimensões, todos os objetos parece esparsos e dissimilares, o que não permite que as estratégias de organização de dados sejam eficientes.

Principal Component Analysis (PCA)

A PCA é uma transformação linear que realiza um mapeamento dos dados para um espaço em que a maior parte da variância dos dados está concentrada em poucas dimensões. Desta forma é possível selecionar apenas algumas dimensões, mantendo a variância dos dados.

PCA - exemplo

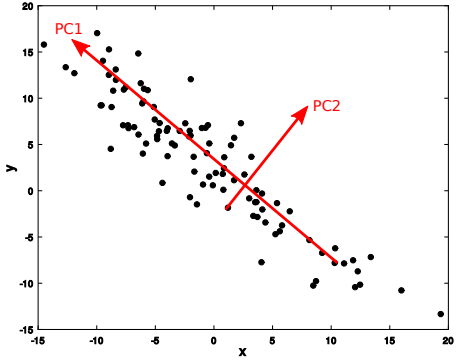


Figura 100: Exemplo PCA.

PCA - exemplo

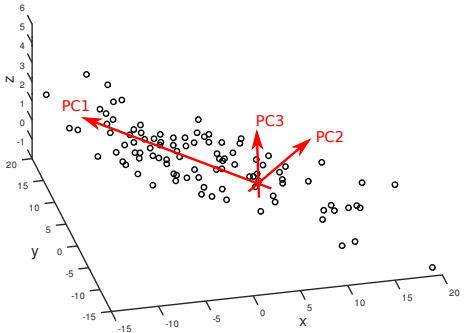


Figura 101: Exemplo PCA.

PCA - exemplo

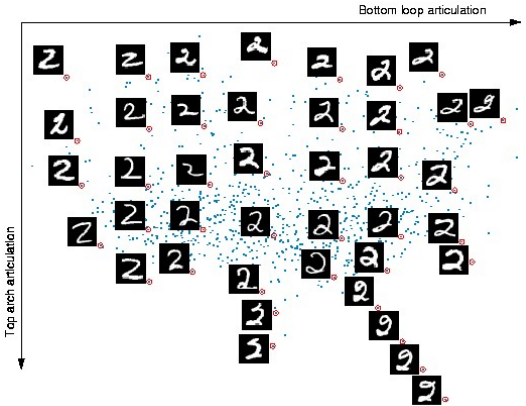


Figura 102: Fonte: http://web.mit.edu/cocosci/isomap/datasets.html

PCA - exemplo

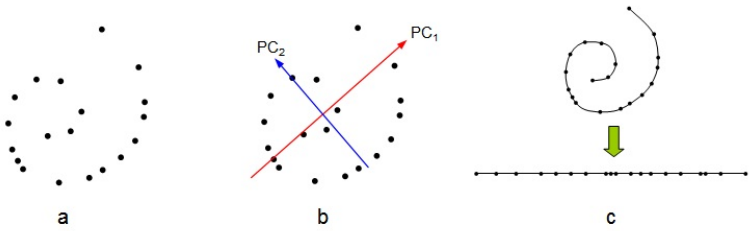


Figura 103: *kernel* PCA - aplicar uma transformação não-linear anterior à PCA.

PCA vs Regressão Linear

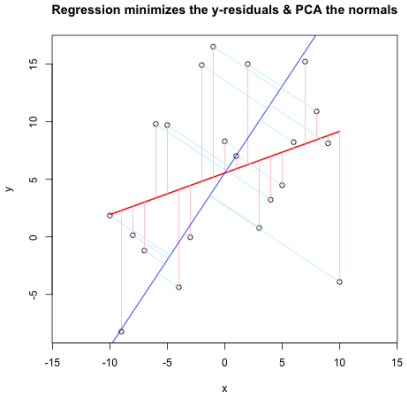


Figura 104: PCA vs Regressão Linear.
https://stackoverflow.com/questions/8457279/visual-comparison-of-regression-pca

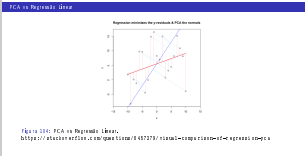


Figura 104: PCA vs Regressão Linear, https://stackoverflow.com/questions/8457270/visual-explanation-of-pca-versus-regression

Na regressão linear teremos y como uma função de x, e desta forma iremos procurar a reta que fornece a melhor predição (menor erro: $e = y - \hat{y}$).

Na PCA iremos encontrar a dimensão sob a qual teremos o menor erro ao realizar a projeção (erro $e = ||\mathbf{p_i} - \mathbf{p_i}^{\mathbf{pca}}||$).

Pré-Processamento I

Dados de treinamento: $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$.
Pré-processamento (escalonamento de cada característica / normalização da média)
Normalização da média:

- calcular a média de cada característica

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)} \tag{319}$$

- substituir cada $x_j^{(i)}$ por $x_j^{(i)} - \mu_j$.

Pré-Processamento II

Cada característica (dimensão) possui uma escala diferente. É necessário realizar uma mudança de escala para que os valores apresentem extensões similares e uma característica não domine outra.

- realizar a seguinte substituição

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j} \tag{320}$$

PCA - algoritmo

1) realizar o pré-processamento, se necessário
2) calcular a matriz de covariância dos dados
3) encontrar os autovetores da matriz de covariância (utilizar a decomposição em valores singulares)

Dataset I

Nosso conjunto de dados possui n amostras.
Cada amostra $\mathbf{x}^{(i)}$ é um vetor em um espaçado de m dimensões, $\mathbf{x}^{(i)} \in \mathbb{R}^m$.

$$\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)}]^T \tag{321}$$

Podemos representar os dados por uma matriz $m \times n$, onde cada coluna é uma amostra $\mathbf{x}^{(i)}$.

$$\mathbb{X} = \begin{bmatrix} \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \ldots & \mathbf{x}^{(n)} \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \ldots & x_1^{(n)} \\ x_2^{(1)} & x_2^{(2)} & \ldots & x_2^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ x_m^{(1)} & x_m^{(2)} & \ldots & x_m^{(n)} \end{bmatrix} \tag{322}$$

Dataset II

$$\mathbb{X} = \begin{bmatrix} \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \dots & \mathbf{x}^{(n)} \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \dots & x_{m,n} \end{bmatrix} \tag{323}$$

Elemento $x_{i,j} = x_i^{(j)}$, i-ésima característica do j-ésima amostra.

Mudança de Base I

Desejamos realizar uma mudança de base.

- Existe uma outra base, que seja uma combinação linear da base original, que melhor represente os nossos dados?

Transformação linear (rotação e estiramento): $\mathbf{P}$.
Dados originais: $\mathbf{X}$.
Após a transformação: $\mathbf{Y}$.

$$\mathbf{Y} = \mathbf{P}\mathbf{X} \tag{324}$$

Mudança de Base II

As linhas de $\mathbf{P}$ são os vetores base para a representação dos dados nas colunas de $\mathbf{X}$.

$$\mathbf{PX} = \begin{bmatrix} \mathbf{p_1} \\ \vdots \\ \mathbf{p_m} \end{bmatrix} \begin{bmatrix} \mathbf{x_1} & \dots & \mathbf{x_n} \end{bmatrix} \tag{325}$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{p_1} \cdot \mathbf{x_1} & \dots & \mathbf{p_1} \cdot \mathbf{x_n} \\ \vdots & \ddots & \vdots \\ \mathbf{p_m} \cdot \mathbf{x_1} & \dots & \mathbf{p_m} \cdot \mathbf{x_n} \end{bmatrix} \tag{326}$$

Mudança de Base III

Cada coluna de $\mathbf{Y}$ é da forma

$$\mathbf{y_i} = \begin{bmatrix} \mathbf{p_1} \cdot \mathbf{x_i} \\ \vdots \\ \mathbf{p_m} \cdot \mathbf{x_i} \end{bmatrix} \tag{327}$$

Observamos que cada coeficiente de $\mathbf{y_i}$ é o produto interno entre $\mathbf{x_i}$ e a linha correspondente de $\mathbf{P}$, ou seja, o j-ésimo coeficiente de $\mathbf{y_i}$ é a projeção de $\mathbf{x_i}$ na j-ésima linha de $\mathbf{P}$.

Variância e Objetivo I

Qual é a melhor base na qual queremos representar nossos dados? Como escolher $\mathbf{P}$?

- ruido deve ser baixo (alta relação sinal-ruido)

$$\text{SNR} = \frac{\sigma^2_{\text{sinal}}}{\sigma^2_{\text{ruido}}} \tag{328}$$

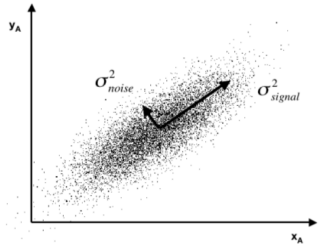


Figura 105: Relação sinal ruído (Shlens, 2014).

Variância e Objetivo II

- redundância

r_1 e r_2 são duas medições arbitrárias

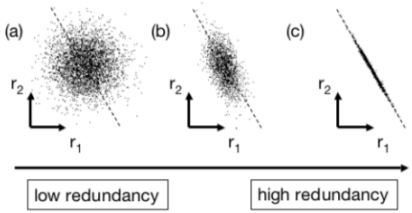


Figura 106: Redundância (Shlens, 2014).

Variância e Objetivo

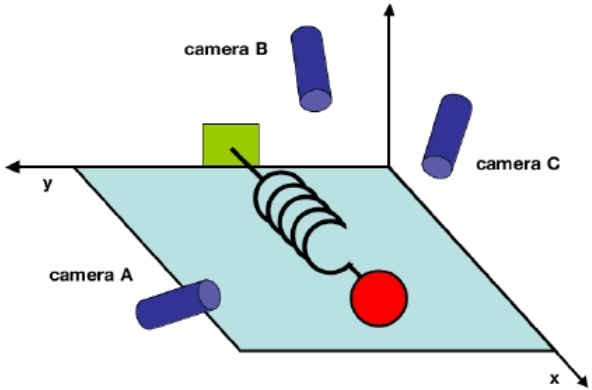


Figura 107: Exemplo hipotético (Shlens, 2014).

Neste exemplo temos medições provenientes de 3 câmeras diferentes, com orientações distintas. Cada uma belas observa a movimentação do objeto através de uma projeção em duas dimensões, Entretanto, sabemos que o objeto movimenta-se ao longo de uma única dimensão.

Matriz de Variância I

A SNR é calculada pela variância.
Considere os conjuntos de medição com média nula

$$A = \{a_1, a_2, \ldots, a_n\} \quad \text{e} \quad B = \{b_1, b_2, \ldots, b_n\} \tag{329}$$

A variância é dada

$$\sigma_A^2 = \langle a_i a_i \rangle_i \quad \text{e} \quad \sigma_B^2 = \langle b_i b_i \rangle_i \tag{330}$$

onde o valor esperado é a média sobre n variáveis.
O valor esperado $\langle \cdot \rangle_i$ é denotado pela média sobre valores indexados por i, pois a média é nula.
$\sigma_X = E[(X - \mu_X)(X - \mu_X)]$

Matriz de Variância II

A covariância entre A e B é dada por

$$\sigma_{AB}^2 = \langle a_i b_i \rangle_i \tag{331}$$

- $\sigma_{AB}^2 = 0$ se e somente se A e B são inteiramente descorrelacionados
- $\sigma_{AB}^2 = \sigma_A^2$ se $A = B$

Variância I

Representando na forma vetorial.

$$\mathbf{a} = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix} \tag{332}$$

$$\mathbf{b} = \begin{bmatrix} b_1 & b_2 & \dots & b_n \end{bmatrix} \tag{333}$$

$$\sigma^2_{\mathbf{ab}} \equiv \frac{1}{n-1}\mathbf{a}\mathbf{b}^T \tag{334}$$

onde o primeiro termo é a constante de normalização[3].

[3] A normalização mais simples é utilizando $\frac{1}{n}$, entretanto, desta forma teríamos um estimador polarizado para a variância, particularmente para n pequeno. Para obter um estimador não polarizado para a variância, devemos utilizar $\frac{1}{n-1}$.

Matriz de Covariância I

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_1} \\ \vdots \\ \mathbf{x_m} \end{bmatrix} \tag{335}$$

Cada linha de $\mathbf{X}$ representa uma amostra (conjuntos de medições/características de um tipo). A matriz de covariância é definida então

$$\mathbf{S_X} \equiv \frac{1}{n-1}\mathbf{X}\mathbf{X}^T \tag{336}$$

- $\mathbf{S_X}$ é uma matriz quadrada simétrica $m \times m$
- os valores na diagonal de $\mathbf{S_X}$ são a variância de uma medição (uma amostra)
- os valores fora da diagonal são valores de covariância entre duas medições (amostras) distintas

Diagonalização da Matriz de Covariância I

Queremos reduzir a redundância, ou seja, queremos que cada variável co-varie o mínimo possível uma com as outras.
O objetivo então é encontrar $\mathbf{Y}$ de forma que $\mathbf{S_Y}$ seja uma matriz diagonal.

- existem diversas formas de diagonalizar $\mathbf{S_Y}$
- PCA é uma dessas forma. A PCA assume que os vetores de base $\{\mathbf{p_1}, \ldots, \mathbf{p_m}\}$ são ortonormais
- a PCA utilizará uma matriz ortonormal $\mathbf{P}$

Autovetores da Covariância I

- Desejamos encontrar uma matriz ortonormal $\mathbf{P}$, onde $\mathbf{Y} = \mathbf{PX}$ esteja sujeito a $\mathbf{S_Y} \equiv \frac{1}{n-1}\mathbf{YY}^T$ ser uma matriz diagonal. As linhas de $\mathbf{P}$ são as componentes principais de $\mathbf{X}$.

$$\begin{aligned} \mathbf{S_Y} &= \frac{1}{n-1}\mathbf{YY}^T \\ &= \frac{1}{n-1}(\mathbf{PX})(\mathbf{PX})^T \\ &= \frac{1}{n-1}\mathbf{P}(\mathbf{XX}^T)\mathbf{P}^T \\ &= \frac{1}{n-1}\mathbf{PAP}^T \end{aligned} \tag{337}$$

Definimos a matriz simétrica $\mathbf{A} \equiv \mathbf{XX}^T$.

Diagonalização - teorema 1 I

Serão utilizados dois teoremas.

uma matriz é simétrica se e somente se é ortogonalmente diagonalizável

1) Se $\mathbf{A}$ é uma matriz ortogonalmente diagonalizável, então, por hipótese, exite $\mathbf{E}$ tal que $\mathbf{A} = \mathbf{EDE}^T$, onde $\mathbf{D}$ é diagonal. Desta forma, teremos

$$\mathbf{A}^T = (\mathbf{EDE}^T)^T = \mathbf{E}^{TT}\mathbf{D}^T\mathbf{E}^T = \mathbf{EDE}^T = \mathbf{A} \tag{338}$$

   e concluímos que a matriz $\mathbf{A}$ é simétrica.

2) mostrar que: Se uma matriz $\mathbf{A}$ é simétrica, então ela será ortogonalmente diagonalizável.

Diagonalização - teorema 2 I

uma matriz simétrica é diagonalizada pela matriz de seus autovetores ortonormais

Seja $\mathbf{A}$ uma matriz simétrica $n \times n$ com autovetores associados $\{\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}\}$. Seja $\mathbf{E}$ tal que cada i-ésima coluna é um autovetor $\mathbf{e_i}$ de $\mathbf{A}$, então $\mathbf{E} = \begin{bmatrix} \mathbf{e_1} & \mathbf{e_2} & \ldots & \mathbf{e_n} \end{bmatrix}$. Então existe uma matriz diagonal $\mathbf{D}$ tal que $\mathbf{A} = \mathbf{EDE}^T$.

Demonstração em duas partes:

1) qualquer matriz pode ser ortogonalmente diagonalizável se e somente se os autovetores desta matriz forem linearmente independentes
2) uma matriz simétrica possui autovetores que são ortogonais

Diagonalização - teorema 2 - parte 1 I

Seja $\mathbf{A}$ uma matriz qualquer e $\mathbf{E} = \begin{bmatrix} \mathbf{e_1} & \mathbf{e_2} & \dots & \mathbf{e_n} \end{bmatrix}$ a matriz de autovetores de $\mathbf{A}$. Vamos criar uma matriz $\mathbf{D}$ em que o i-ésimo autovalor de $\mathbf{A}$ seja colocado na posição (i, i) da matriz $\mathbf{D}$. Pela definição de autovalor e autovetor, temos

$$\mathbf{Ae_i} = \lambda_i \mathbf{e_i} \tag{339}$$

Desta forma, podemos escrever

$$\begin{aligned} \begin{bmatrix} \mathbf{Ae_1} & \mathbf{Ae_2} & \dots & \mathbf{Ae_n} \end{bmatrix} &= \begin{bmatrix} \lambda_1 \mathbf{e_1} & \lambda_2 \mathbf{e_2} & \dots & \lambda_n \mathbf{e_n} \end{bmatrix} \\ \mathbf{AE} &= \mathbf{ED} \end{aligned} \tag{340}$$

e assim teremos

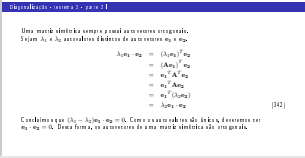
$$\mathbf{A} = \mathbf{EDE}^{-1} \tag{341}$$

Diagonalização - teorema 2 - parte 2 I

Uma matriz simétrica sempre possui autovetores ortogonais.
Sejam λ_1 e λ_2 autovalores distintos de autovetores $\mathbf{e_1}$ e $\mathbf{e_2}$.

$$\begin{aligned} \lambda_1 \mathbf{e_1} \cdot \mathbf{e_2} &= (\lambda_1 \mathbf{e_1})^T \mathbf{e_2} \\ &= (\mathbf{A}\mathbf{e_1})^T \mathbf{e_2} \\ &= \mathbf{e_1}^T \mathbf{A}^T \mathbf{e_2} \\ &= \mathbf{e_1}^T \mathbf{A} \mathbf{e_2} \\ &= \mathbf{e_1}^T (\lambda_2 \mathbf{e_2}) \\ &= \lambda_2 \mathbf{e_1} \cdot \mathbf{e_2} \end{aligned} \tag{342}$$

Concluímos que $(\lambda_1 - \lambda_2)\mathbf{e_1} \cdot \mathbf{e_2} = 0$. Como os autovalores são únicos, deveremos ter $\mathbf{e_1} \cdot \mathbf{e_2} = 0$. Desta forma, os autovetores de uma matriz simétrica são ortogonais.


Diagonalização - teorema 2 - parte 2

Uma matriz simétrica sempre possui autovetores ortogonais.
Sejam $\lambda_1 \neq \lambda_2$ autovalores distintos de uma matriz A e e_1 e e_2 seus autovetores.

$$\begin{aligned} \lambda_1 e_1 \cdot e_2 &= (\lambda_1 e_1)^T e_2 \\ &= (Ae_1)^T e_2 \\ &= e_1^T A^T e_2 \\ &= e_1^T A e_2 \\ &= e_1^T (\lambda_2 e_2) \\ &= \lambda_2 e_1 \cdot e_2 \end{aligned} \quad [1.2]$$

Concluímos que $(\lambda_1 - \lambda_2) e_1 \cdot e_2 = 0$. Como os autovalores são iguais, devemos ter $e_1 \cdot e_2 = 0$. Dessa forma, os autovetores de uma matriz simétrica são ortogonais.

Um autovetor multiplicado por um escalar será também autovetor. Logo, por definição iremos escolher autovetores com norma unitária, assim podemos afirmar que os autovetores de uma matriz simétrica são ortonormais.

Diagonalização - teorema 2 I

Como $\mathbf{E}$ é uma matriz ortogonal, teremos $\mathbf{E}^T = \mathbf{E}^{-1}$ e, desta forma,

$$\mathbf{A} = \mathbf{E}\mathbf{D}\mathbf{E}^T \tag{343}$$

Uma matriz simétrica é diagonalizável por uma matriz de seus autovetores.

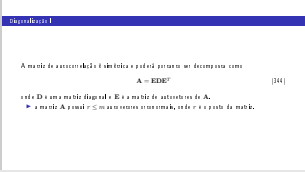
Diagonalização I

A matriz de autocorrelação é simétrica e poderá portanto ser decomposta como

$$\mathbf{A} = \mathbf{EDE}^T \tag{344}$$

onde $\mathbf{D}$ é uma matriz diagonal e $\mathbf{E}$ é a matriz de autovetores de $\mathbf{A}$.

- a matriz $\mathbf{A}$ possui $r \leq m$ autovetores ortonormais, onde r é o posto da matriz.



O posto de $\mathbf{A}$ é menor do que m quando $\mathbf{A}$ é degenerada, ou quando todos os dados ocupam um subespaço de dimensão $r \leq m$. Para remediar este empecilho, podemos adicionar outros $(m - r)$ vetores ortonormais para preencher a matriz $\mathbf{E}$. Estes vetores não terão efeito na solução final pois a variância associada a eles será nula.

O posto de uma matriz $\mathbf{A}$ é o número de linhas ou colunas linearmente independentes de $\mathbf{A}$, ou equivalentemente, é o número de linhas não-nulas quando a mesma está escrita na forma reduzida escalonada por linhas.

Diagonalização I

Vamos selecionar a matriz $\mathbf{P}$ de forma que cada linha $\mathbf{p_i}$ seja um autovetor de $\mathbf{X}\mathbf{X}^T$, ou seja, $\mathbf{P} \equiv \mathbf{E}^T$. Teremos então

$$\begin{aligned}
\mathbf{S_Y} &= \frac{1}{n-1}\mathbf{P}\mathbf{A}\mathbf{P}^T \\
&= \frac{1}{n-1}\mathbf{P}(\mathbf{P}^T\mathbf{D}\mathbf{P})\mathbf{P}^T \\
&= \frac{1}{n-1}(\mathbf{P}\mathbf{P}^T)\mathbf{D}(\mathbf{P}\mathbf{P}^T) \\
&= \frac{1}{n-1}(\mathbf{P}\mathbf{P}^{-1})\mathbf{D}(\mathbf{P}\mathbf{P}^{-1}) \\
&= \frac{1}{n-1}\mathbf{D} \qquad (345)
\end{aligned}$$

Fica assim evidente que a escolha feita para $\mathbf{P}$ diagonaliza a matriz $\mathbf{S_Y}$, o que é o objetivo da PCA.

Resultado da PCA I

Como resultado da PCA teremos

- as componentes principais de $\mathbf{X}$ são autovetores de $\mathbf{X}\mathbf{X}^T$; que serão linhas de $\mathbf{P}$.
- o i-ésimo valor na diagonal de $\mathbf{S_Y}$ representa a variância de $\mathbf{X}$ ao longo da direção de $\mathbf{p_i}$.

Notebook - PCA

https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/pca.ipynb

Ahmed, N., Natarajan, T., and Rao, K. (1974). Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93.

Alakuijala, J., Obryk, R., Stoliarchuk, O., Szabadka, Z., Vandevenne, L., and Wassenberg, J. (2017). Guetzli: Perceptually guided jpeg encoder.

Araújo, L., Sansão, J., and Fasolo, S. (2018). Quantização vetorial de cores em imagens digitais. In *Anais de XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*. Sociedade Brasileira de Telecomunicações.

Chen, W.-H., Smith, C., and Fralick, S. (1977). A fast computational algorithm for the discrete cosine transform. *IEEE Transactions on Communications*, 25(9):1004–1009.

Gallager, R. G. (2008). *Principles of Digital Communication*. Cambridge University Press.

Gray, H. (1918). *Anatomy of the Human Body*. Lea and Febiger.

Kerr, D. A. (2012). Chrominance subsampling in digital images. [Online; posted January 19, 2012].

Kondoz, A. M. (2004). *Digital Speech: Coding for Low Bit Rate Communication Systems*. Wiley.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580.

Ogundunmi, T. and Narasimha, M. (2010). *Principles of Speech Coding*. CRC Press.

Oppenheim, A. V. (2009). *Discrete-Time Signal Processing*. Pearson.

Salomon, D. (2000). *Data Compression: The Complete Reference*. Springer New York.

Salomon, D., Bryant, D., and Motta, G. (2010). *Handbook of Data Compression*. Springer London.

Shlens, J. (2014). A tutorial on principal component analysis.

Yehia, H. C. (1993). Análise de funções de erro em sistemas de codificação lpc. Master's thesis, ITA.