



Teoria da Informação

- Componente Teórica
 - Exame
 - 3 Séries de Problemas
- Componente Laboratorial
 - 5 Trabalhos Práticos
- Projecto
 - Algoritmo de Codificação/Descodificação

$$\text{Nota Final} = 0,5 \text{ Exame} + 0,1 \text{ Séries} + 0,2 \text{ C. Lab.} + 0,2 \text{ Projecto}$$

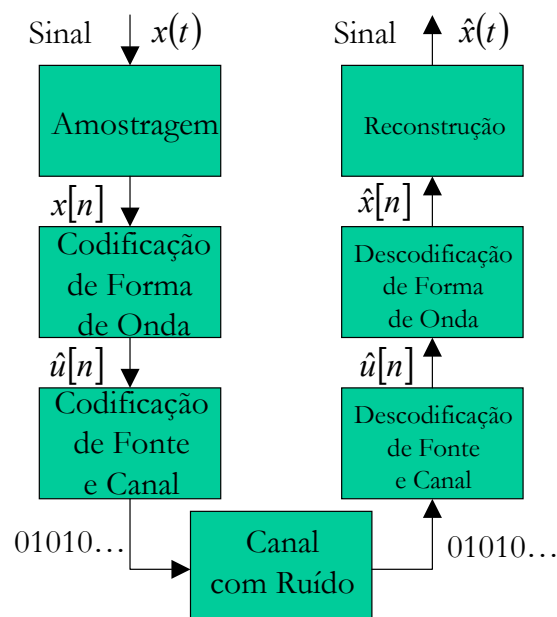


- 3 Séries de Problemas
 - Codificação de Forma de Onda
 - Probabilidades e Entropia
 - Codificação de Fonte
 - 5 Trabalhos Laboratoriais
 - *Discrete Cosine Transform* - DCT (2 partes)
 - Telecópia
 - Norma JPEG
 - Codificação Videotelefónica Norma ITU-T.H261
- Trabalhos elaborados pelo Prof. Fernando Pereira (IST/IT)



Resumo

1. Codificação de Forma de Onda
2. Teoria de Comunicação
 - 2.1 Codificação de Fonte
 - 2.2 Codificação de Canal
3. Codificação de Imagem

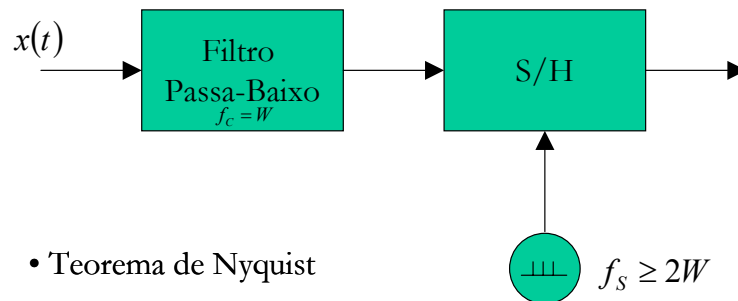


Bibliografia

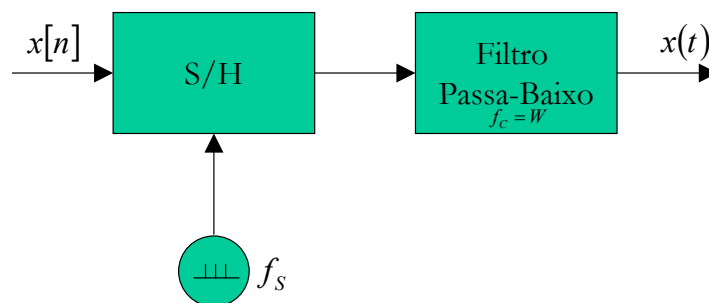
- ➡ • A. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989
- N. Jayant & P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, 1984
- ➡ • D. MacKay, *Information Theory, Inference, and Learning Algorithms*, 1998
- S. Haykin, *Communication Systems*, John Wiley & Sons, 1994
- ➡ • A. Tekalp, *Digital Video Processing*, Prentice-Hall, 1995
- R. Clarke, *Digital Compression of Still Images and Video*, Academic Press, 1996



Amostragem

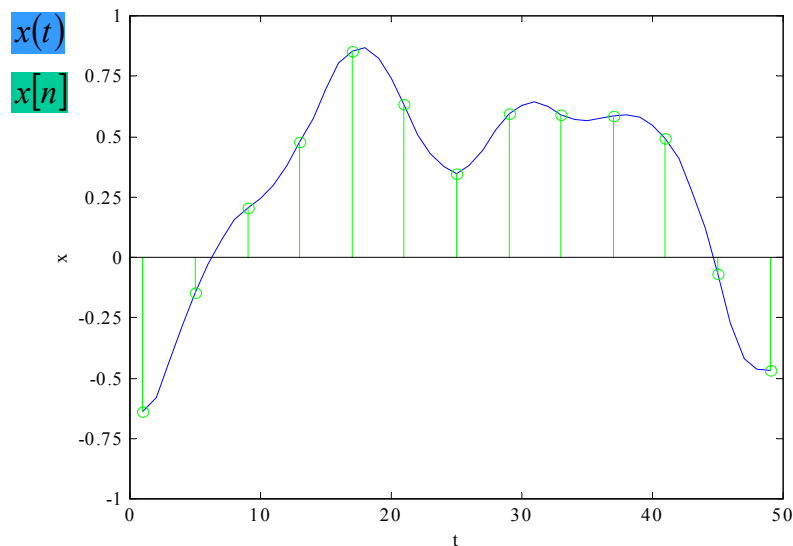


Reconstrução





Amostragem/Reconstrução



1. Codificação de Forma de Onda

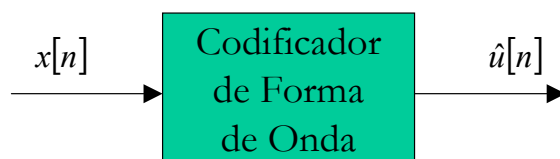
- 1.1 Quantificação Uniforme
- 1.2 Quantificação não-Uniforme
- 1.3 Quantificação Vectorial
- 1.4 Codificação Preditiva
- 1.5 Codificação Por Transformada
 - 1.5.1 Discrete Fourier Transform *DFT*
 - 1.5.2 Discrete Cosine Transform *DCT*
 - 1.5.3 Transformada de Karhunen-Loeve



1. Codificação de Forma de Onda

Objectivo

Transformar sinais discretos numa sequência de valores quantificados com vista à sua transmissão por um canal digital.



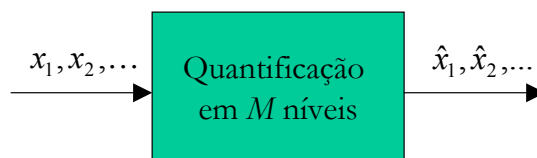
1.1 Quantificação Uniforme

- Quantificação em M níveis

A gama de variação é dividida em M intervalos de igual comprimento

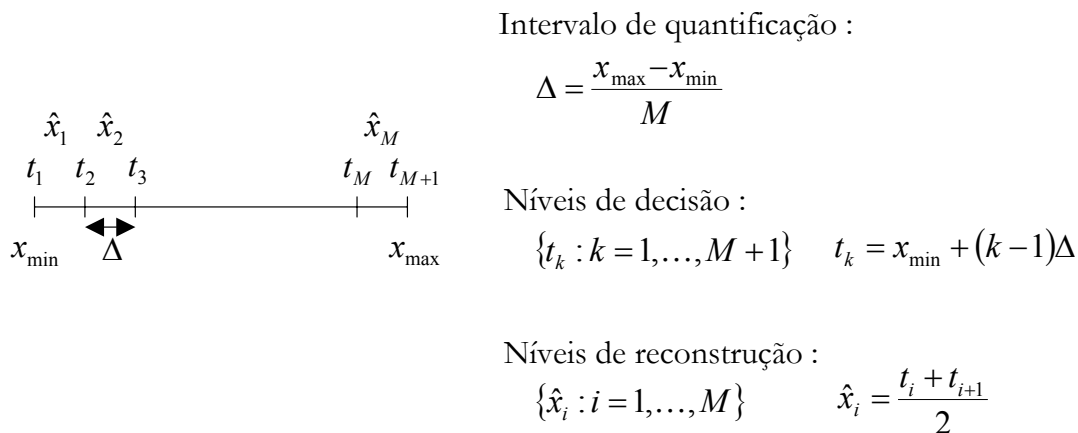
A cada intervalo é atribuído um nível de reconstrução

A cada amostra é atribuído o nível de reconstrução do intervalo a que pertence





1.1 Quantificação Uniforme



1.1 Quantificação Uniforme

Q? - Que informação é necessário enviar para o receptor ?

- níveis de reconstrução
- para cada amostra é enviado o índice do nível de reconstrução

Q? - Quantos bits são necessários para enviar a informação desejada ?

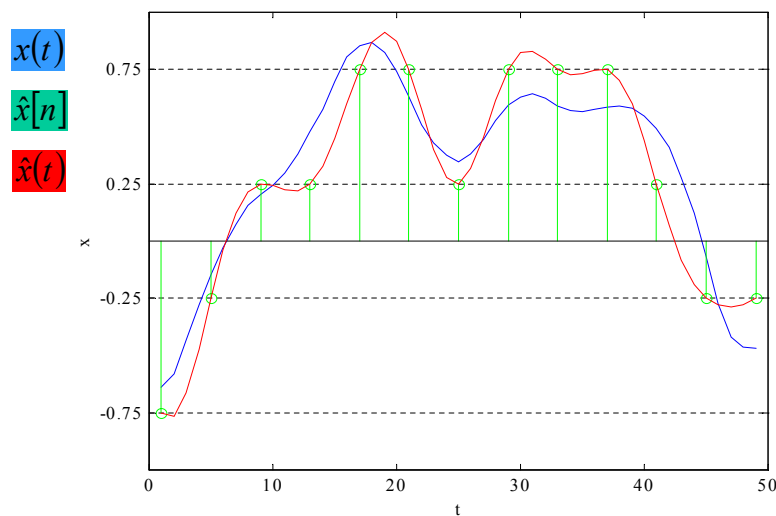
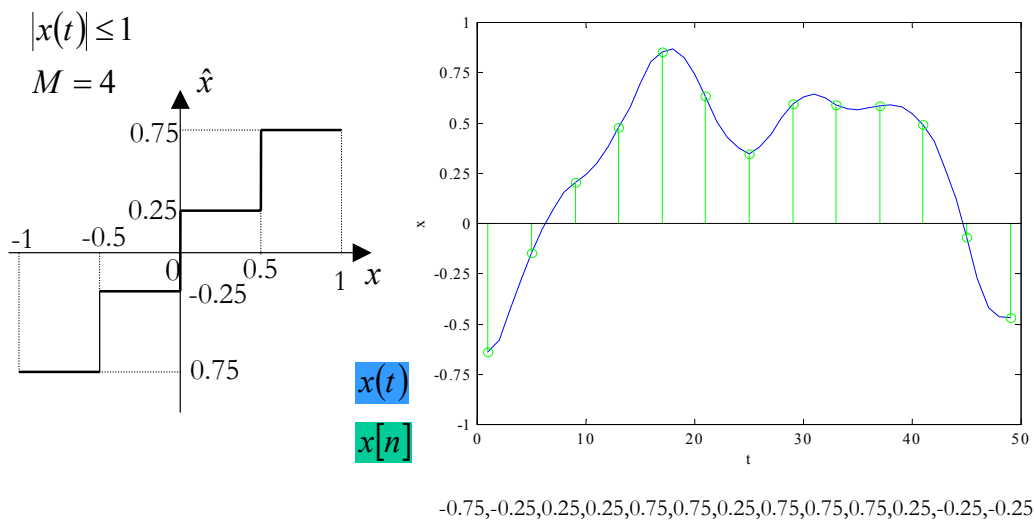
$$H = \log_2 M \quad \text{bits/símbolo}$$

r número médio de símbolos por segundo

$$R = rH \quad \text{bits/seg.}$$



Exemplo : *PCM* Uniforme



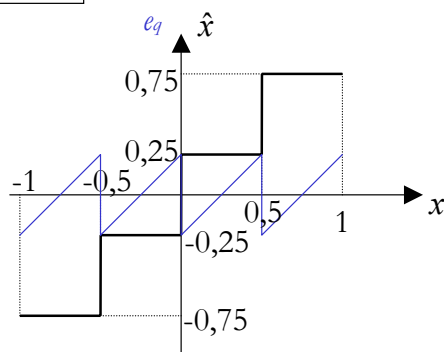
Q? - Será $x(t)$ igual a $\hat{x}(t)$?



1.1 Quantificação Uniforme - Erro de Quantificação

R! - Não, dado que quase sempre $\hat{x} \neq x$.

$$e_q = x - \hat{x}$$



1.1 Quantificação Uniforme - Erro de Quantificação

- Variância do erro de quantificação

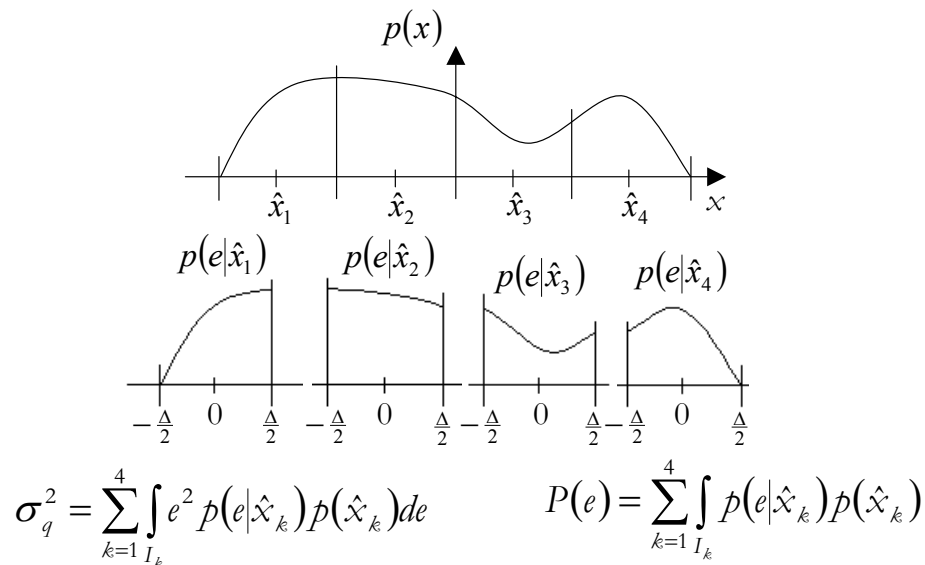
$$N_q = \sigma_q^2 = E[e^2]$$

- Relação Sinal/Ruído

$$\left(\frac{S}{N}\right)_D = \frac{S_x}{\sigma_q^2}$$



1.1 Quantificação Uniforme - Erro de Quantificação



1.2 Quantificação não-Uniforme

Q? - Será que todos os sinais devem ter os mesmos intervalos de quantificação com vista a minimizar o erro de quantificação ?

Uma estratégia para encontrar os intervalos de quantificação e os níveis de reconstrução consiste em minimizar a variância do erro de quantificação (erro quadrático médio) - **Quantificador de Lloyd-Max**

$$E[e^2] = \int_{t_1}^{t_{M+1}} (x - x_i)^2 p_X(x) dx$$

$$\min \xi \Rightarrow \frac{\partial \xi}{\partial t_i} = 0, \frac{\partial \xi}{\partial x_i} = 0$$



1.2 Quantificação não-Uniforme

Quantificador de Lloyd-Max

1. Inicialização

Exemplo: quantificador uniforme



2. Iteração

níveis de reconstrução

$$\hat{x}_i = \frac{\int_{t_i}^{t_{i+1}} xp(x)dx}{\int_{t_i}^{t_{i+1}} p(x)dx}$$

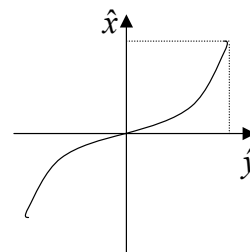
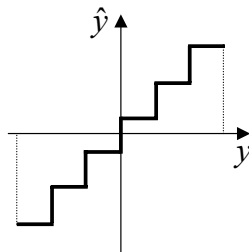
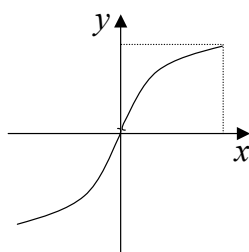
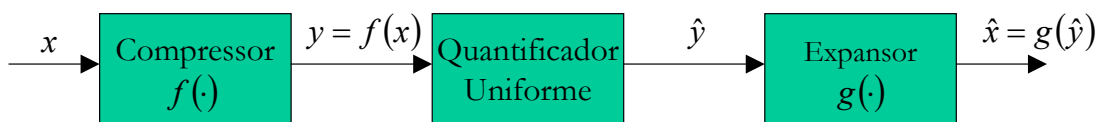
intervalos de quantificação

$$t_i = \frac{\hat{x}_{i-1} + \hat{x}_i}{2}$$



1.2 Quantificação não-Uniforme

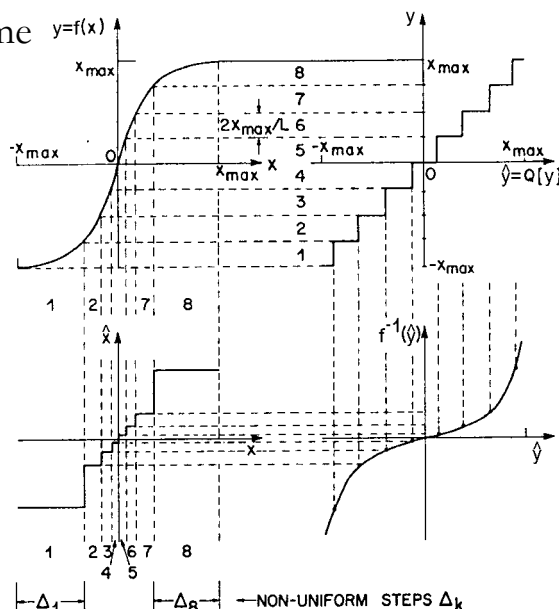
Compressor-Expansor (*Compandor*)





1.2 Quantificação não-Uniforme

Compandor



1.2 Quantificação não-Uniforme

Compandor

A escolha das funções f e g deve ser feita de forma a que o quantificador não uniforme resultante seja aproximado do quantificador de Lloyd-Max.

$$g(x) = f^{-1}(x)$$

$$f(x) = 2a \left\{ \frac{\int_{t_1}^x [p_U(u)]^{\frac{1}{3}} du}{\int_{t_1}^{t_{L+1}} [p_U(u)]^{\frac{1}{3}} du} \right\} \quad \text{Onde } [-a, a] \text{ define o intervalo de variação do quantificador uniforme e } p_U(u) \text{ define a f.d.p. de } x.$$



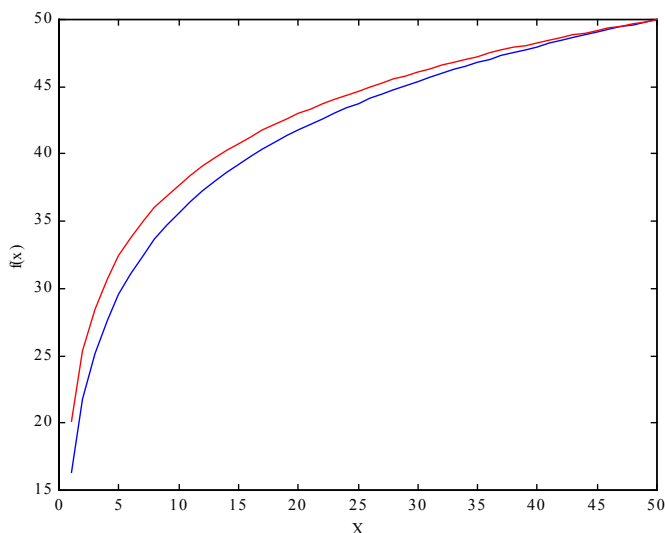
1.2 Quantificação não-Uniforme

Compandor

Exemplos

• PCM Lei A

• PCM Lei μ



1.3 Quantificação Vectorial

Problema : Como quantificar dados N dimensionais ?

Utilizar um conjunto de M vectores típicos - *Livro de Código*

$$L = \{\hat{x}_i, i = 1 \dots M\} \quad \text{onde } \hat{x}_i \in R^N$$

Codificação

Dado um conjunto de dados $X = \{x : x \in R^N\}$, encontrar no livro de código o vector típico \hat{x}_i que melhor representa o dado x_i segundo um determinado critério.

Transmite-se o índice i .

Descodificação

O vector reconstruído é o elemento i do livro de código $L(i) = \hat{x}_i$



1.3 Quantificação Vectorial

Questões por resolver :

Q1? - Qual o livro de código que minimiza o erro de representação dos dados ?

R1! - P. ex. C-médias.

Q2? - Qual o critério para escolher o vector do livro de código ?

R2! - P. ex. Distância Euclidiana.

Q3? - Qual o algoritmo de pesquisa ?

R3! - P. ex. Pesquisa exaustiva.

Desvantagem - Peso computacional.

Método suboptimo : Procura não exaustiva em árvore binária.



1.3 Quantificação Vectorial

Algoritmo de C-Médias

Inicialização do livro de código

Passo1

Quantificam-se os vectores de X

$$\hat{x} = \arg \min_{\hat{x}_c} \|x - \hat{x}_c\|$$

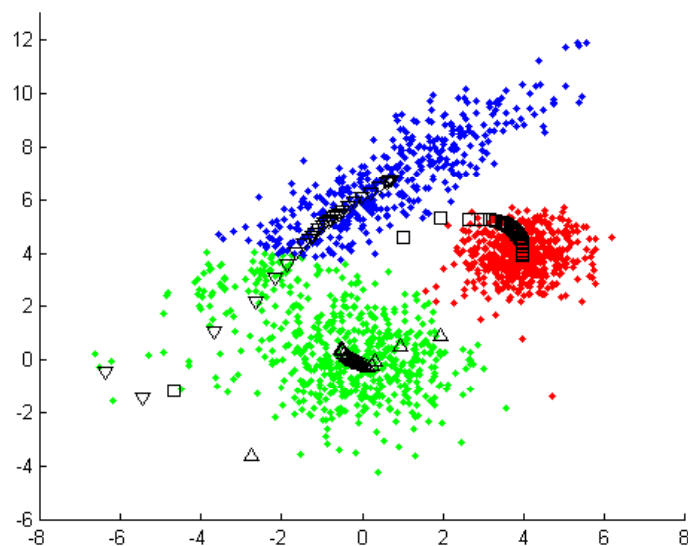
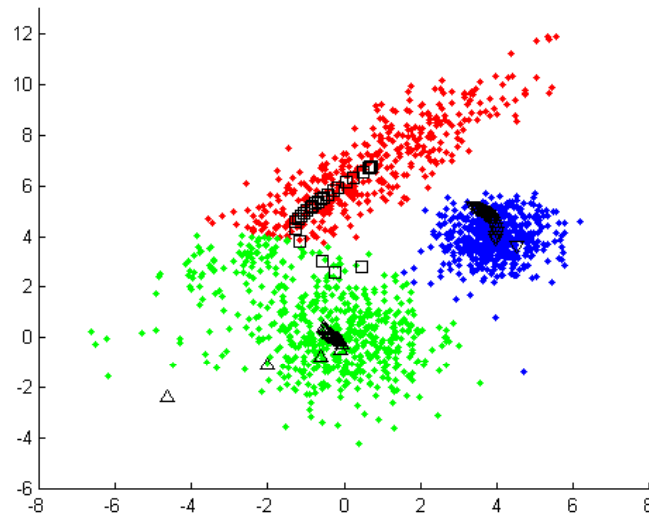
Passo2

Define-se \hat{x}_c como o centro de massa (valor médio) dos padrões da classe c

$$\hat{x}_c = \frac{1}{N_C} \sum_{x \in X_C} x$$



1.3 Quantificação Vectorial





1.3 Quantificação Vectorial

Procura não exaustiva em árvore binária

Dividir o livro de código L numa sequência de $L_i, i = 1, \dots, \log_2 M$ livros de código mais pequenos.

Os livros de código dos níveis inferiores são versões reduzidas de L .

Para $j=1$ o conjunto de todos os vectores do livro de código é dividido em duas regiões com $\frac{M}{2}$ elementos cada. Os centros de massa de cada região define o livro de código L_1

Quando é identificada a região correspondente ao vector de entrada x , a região escolhida é outra vez dividida.

$$\underbrace{\hat{x}_1 \quad \dots \quad \hat{x}_{\frac{M}{2}}}_{\hat{x}_2^1} \mid \underbrace{\hat{x}_{\frac{M}{2}+1} \quad \dots \quad \hat{x}_M}_{\hat{x}_2^1}$$



1.4 Codificação Preditiva

As amostras de sinais reais derivam de processos físicos que exibem por vezes redundância. Nestes casos é possível tirar partido da sequência de amostras e prever uma amostra com base nas anteriores.

$$\tilde{x}[n] = \sum_k a_k x[n-k]$$

$$\text{Erro de predição : } e[n] = x[n] - \tilde{x}[n]$$

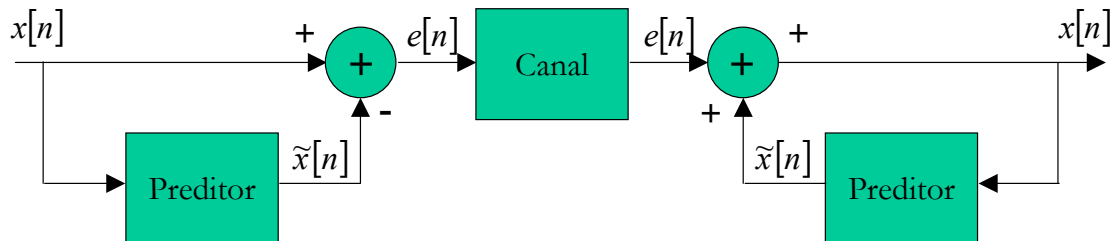
$$x[n] = \sum_k a_k x[n-k] + e[n]$$

Vantagem : A gama dinâmica do erro de predição é menor que a do sinal, permitindo diminuir o erro de quantificação ou o número de bits para enviar o sinal com a mesma relação sinal/ruído.



1.4 Codificação Preditiva

Análise/Síntese com predição

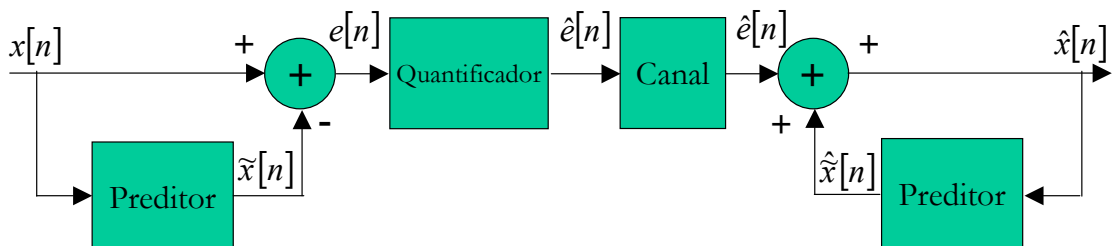


Problema - Não se envia $e[n]$ mas o seu valor quantificado !



1.4 Codificação Preditiva

Análise/Síntese com quantificação



Não funciona - $\hat{\hat{x}}[n] \neq \tilde{x}[n]$

$\hat{x}[n]$ vai acumulando os erros de quantificação !



Exemplo : Análise/Síntese com quantificação

$$x[n] = 0,1 \cdot 2^n + w$$

$w \rightarrow v.a. \text{ uniforme}$

$$[-0,025; -0,025]$$

$$\tilde{x}[n] = 2x[n-1]$$

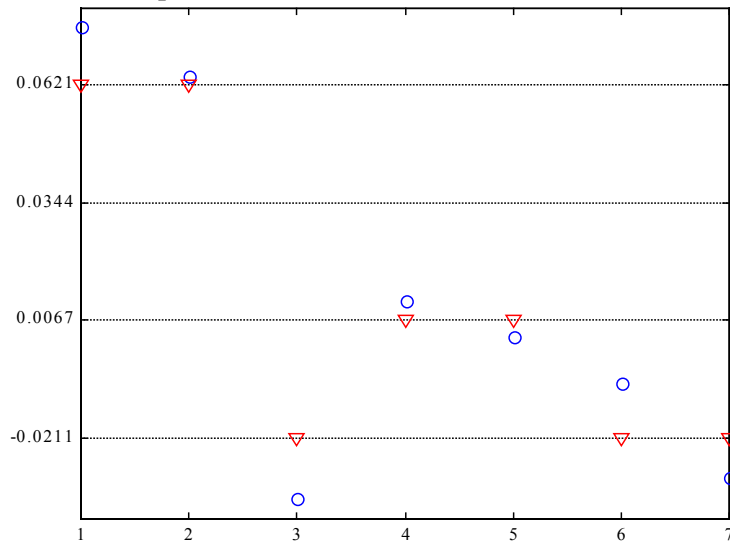
quantificação

uniforme

2 bits

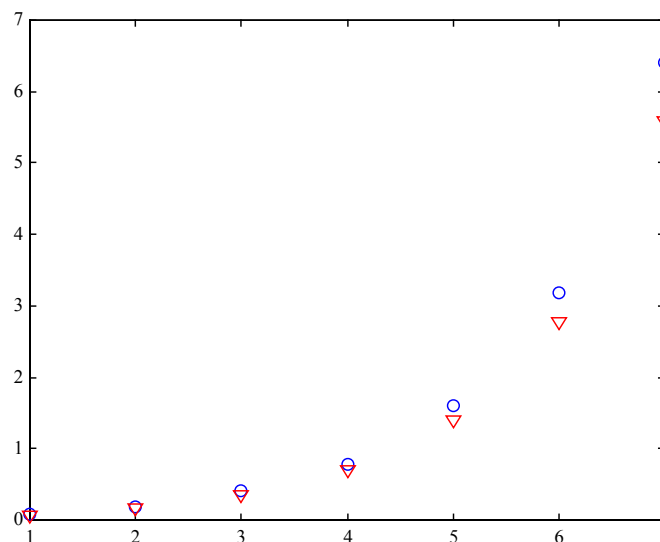
$e[n]$

$\hat{e}[n]$



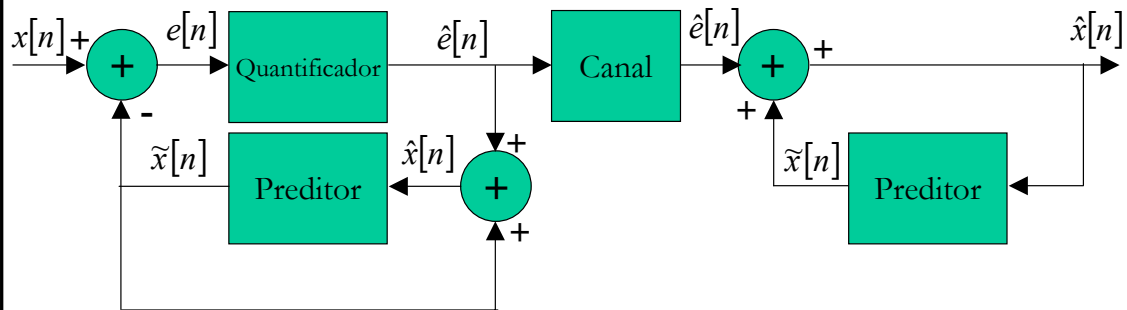
$x[n]$

$\hat{x}[n]$



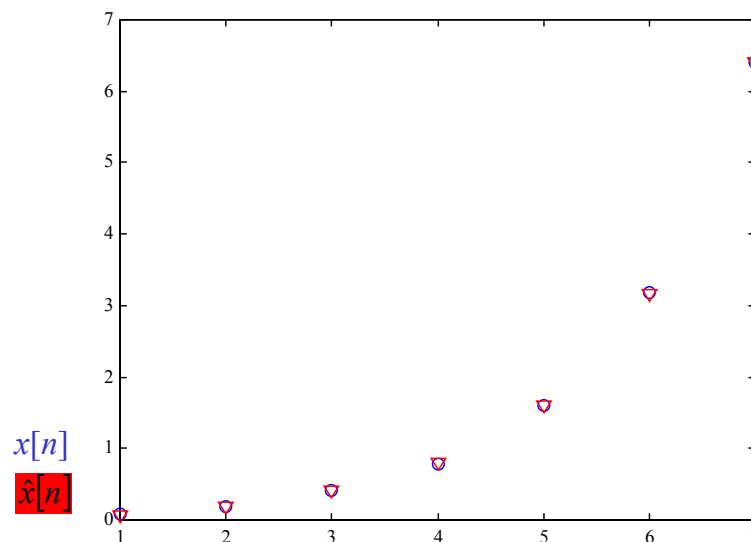


1.4 Codificação Preditiva



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

Nota - O emissor contém uma réplica do receptor.





1.4 Codificação Preditiva

Ganho de Predição

$$G_p = 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_e^2} \right)$$

Exemplo de aplicações

Delta Modulation - *DM* $\hat{e}[n] = \pm \Delta$

Diferencial *PCM* - *DPCM*

GSM FR e GSM EFR



Exemplo : Ganho de Predição

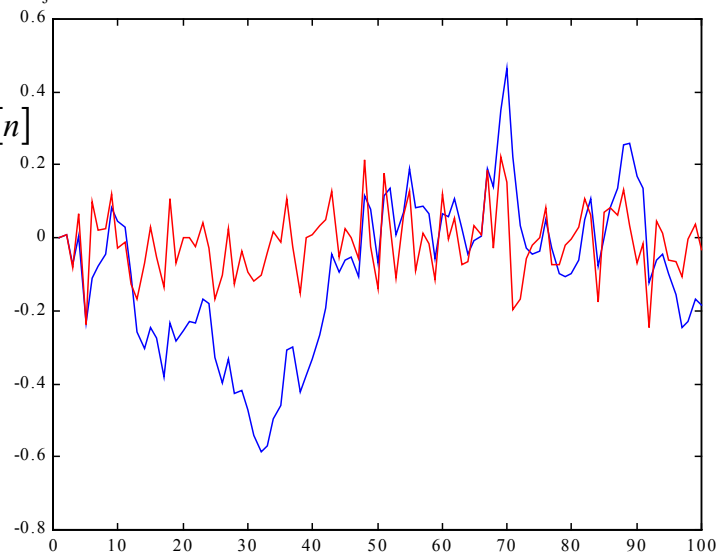
$$y[n] = 0.9y[n-1] + 0.1w[n]$$

$$w[n] \sim N(0,1)$$

$$G_p = 10 \log \frac{0.043}{0.008} \\ = 6.8 \text{ dB}$$

$y[n]$

$e[n]$

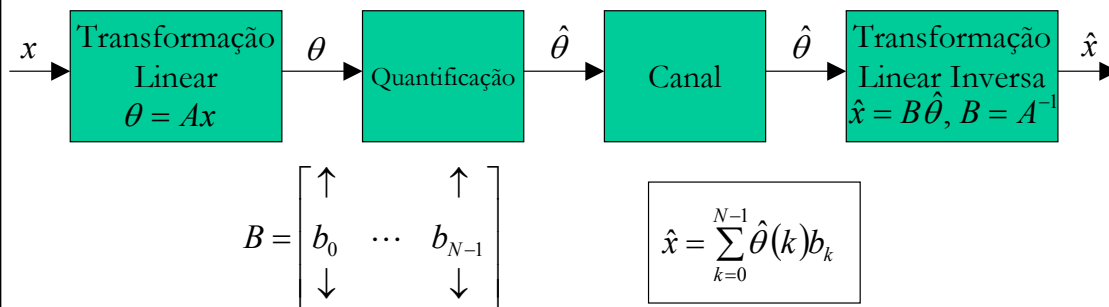




1.5 Codificação por Transformada

Transformação de N amostras do sinal x em N coeficientes da transformação.

A transformação linear pode ser interpretada como uma decomposição do sinal x numa combinação linear de vectores que formam uma outra base.



1.5 Codificação por Transformada

Transformações Ortogonais

- $A^{-1} = A^T \Rightarrow B = A^T$

Para a reconstrução do sinal não é necessário inverter a matriz de transformação

- $b_i^T b_i = b_i b_i^T = 1 \Leftrightarrow \|b_i\| = 1 \quad \text{e} \quad b_i^T b_j = 0 \quad i \neq j \Leftrightarrow \langle b_i, b_j \rangle = 0$

A base da transformação é ortonormada (componentes não correlacionadas)

- $\sigma_e^2 = \sigma_q^2$

A variância do erro de reconstrução é igual à introduzida pelo processo de quantificação dos coeficientes



1.5 Codificação por Transformada

Transformação Óptima (Transformação de Karhunen-Loeve)

Corresponde à descorrelação das amostras do sinal de entrada

$R_X = \{R_X(k, l) : k, l = 0, 1, \dots, N-1\}$ Matriz de covariância da entrada

$L_X = \{L_X(k, l) : k, l = 0, 1, \dots, N-1\}$ Matriz dos vectores próprios de R_X

$D_X = \{D_X(k) : k = 0, 1, \dots, N-1\}$ Matriz diagonal dos valores próprios de R_X

$$\begin{array}{l} \theta = L_X^T x \\ \hat{x} = L_X \hat{\theta} = \sum_{k=0}^{N-1} \hat{\theta}(k) l_k \end{array} \quad L_X = \begin{bmatrix} \uparrow & & \uparrow \\ l_0 & \cdots & l_{N-1} \\ \downarrow & & \downarrow \end{bmatrix} \quad R_\theta = L_X^T R_X L_X = D_X$$



1.5 Codificação por Transformada

A transformação de KL envolve um peso computacional muito grande.

É melhor do ponto de vista prático utilizar métodos subóptimos que têm menor peso computacional mas que no entanto produzem bons resultados.

- Discrete Fourier Transform - *DFT*
- Discrete Cosine Transform - *DCT*



1.5 Codificação por Transformada

Discrete Fourier Transform - *DFT*

$$\theta = Fx \quad x = F^* \theta$$

$$F = \left\{ \frac{1}{\sqrt{N}} e^{-j2\pi kn/N} \right\}_{k,n=0,1,\dots,N-1}$$

$$\theta(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1$$

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \theta(k) e^{-j2\pi kn/N}, \quad n = 0, 1, \dots, N-1$$

A *DFT* também descorrelaciona os dados da entrada



1.5 Codificação por Transformada

Discrete Fourier Transform - *DFT*

Poderemos interpretar os vectores da base da seguinte forma

$$b_k = \left\{ \frac{1}{\sqrt{N}} e^{-j2\pi kl/N} \right\}_{l=0,1,\dots,N-1}$$

$$\text{Re}\{b_k\} = \frac{1}{\sqrt{N}} \cos\left(2\pi \frac{kl}{N}\right), \quad l = 0, 1, \dots, N-1$$

$$\text{Im}\{b_k\} = \frac{1}{\sqrt{N}} \sin\left(2\pi \frac{kl}{N}\right), \quad l = 0, 1, \dots, N-1$$



1.5 Codificação por Transformada

Discrete Cosine Transform - *DCT*

$$\theta = Cx \quad x = C^T \theta$$

$$C = \left\{ \sqrt{\frac{2}{N}} \alpha(k) \cos \frac{(2n+1)k\pi}{2N} \right\}_{k,n=0,1,\dots,N-1}$$

$$\alpha(0) = \frac{1}{\sqrt{2}} \quad \alpha(k) = 1, k \neq 0$$

$$\theta(k) = \sqrt{\frac{2}{N}} \alpha(k) \sum_{n=0}^{N-1} x(n) \cos \frac{(2n+1)k\pi}{2N}, \quad k = 0, 1, \dots, N-1$$

$$x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \alpha(k) \theta(k) \cos \frac{(2n+1)k\pi}{2N}, \quad n = 0, 1, \dots, N-1$$



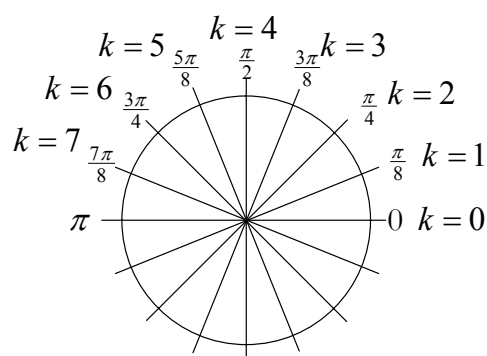
1.5 Codificação por Transformada

Discrete Cosine Transform - *DCT*

N=8

Vectores da base

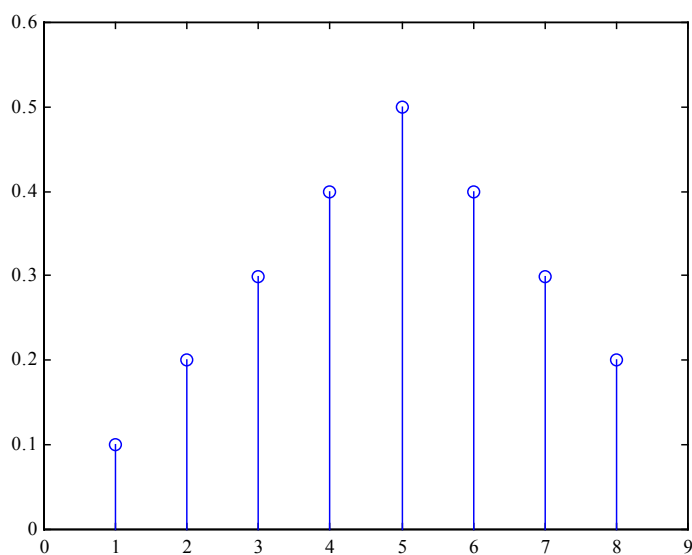
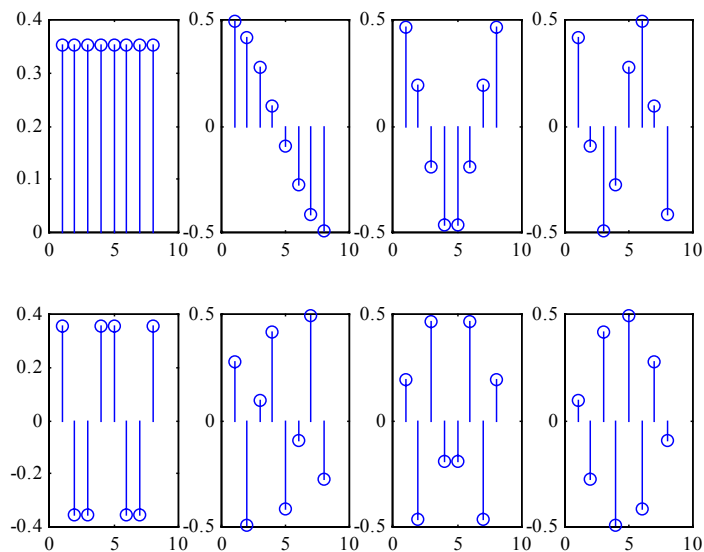
$$b_k = \left\{ \sqrt{\frac{2}{N}} \alpha(k) \cos \frac{(2n+1)k\pi}{2N} \right\}_{n=0,1,\dots,N-1}$$



A *DCT* também descorrelaciona os dados da entrada



Exemplo : $DCTN=8$



$$\theta = \begin{bmatrix} 0.8485 \\ -0.1281 \\ -0.3154 \\ 0.0450 \\ 0.0000 \\ -0.0301 \\ -0.0224 \\ 0.0255 \end{bmatrix}$$



1.5 Codificação por Transformada

Transformada Bidimensionais

$$\theta(k, l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n) a(k, l, m, n)$$

$$x(m, n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \hat{\theta}(k, l) b(k, l, m, n)$$

Transformada Bidimensionais Separáveis

$$\begin{aligned} a(k, l, m, n) &= a_v(k, m) a_h(l, n) & A_h &= \{a_v(i, j)\}_{i,j=0,1,\dots,N-1} \\ & & A_v &= \{a_h(i, j)\}_{i,j=0,1,\dots,N-1} \\ b(k, l, m, n) &= b_v(k, m) b_h(l, n) & B_h &= \{b_v(i, j)\}_{i,j=0,1,\dots,N-1} \\ & & B_v &= \{b_h(i, j)\}_{i,j=0,1,\dots,N-1} \end{aligned}$$



1.5 Codificação por Transformada

Transformada Bidimensionais Separáveis

$$\theta(k, l) = \sum_{m=0}^{N-1} a_v(k, m) \sum_{n=0}^{N-1} x(m, n) a_h(l, n)$$

$$\theta = A_v X A_h^T$$

$$x(m, n) = \sum_{k=0}^{N-1} b_v(k, m) \sum_{l=0}^{N-1} \theta(k, l) b_h(l, n)$$

$$X = B_v^T \theta B_h$$

$$A_v = A_h = A$$

$$B_v = B_h = B$$

$$\begin{aligned} X &= \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \theta(k, l) B_{kl} \\ B_{kl} &= b_k b_l^T \end{aligned}$$

O sinal reconstruído (imagem) é uma combinação linear das imagens de base B_{kl} pesadas pelos parâmetros $\theta(k, l)$



1.5 Codificação por Transformada

DCT Bidimensional

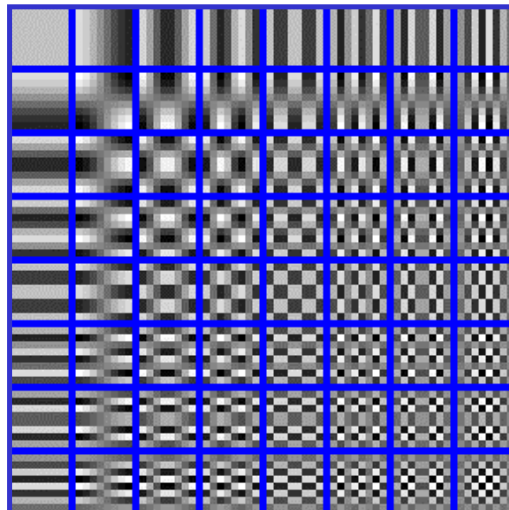
$$\theta(k,l) = \frac{2}{N} \alpha(k) \alpha(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) \cos \frac{(2m+1)k\pi}{2N} \cos \frac{(2n+1)l\pi}{2N}, \quad k,l = 0,1,\dots,N-1$$

$$x(m,n) = \frac{2}{N} \sum_{k=0}^{N-1} \alpha(k) \alpha(l) \theta(k,l) \cos \frac{(2m+1)k\pi}{2N} \cos \frac{(2n+1)l\pi}{2N}, \quad m,n = 0,1,\dots,N-1$$

$$X = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \theta(k,l) B_{kl} \quad B_{kl} = b_k b_l^T \quad b_k = \left\{ \sqrt{\frac{2}{N}} \alpha(k) \cos \frac{(2n+1)k\pi}{2N} \right\}_{n=0,1,\dots,N-1}$$

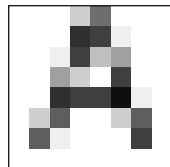


Exemplo : Matrizes de base para a *DCT* de 8x8

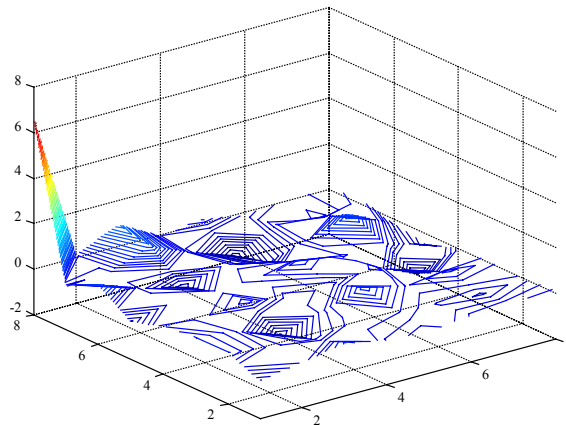




Exemplo : Decomposição de uma imagem



$$\theta(k,l) = \begin{bmatrix} 6.56 & 0.12 & 0.96 & -0.05 & 0.19 & -0.15 & 0.01 & 0.06 \\ -0.07 & 0.01 & 0.66 & -0.12 & -0.76 & 0.12 & -0.07 & 0.17 \\ 0.63 & -0.06 & -0.46 & 0.01 & -0.40 & 0.17 & 0.69 & -0.23 \\ -0.34 & 0.02 & -0.02 & 0.07 & 0.31 & 0.03 & 0.09 & -0.48 \\ 0.09 & 0.02 & 0.23 & -0.17 & -0.06 & 0.22 & -0.60 & 0.01 \\ 0.26 & 0.03 & -0.63 & -0.06 & 0.45 & 0.02 & 0.21 & 0.05 \\ 0.26 & 0.00 & -0.16 & 0.09 & 0.05 & -0.12 & -0.20 & -0.06 \\ -0.33 & -0.01 & 0.34 & -0.05 & -0.17 & 0.05 & 0.10 & 0.13 \end{bmatrix}$$



1.5 Codificação por Transformada

Atribuição de bits

Q? - Devemos distribuir o número de bits uniformemente pelos N coeficientes ?

R! - Devemos atribuir mais bits aos parâmetros com maior variância.

Condição

Devemos distribuir o número de bits de acordo com

$$R = \frac{1}{N} \sum_{k=0}^{N-1} R_k = \text{constante}$$



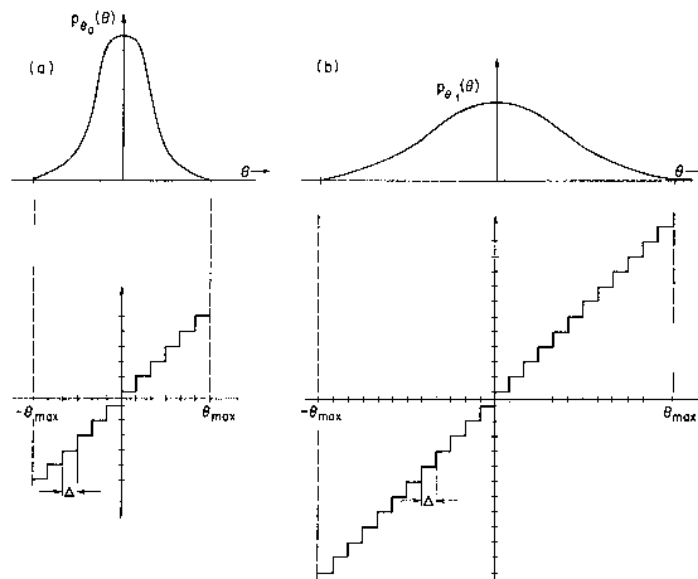
1.5 Codificação por Transformada

Atribuição de bits

Os intervalos de quantificação são iguais para todos os coeficientes

$$R_k = R + \frac{1}{2} \log_2 \frac{\sigma_k^2}{\left[\prod_{j=0}^{N-1} \sigma_j^2 \right]^{\frac{1}{N}}}$$

em que $\sigma_k^2, k = 0, 1, \dots, N-1$ Definem as variâncias dos coeficientes da transformação





1.5 Codificação por Transformada

Em vez de se transmitir todos os N coeficientes poderemos transmitir somente P coeficientes menor que N

$$\hat{x} = \sum_{k=0}^{P-1} \hat{\theta}(k) b_k$$

Vantagem - Diminui o débito binário

Desvantagem - Aumenta o erro de quantificação

$$e = x - \hat{x} = e_q + \sum_{k=P}^{N-1} \theta(k) b_k$$

A escolha dos coeficientes a enviar poderá ser realizada com base nas suas amplitudes

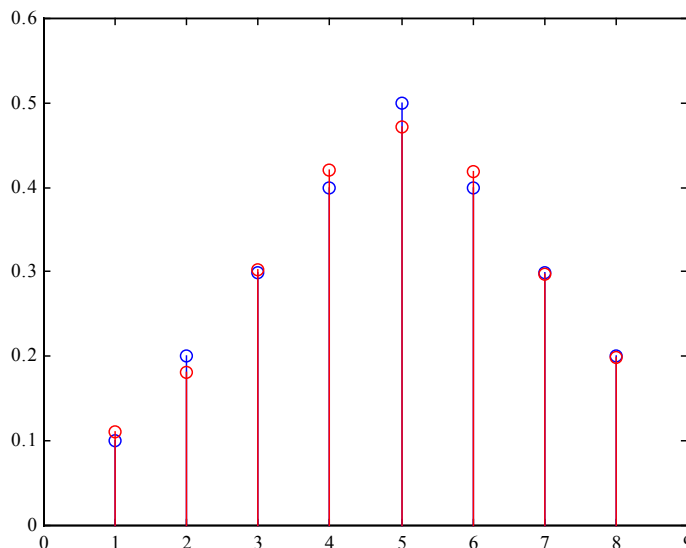


Considerando somente os 4 primeiros coeficientes (menor frequência).

$$e^2 = \sum_{i=0}^{N-1} (x[n] - \hat{x}[n])^2$$

$$e^2 = 0.0021$$

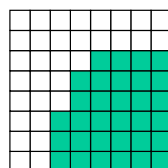
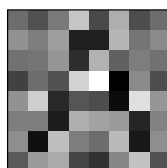
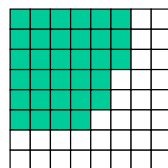
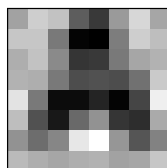
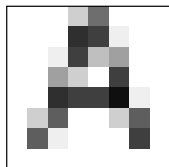
$x[n]$
 $\hat{x}[n]$





1.5 Codificação por Transformada

DCT Bidimensional



Imagens reconstruídas com 1/2 dos
coeficientes

