Source Coding for Compression

Types of data compression:

1. **Lossless** - removes redundancies (reversible)
2. **Lossy** - removes less important information (irreversible)

Lossless “Entropy” Coding, e.g. “Huffman” Coding

Example – 4 possible messages, 2 bits uniquely specifies each

$$\left.\begin{array}{lllll} \text{if} & p(A) & = & 0.5 & \triangleq & \text{“0”} \\ & p(B) & = & 0.25 & \triangleq & 1\quad 1 \\ & p(C) & = & 0.125 & \triangleq & 1\quad 0\quad 0 \\ & p(D) & = & 0.125 & \triangleq & 1\quad 0\quad 1 \end{array}\right\}\ \text{forms “comma-less” code}$$

Example of comma-less message:

$\underline{11}\ \ \underline{0}\ \ \underline{0}\ \ \underline{101}\ \ \underline{0}\ \ \underline{0}\ \ \underline{0}\ \ \underline{11}\ \ \underline{100}\ \leftarrow$ codeword grouping (unique)

Then the average number of bits per message (“rate”)

$$\langle R\rangle = 0.5\times 1\text{bit} + 0.25\times 2 + 0.25\times 3 = 1.75 \text{ bits/message} \geq H$$

Define “entropy” $H = -\sum_i p_i \log_2 p_i$ of a message ensemble

Define coding efficiency (here H = 1.75) $\eta_c \triangleq H/\langle R\rangle \leq 1$

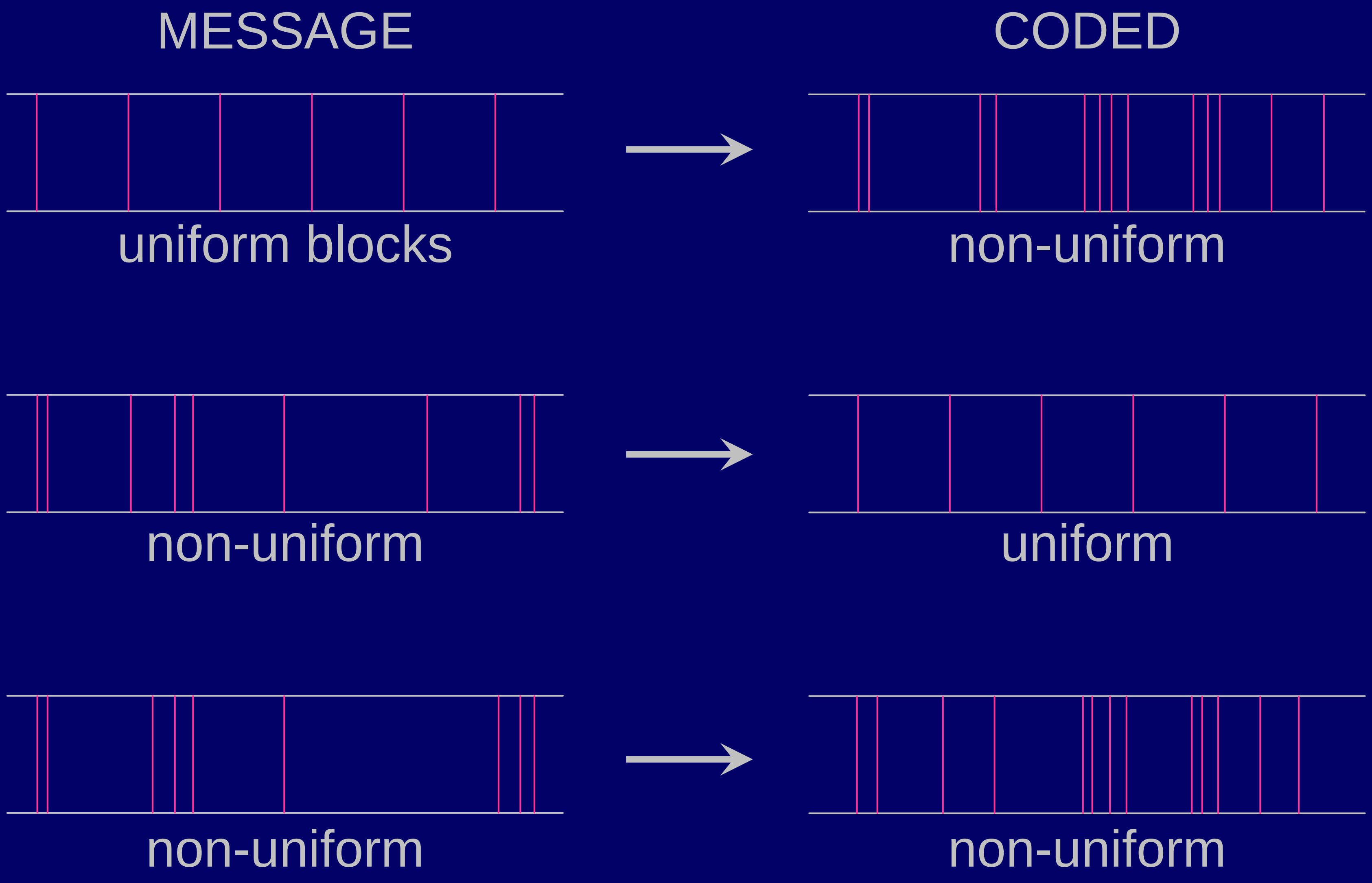
Simple Huffman Example

Simple binary message with p{0} = p{1} = 0.5;
each bit is independent, then entropy (H):

$$H = -2(0.5\log_2 0.5) = 1\,\text{bit/symbol} = R$$

If there is no predictability to the message, then there is no opportunity for further lossless compression

Lossless Coding Format Options



MESSAGE

CODED

uniform blocks

non-uniform

non-uniform

uniform

non-uniform

non-uniform

Run-length Coding

Example:

$$\underbrace{\ldots 1\ \underbrace{000}_{3}\ \underbrace{1111111}_{7}\ \underbrace{00}_{2}\ \underbrace{1111}_{4}\ \underbrace{000000000}_{9}\ 11\ldots}_{\text{Huffman code in this sequence } n_i\ (\ldots 3,\ 7,\ 2,\ 4,\ 9,\ \ldots)}$$

← run-length coding

Huffman code in this sequence n_i (…3, 7, 2, 4, 9, …) ← optional

Note: opportunity for compression here comes from tendancy for long runs of 0’s or 1’s

Simple: $p\{n_i\,;\ i = 1, \infty\} \Rightarrow$ code

Better: $p\{n_i | n_{i-1}\,;\ i = 1, ..\infty\} \Rightarrow$ code

↓

If n_i correlated with n_{i-1}

Other Popular Codes

Arithmetic codes:

(e.g. see Feb. ’89, *IEEE Trans. Comm.*, **37**, 2, pp. 93-97)

One of the best entropy codes for it adapts well to the message, but it involves some computation in real time.

Lempel-Ziv-Welch (LZW) Codes: Deterministically compress digitial streams adaptively, reversibly, and efficiently

Information-Lossy Source Codes

Common approaches to lossy coding:

1) Quantization of analog signals
2) Transform signal blocks; quantize the transform coefficients
3) Differential coding: code only derivatives or changes most of the time; periodically reset absolute value
4) In general, reduce redundancy and use predictability; communicate only unpredictable parts, assuming prior message was received correctly
5) Omit signal elements less visible or useful to recipient

Transform Codes - DFT

Discrete Fourier Transform (DFT):

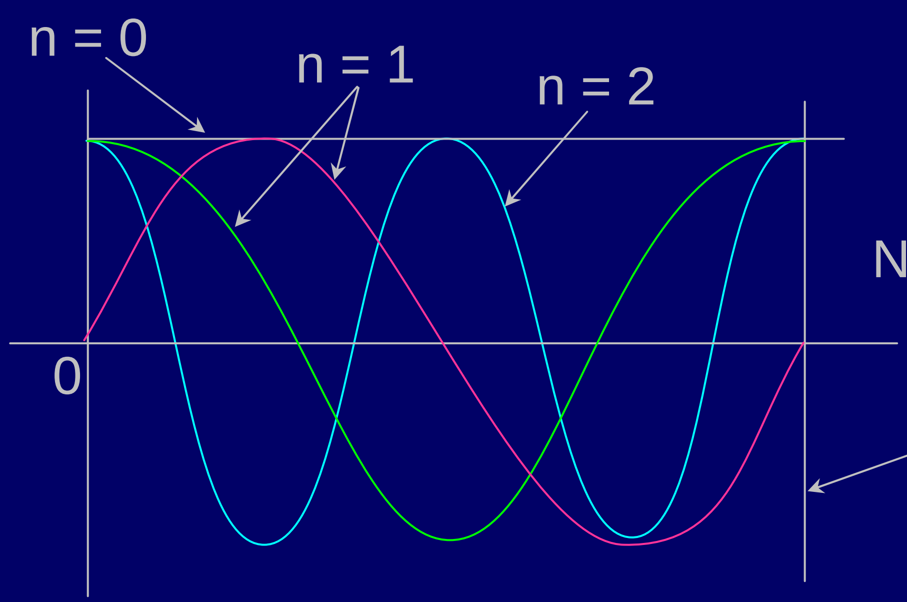
e.g

$$\underline{X}(n) = \sum_{k=0}^{N-1} x(k)e^{-jn2\pi(k/N)}$$

[n = 0, 1, …, N – 1]

$$x(k) = \frac{1}{N}\sum_{n=0}^{N-1} \underline{X}(n)e^{jn2\pi(k/N)}$$

Inverse DFT “IDFT”

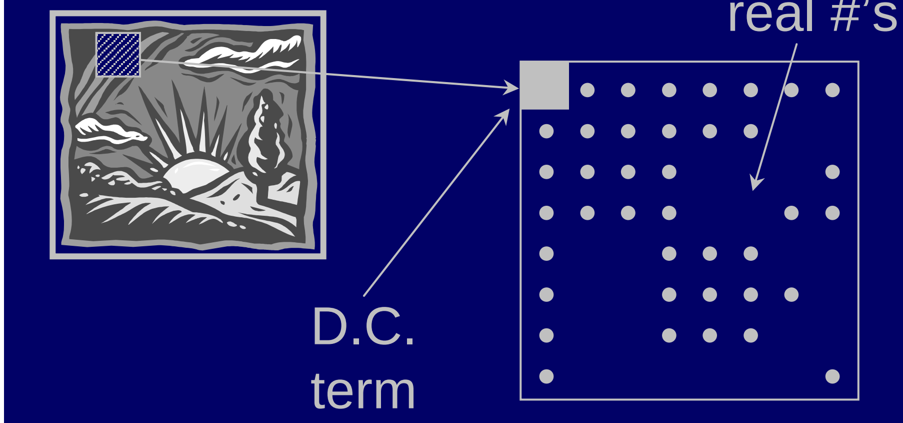


Note: $\underline{X}$(n) is complex ↔ 2N #’s

sharp edges of window ⇒ “ringing” or “sidelobes in the reconstructed decoded signal

Example of DCT Image Coding

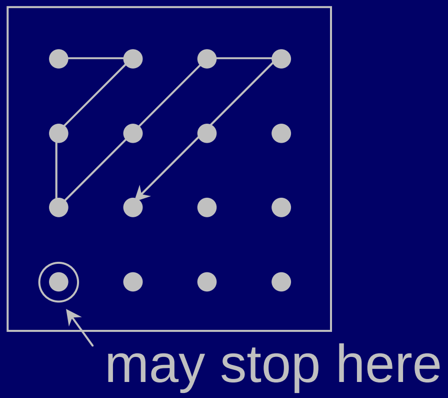
Say 8 × 8 block:



8 × 8 real #'s

D.C. term

Can sequence coefficients, stopping when they are too small, e.g.:



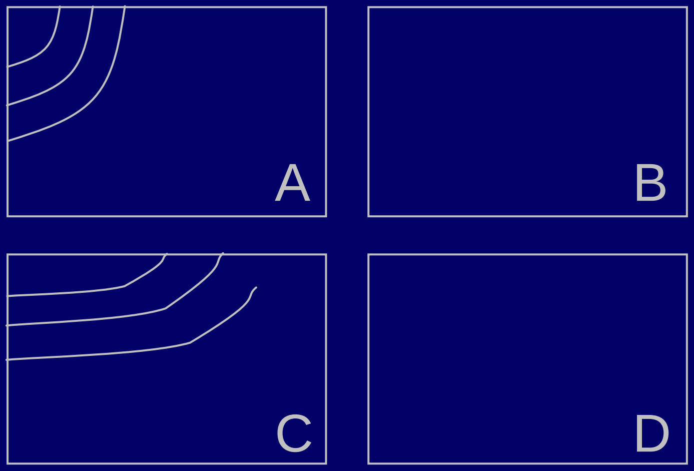
may stop here

Can classify blocks, and assign bits correspondingly

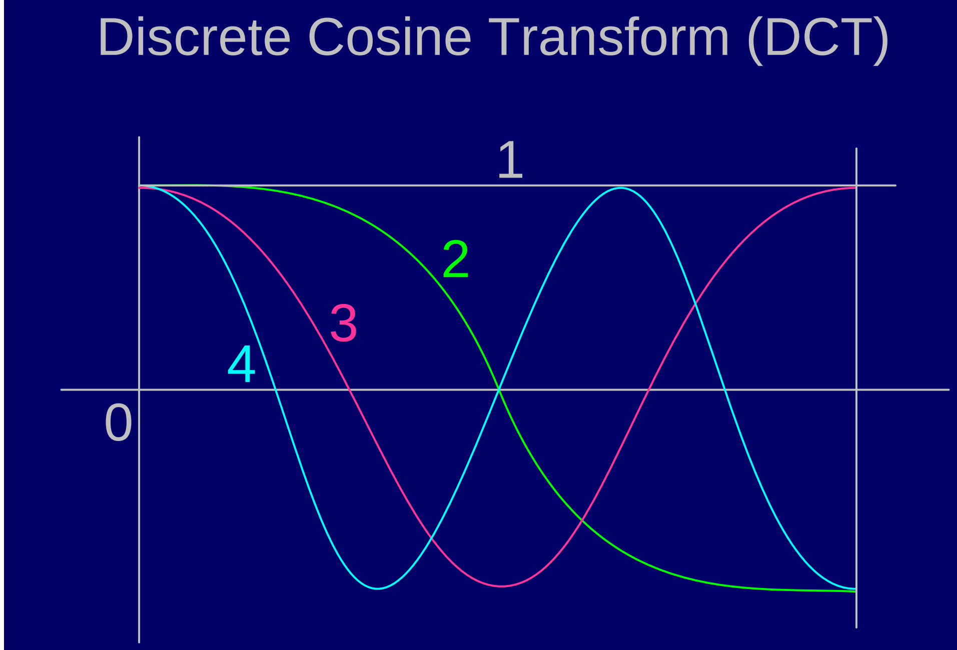
Image Types:

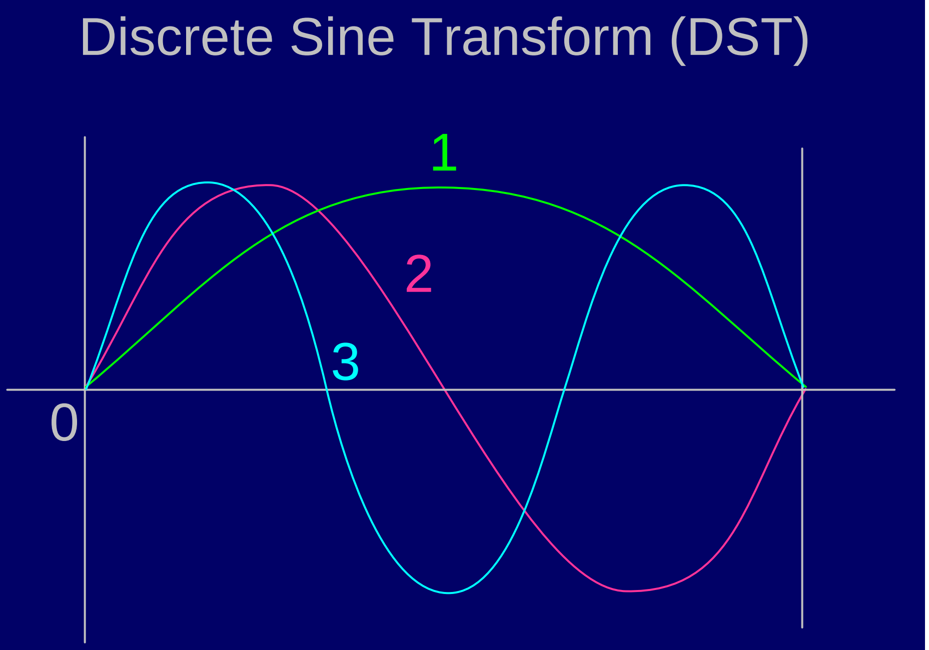
A. Smooth image
B. Horizontal striations
C. Vertical striations
D. Diagonals (utilize correlations)

Contours of typical DCT coefficient magnitudes



Discrete Cosine and Sine Transforms (DCT and DST)

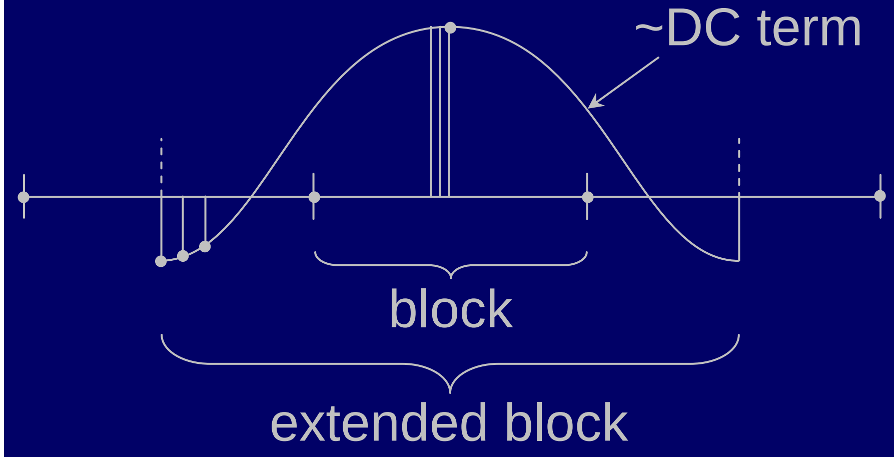




The DC basis function (n = 1) is an advantage, but the step functions at the ends produce artifacts at block boundaries of reconstructions unless n → ∞

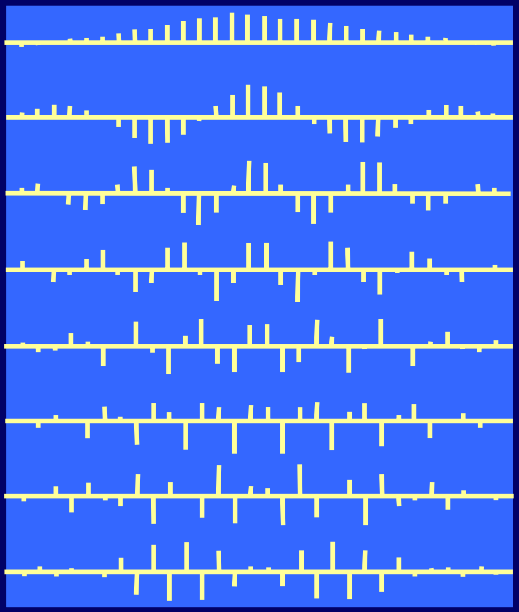
Lack of a DC term is a disadvantage, but zeros at end often overlap

Lapped Transforms

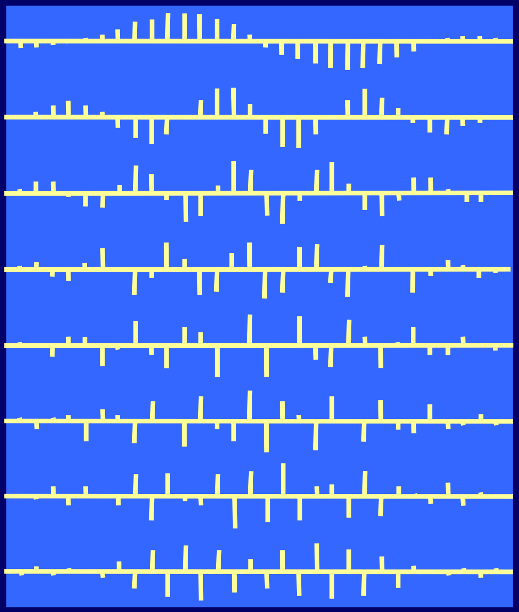


Lapped Orthogonal Transform = (LOT)
(1:1, invertible; orthogonal between blocks)

Reconstructions ring less, but ring outside the quantized block



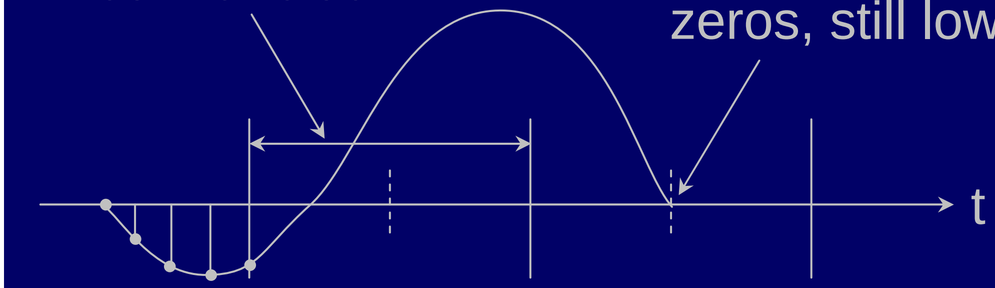
(a) Even Basis Functions



(b) Odd Basis Functions

An optimal LOT for N =16, L = 32, and ρ = 0.95

Lapped Transforms



First basis function for MLT

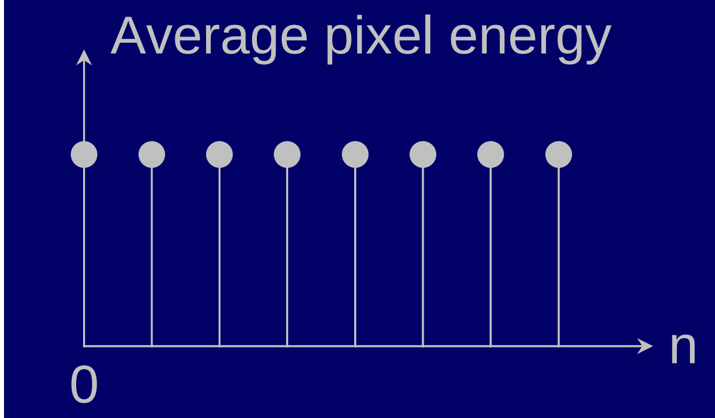
MLT = Modulated Lapped Transform

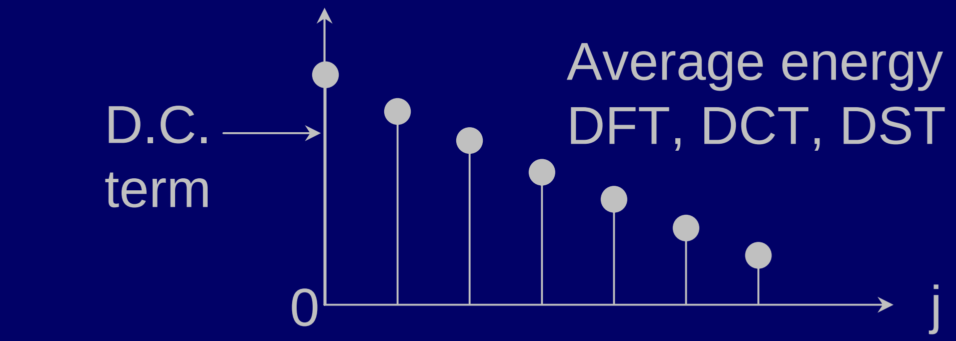
Ref: Henrique S. Malvar and D.H. Staelin, “The LOT: Transform Coding Without Blocking Effects,” *IEEE Trans. on Acous., Speech, and Sign. Proc.*, **37**(4), (1989).

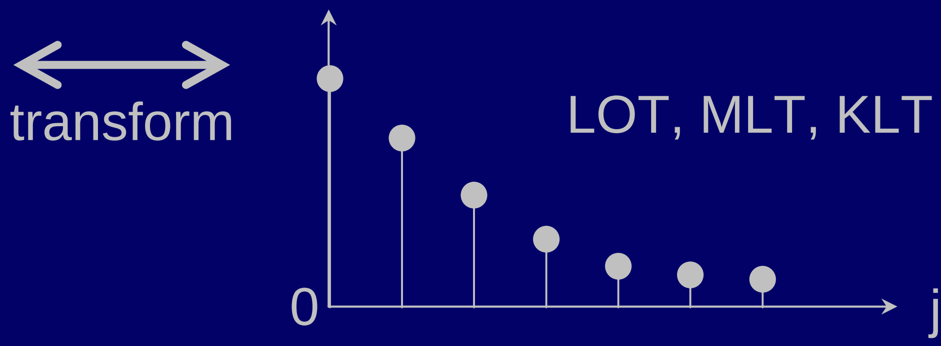
Karhounen-Loéve Transform (KLT)

Maximizes energy compaction within blocks for jointly gaussian processes

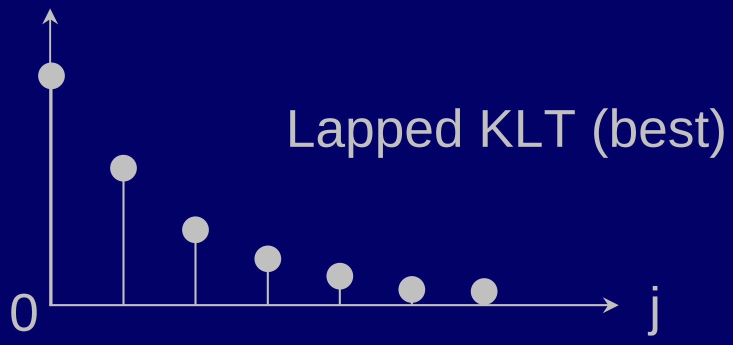
Example:



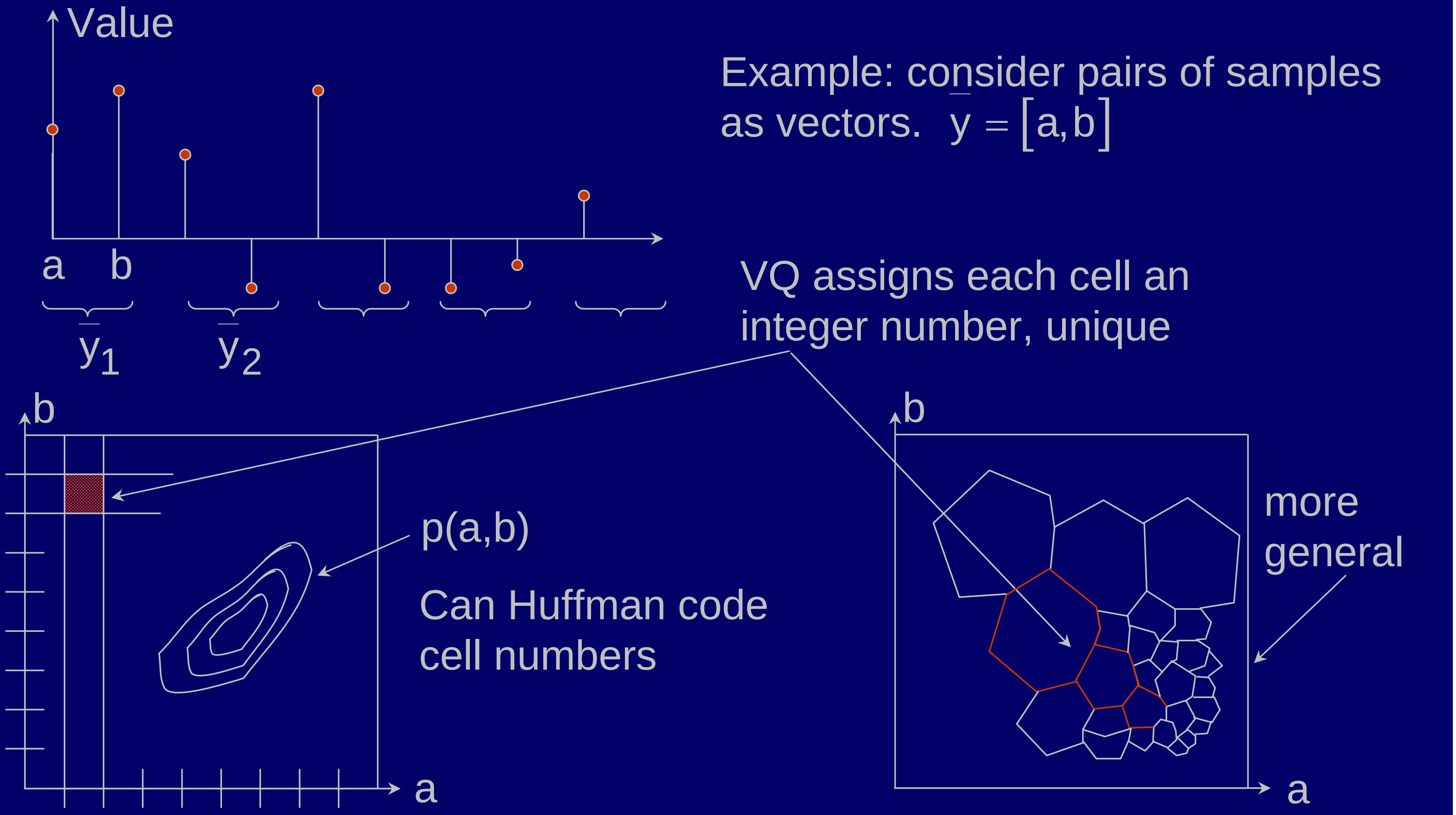




Note: The KLT for a first order Markov process is the DCT



Vector Quntization (“VQ”)



Example: consider pairs of samples as vectors. $\overline{y} = [a, b]$

VQ assigns each cell an integer number, unique

Can Huffman code cell numbers

more general

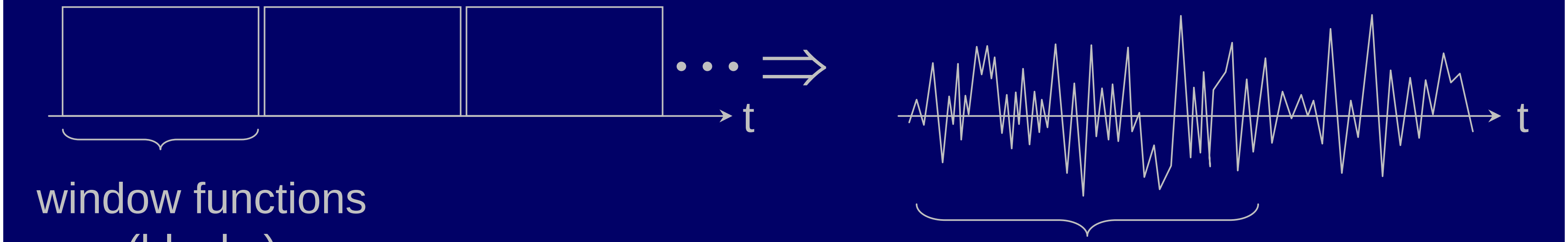
VQ is better because more probable cells are smaller and well packed.

VQ is n-dimensional (n = 4 to 16 is typical). There is a tradeoff between performance and computation cost

Reconstruction Errors

When such “block transforms” are truncated (high frequency terms omitted) or quantized, their reconstructions tend to ring

The reconstruction error is the superposition of the truncated (omitted or imperfectly quantized) sinusoids.



window functions
(blocks)
[original f(t)]

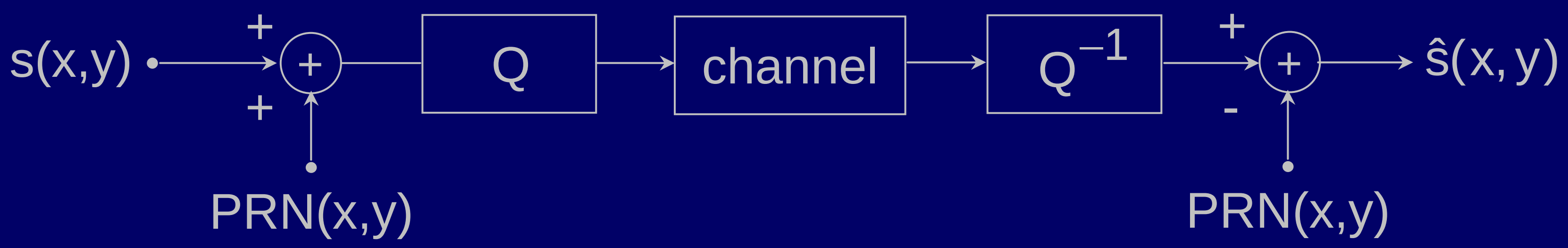
Reconstructed signal from truncated coefficients and quantization errors

Ringing and block-edge errors can be reduced by using orthogonal overlapping tapered transforms (e.g., LOT, ELT, MLT, etc.)

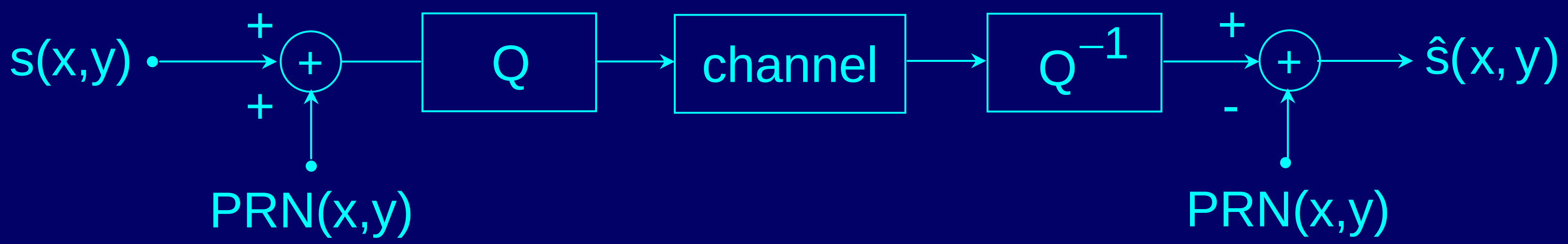
Smoothing with Pseudo-Random Noise (PRN)

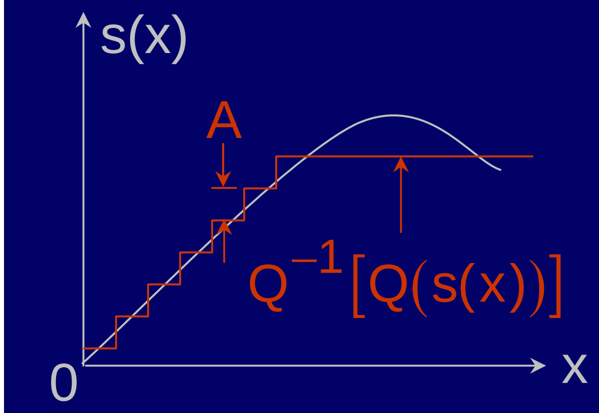
Problem: Coarsely quantized images are visually unacceptable

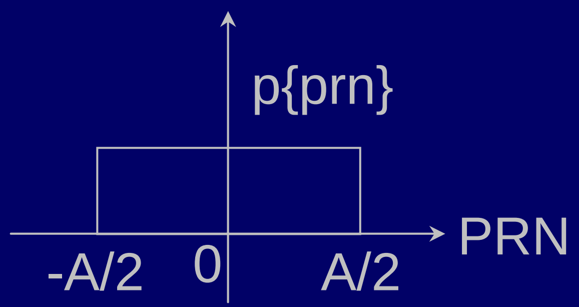
Solution: Add spatially white PRN to image before quantization, and subtract identical PRN from quantized reconstruction; result shows no quantization contours (zero!). PRN must be uniformly distributed, zero mean, with range equal to quantization interval.

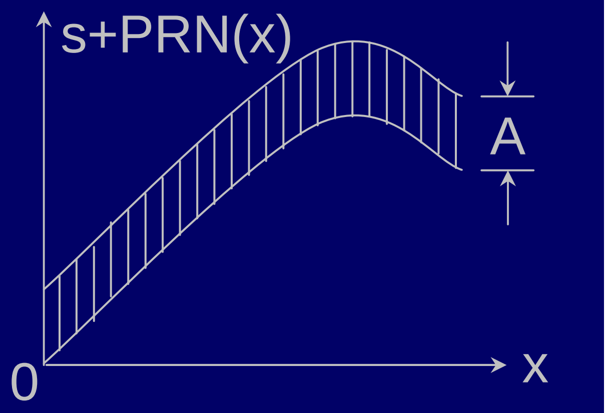


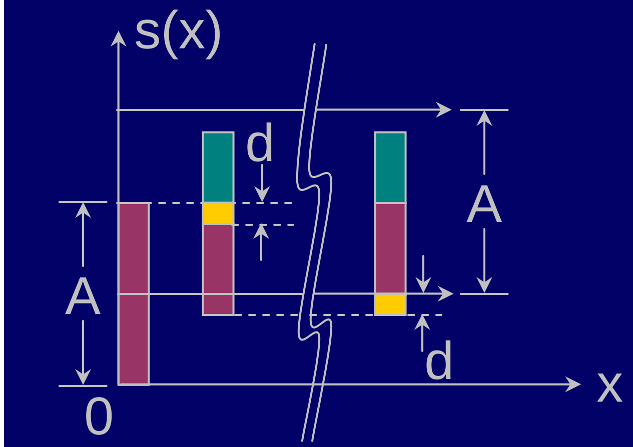
Smoothing with Pseudo-Random Noise (PRN)

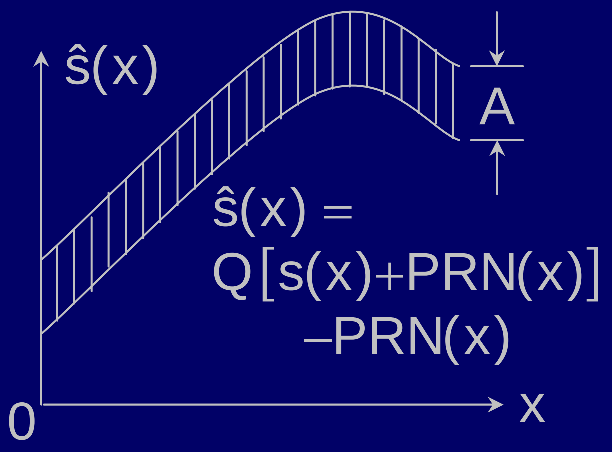


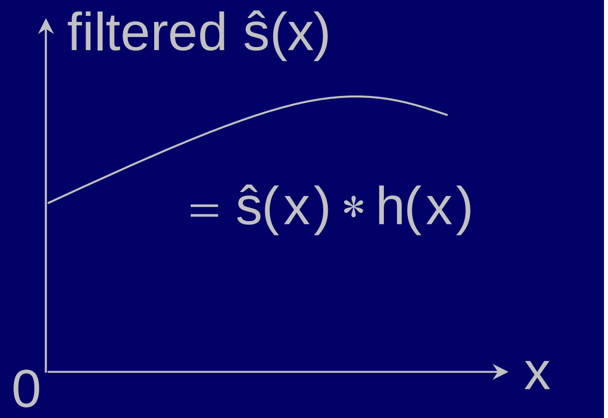




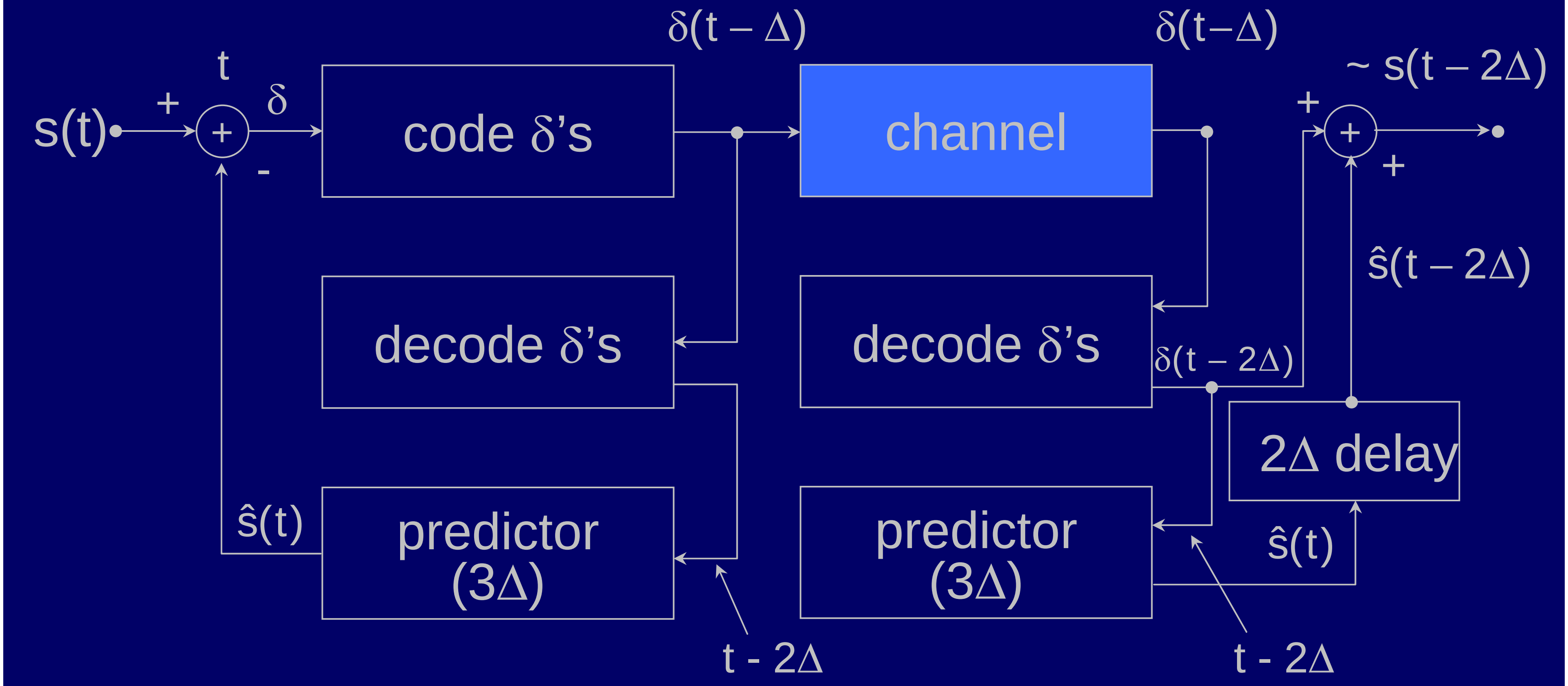








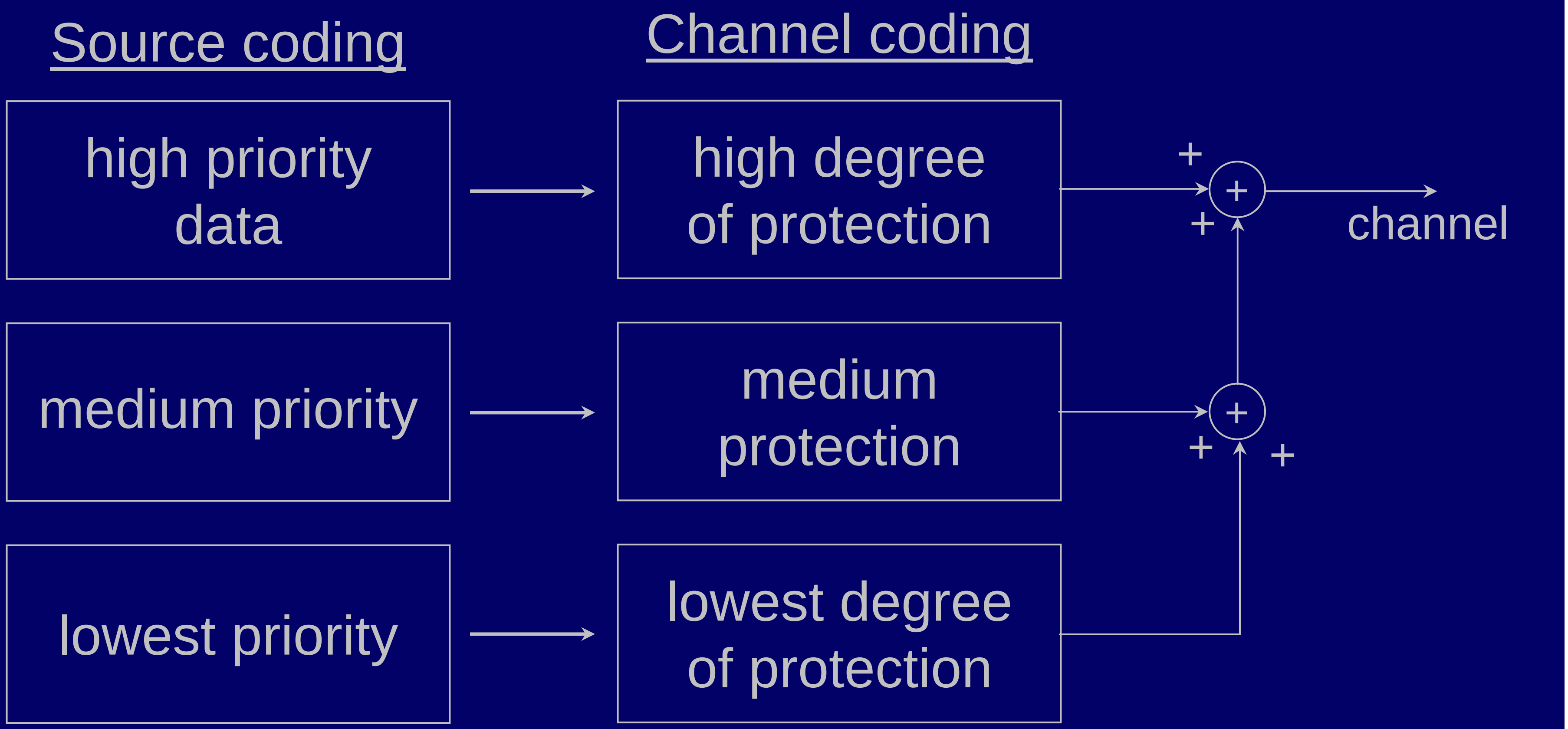
Example of Predictive Coding



Δ = computation time

The predictor can simply predict using derivatives, or can be very sophisticated, e.g. full image motion compensation.

Joint Source and Channel Coding



Source coding

Channel coding

high priority data

high degree of protection

medium priority

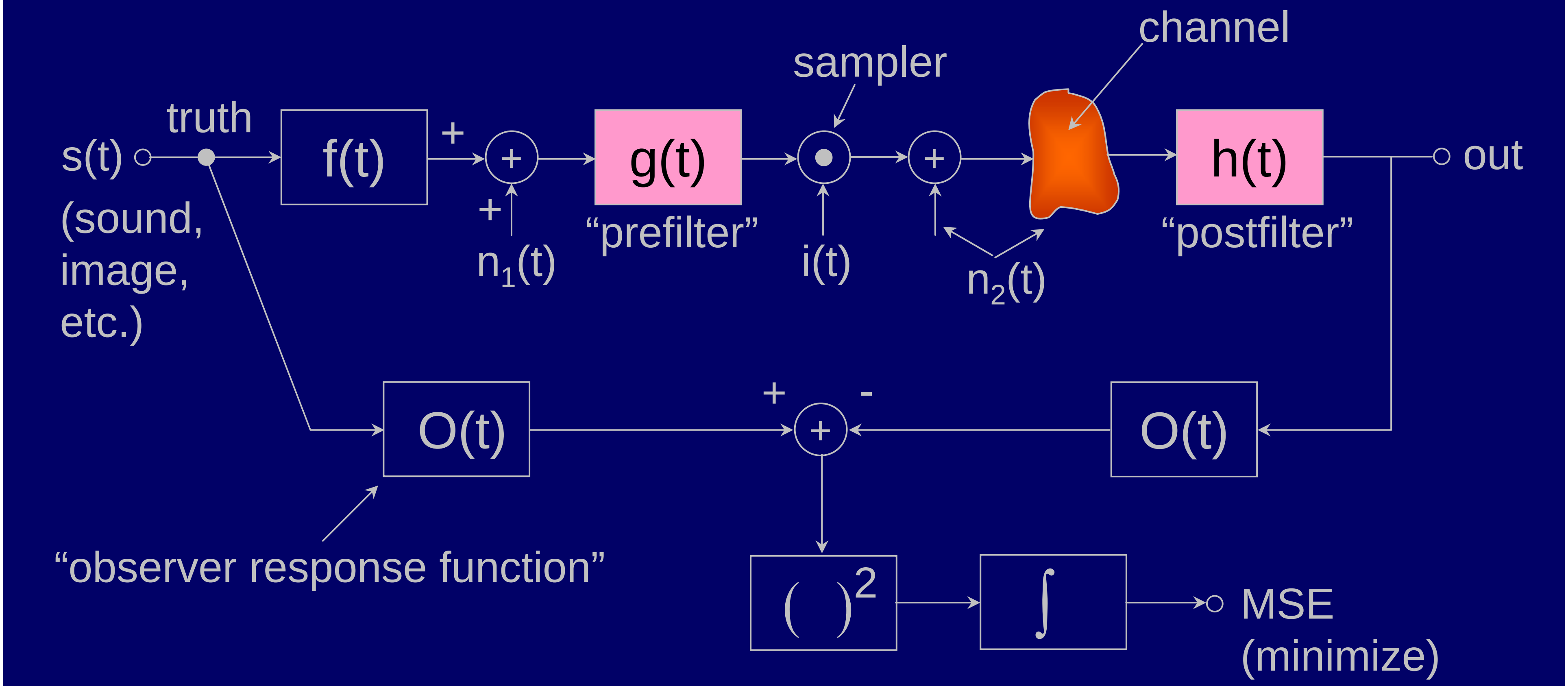
medium protection

lowest priority

lowest degree of protection

For example: lowest priority data may be highest spatial (or time) frequency components.

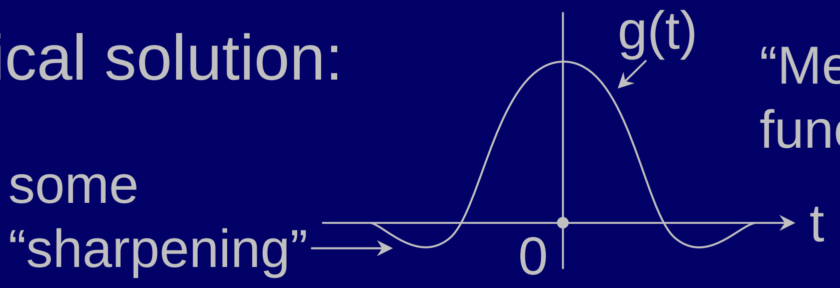
Prefiltering and Postfiltering Problem



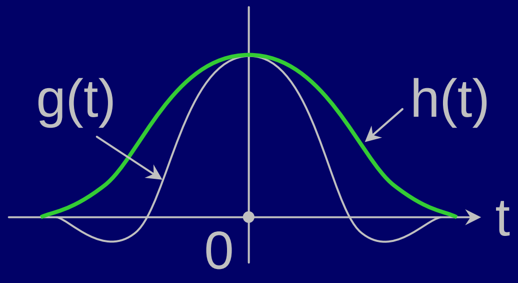
Given s(t), f(t), i(t), O(t), $n_1(t)$, and $n_2(t)$ [channel plus receiver plus quantization noise], choose g(t), h(t) to minimize MSE.

Prefiltering and Postfiltering

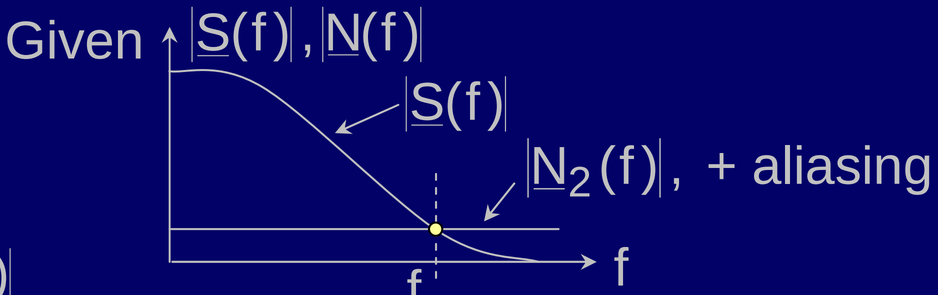
Typical solution:



"Mexican-hat function"

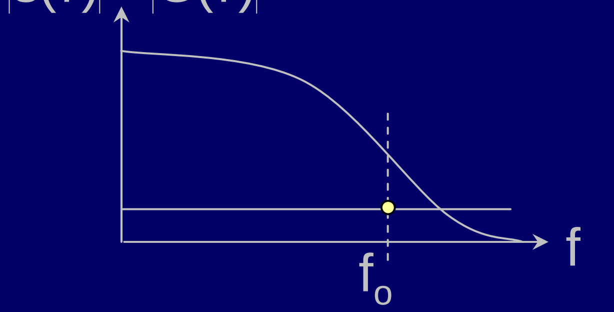


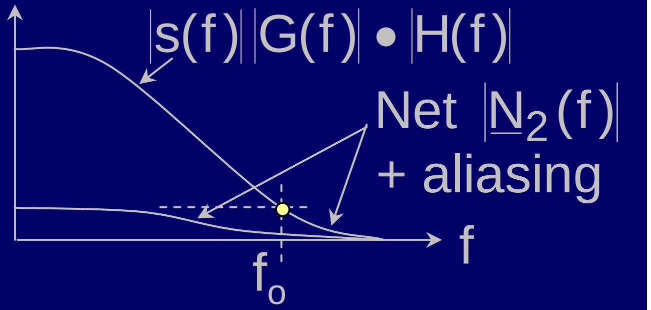
Interpretation:


Given $|\underline{S}(f)|, |\underline{N}(f)|$
$|\underline{S}(f)|$
$|\underline{N}_2(f)|$, + aliasing
f
f_o

Solution:

$|s(f)| \bullet |G(f)|$





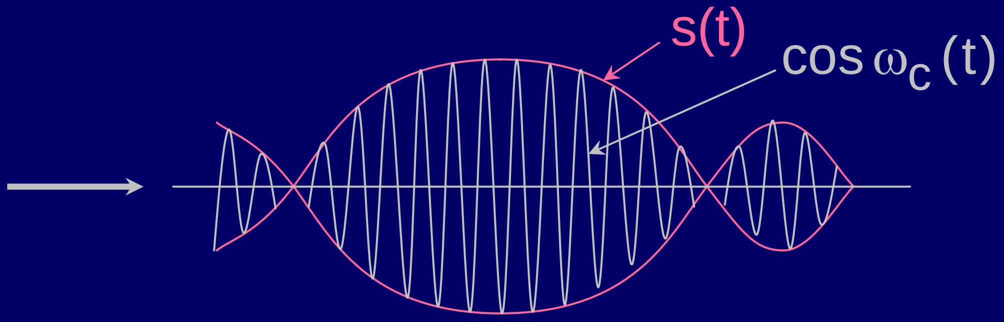
By boosting the weaker signals relative to the stronger ones prior to adding aliasing and $n_2(t)$, better weak-signal (high-frequency) performance follows. Prefilters and postfilters first boost and then attenuate weak signal frequencies.

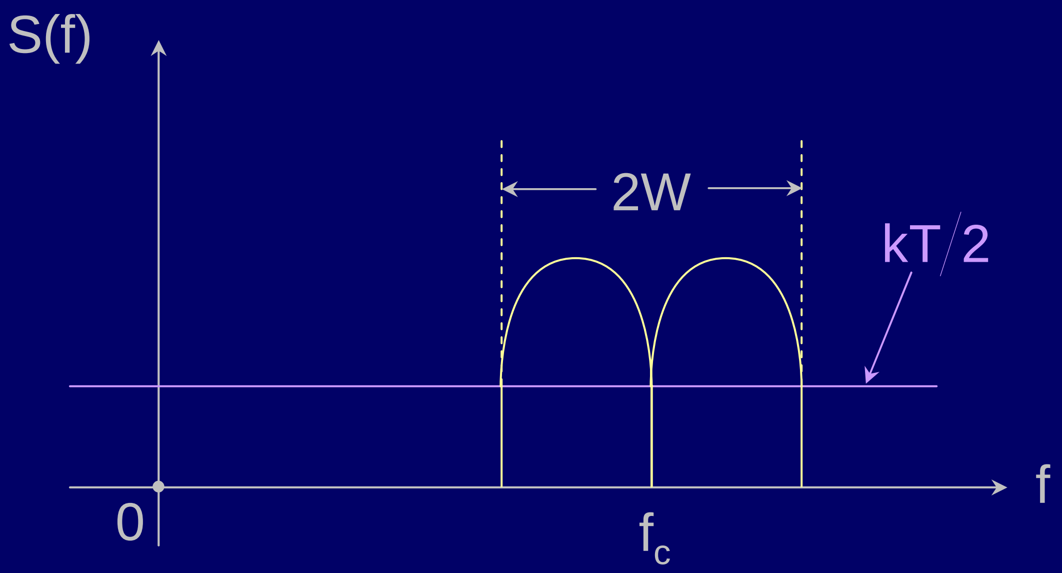
(Ref: H. Malvar, MIT EECS PhD thesis, 1987)

Analog Communications

Double Sideband Synchronous Carrier “DSBSC”:

$$\text{Received signal} = A_c s(t)\cos\omega_c t + n(t)$$

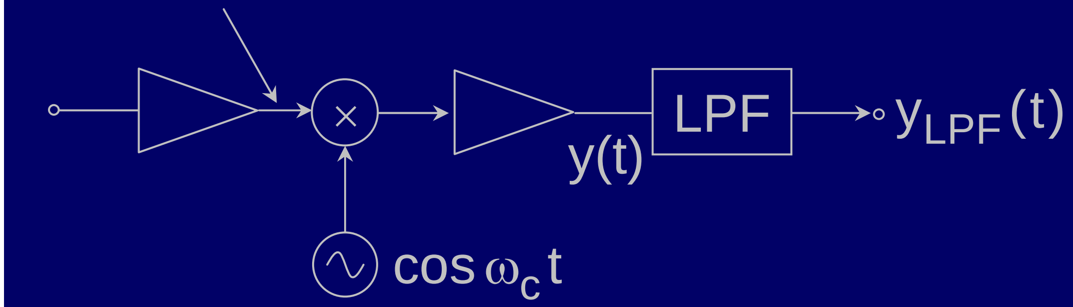


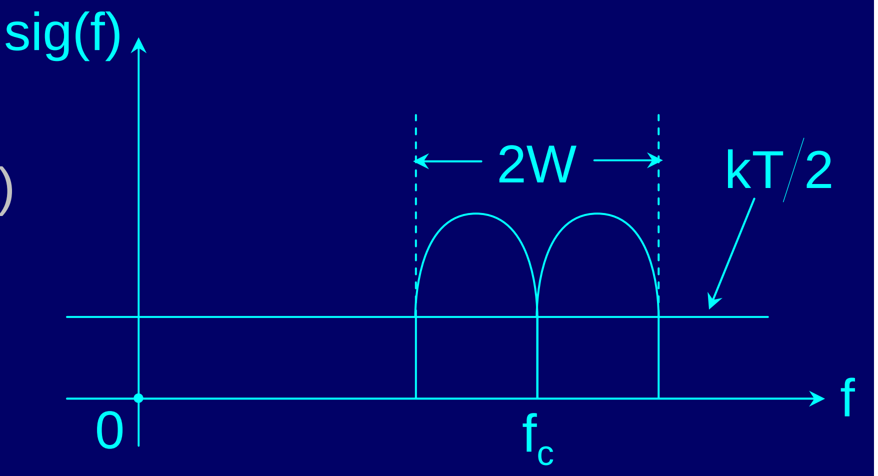


$$\left\langle n^2(t)\right\rangle = \underbrace{\frac{kT_R}{2}}_{\triangleq N_o} \bullet 4W$$

DSBSC Receiver

$[A_c s(t) + n_c(t)]\cos\omega_c t - n_s(t)\sin\omega_c t$





$$\mathrm{SNR}_{out} = ?$$

Let $n(t) \triangleq \underset{\substack{\uparrow \\ \text{slowly varying}}}{n_c}(t)\cos\omega_c t - \underset{\substack{\uparrow \\ \text{slowly varying}}}{n_s}(t)\sin\omega_c t$

So: $\overline{n^2(t)} = \left[\overline{n_c^2(t)} + \overline{n_s^2(t)}\right]\Big/2 = \overline{n_c^2} = \overline{n_s^2} = 2N_o 2W$

$$y(t) = [A_c s(t) + n_c(t)]\cos^2\omega_c t - \underbrace{n_s(t)(\sin\omega_c t)(\cos\omega_c t)}_{=\frac{n_s(t)}{2}\sin 2\omega_c t}$$

(filtered out by low-pass filter)

DSBSC Carrier

So: $\overline{n^2(t)} = \left[\overline{n_c^2(t)} + \overline{n_s^2(t)}\right]/2 = \overline{n_c^2} = \overline{n_s^2} = 2N_o 2W$

$$y(t) = [A_c s(t) + n_c(t)]\cos^2\omega_c t - \underbrace{n_s(t)\sin\omega_c t\cos\omega_c t}_{=\frac{n_s(t)}{2}\sin 2\omega_c t}$$

(filtered out by low-pass filter)

$$\cos^2\omega_c t = \frac{1}{2}(1 + \cos 2\omega_c t)$$

Therefore $y_{LPF}(t) = \frac{1}{2}[A_c s(t) + n_c(t)]$ (low-pass filtered)

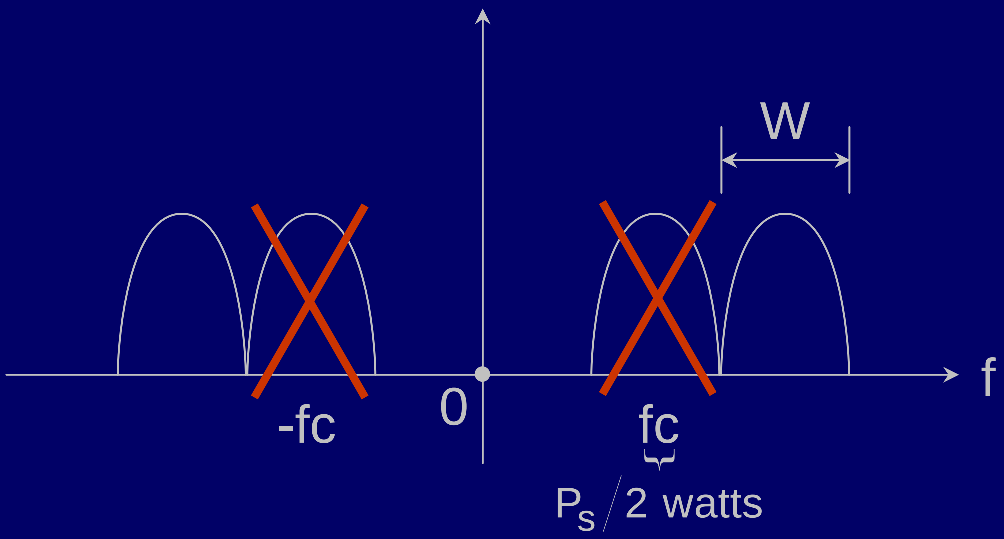
$$S_{out}/N_{out} = A_c^2 \underbrace{\overline{s^2(t)}}_{\text{let max = 1}} / \underbrace{\overline{n_c^2(t)}}_{4N_oW} = [P_c/2N_oW]\overline{s^2(t)}$$

(where carrier power $P_c = A_c^2/2$)

$\triangleq$ "CNR"$_{DSBSC}$ = "Carrier-to-Noise Ratio" (for $\overline{s^2} = 1$)

Single-sideband “SSB” Systems

(Synchronous carrier)



$$S_{out} \Big/ N_{out} = \frac{P_c\,\overline{s^2(t)}}{2N_o W}$$

Note: Both signal and noise are halved, so

$$S_{out} \Big/ N_{out_{SSBSC}} = S_{out} \Big/ N_{out_{DSBSC}}$$