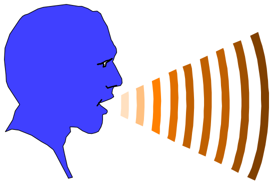
   

EE3414
Multimedia Communication Systems - I

Midterm Review

Yao Wang
Polytechnic University, Brooklyn, NY11201
http://eeweb.poly.edu/~yao

Frequency Domain Characterization of Signals

- Sinusoid signals:
  – Can determine the period, frequency, magnitude and phase of a sinusoid signal from a given formula or plot
- Fourier series for periodic signals
  – Understand the meaning of Fourier series representation
  – Can calculate the Fourier series coefficients for simple signals
  – Can sketch the line spectrum from the Fourier series coefficients
  – You only need to know the double sided Fourier series
- Fourier transform for non-periodic signals
  – Understand the meaning of the inverse Fourier transform
  – Can calculate the Fourier transform for simple signals
  – Can sketch the spectrum
  – Can determine the bandwidth of the signal from its spectrum
  – Know how to interpret a spectrogram plot

Fourier series for periodic signals

- Any periodic signal with fundamental frequency f_0 can be represented as a sum of many sinusoid signal with frequencies $f_k = k f_0$
  - Forward transform: determining coefficients for different frequency components
  - Inverse transform: reconstruct the original signal from the coefficients
  - The k-th coefficient (S_k) indicates the energy of the sinusoid with frequency $f_k = k f_0$
- You only need to know the double sided Fourier series representation

Inverse transform (Fourier synthesis):

$$s(t) = \sum_{k=-\infty}^{\infty} S_k \exp(j2\pi k f_0 t)$$

Forward transform (Fourier analysis):

$$S_k = \frac{1}{T_0} \int_0^{T_0} s(t) \exp(-j2\pi k f_0 t)\, dt; k = 0, \pm 1, 2, \ldots$$

For real signals, $S_k = S^*_{-k}$ **;** $|S_k| = |S_{-k}|$ **(Symmetric spectrum)**

Fourier series for periodic signals

- Given a simple periodic signal (in sketch or function form), you should be able to determine its fundamental period and frequency, compute its Fourier series coefficients, and plot the line spectrum
- Given a function consisting of sinusoidal signals only, you should be able to find the Fourier series coefficients by writing the function as sum of complex exponential signals.

$$\text{Euler formula:}$$
$$\exp(j\alpha) + \exp(-j\alpha) = 2\cos(\alpha);$$
$$\exp(j\alpha) - \exp(-j\alpha) = j2\sin(\alpha)$$

Fourier transform for aperiodic signals

- Any aperiodic signal can be represented as a sum of infinitely many complex exponential functions with frequency from 0 to \infty.
- Double sided Fourier transform:

$$\text{Fourier synthesis (inverse transform)}:$$

$$s(t) = \int_{-\infty}^{\infty} S(f)\exp(j2\pi ft)df$$

$$\text{Fourier analysis (forward transform)}:$$

$$\mathrm{S(f)} = \int_{-\infty}^{\infty} s(t)\exp(-j2\pi ft)dt$$

- Can compute the Fourier transform for simple functions (pulse, triangle, exponential) in sketch or function form
- Can plot the magnitude spectrum of the Fourier transform and understand its meaning

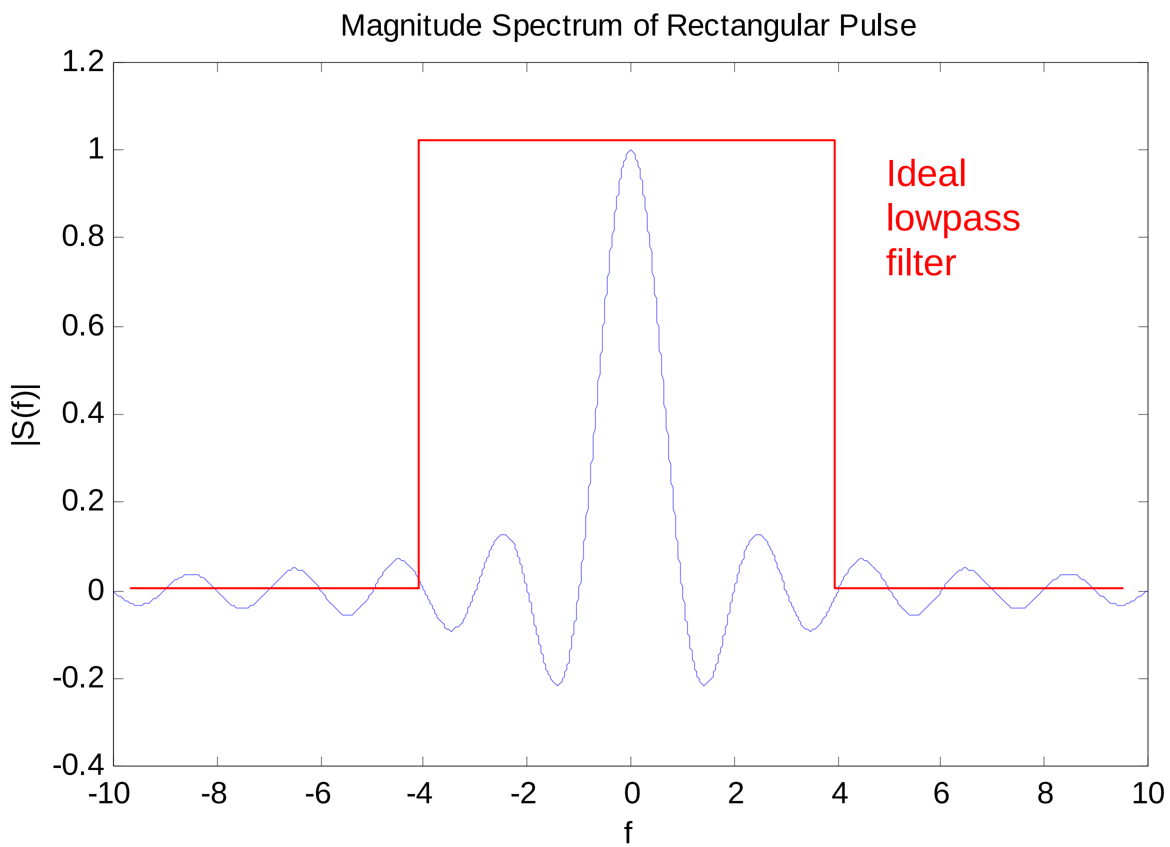
Frequency content of a signal

- Bandwidth: can estimate the bandwidth of a signal from its spectrum, given the threshold
- Can estimate the fundamental period and frequency of a periodic signal
- Can estimate the maximum frequency of a signal from its waveform
  - Find the shortest cycle in the waveform
- Understand the function of a filter from its frequency response (low-pass, high-pass, band-pass)

Speech & Music Signals

- Typical speech and music waveforms are semi-periodic
  - The fundamental period is called pitch period
  - The inverse of the pitch period is the fundamental frequency (f0)
  - Pure music notes are sinusoids with different frequencies
- Spectral content
  - Within each short segment, a speech or music signal can be decomposed into a pure sinusoidal component with frequency f0, and additional harmonic components with frequencies that are multiples of f0.
  - The maximum frequency is usually several times of the fundamental frequency
  - Speech has a frequency span up to 4 KHz
  - Audio has a much wider spectrum, up to 22KHz

Filtering in Frequency Domain



Filtering is done by a simple multiplication:

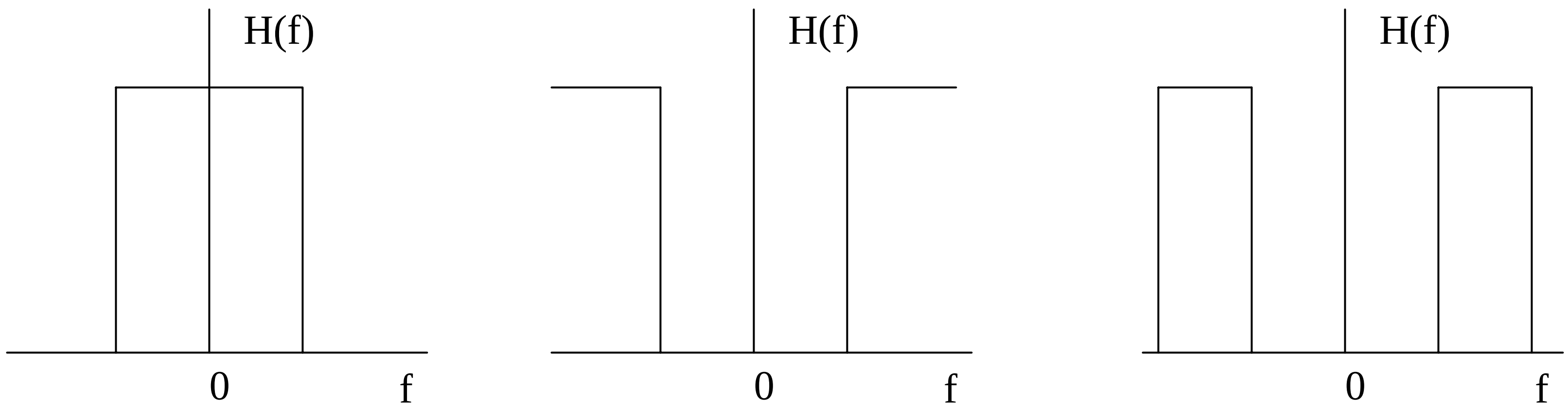
Y(f)= X(f) H(f)

H(f) is designed to magnify or reduce the magnitude (and possibly change phase) of the original signal at different frequencies.

A pulse signal after low pass filtering (left) will have rounded corners.

Typical Filters

- Lowpass -> smoothing, noise removal
- Highpass -> edge/transition detection
- Bandpass -> Retain only a certain frequency range

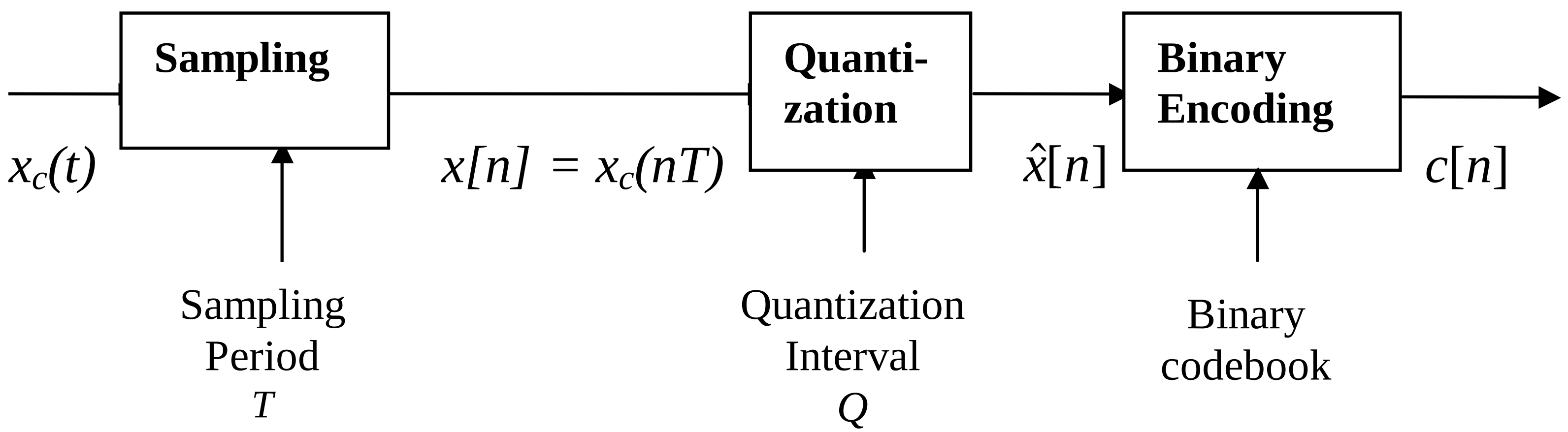


Implementation of Filtering

- Frequency Domain
  - FT -> Filtering by multiplication with H(f) -> Inverse FT
- Time Domain
  - Convolution using a filter h(t) (inverse FT of H(f))

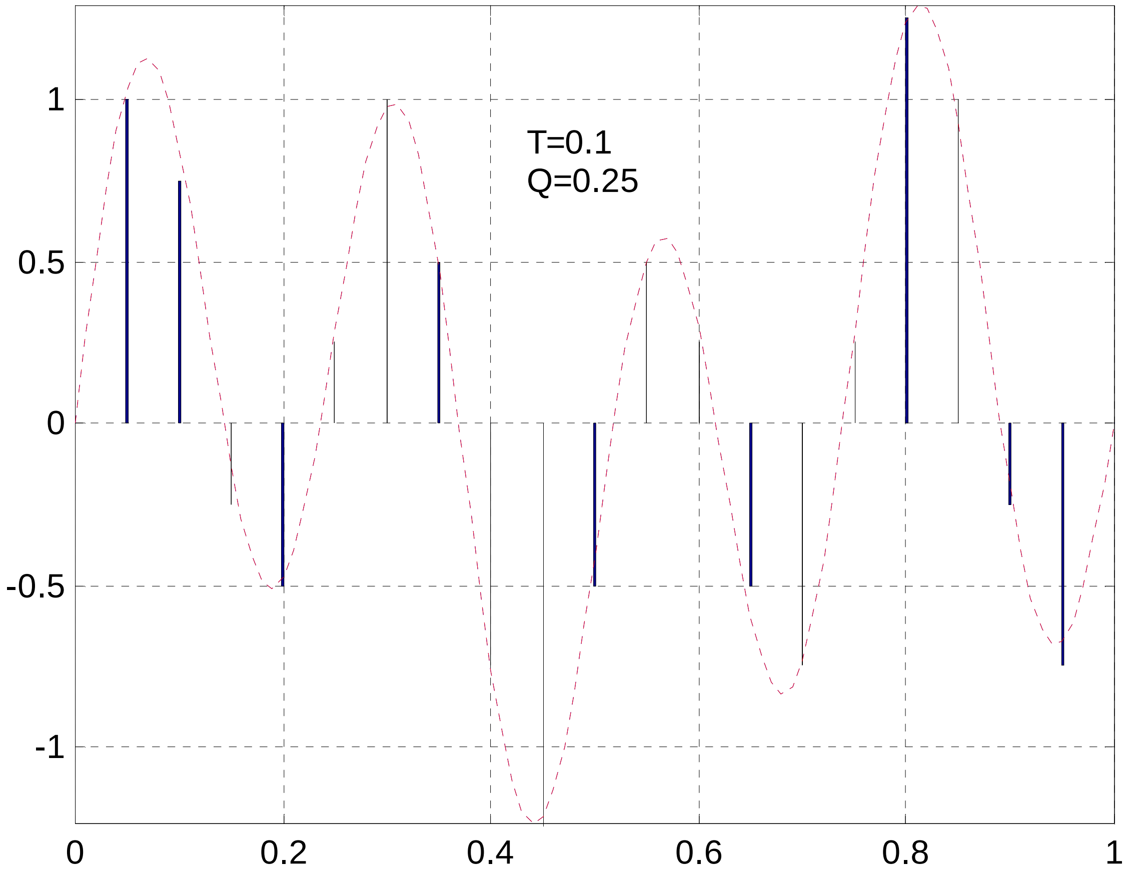
- You should understand how to perform filtering in frequency domain, given a filter specified in frequency domain
- Should know the function of the filter given H(f)
- Computation of convolution is not required in this course
- Filter design is not required.

Three Processes in A/D Conversion



- Sampling: take samples at time *nT*
  - T: sampling period;
  - f_s = 1/T: sampling frequency
- Quantization: map amplitude values into a set of discrete values *kQ*
  - Q: quantization interval or stepsize
- Binary Encoding
  - Convert each quantized value into a binary codeword

Analog to Digital Conversion

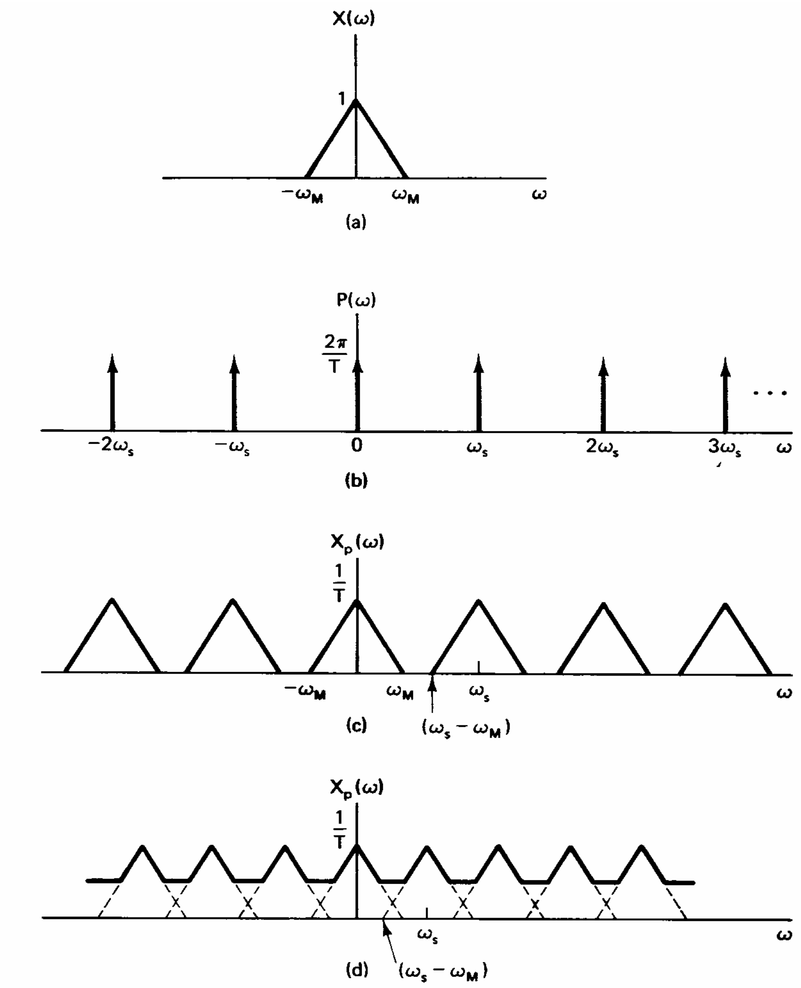


A2D_plot.m

Sampling

- Sampling:
  - Know the minimally required sampling rate (Nyquist sampling rate):
    - $f_s > 2 f_{max}$, $T_s < T_{0,min} /2$
    - Can estimate $T_{0,min}$ from signal waveform
  - Can illustrate samples on a waveform and observe whether the signal is under-sampled.
  - Can plot the spectrum of a sampled signal
    - The sampled signal spectrum contains the original spectrum and its replicas (aliases) at kf_s, k=+/- 1,2,….
    - Can determine whether the sampled signal suffers from aliasing
  - Understand why do we need a prefilter when sampling a signal
    - To avoid alising
    - The filter should be an ideal lowpass filter with cutoff frequency at $f_s /2$.
  - Can show the aliasing phenomenon when sampling a sinusoid signal using both temporal and frequency domain interpretation

Frequency Domain Interpretation of Sampling



Original signal

Sampling impulse train

Sampled signal $\omega_s > 2\ \omega_m$

Sampled signal $\omega_s < 2\ \omega_m$ (Aliasing effect)

The spectrum of the sampled signal includes the original spectrum and its aliases (copies) shifted to *$k\ f_s$, k=+/- 1,2,3,…* The reconstructed signal from samples has the frequency components upto *$f_s/2$.*

When $f_s < 2f_m$, aliasing occur.

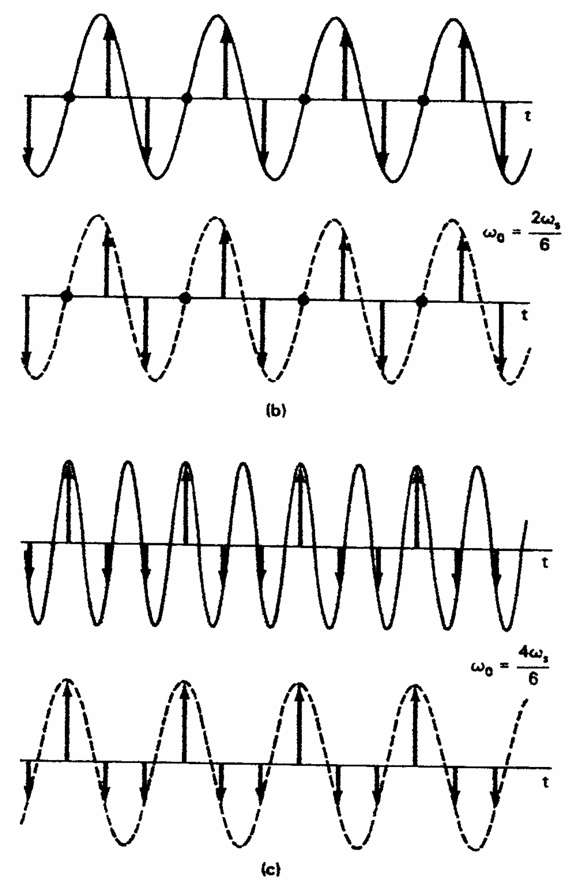
Sampling Sinusoid Signals: Temporal Domain Interpretation

Sampling above
Nyquist rate
$\omega_s = 3\omega_m > \omega_{s0}$

Reconstructed
=original

Sampling under
Nyquist rate
$\omega_s = 1.5\omega_m < \omega_{s0}$

Reconstructed
\= original



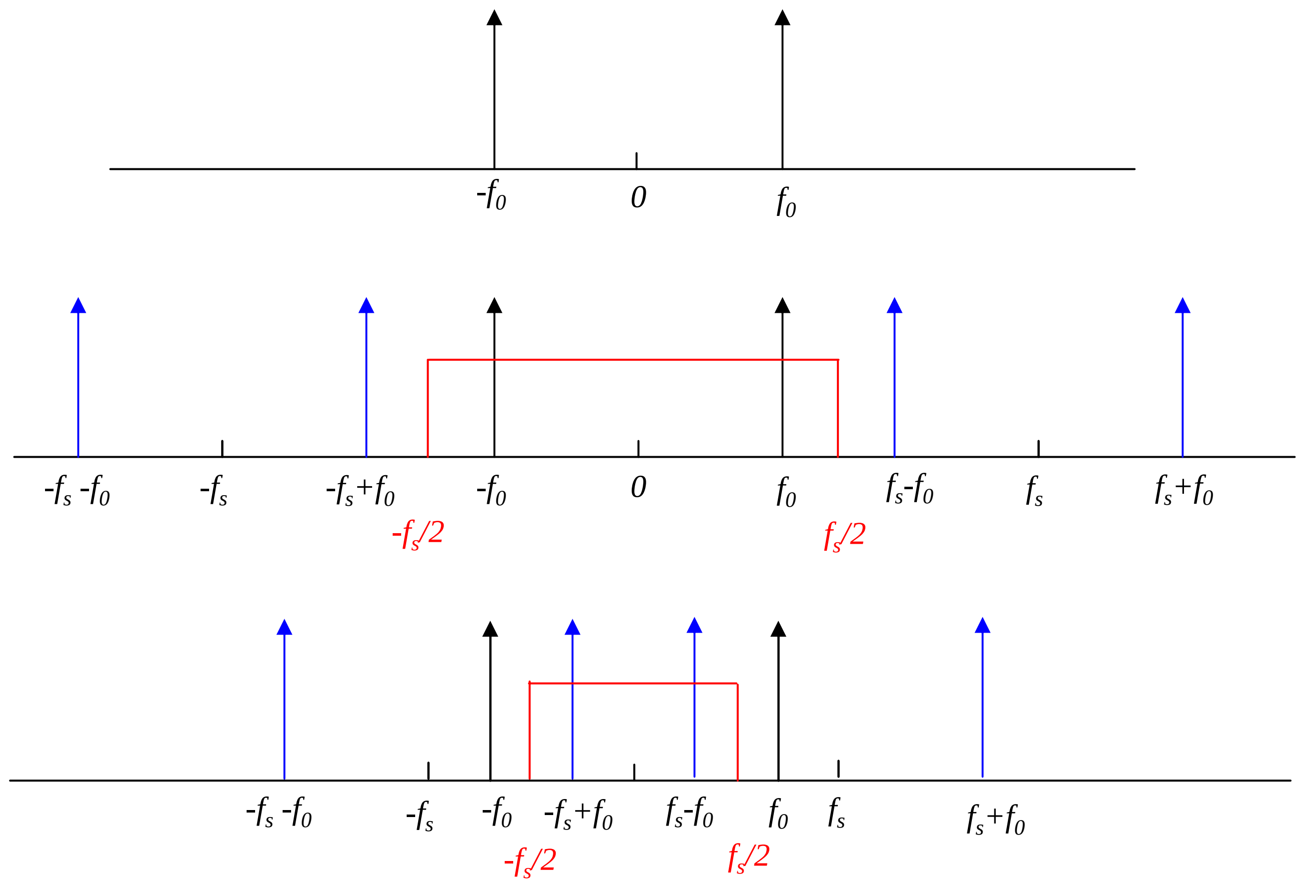
Aliasing: The reconstructed sinusoid has a lower frequency than the original!

Sampling of Sinusoid: Frequency Domain Interpretation

Spectrum of $cos(2\pi f_0 t)$

No aliasing
$f_s > 2f_0$
$f_s - f_0 > f_0$
Reconstructed signal: f_0

With aliasing
$f_s < 2f_0$
$f_s - f_0 < f_0$
Reconstructed signal: $f_s - f_0$

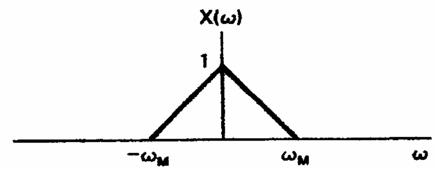


Interpolation

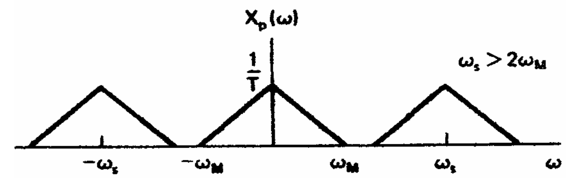
- Interpolation:
  - Can illustrate sample-and-hold and linear interpolation from samples.
  - Understand why the ideal interpolation filter is a lowpass filter with cutoff frequency at $f_s/2$.
  - Know the ideal interpolation kernel is the sinc function.

Frequency domain interpretation

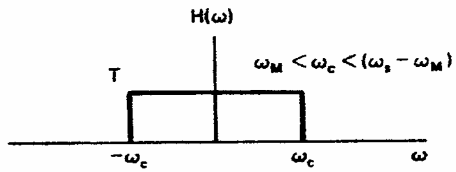
Original signal



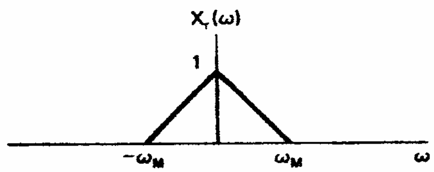
Sampled signal
$\omega_s > 2\,\omega_m$



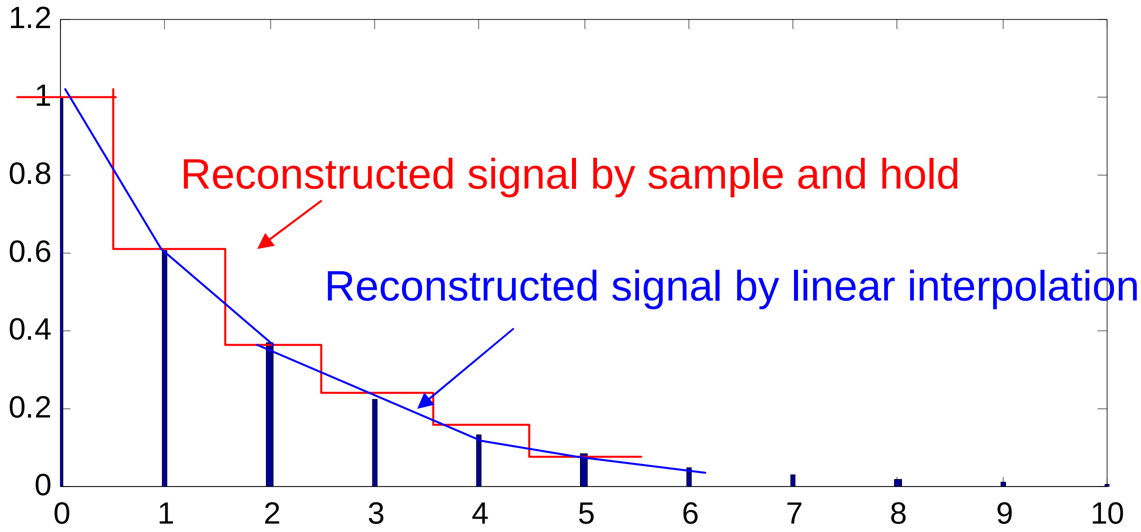
Ideal
reconstruction filter
(low-pass)



Reconstructed signal
(=Original signal)



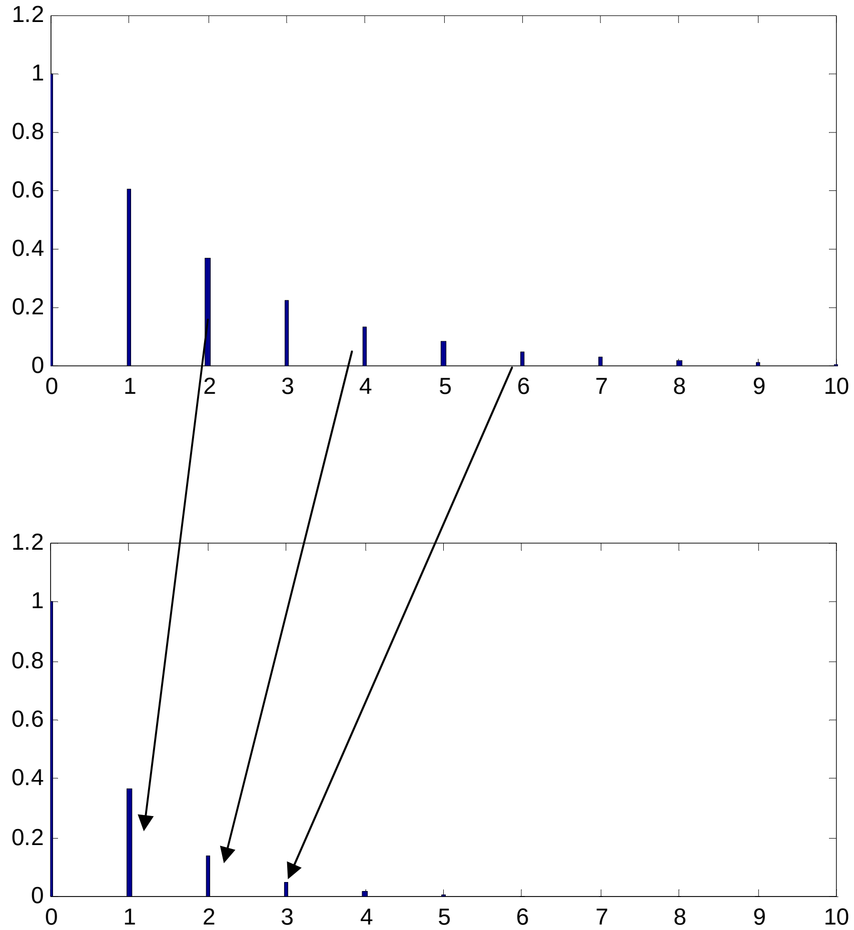
Sample-and-Hold vs. Linear Interpolation



Sampling Rate Conversion

- Know the meaning of down-sampling and upsampling
- Understand the need for prefiltering before down-sampling
  - To avoid aliasing
  - Know how to do factor of 2 down sampling without prefiltering and with averaging filter
- Can illustrate up-sampling by sample-and-hold and linear interpolation

Down-Sampling Illustration



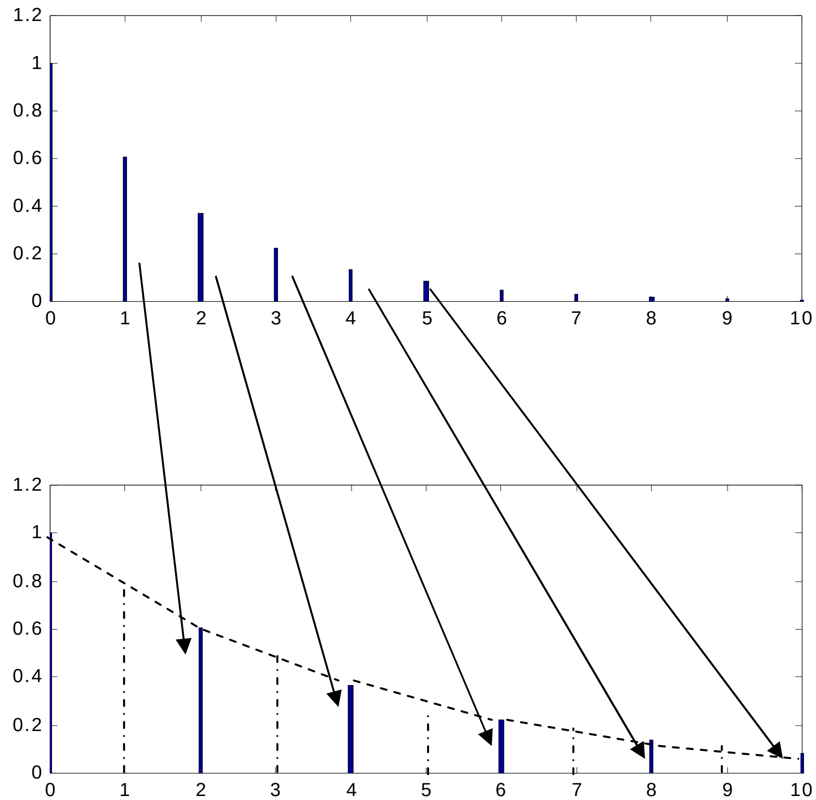
Down-sampling by a factor of 2 = take every other sample

To avoid aliasing of any high frequency content in the original signal, should smooth the original signal before down-sampling -- Prefiltering

Down-Sampling Example

- Given a sequence of numbers, down-sample by a factor of 2,
  - Original sequence: 1,3,4,7,8,9,13,15…
  - Without prefiltering, take every other sample:
    - 1,4,8,13,…
  - With 2-sample averaging filter
    - Filtered value=0.5*self+0. 5*right, filter h[n]=[0.5,0.5]
    - Resulting sequence:
      - 2, 5.5,8.5,14,…
  - With 3-sample weighted averaging filter
    - Filtered value=0.5*self+0.25*left+0.25*right, filter h[n]=[0.25,0.5,0.25]
    - Resulting sequence (assuming zeros for samples left of first):
      - 1.25, 4.5,8,12.5,…

Upsampling by linear interpolation



Missing samples need to be filled from neighboring available samples using interpolation filter

Up-Sample Example

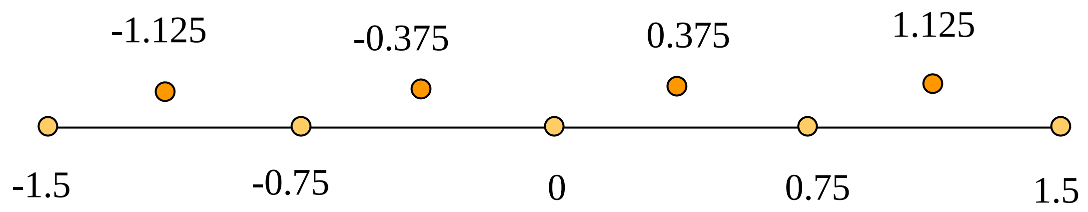
- Given a sequence of numbers, up-sample by a factor of 2
  - Original sequence: 1,3,4,7,8,9,13,15…
  - Zero-padding:
    - 1,0,3,0,4,0,7,0,…
  - Sample and hold
    - Repeat the left neighbor, filter h[n]=[1,1]
    - 1,1,3,3,4,4,7,7,…
  - With linear interpolation
    - New sample=0.5*left+0. 5*right, filter h[n]=[0.5,1,0.5]
    - Resulting sequence:
      - 1,2,3,3.5,4,5.5,7,8,…,…

Quantization

- Understand the general concept of quantization
- Can design a uniform quantizer given the signal range and the number of quantizer levels
- Can perform uniform quantization on a given sequence, generate the resulting quantized sequence and its binary representation
- Understand the principle of non-uniform quantization
  - Smaller intervals for more frequent signal values
  - Understand how the mu-law quantizer works
- Can represent quantized signals using binary codewords
- Can calculate bit rate given sampling rate and quantization levels

Uniform Quantization Example

- For the following sequence {1.2,-0.2,-0.5,0.4,0.89,1.3…}, Quantize it using a uniform quantizer in the range of (-1.5,1.5) with 4 levels, and write the quantized sequence.
- Solution: Q=3/4=0.75. Quantizer is illustrated below.



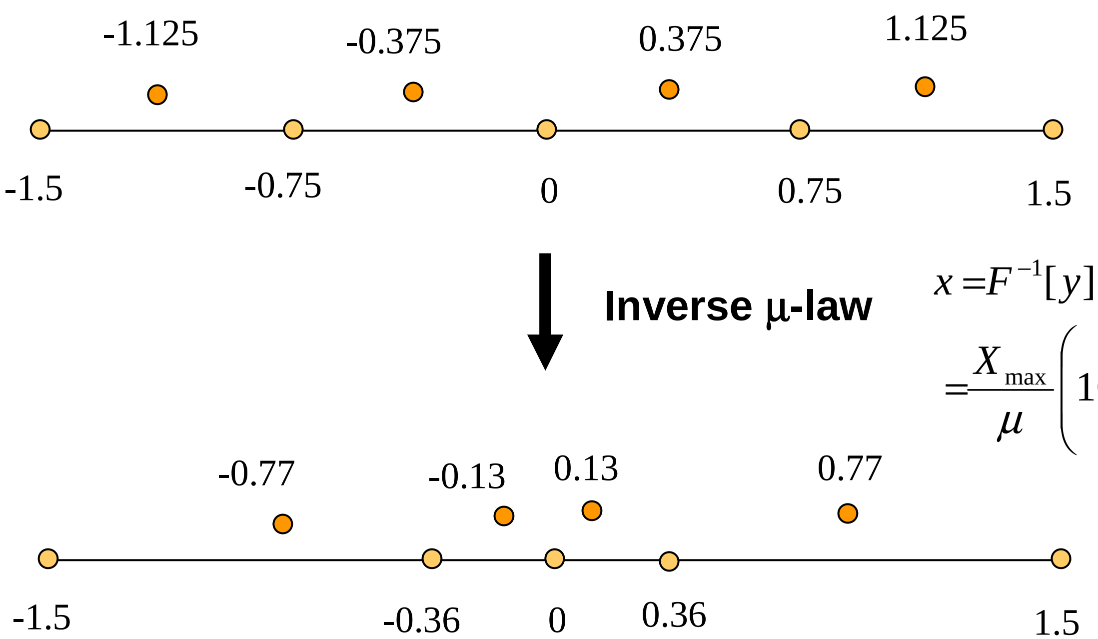
Yellow dots indicate the partition levels (boundaries between separate quantization intervals)
Red dots indicate the reconstruction levels (middle of each interval)

1.2 fall between 0.75 and 1.5, and hence is quantized to 1.125

- Quantized sequence:

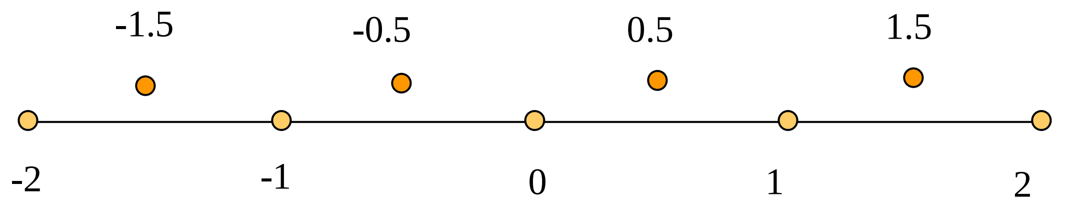
{1.125,-0.375,-0.375,0.375,1.125,1.125}

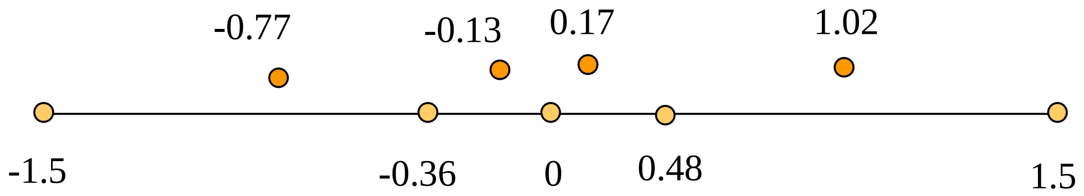
μ-Law Qunatization Example



$$x = F^{-1}[y] = \frac{X_{\max}}{\mu}\left(10^{\frac{\log(1+\mu)}{X_{\max}}|y|} - 1\right)\mathrm{sign}(y)$$

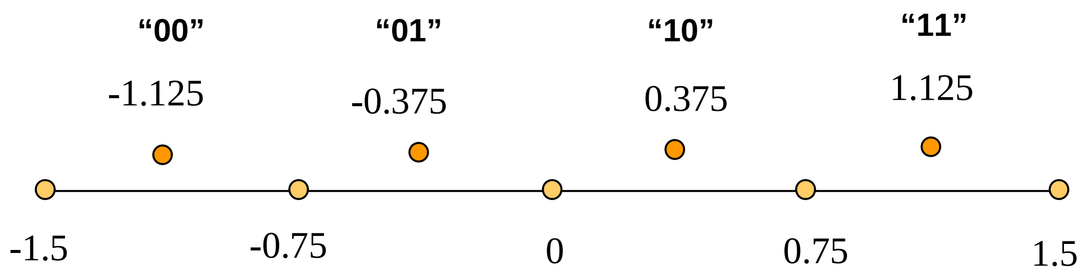
- Original sequence: {1.2,-0.2,-0.5,0.4,0.89,1.3…}
- Quantized sequence
  - {0.77,-0.13,-0.77,0.77,0.77,0.77}


-1.5
-0.5
0.5
1.5
-2
-1
0
1
2


-0.77
-0.13
0.17
1.02
-1.5
-0.36
0
0.48
1.5

Fixed Length Binary Encoding Example

- Following previous example: 4 levels -> 2 bits/level

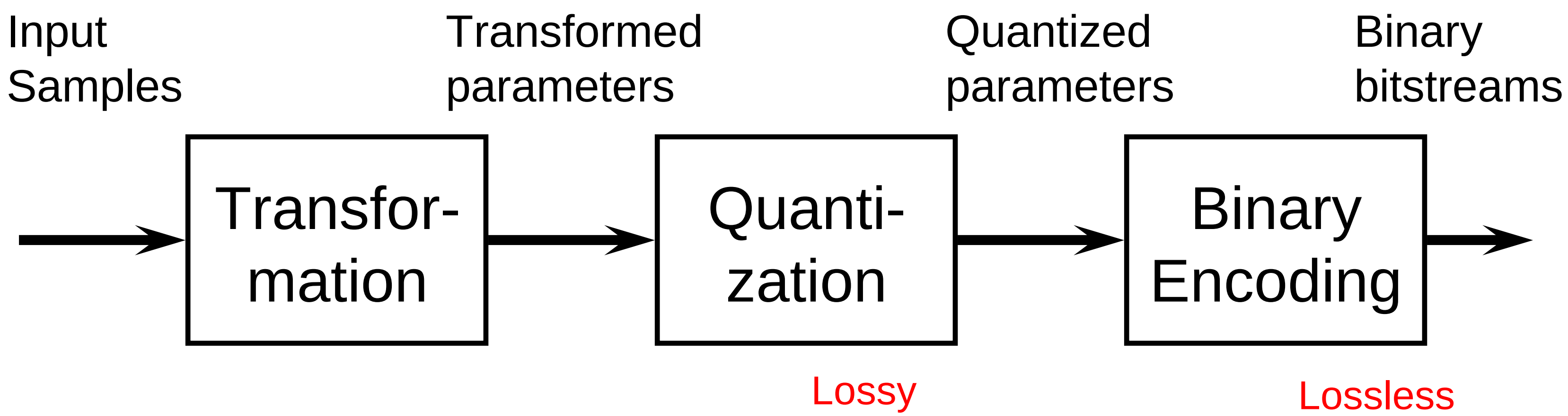


- Original sequence: {1.2,-0.2,-0.5,0.4,0.89,1.3…}
- Quantized sequence: {1.125,-0.375,-0.375,0.375,1.125,1.125,…}
- Binary representation: {11 01 01 10 11 11…}

Speech and Audio Coding

- Basic components in a coding system
- Binary encoding and entropy bound
- Predictive coding (speech)
- Model of human hearing and sight
  - Model of vocal track for speech production
  - Model of human ear: masking effect
  - Model of visual perception: frequency dependent threshold
- Subband coding (audio)
- Coding Standards
  - Speech coders, MPEG audio including MP3

Basic Components in a Source Coding System



Prediction
Transforms
Model fitting
……

Scalar Q
Vector Q

Fixed length
Variable length (Huffman, arithmetic, LZW)

- Motivation for transformation ---
  To yield a more efficient representation of the original samples.

Entropy Bound on Bit Rate

- Shannon's Theorem:
  - A source with finite number of symbols $\{s_1, s_2, \ldots, s_N\}$
  - Symbol s_n has a probability Prob(s_n)= p_n
  - If s_n is given a codeword with l_n bits, average bit rate (bits/symbol) is

$$\bar{l} = \sum p_n l_n;$$

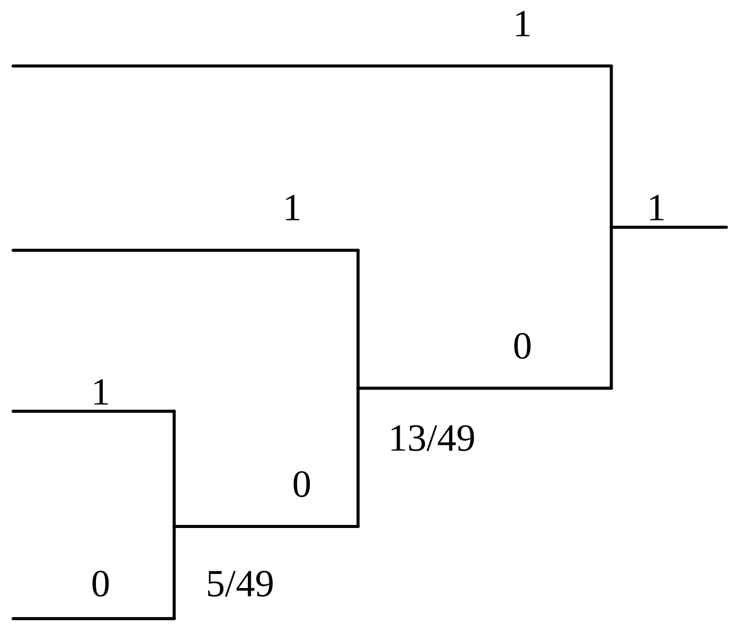
  - Average bit rate is bounded by entropy of the source (H)

$$H \leq \bar{l} \leq H + 1;$$
$$H = -\sum p_n \log_2 p_n;$$

  - For this reason, variable length coding is also known as entropy coding. The goal in VLC is to reach the entropy bound as closely as possible.

Huffman Coding Example

Symbol	Prob	Codeword	Length
" 2 "	36/49	" 1 "	1
" 3 "	8/49	" 01 "	2
" 1 "	4/49	" 001 "	3
" 0 "	1/49	" 000 "	3



$$l = \frac{36}{49}\cdot 1 + \frac{8}{49}\cdot 2 + \left(\frac{4}{49}+\frac{1}{49}\right)\cdot 3 = \frac{67}{49} = 1.4; \qquad H = -\sum p_k \log p_k = 1.16.$$

Huffman Coding Example Continued

- Code the sequence of symbols {3,2,2,0,1,1,2,3,2,2} using the Huffman code designed previously
- Code Table

symbol	codeword
0	000
1	001
2	1
3	01

- Coded sequence: {01,1,1,000,001,001,1,01,1,1}
  - Average bit rate: 18 bits/10=1.8 bits/symbol
  - Fixed length coding rate: 2 bits/symbol
  - Saving is more obvious for a longer sequence of symbols
- Decoding: table look-up

What you need to know

- Huffman coding:
  - Given a probability table, can calculate the entropy
  - Given a probability table, can construct a huffman code and compare the result to the entropy
  - Can apply the code you generated to a string of symbols for coding. Can also decode from the bit stream

Predictive Coding (DPCM)

- Observation:
  - Adjacent samples are often similar
- Predictive coding:
  - Predict the current sample from previous samples, quantize and code the prediction error, instead of the original sample.
  - If the prediction is accurate most of the time, the prediction error is concentrated near zeros and can be coded with fewer bits than the original signal
  - Usually a linear predictor is used (linear predictive coding)

$$x_p(n) = \sum_{k=1}^{P} a_k x(n-k)$$

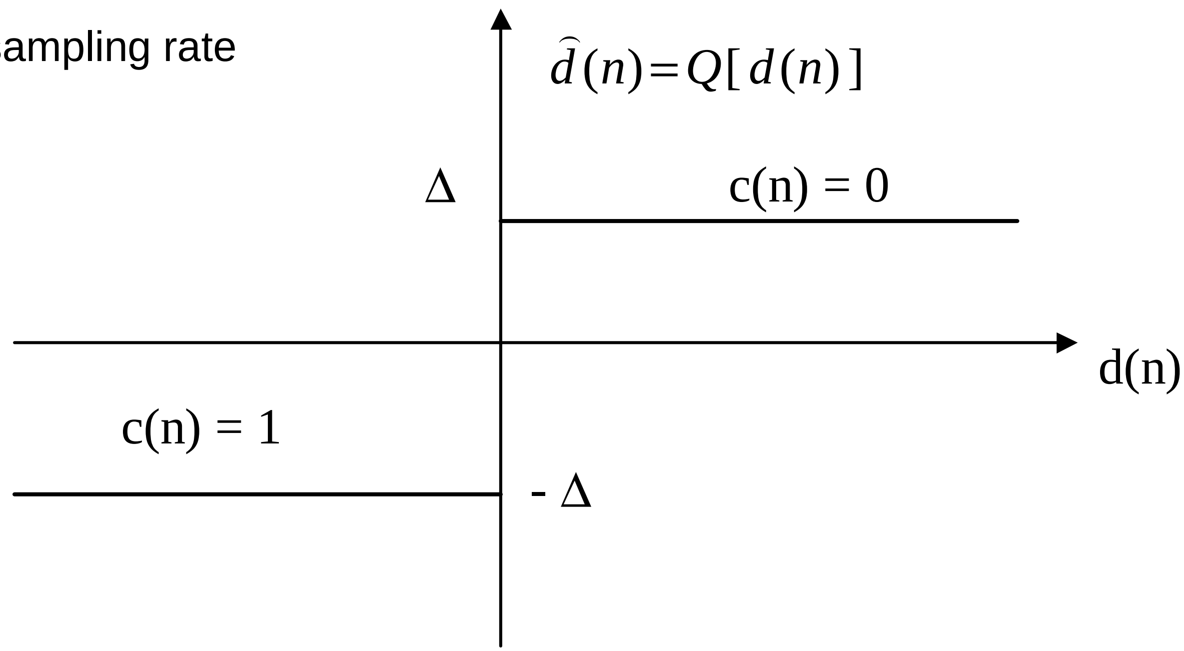
- Other names:
  - Non-predictive coding (uniform or non-uniform) -> PCM
  - Predictive coding -> Differential PCM or DPCM

Example

- Code the following sequence of samples using a DPCM coder:
  - Sequence: {1,3,4,4,7,8,6,5,3,1,…}
  - Using a simple predictor: predict the current value by the previous one
  - Using a 3-level quantizer, $Q(d)=\begin{cases} 2 & d>=1 \\ 0 & |d|<1 \\ -2 & d<=-1 \end{cases}$
  - Show the reconstructed sequence
  - For simplicity, assume the first sample is known.
  - Show the coded binary stream if the following code is used to code the difference signal
    - Error “0”-> “1”, Error “2” -> “01”, Error “-2” -> 00”

Delta-Modulation (DM)

- Delta-Modulation
  - Quantize the prediction error to 2 levels, Δ and $-\Delta$.
  - Each sample takes 1 bit
    - Bit rate=sampling rate



Speech Production Model

- Speech is produced when air is forced from the lungs through the vocal cords and along the vocal tract.
- Speech production in human can be crudely modeled as an excitation signal driving a linear system (modeled by a IIR filter)
- The filter coefficients depend on shape of the vocal tract, which changes when producing different sounds, by changing the positions of the tongue and jaw.
- Vowels are produced when the excitation signal is a periodic pulse with different periods
- Consonants are generated when the excitation signal is noise like

Vocoder Principle

- For every short frame
  - Code the filter coefficient
  - Code the excitation signal
    - Send a indicator for “voiced” “unvoiced”
    - For “voiced” send the period
- Produce “unnatural” but “intelligible” sounds at very low bit rate (<=2.4 kbps)

Hybrid Coder

- To produce more natural sounds, let the excitation signal being arbitrary, chosen so that the produced speech waveform matches with the actual waveform as closely as possible
- Hybrid coder: code the filter model plus the excitation signal as a waveform
- Code-excited linear prediction (CELP) coder: choose the excitation signal from codewords in a predesigned codebook
- This principle leads to acceptable speech quality in the rate range 4.8-16 kbps, used in various wireless cellular phone systems

What Should You Know (Predictive Coding and Speech Coding)

- Predictive coding
  - Understand the predictive coding process (the encoder and decoder block diagram)
  - Understand how the simple predictive coder (including delta modulator) works and can apply it to a sequence
  - Understand the need for adaptation of both the predictor and the quantizer, and why ADPCM improves performance (not required in the exam)
  - Understand the basic model of speech production, and difference between vocoder, hybrid coder, and waveform coders.

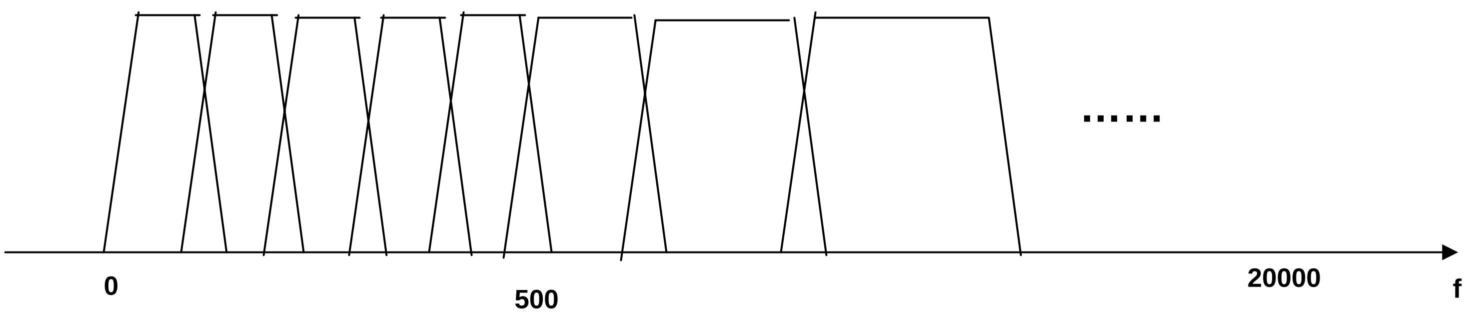
Psychoacoustic Model of Human Hearing

- Ear as a filter bank
- Three masking effects:
  - Threshold in quiet
  - Frequency masking
  - Temporal masking

Ear as a Filterbank

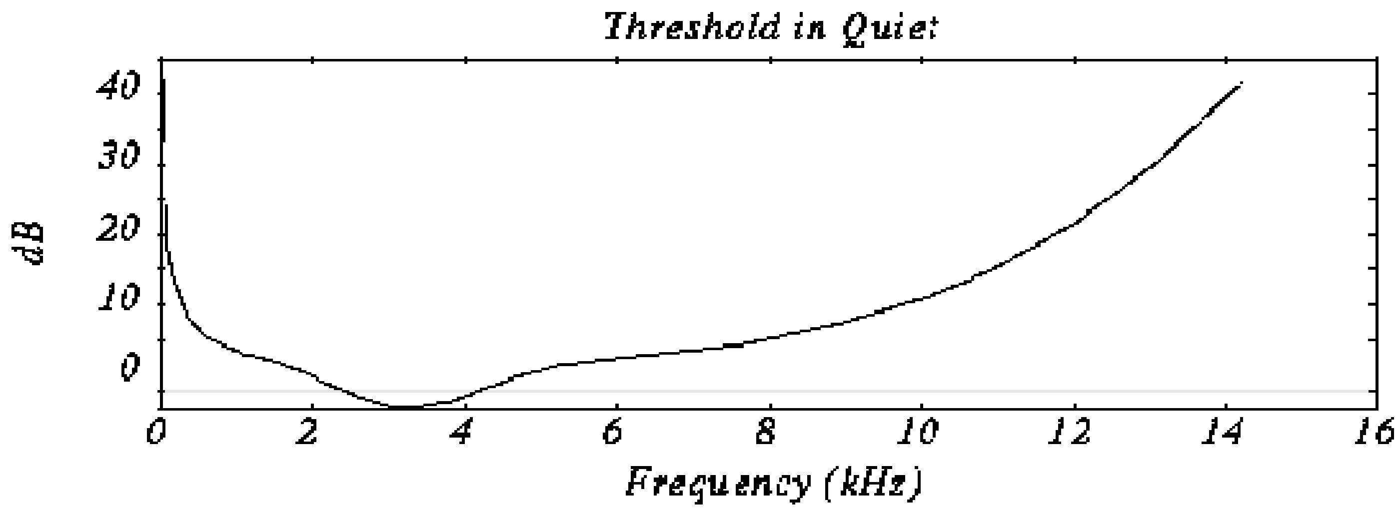
- The auditory system can be roughly modeled as a filterbank, consisting of 25 overlapping bandpass filters, from 0 to 20 KHz
  - The ear cannot distinguish sounds within the same band that occur simultaneously.
  - Each band is called a critical band
  - The bandwidth of each critical band is about 100 Hz for signals below 500 Hz, and increases linearly after 500 Hz up to 5000 Hz
  - 1 bark = width of 1 critical band

$$\text{Bark} = \begin{cases} f/100, & f \le 500\text{Hz} \\ 9 + 4\log_2(f/1000), & f > 500\text{Hz} \end{cases}$$



Threshold in Quiet

Put a person in a quiet room. Raise level of 1 kHz tone until just barely audible. Vary the frequency and plot

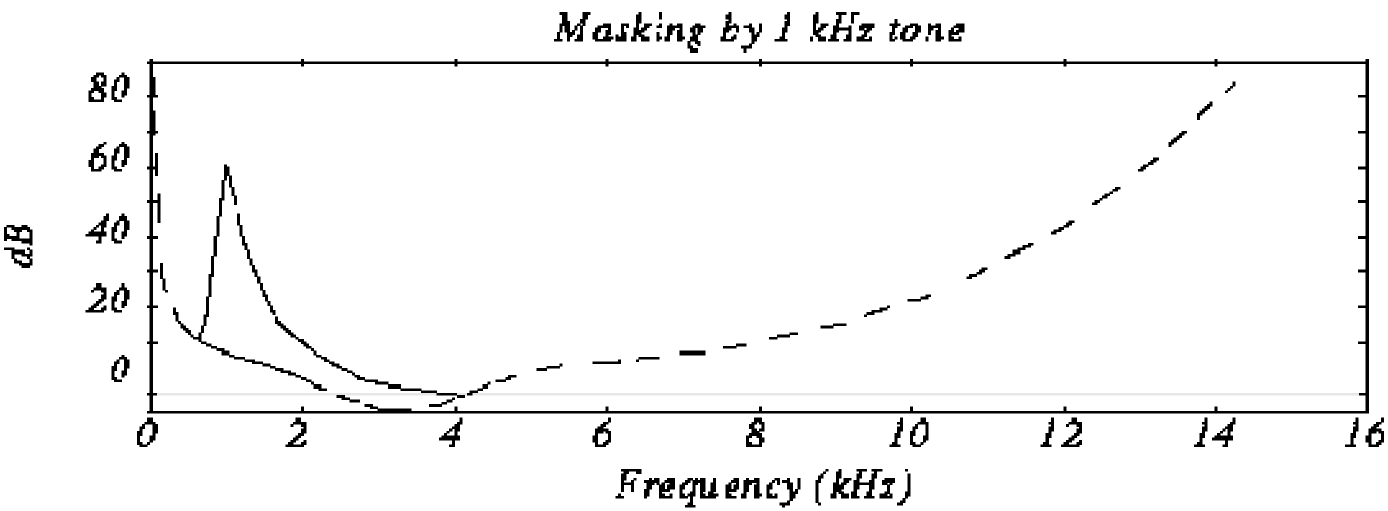


The threshold levels are frequency dependent. The human ear is most sensitive to 2-4 KHz.

From http://www.cs.sfu.ca/fas-info/cs/CC/365/li/material/notes/Chap4/Chap4.4/Chap4.4.html

Frequency Masking

Play 1 kHz tone (*masking tone*) at fixed level (60 dB). Play *test tone* at a different level (e.g., 1.1kHz), and raise level until just distinguishable. Vary the frequency of the test tone and plot the threshold when it becomes audible

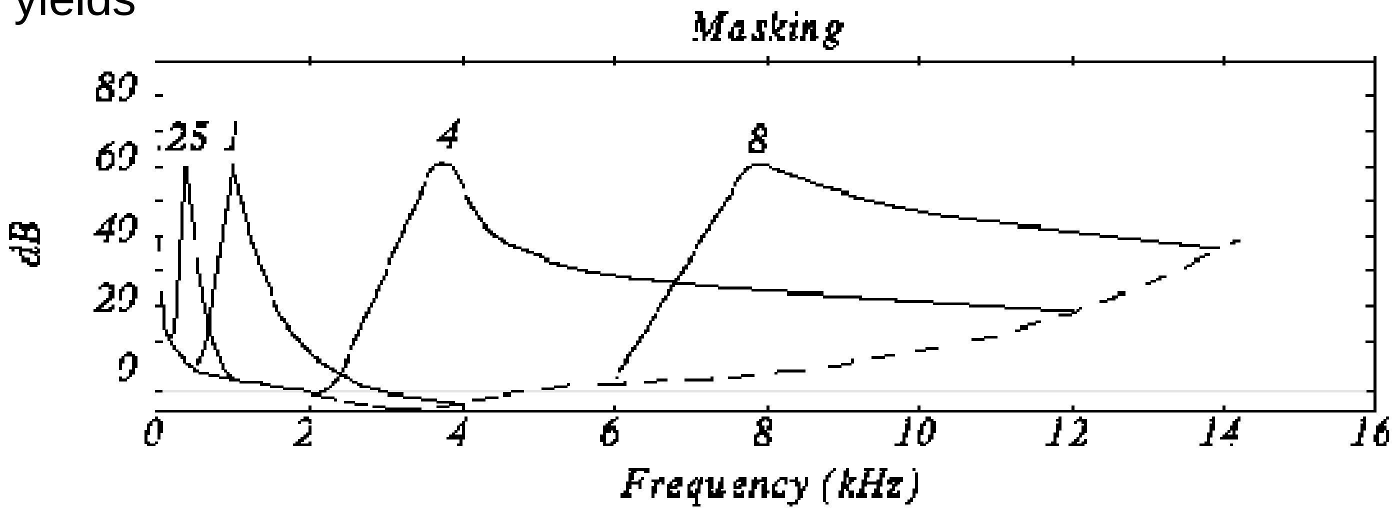


The threshold for the test tone is much larger than the threshold in quiet, near the masking frequency

From http://www.cs.sfu.ca/fas-info/cs/CC/365/li/material/notes/Chap4/Chap4.4/Chap4.4.html

Frequency Masking

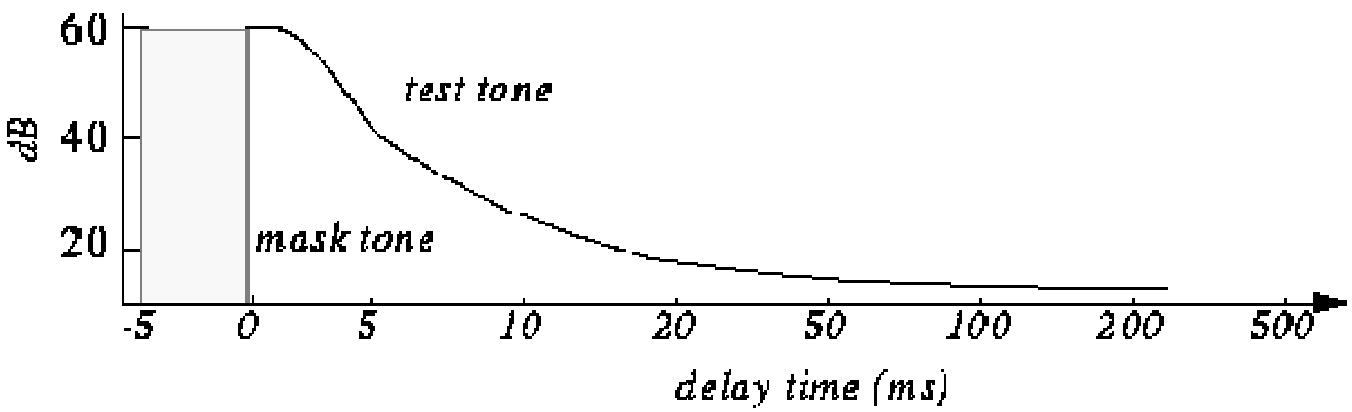
Repeat the previous experiment for various frequencies of masking tones yields



From http://www.cs.sfu.ca/fas-info/cs/CC/365/li/material/notes/Chap4/Chap4.4/Chap4.4.html

Temporal Masking

- If we hear a loud sound, then it stops, it takes a little while until we can hear a soft tone nearby
- Play 1 kHz *masking tone* at 60 dB, plus a *test tone* at 1.1 kHz at 40 dB. Test tone can't be heard (it's masked). Stop masking tone, and measure the shortest delay time after which the test tone can be heard (e.g., 5 ms). Repeat with different level of the test tone and plot. The weaker is the test tone, the longer it takes to hear it.

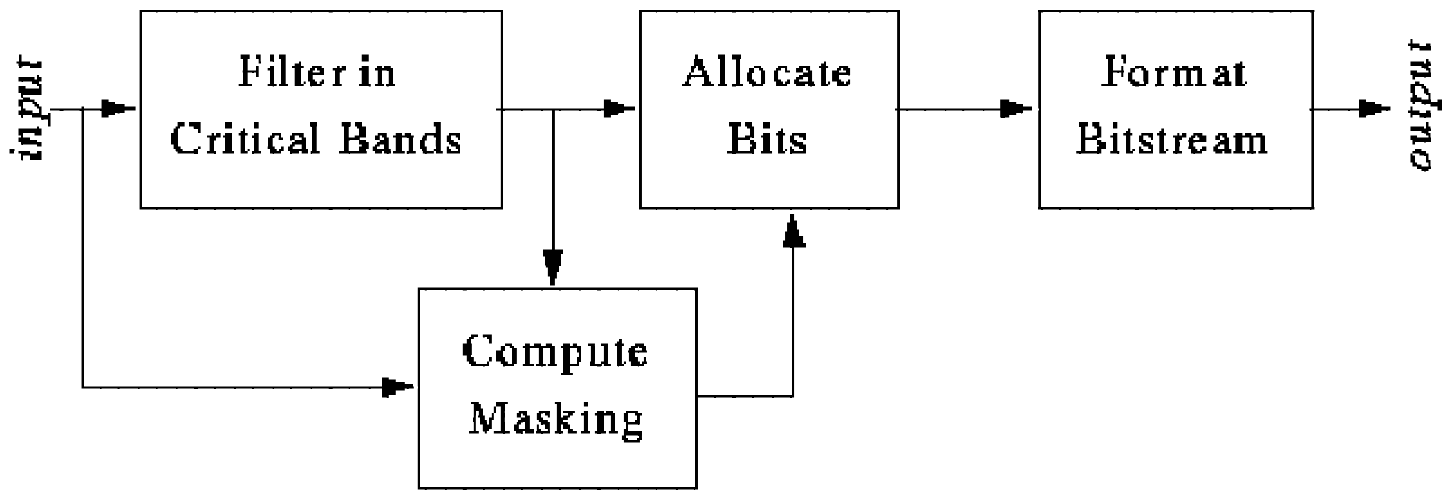


From http://www.cs.sfu.ca/fas-info/cs/CC/365/li/material/notes/Chap4/Chap4.4/Chap4.4.html

Perceptual Audio Coding: Basic Ideas

- Decompose a signal into separate frequency bands by using a filter bank
- Analyze signal energy in different bands and determine the total masking threshold of each band because of signals in other band/time
- Quantize samples in different bands with accuracy proportional to the masking level
  - Any signal below the masking level does not need to be coded
  - Signal above the masking level are quantized with a quantization step size according to masking level and bits are assigned across bands so that each additional bit provides maximum reduction in perceived distortion.

Perceptual Audio Coding Block Diagram



From http://www.cs.sfu.ca/fas-info/cs/CC/365/li/material/notes/Chap4/Chap4.4/Chap4.4.html

Quantization Basics

- The quantization error for a uniform quantizer with stepsize *Q* is approximately uniformly distributed in (-Q/2,Q/2), or with a variance of Q^2/12 (this is the quantization noise).
- Assuming the original signal is uniformly distributed over a range of *B*. With *R* bits/sample, we can use 2^R levels. The stepsize *Q* is related to the bit rate *R* by *Q=B/(2^R*)
- The quantization noise is reduced by 6 dB for every additional bit (Q -> Q/2)

Example

- Assume that the levels of 16 of the 32 bands are :

Band	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Level (db)	0	8	12	10	6	2	10	60	35	20	15	2	3	5	3	1

- Assume that If the level of the 8th band is 60dB, it gives a masking of 12 dB in the 7th band, 15dB in the 9th.
- Level in 7th band is 10 dB (< 12 dB), so ignore it.
- Level in 9th band is 35 dB (> 15 dB), so send it.
- --> Can encode with up to 2 bits (= 12 dB) of quantization error. If the original sample is represented with 8 bits, then we can reduce it to 6 bits.

From http://www.cs.sfu.ca/fas-info/cs/CC/365/li/material/notes/Chap4/Chap4.4/Chap4.4.html

Basic Steps in MPEG-1 Audio Coding

1. Use convolution filters to divide the audio signal into 32 frequency subbands --> *sub-band filtering*.
2. Determine amount of masking for each band based on its frequency (*threshold-in-quiet*), and the energy of its neighboring band in frequency and time (*frequency and temporal masking*) (this is called the *psychoacoustic model*).
3. If the energy in a band is below the masking threshold, don't encode it.
4. Otherwise, determine number of bits needed to represent the coefficient in this band such that the noise introduced by quantization is below the masking effect (Recall that 1 additional bit reduces the quantization nosie by 6 dB).
5. Format bitstream: insert proper headers, code the side information, e.g., quantization scale factors for different bands, and finally code the quantized coefficient indices, generally using variable length encoding, e.g. Huffman coding.

MPEG-1 Audio Layers

- Layer 1: DCT type filter with equal frequency spread per band. Psychoacoustic model only uses frequency masking.
- Layer 2: Same filter bank as layer 1. Psychoacoustic model uses a little bit of the temporal masking.
- Layer 3 (MP3): Layer 1 filterbank followed by MDCT per band to obtain non-uniform frequency division similar to critical bands. Psychoacoustic model includes temporal masking effects, takes into account stereo redundancy, and uses Huffman coder.
- At the time of MPEG1 audio development (finalized 1992), Layer 3 was considered too complex to be practically useful. But today, layer 3 is the most widely deployed audio coding method (known as MP3), because it provides good quality at an acceptable bit rate. It is also because the code for layer 3 is distributed freely.

What should you know? (Audio Coding)

- The properties of the auditory system
  - Ear as a filterbank
  - Masking effects: threshold-in-quiet, frequency/temporal masking
    - Should know what each means, and how to interpret a given plot of the threshold changes in freq/time
- Basic components in perceptual audio coding
  - Subband decomposition, bit allocation based on psychoacoustic model, quantization and coding
- MPEG codecs (MPEG1/2/4 and different options)
  - You should know their target applications and corresponding difference in terms of the type of audio they can handle and their relative performance
    - stereo vs. 5.1 channel, natural vs. synthetic vs. speech
  - Differences in detailed techniques in different standards/layers/profiles are not required for the exam

Midterm Exam

- Time
  - Thursday 3/2 11-12:50
- Closed book
  - 1 sheet of notes (double sided) allowed, can use a calculator
  - No peak into neighbors
  - Cheating will result in a failing grade!
- What you should know:
  - Should be able to do all the HW problems (HW1-HW5) without checking into the solutions!
- Office hours:
  - Tuesday (2/28): Yao Wang 1-3 PM (LC256)
  - Wed (3/1): Zhengye Liu 2-6 PM (LC220)