

Processamento de Áudio e Vídeo

Prof. Leonardo Araújo



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

1 Introdução

2 Técnicas Básicas de Compressão

- RLE

3 Quantização Escalar

- Conversão AD/DA
- Quantização Escalar
- Entropia na saída do quantizador
- Quantização Escalar Uniforme
- Quantização Escalar Não-Uniforme
- Lloyd-Max
- Performance do Quantizador
- Quantizador Uniforme
- Quantizador Não Uniforme
- Compressor e Expansor
- Quantização Vetorial
- Processamento digital de sinais analógicos
- Conversão discreto-contínuo
- Oversampling e Noise-Shaping

4 Mudança de frequência de amostragem

- Downsampling
- Upsampling
 - Interpolador Linear

- Mudança da frequência de amostragem por um fator não inteiro
- Processamento Multi-taxa de Sinais
- Decimação e Interpolação com múltiplos estágios
- Decomposição Polifásica
- Implementação de decimadores usando decomposição polifásica
- Implementação de interpoladores usando decomposição polifásica
- Banco de Filtros Multi taxa

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

- Compressão
- Com/Sem perdas
- mp3, jpeg, mpeg, flac, etc, gif, png, etc
- dados de áudio, íd3, imagens e vídeo
- qualidade, taxa de compressão, etc

- 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

- Compressão
- Com/Sem perdas
- mp3, jpeg, mpeg, flac, dv, gH, png, etc
- Áudio de áudio, ía b, imagens e vídeo
- Qualidade, taxa de compressão, etc

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

└─ Introdução

└─ Processamento de Áudio e Vídeo

- Compressão
- Com/Sem perdas
- mp3, jpeg, mpeg, flac, dv, gif, png, etc
- sinais de áudio, ímagem e vídeo
- qualidade, taxa de compressão, tempo

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

- Compressão
- Com/Sem perdas
- mp3, jpeg, mpeg, flac, dv, gif, png, etc
- Áudio de áudio, ímagem e vídeo
- Qualidade, custo de compressão, etc

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

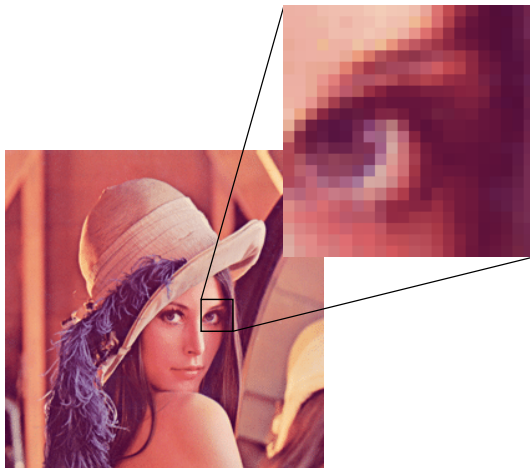


Figura 1: Lena - detalhe.

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?

Processamento de Áudio e Vídeo

└─ Introdução

└─ Dilema de compressão

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.

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

resultos:

- ▶ menor tamanho da representação digital (em bytes)
- ▶ eficiência computacional (compressão e/ou decompressão)
- ▶ simplicidade de algoritmos

Qual é o limite de compressão para um determinado dado?

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

resultat:

- ▶ menor tamanho da representação digital resultante
- ▶ eficiência computacional (compressão e/ou decompressão)
- ▶ simplicidade de algoritmos

Qual é o **limite** de compressão para um determinado dado?

Conjecturas¹.

- 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.

Compressão perceptiva ocorre quando apenas a informação imperceptível pelos nossos 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 Compressão = 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

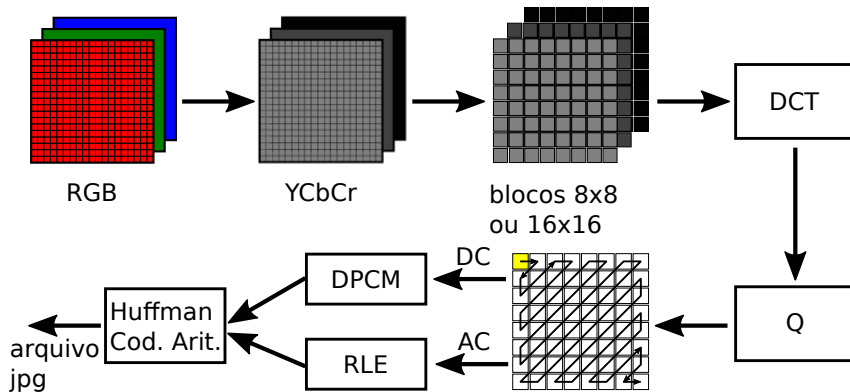


Figura 2: Esquema de compressão JPEG.

Slides- introdução ao GNU Octave



https://drive.google.com/open?id=1ew5fl9v_0Iybsy3KdEgIvohLTcuwuru_

Notebook - introdução



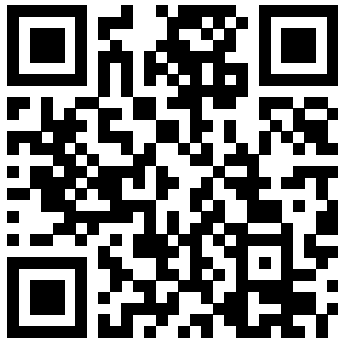
`https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/
introducao.ipynb`

Notebook - imagem colorida



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

Processamento de Áudio e Vídeo

└ Tecnicas Básicas de Compressão

└ RLE

└ Compressão RLE

Exemplo:
original: "2, a016 m 0 2s m017"
codificação: "2, a016 016 m017"

Método MNP 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: **abcd dcbam nopp onm**

$C = (0, 1, 2, 3, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3)$

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

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
\vdots	\vdots	\vdots
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

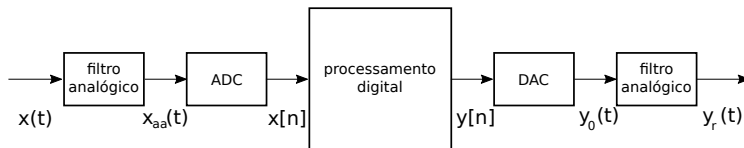


Figura 4: Processamento digital de sinais. Conversão AD e DA.

Quantização

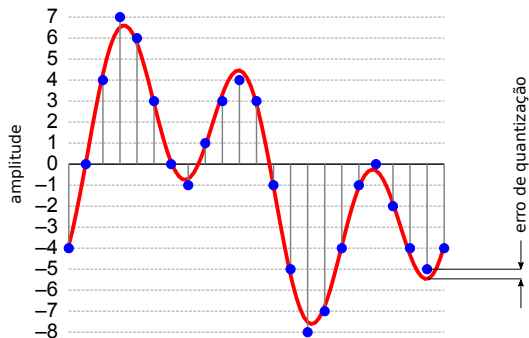


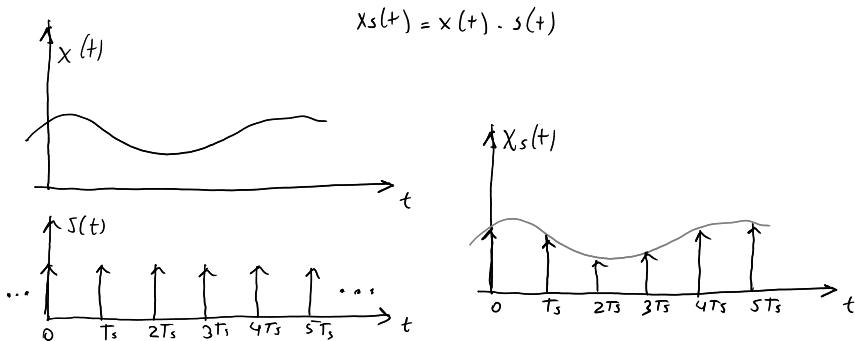
Figura 5: Quantização.

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



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:

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

onde

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

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

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

Amostragem II

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

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

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

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

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

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

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 $x(t)$:

$$X_r(j\Omega) = H_r(j\Omega)X_s(j\Omega) \quad (8)$$

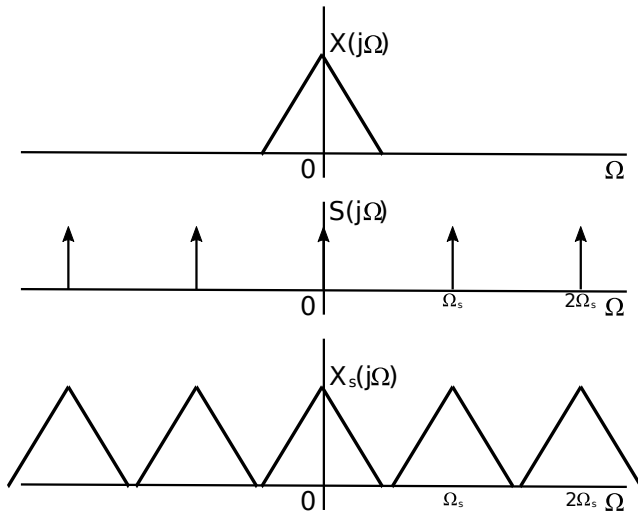
onde H_r é um filtro passa-baixas ideal com $\Omega_N < \Omega_c < \Omega_s$.

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

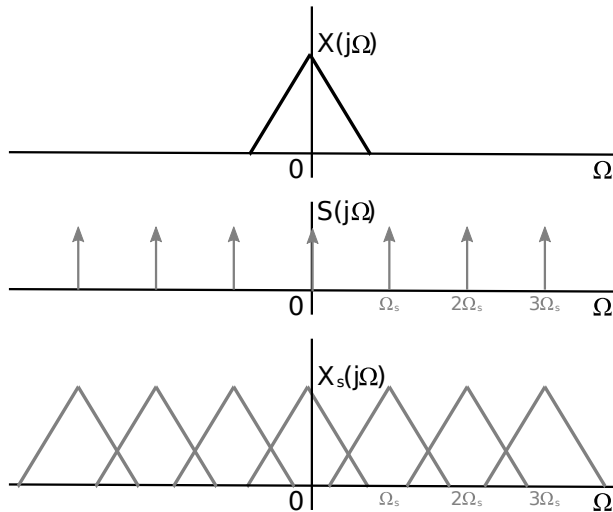
(Teorema da Amostragem)

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 $Y = y_1, y_2, \dots, y_M$. Os valores y_i , $i = 1, 2, \dots, M$, 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 $\mathcal{T} = \{t_i\}$, $i = 0, 1, \dots, M$ e pelo conjunto de pontos de representação $\mathcal{Y} = \{y_i\}$, $i = 1, \dots, M$. 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

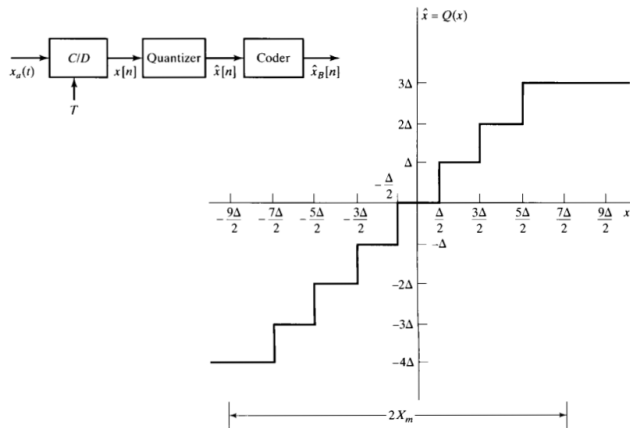


Figura 6: Quantização. Fonte: Oppenheim (2009).

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

$$D(Q) = E \{d(x, Q(x))\}, \quad (10)$$

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

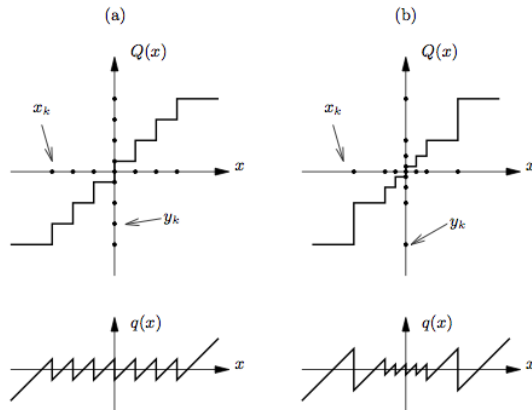


Figura 7: (a) linear (b) logarítmico. Fonte: Oppenheim (2009).

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

$$P(y_i) = \int_{t_{i-1}}^{t_i} f(x) dx. \quad (11)$$

A entropia da saída do quantizador é igual a

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

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

Processamento de Áudio e Vídeo

└─ Quantização Escalar

└─ Entropia na saída do quantizador

└─ Entropia na saída do quantizador

As probabilidades da saída 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 de obtenção da saída de representação (a probabilidade de $x \in \Delta_k$) como

$$P(y_k) = \int_{\Delta_k} f(x) dx. \quad [11]$$

A entropia da saída de quantizador é igual a

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

Um código de comprimento variável pode ser utilizado para representar a saída de 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,

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

Quantização escalar uniforme II

(*obs.: considerando apenas valores positivos)

Dada a entrada x , a célula associada a x é determinada por

$$i = [x/\delta], \quad (14)$$

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

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

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

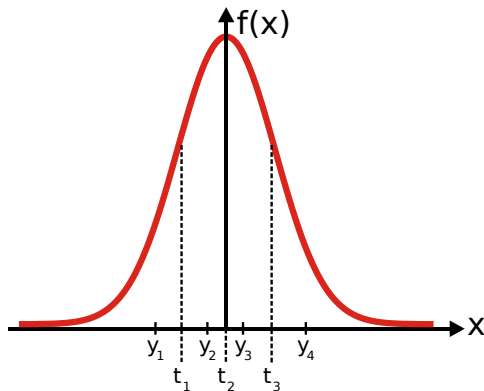
$$D(Q) = \int_{-\infty}^{\infty} f(x) d(x, Q(x)) dx = \sum_i \int_{t_{i-1}}^{t_i} f(x) d(x, y_i) dx. \quad (16)$$

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

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

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:

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

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

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

Algoritmo de Lloyd-Max II

Algoritmo:

- 1) Escolher um conjunto inicial arbitrário com M pontos de representação $y_1 < y_2 < \dots y_M$.
- 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.

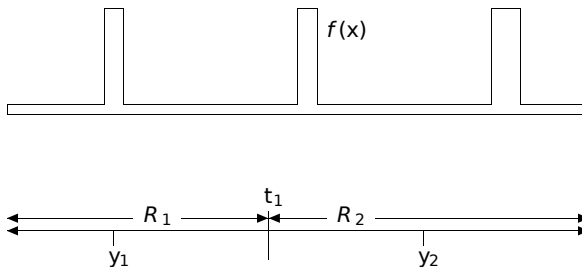


Figura 8: Exemplo Lloyd-Max: regiões e pontos de representação que satisfazem a condição de parada do algoritmo nas não minimizam a distorção média quadrática. Fonte: Gallager (2008)

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

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

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

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

e assim, a equação 20 poderá ser reescrita como

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

Performance do Quantizador II

Mostra-se que

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

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

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

Performance do Quantizador III

e assim podemos reescrever a equação 22 como

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

Performance do Quantizador Uniforme I

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

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

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),

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

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

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

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

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

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.

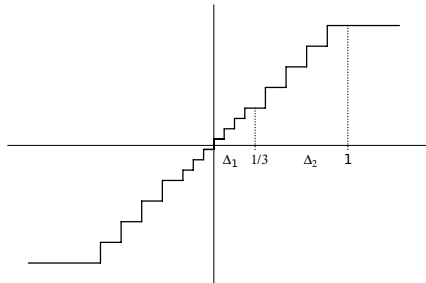


Figura 9: Exemplo de quantizador não uniforme de 4 bits, com $\Delta_1 = \Delta_2/2$ (Ogundunmi and Narasimha, 2010).

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.

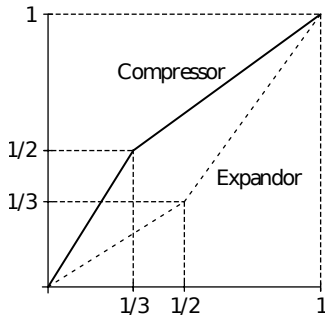


Figura 10: Exemplo de compressor e expansor (Ogundunmi and Narasimha, 2010).

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

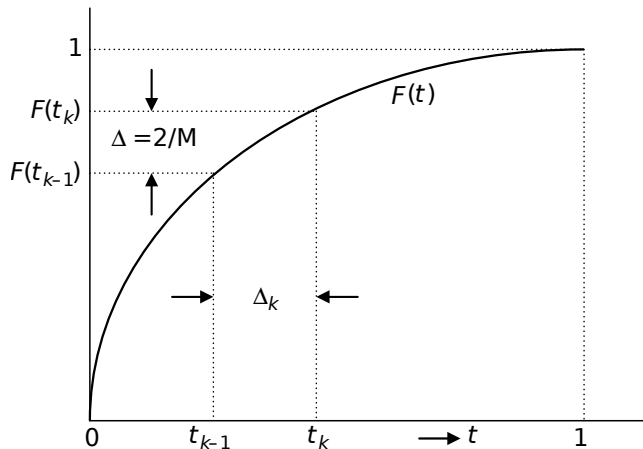


Figura 11: Exemplo de compressor (Ogundunmi and Narasimha, 2010).

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 ,

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

Substituindo Δ_k na Equação 25, teremos

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

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

$$\sigma_q^2 = \frac{1}{3M^2} \int_{-1}^{+1} \frac{p(x)}{(F'(x))^2} dx = \frac{2}{3M^2} \int_0^{+1} \frac{p(x)}{(F'(x))^2} dx, \quad (32)$$

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

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

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.

$$\text{SNR} = \frac{\sigma_x^2}{\sigma_q^2} = \frac{2 \int_0^1 x^2 p(x) dx}{\frac{2}{3M^2} \int_0^1 \frac{p(x)}{(F'(x))^2} dx}. \quad (34)$$

A expressão em 34 pode ser feita constante escolhendo

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

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

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

Compressão Logarítmica II

Para este caso a SNR obtida será

$$\text{SNR} = \frac{3M^2}{k^2}. \quad (37)$$

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

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

Compressão Logarítmica III

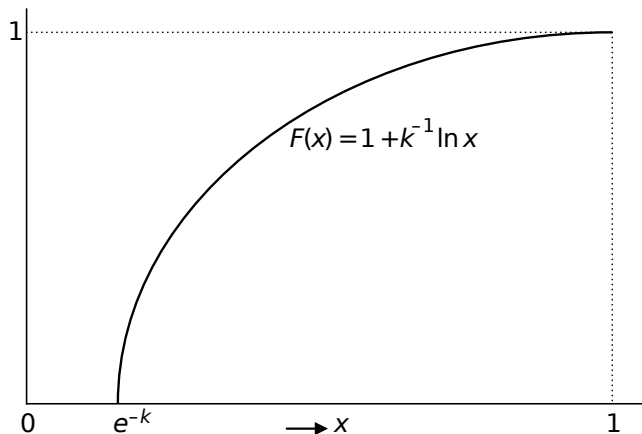


Figura 12: Gráfico de $F(x) = 1 + k^{-1} \ln x$ (Ogundunmi and Narasimha, 2010).

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

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

Para valores próximo da origem, esta proporcionalidade não poderá ser mantida, pois $\ln x$ diverge quando $x \rightarrow 0$.

Uma lei de compressão prática não pode ter tal descontinuidade, devendo também especificar a compressão dada a sinais de baixa amplitude.

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

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

onde a base do logaritmo é irrelevante.

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

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

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

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

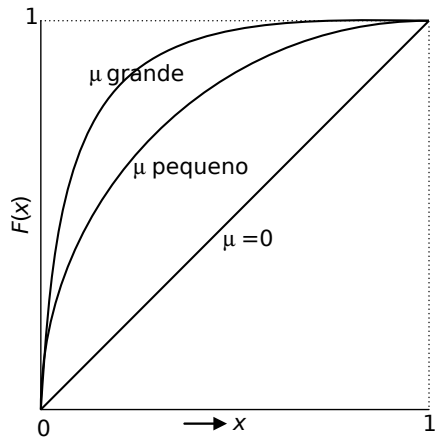
lei μ III

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

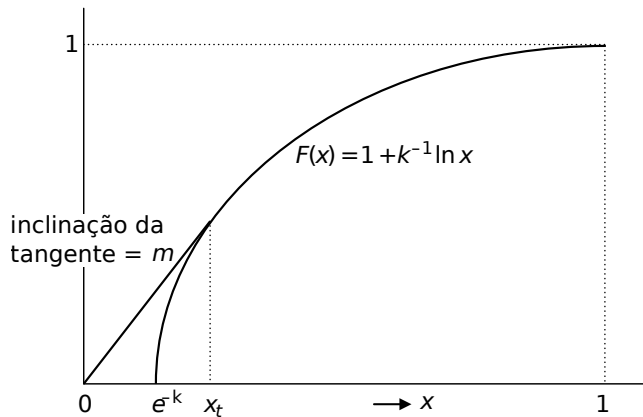


Figura 14: Curva de compressores segundo a lei A (Ogundunmi and Narasimha, 2010).

$$k = 1 + \ln A. \quad (43)$$

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

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 original (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 $\mathbf{x} = (x_1, x_2, \dots, x_n) \in R^n$.

Quantização Vetorial II

Uma quantização vetorial é um mapeamento Q de vetores de entrada $\mathbf{x} = (x_1, x_2, \dots, x_n)$, onde $\mathbf{x} \in \mathcal{X}^n$, nos valores $\mathbf{y} = (y_1, y_2, \dots, y_n) \in \mathcal{X}^n$ mais próximos (com relação a alguma medida de distorção), onde $\mathbf{y} \in \mathcal{Y} \subseteq \mathcal{X}^n$, sendo $\mathcal{Y} = \{y_1, y_2, \dots, y_M\}$, um subconjunto constituído por M elementos em \mathcal{X}^n . O parâmetro n indica a dimensionalidade dos dados e do quantizador. O conjunto 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 , $i = 1, 2, \dots, M$, 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

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

Utilizando a distância Euclideana normalizada como métrica

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

teremos

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

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

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

e a probabilidade do vetor de representação \mathbf{y}_i é dada por

$$P(\mathbf{y}_i) = \int_{S_i} f(\mathbf{x}) d\mathbf{x}. \quad (49)$$

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

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

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

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

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*).

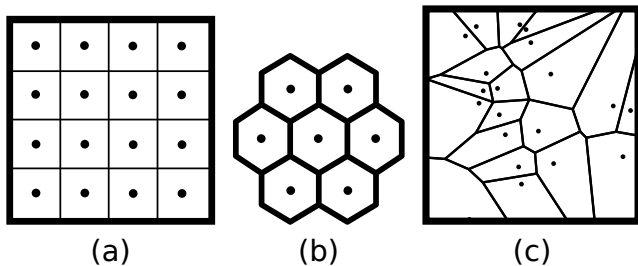


Figura 15: Diagramas de Voronoi. Estruturas em treliça em (a) e (b).

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

Processamento digital de sinais analógicos

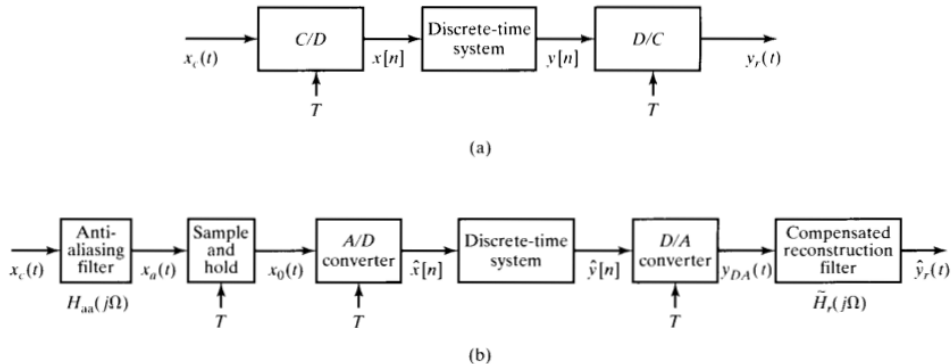


Figura 16: Processamento digital de sinais analógicos (Oppenheim, 2009).

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

Processamento de Áudio e Vídeo

└ Quantização Escalar

└ Processamento digital de sinais analógicos

└ Processamento digital de sinais analógicos

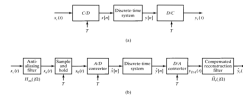


Figura 3.6: Processamento digital de sinais analógicos (Oppenheim, 2003).

Oppenheim, A. V. (2003). *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).

Processamento de Áudio e Vídeo

- Quantização Escalar
- Processamento digital de sinais analógicos
- Processamento digital de sinais analógicos

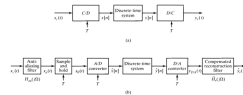


Figura 10: Processamento digital de sinais analógicos (Oppenheim, 2008).

Oppenheim, A. V. (2008). *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:

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

Processamento digital de sinais analógicos I

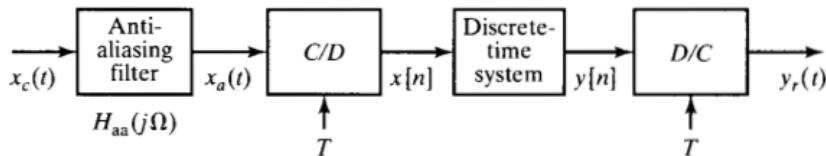


Figura 17: Processamento digital de sinais analógicos (Oppenheim, 2009).

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

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

Processamento digital de sinais analógicos II

ou seja,

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

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 de Áudio e Vídeo

└─ Quantização Escalar

└─ Processamento digital de sinais analógicos

└─ Processamento digital de sinais analógicos

ou seja,

$$H_{eq}(j\Omega) \approx \begin{cases} H(e^{j\Omega T}) & , |\Omega| < \Omega_c \\ 0 & , |\Omega| > \Omega_c \end{cases} \quad [54]$$

Na prática, ocorre uma aproximação a esta pela resposta em frequência de $H_{eq}(j\Omega)$ e da ideia linear ou limitada em frequência, mas podemos fazer $H_{eq}(j\Omega)$ pequena para $|\Omega| > \pi/T$, em vez de atingir a algarizagem.

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

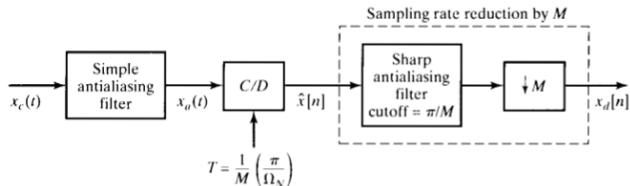


Figura 18: Utilizando uma conversão A/D com sobre-amostragem para simplificar o filtro analógico *antialiasing* (Oppenheim, 2009).

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

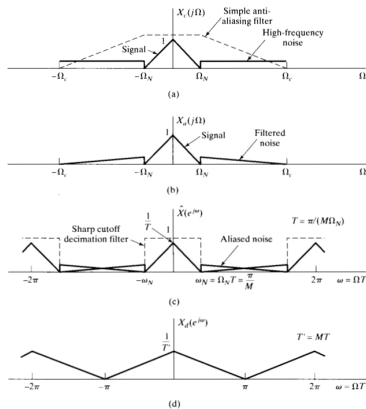
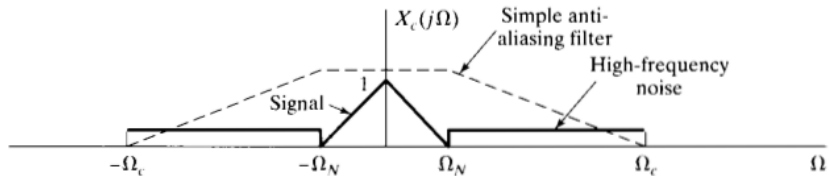
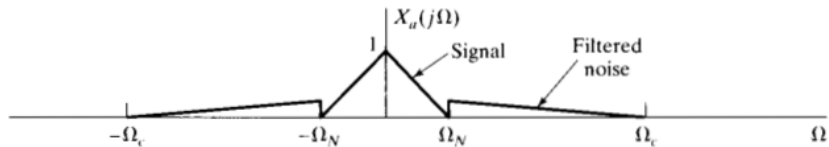


Figura 19: Exemplo (Oppenheim, 2009)

Sobre-amostragem e decimação II

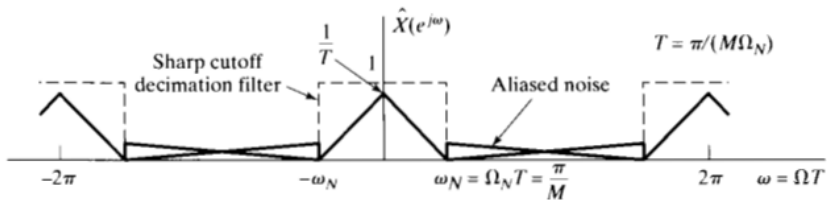


(a)

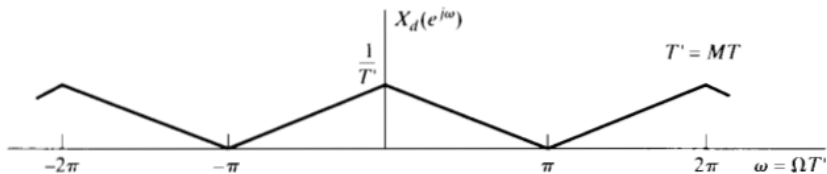


(b)

Sobre-amostragem e decimação III



(c)



(d)

Configuração física para conversão analógico-digital

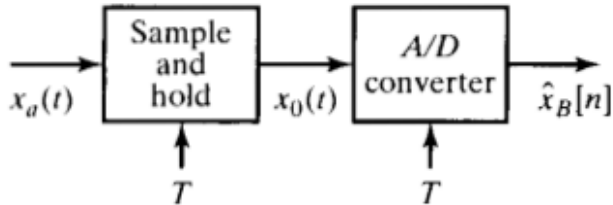


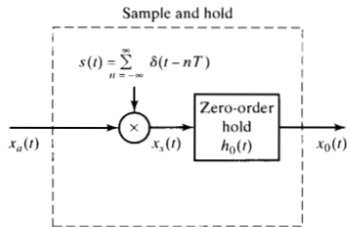
Figura 20: Configuração física para conversão analógico-digital (Oppenheim, 2009).

$x_a(t)$: sinal analógico

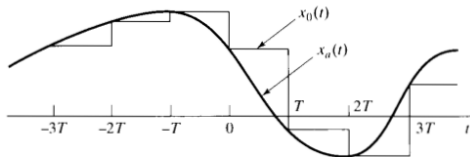
$\hat{x}_B[n]$: sinal digital (amostrado e quantizado)

T : período de amostragem

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

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

(a)



(b)

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

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

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 ideais (não quantizadas) de $x_a(t)$ e $h_0(t)$ é a resposta ao impulso do *hold* de ordem zero, i.e.,

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

A Equação 55 é equivalente a

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

Processamento de Áudio e Vídeo

└ Quantização Escalar

└ Processamento digital de sinais analógicos

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

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.

Representação de um *sample-and-hold* ideal IIIA 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 ideais (isto quer dizer que $x_a(t)$ e $h_0(t)$ é a resposta ao impulso de hold de ordem zero, i.e.,

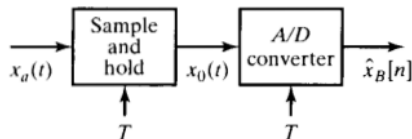
$$h_0(t) = \begin{cases} 1 & , 0 \leq t < T, \\ 0 & , \text{caso contrário.} \end{cases} \quad [56]$$

A Equação 55 é equivalente a

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

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

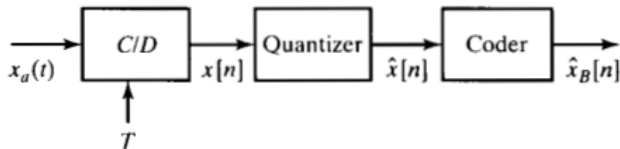


Figura 22: Sistema equivalente (Oppenheim, 2009).

Quantizador Uniforme I

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

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

Quantizador Uniforme II

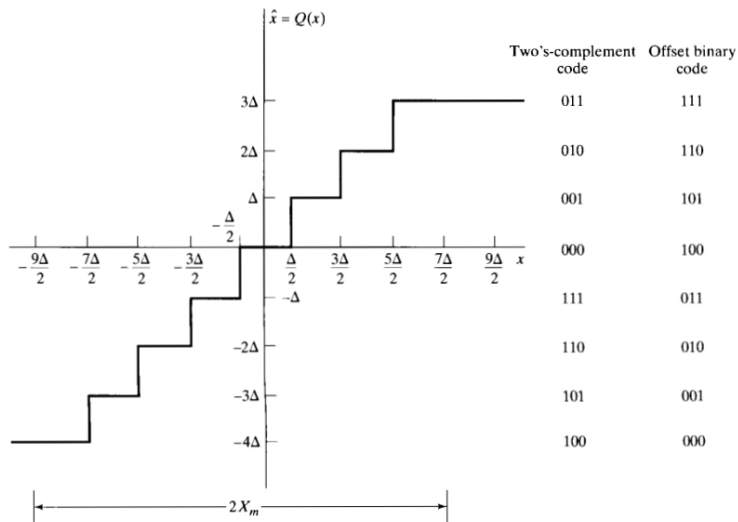


Figura 23: Quantizador uniforme de 3 bits (Oppenheim, 2009).

Erro de quantização I

O erro de quantização é definido por

$$e[n] = \hat{x}[n] - x[n] . \quad (62)$$

O erro de quantização satisfaz

$$\Delta/2 < e[n] \leq \Delta/2 \quad (63)$$

sempre que

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

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

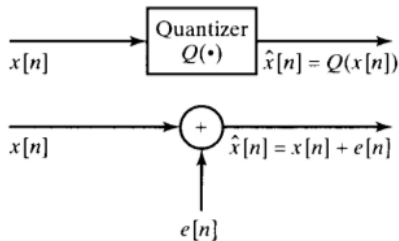


Figura 25: Modelo aditivo do erro de quantização (Oppenheim, 2009).

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 $[-\Delta/2, \Delta/2]$.

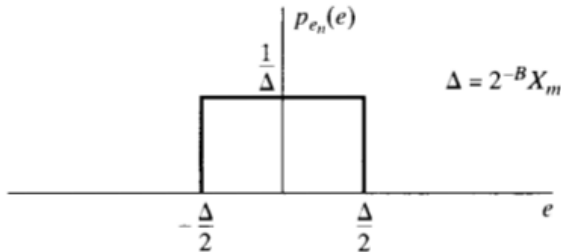


Figura 26: Modelo do ruído (Oppenheim, 2009).

média : $\mu_e = 0$;

Modelo aditivo do erro de quantização III

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

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

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

$$\sigma_e^2 = \frac{2^{-2B} X_m^2}{12} , \quad (66)$$

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-continuo I

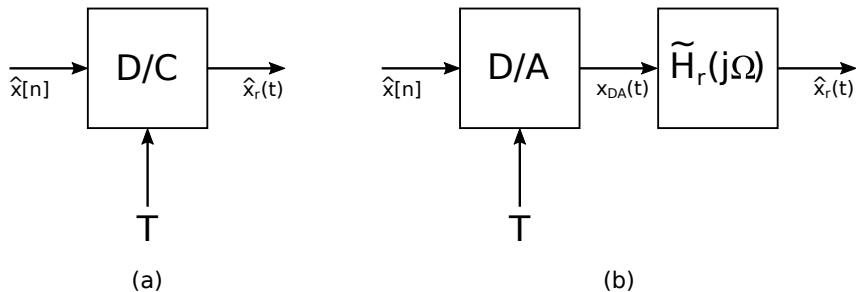


Figura 27: Conversão discreto para contínuo.

Conversão discreto-continuo II

A reconstrução é representada por

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

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

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

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

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

Conversão discreto-continuo III

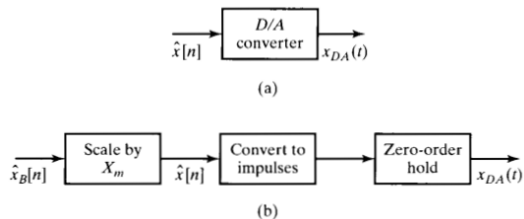


Figura 28: Modelo do conversor D/A (Oppenheim, 2009).

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

Conversão discreto-continuo IV

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

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

Definimos

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

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

de forma que

$$x_{\text{DA}}(t) = x_0(t) + e_0(t) . \quad (75)$$

Conversão discreto-continuo V

A transformada de Fourier da Equação 73 é

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

Como

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

segue que

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

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

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

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

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

Desta forma, o filtro de reconstrução compensado é dado por

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

Filtro de Reconstrução II

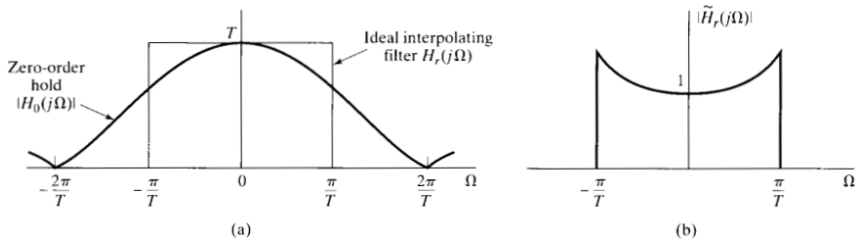


Figura 29: Filtro de reconstrução compensando o efeito do hold (Oppenheim, 2009).

Configuração física da conversão digital analógico I

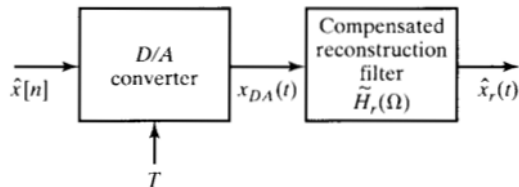


Figura 30: Configuração física para a conversão digital-analógico (Oppenheim, 2009).

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

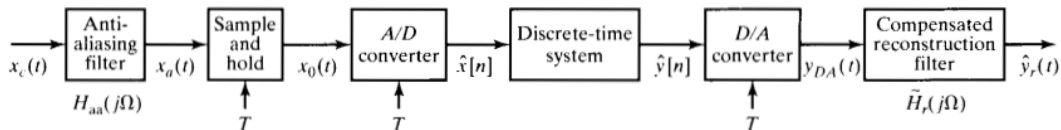


Figura 31: Sistema para processamento digital de sinais analógicos (Oppenheim, 2009).

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

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

onde

- ▶ $H_{aa}(j\Omega)$ 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

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

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

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

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 (SQNR²). Mantendo número de bits do quantizador, é possível reduzir a SQNR.

²Signal-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

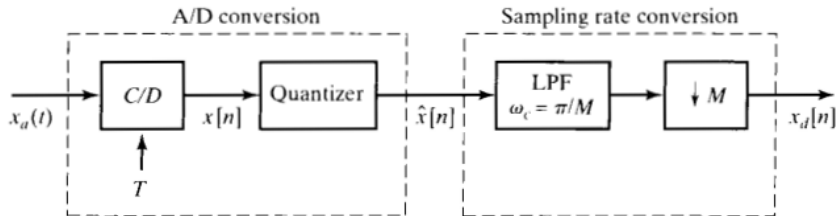


Figura 32: Conversão A/D com sobre-amostragem (Oppenheim, 2009).

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

Utilizando o modelo do ruído aditivo.

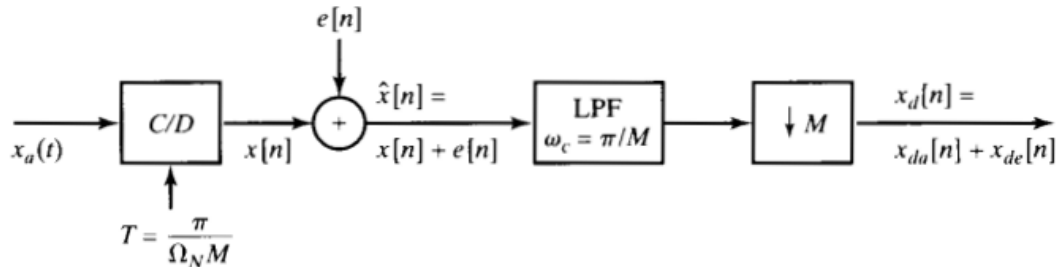


Figura 33: Modelo do ruído aditivo na conversão A/D com sobre-amostragem (Oppenheim, 2009).

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

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

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

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

Assim

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

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

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

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

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

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)

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

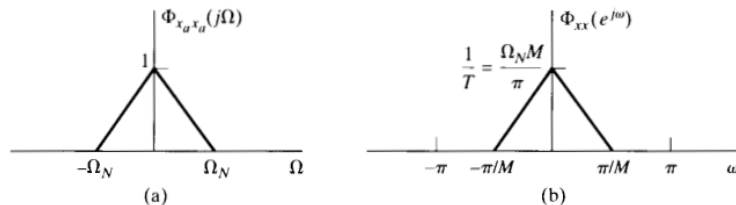


Figura 34: Densidade espectral de potência (Oppenheim, 2009).

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

A potência total do sinal analógico é dada por

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

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

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

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 $|\omega| < \pi/M$,

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

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

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

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

$$\sigma_e^2 = \frac{\Delta^2}{12}. \quad (98)$$

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

$$\phi_{ee}[m] = \sigma_e^2 \delta[m] \quad (99)$$

e

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

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

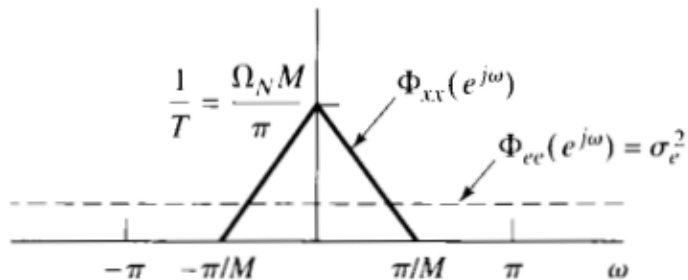


Figura 35: Densidade espectral de potência do sinal e do ruído (Oppenheim, 2009).

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

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

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

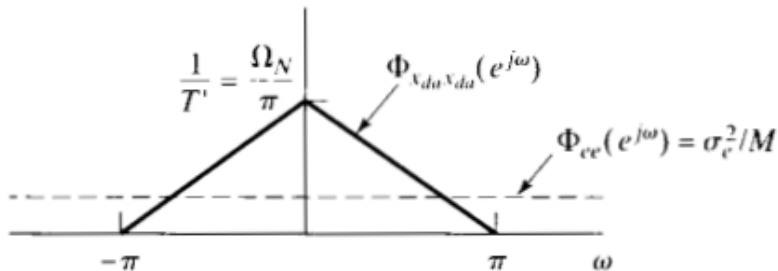


Figura 36: Densidades espectrais de potência do sinal e ruído, após a decimação (Oppenheim, 2009).

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

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

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 .

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

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 $P_{de} = \varepsilon\{x_{de}^2[n]\}$,

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

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 técnica 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 $|\omega| < \pi/M$.

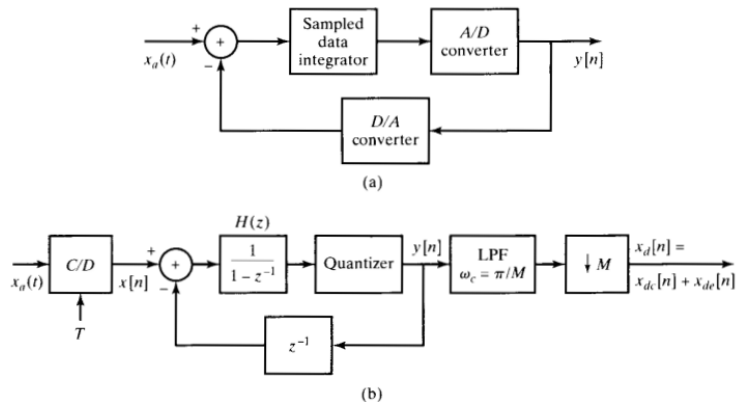
Conversão A/D com *Noise Shaping* I

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

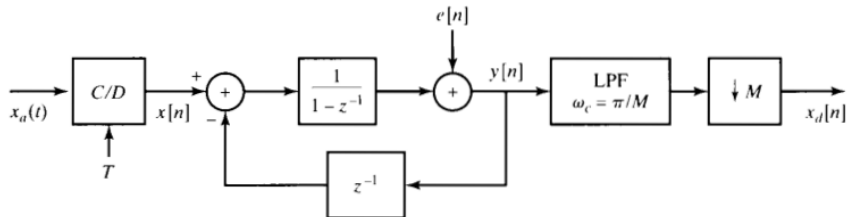
Conversão A/D com *Noise Shaping* II

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

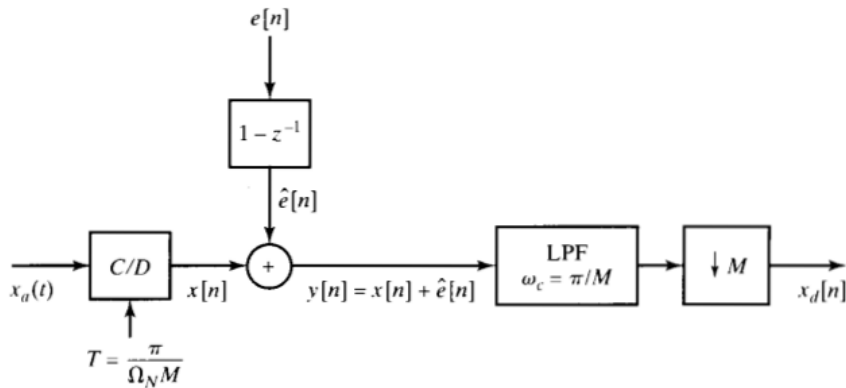
Conversão A/D com *Noise Shaping* III

Figura 39: Simplificação (Oppenheim, 2009).

Conversão A/D com *Noise Shaping* IV

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

$$H_x(z) = 1 \quad (105)$$

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

Consequently

$$y_x[n] = x[n] \quad (107)$$

$$\hat{e}[n] = e[n] - e[n-1] \quad (108)$$

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

$$y[n] = x[n] + \hat{e}[n] \quad (109)$$

Conversão A/D com *Noise Shaping* V

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

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

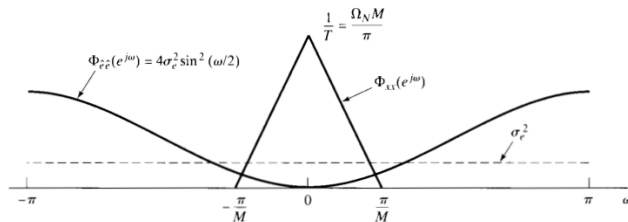


Figura 40: Densidade espectral de potência do ruído de quantização com *noise shaping* (Oppenheim, 2009).

Conversão A/D com *Noise Shaping* VI

Densidade espectral de potência após do *downsampling*.

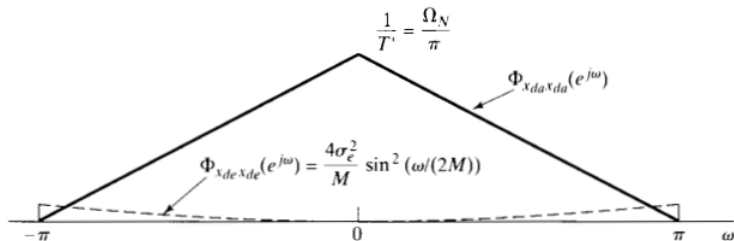


Figura 41: Densidade espectral de potência final (Oppenheim, 2009).

Conversão A/D com *Noise Shaping* VII

A potência do sinal $x_{da}[n]$ é

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

A potência do ruído de quantização final na saída é

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

Podemos assumir que M é suficientemente grande e assim

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

Assim, poderemos considerar

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

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

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

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).

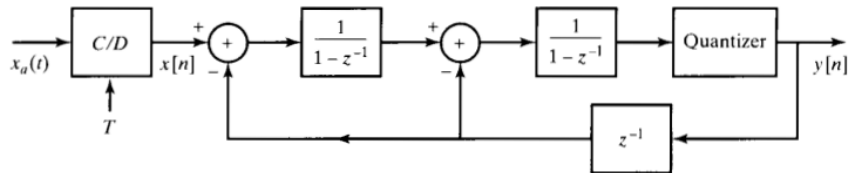
Conversão A/D com *Noise Shaping* X

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

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

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

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

De forma geral, para p estágios, teremos

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

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.

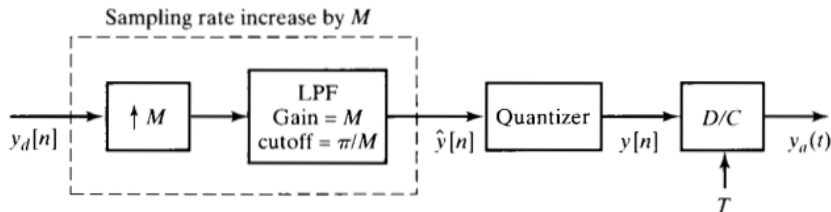


Figura 45: Utilização do *oversampling* com um conversor D/A simples (Oppenheim, 2009).

Oversampling e Noise Shaping na Conversão D/A II

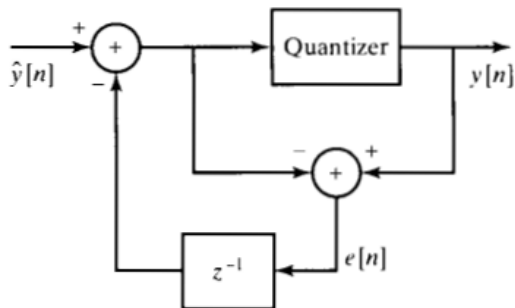


Figura 46: Sistema de *noise shaping* de primeira ordem (Oppenheim, 2009).

Oversampling e Noise Shaping na Conversão D/A III

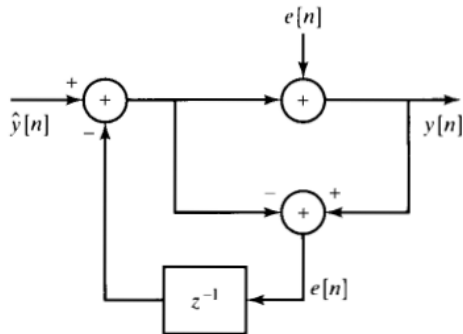


Figura 47: Modelo de ruído aditivo para o quantizador (Oppenheim, 2009).

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

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

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

Oversampling e Noise Shaping na Conversão D/A V

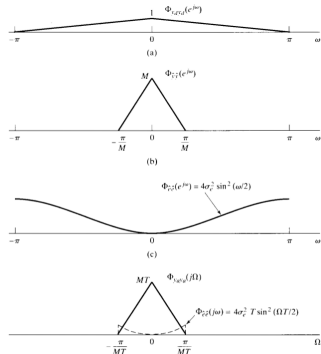


Figura 48: Densidade espectral de potência do sinal e ruído de quantização, na saída (Oppenheim, 2009).

Oversampling e Noise Shaping na Conversão D/A VI

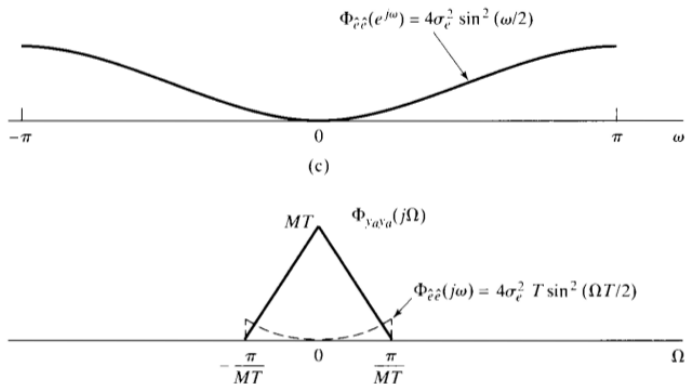


Figura 49: Detalhe: densidade espectral de potência do ruído de quantização, na saída (Oppenheim, 2009).

Oversampling e Noise Shaping na Conversão D/A VII

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

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

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).

$$x[n] = x_c(nT) \quad (121)$$

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

$$x'[n] = x_c(nT') \quad (122)$$

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

$$x_d[n] = x[nM] = x_c(nMT) \quad (123)$$

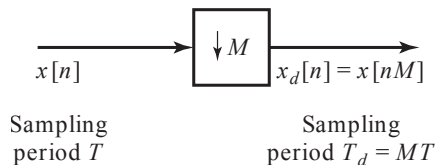


Figura 50: Representação de um compressor ou amostrador discreto. (Oppenheim, 2009).

Downsample por um fator inteiro II

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

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

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

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

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

Downsample por um fator inteiro III

Podemos fazer o índice do somatório da seguinte forma

$$r = i + kM \quad (127)$$

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:

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

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

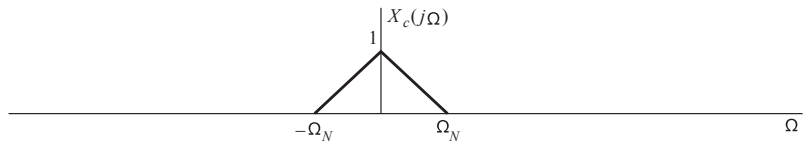
Teremos M cópias de $X(e^{j\omega})$ dilatadas por um fator M e transladadas por $2\pi i/M$.

Downsample por um fator inteiro IV

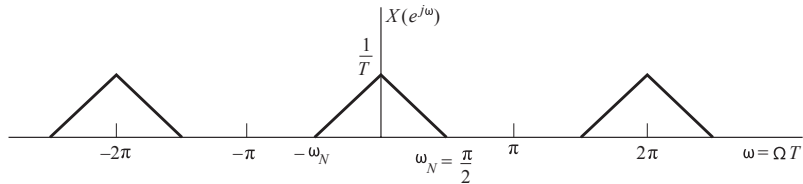
Não haverá *aliasing* se $X(e^{j\omega})$ for limitado em frequência

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

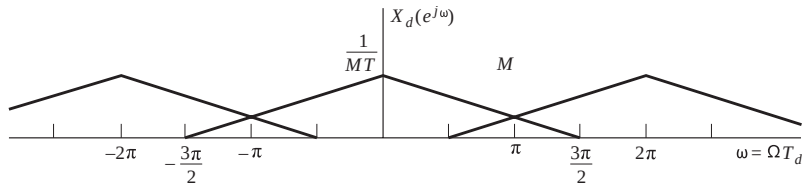
e $2\pi/M \geq 2\omega_N$.

Downsample por um fator inteiro V 

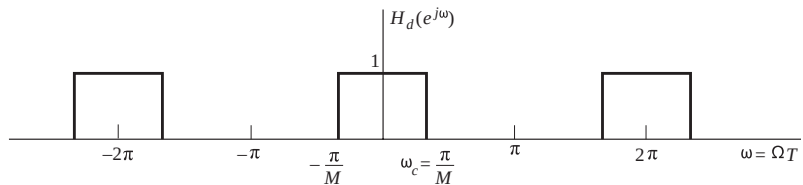
(a)



(b)

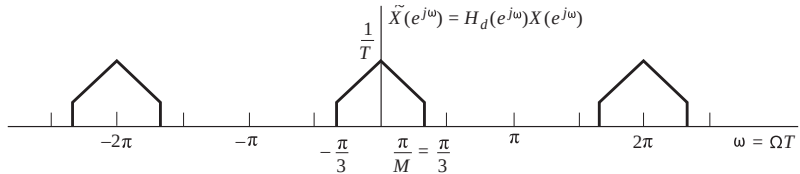
Downsample por um fator inteiro V 

(c)

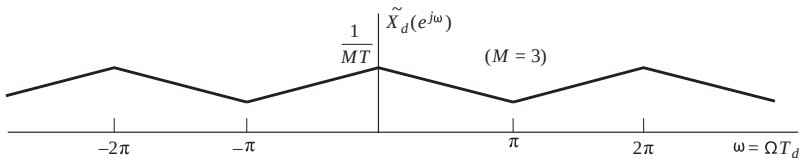


(d)

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

$$\omega_N M \leq \pi \quad \text{ou} \quad \omega_N \leq \pi/M. \quad (131)$$

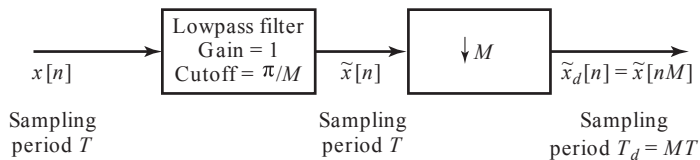


Figura 51: Sistema geral para redução da frequência de amostragem por um fator M . (Oppenheim, 2009).

upsample por um fator inteiro L

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

$$x_i[n] = x_c(nT') \quad (132)$$

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

$$x[n] = x_c(nT) \quad (133)$$

Das equações acima, segue que

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

upsample por um fator inteiro L

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

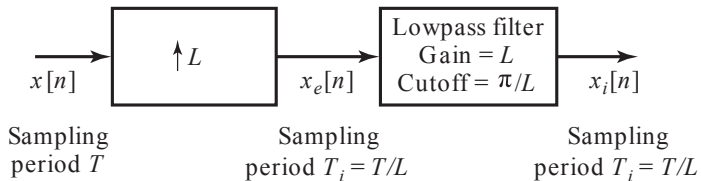


Figura 52: Sistema geral para aumentar a frequência de amostragem por um fator L . (Oppenheim, 2009).

upsample por um fator inteiro III

O sistema da esquerda é chamado expansor. Sua saída será

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

ou de forma equivalente,

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

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

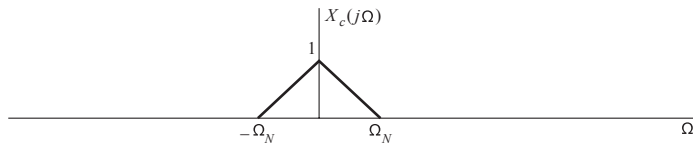
$$X_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} x[k] \delta[n - kL] \right) e^{-j\omega n} \quad (137)$$

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

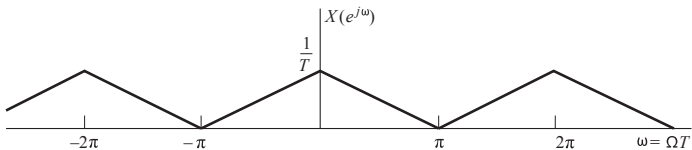
upsample por um fator inteiro L

Ou seja, $X_e(e^{j\omega})$ é obtido através de $X(e^{j\omega})$ por uma compressão de fator L .

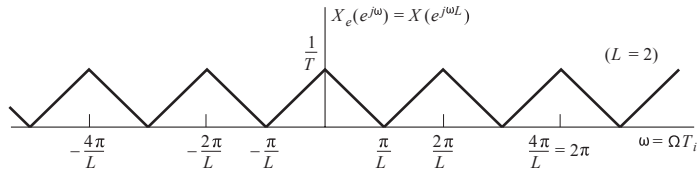
$X_i(e^{j\omega})$ pode ser obtido a partir de $X_e(e^{j\omega})$ realizando a correção da amplitude de $1/T$ para $1/T'$ e removendo todas as réplicas escalonadas de $X_c(j\Omega)$, exceto aquelas nos múltiplos de 2π .

upsample por um fator inteiro V 

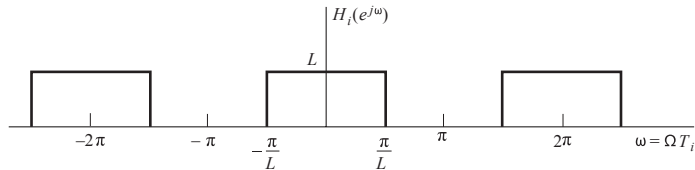
(a)



(b)

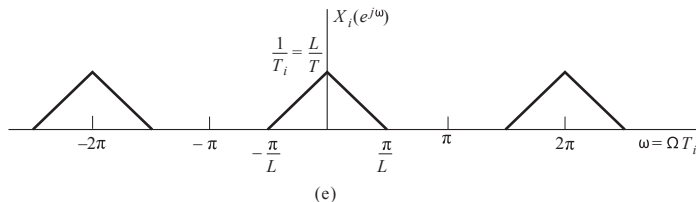
upsample por um fator inteiro V 

(c)



(d)

upsample por um fator inteiro VII



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

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

Obteremos então

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

upsample por um fator inteiro VIII

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

$$h_i[0] = 1, \quad (141)$$

$$h_i[n] = 0, \quad n = \pm L, \pm 2L, \dots \quad (142)$$

Teremos assim

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

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:

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

A saída do filtro interpolador será

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

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

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

Interpolador Linear III

Para $L = 5$ teremos o seguinte exemplo.

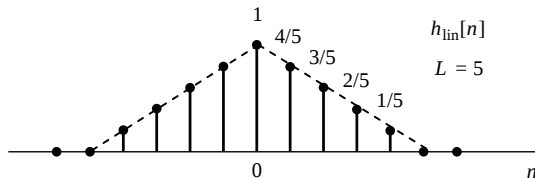


Figura 53: Resposta ao impulso do interpolador linear com $L = 5$. (?).

Interpolador Linear IV

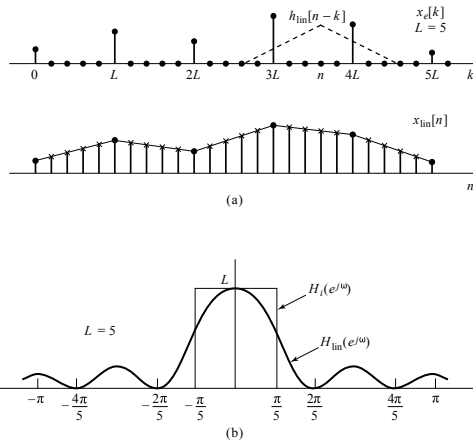


Figura 54: Filtragem com o interpolador linear e sua resposta em frequência. (Oppenheim, 2009).

Mudança da frequência de amostragem por um fator não inteiro I

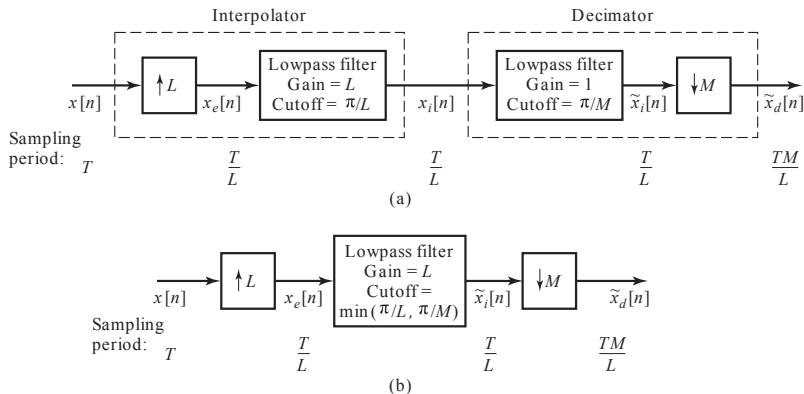


Figura 55: Sistema para mudar a frequência de amostragem por um fator não inteiro. (Oppenheim, 2009).

Mudança da frequência de amostragem por um fator não inteiro II

Exemplo (Mudança da frequência de amostragem por um fator não 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

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

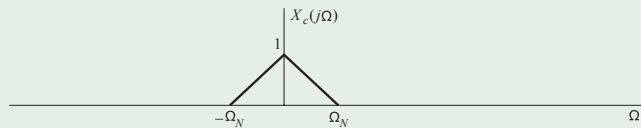
também ilustrado abaixo. Para mudar o período de amostragem para $(3/2)T$, iremos primeiramente interpolar por um fator $L = 2$ e depois decimar por um fator $M = 3$.

...

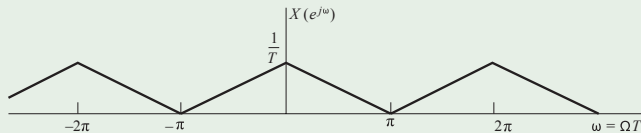
Mudança da frequência de amostragem por um fator não inteiro III

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

continuação...



(a)



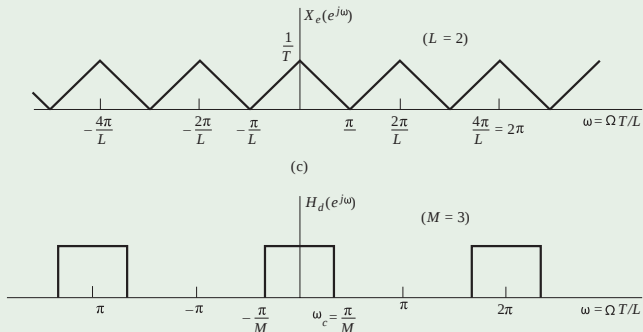
(b)

...

Mudança da frequência de amostragem por um fator não inteiro IV

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

continuação...

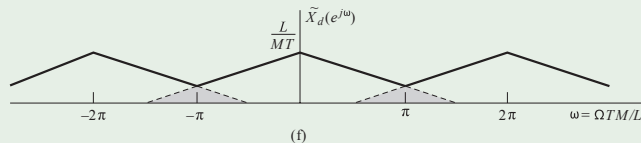
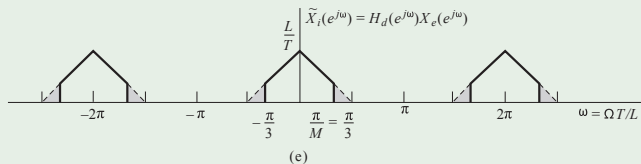


...

Mudança da frequência de amostragem por um fator não inteiro V

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

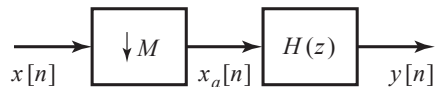
continuação...



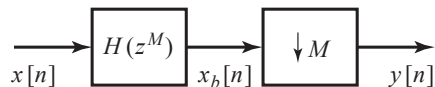
Processamento Multi-taxa de Sinais I

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



(a)



(b)

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:

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

e

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

logo

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

Intercâmbio entre filtro e downsampling/upsampling III

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

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

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

o que corresponde ao sistema (a) da figura.

Intercâmbio entre filtro e downsampling/upsampling IV

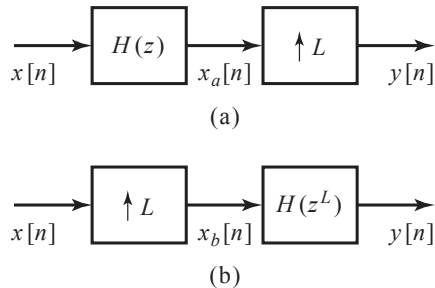


Figura 57: Dois sistemas equivalentes, com base nas identidades do *upsample*. (Oppenheim, 2009).

Intercâmbio entre filtro e downsampling/upsampling V

$$Y(e^{j\omega}) = X_a(e^{j\omega L}) \quad (153)$$

$$= X(e^{j\omega L})H(e^{j\omega L}) \quad (154)$$

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

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

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

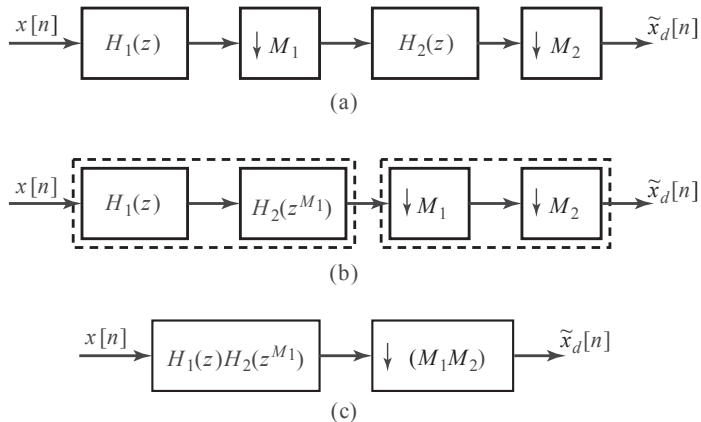


Figura 58: Decimação em múltiplos estágios. (Oppenheim, 2009).

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

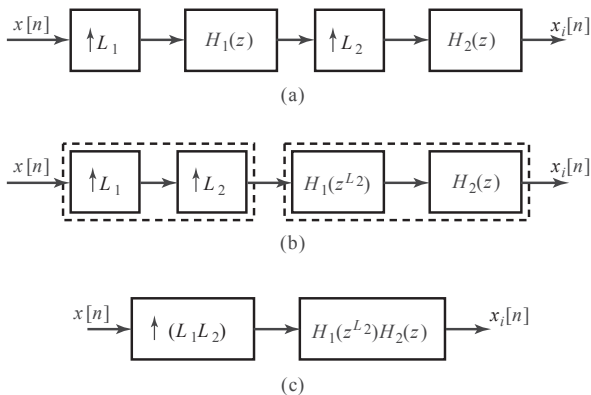


Figura 59: Interpolação em múltiplos estágios. (Oppenheim, 2009).

Decomposição Polifásica I

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

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

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

Decomposição Polifásica II

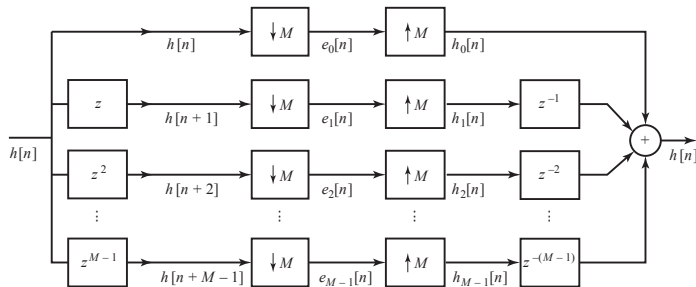


Figura 60: Decomposição polifásica de $h[n]$ usando as componentes $e_k[n]$. (Oppenheim, 2009).

As sequências $e_k[n]$ são dadas por

$$e_k[n] = h[nM + k] = h_k[nM] \quad (159)$$

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

Decomposição Polifásica III

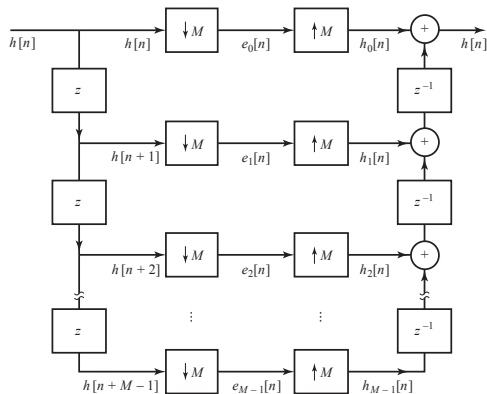


Figura 61: Representação com atrasos em cadeia. (Oppenheim, 2009).

Decomposição Polifásica IV

Representando no domínio da frequência ou no domínio z , teremos

$$Z\{h[n]\} = Z\left\{\sum_{k=0}^{M-1} h_k[n - k]\right\} \quad (160)$$

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

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

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

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.

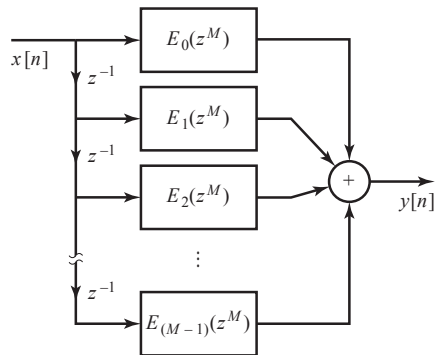


Figura 62: Estrutura da decomposição polifásica de $h[n]$. (Oppenheim, 2009).

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).

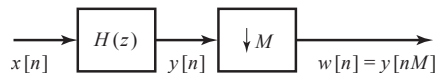


Figura 63: Sistema de decimação. (Oppenheim, 2009).

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

$$e_k[n] = h[nM + k] \quad (164)$$

Implementação de decimadores usando decomposição polifásica II

e assim podemos representar $H(z)$ como

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

Implementação de decimadores usando decomposição polifásica III

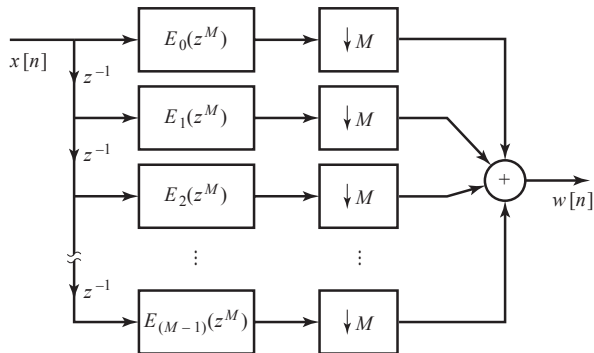


Figura 64: Implementação utilizando decomposição polifásicas. (Oppenheim, 2009).

Implementação de decimadores usando decomposição polifásica IV

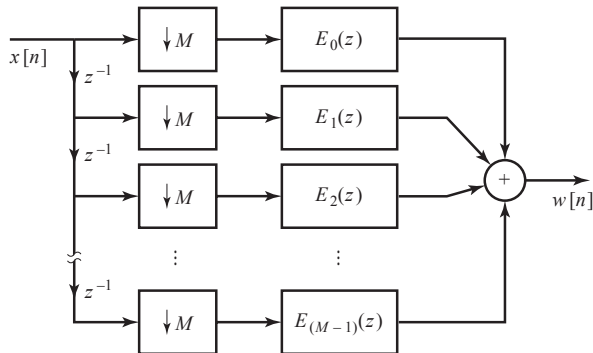


Figura 65: Implementação após utilizar a identidade do *downsapling*. (Oppenheim, 2009).

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

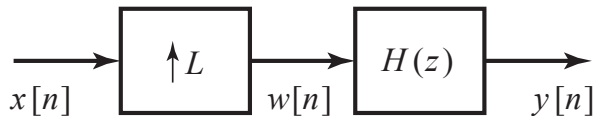


Figura 66: Sistema de interpolação. (Oppenheim, 2009).

Implementação de interpoladores usando decomposição polifásica II

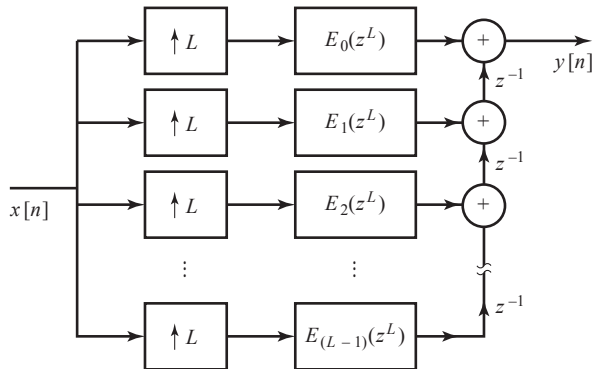
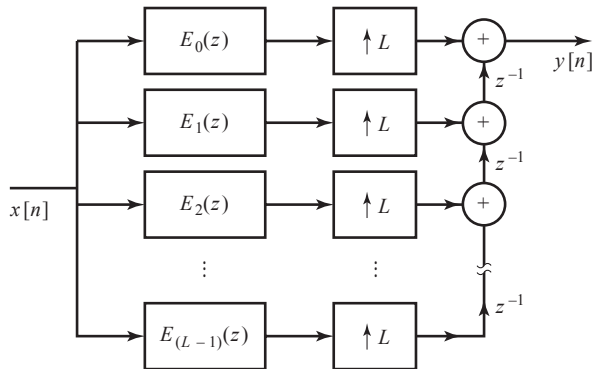


Figura 67: Decomposição polifásica do filtro de interpolação. (Oppenheim, 2009).

Implementação de interpoladores usando decomposição polifásica III

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

Banco de Filtros Multi taxa I

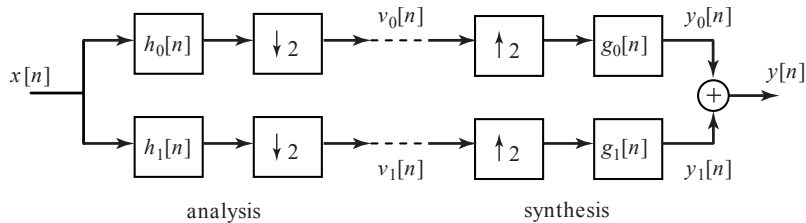


Figura 69: Banco de filtro de análise e síntese com dois canais. (Oppenheim, 2009).

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

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

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

Note que teremos reconstrução perfeita se os filtros forem ideais, entretanto também é possível obter reconstrução perfeita com filtros não ideais, 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

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

Banco de Filtros Multi taxa III

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

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

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

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

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

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

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

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

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

$$h_0[n] = c_0\delta[n - 2n_0] + c_1\delta[n - 2n_1 - 1], \quad (174)$$

onde n_0 e n_1 são inteiros arbitrários e $c_0c_1 = 1/4$.

Por exemplo, considere

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

com resposta em frequência

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

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

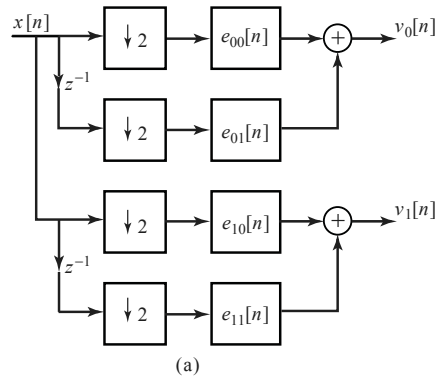


Figura 70: Representação polifásica do banco de filtros de análise. (Oppenheim, 2009).

Banco de Filtros Multi taxa VI

Usando as relações

$$e_{00}[n] = h_0[2n] \quad (177)$$

$$e_{01}[n] = h_0[2n + 1] \quad (178)$$

$$e_{10}[n] = h_1[2n] = e^{j2\pi n} h_0[2n] = e_{00}[n] \quad (179)$$

$$e_{11}[n] = h_1[2n + 1] = e^{j2\pi n} e^{j\pi} h_0[2n + 1] = -e_{01}[n]. \quad (180)$$

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

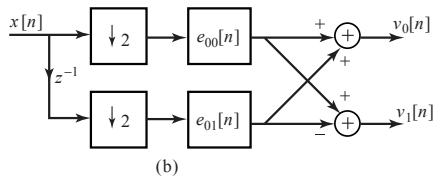


Figura 71: Representação simplificada. (Oppenheim, 2009).

Banco de Filtros Multi taxa VII

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

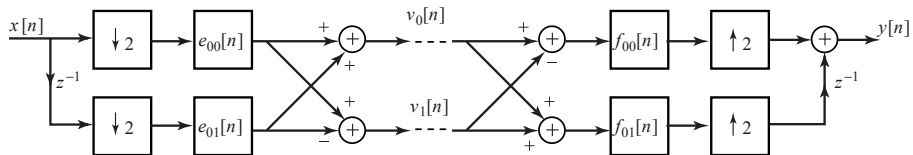


Figura 72: Representação polifásica do banco de filtros de análise e síntese. (Oppenheim, 2009).

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

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.