Professor Brendan Morris, SEB 3216, brendan.morris@unlv.edu

EE482: Digital Signal Processing Applications

Spring 2014
TTh 14:30-15:45 CBC C222

Lecture 14
Quiz 04 Review
14/04/07

http://www.ee.unlv.edu/~b1morris/ee482/

Outline

- Random Processes
  - Autocorrelation, white noise, expectation
- Adaptive Signal Processing
  - Adaptive filtering, LMS, applications
- Speech Signal Processing
  - LPC, CELP, noise subtraction, recognition
- Audio Signal Processing
  - Masking, MDCT, coding systems, equalizers

Outline

- Random Processes
  - Autocorrelation, white noise, expectation
- Adaptive Signal Processing
  - Adaptive filtering, LMS, applications
- Speech Signal Processing
  - LPC, CELP, noise subtraction, recognition
- Audio Signal Processing
  - Masking, MDCT, coding systems, equalizers

Autocorrelation

- Specifies statistical relationship of signal at different time lags $(n-k)$
  - $r_{xx}(n,k) = E[x(n)x(k)]$
  - Similarity of observations as a function of the time between them (repeating pattern, time-delay, etc.)
- We consider wide sense stationary (WSS) processes
  - Statistics do not change with time
  - Mean independent of time
  - Autocorrelation only depends on time lag
    - $r_{xx}(k) = E[x(n+k)x(n)]$

Expected Value

- Value of random variable "expected" if random variable process repeated infinite number of times
  - Weighted average of all possible values
- Expectation operator
  - $E[.] = \int_{-\infty}^{\infty} . f(x) dx$
  - $f(x)$ – probability density function of random variable X
- Favorites are mean and variance
  - Mean - $E[x(n)] = \int_{-\infty}^{\infty} x(n) f(x) dx = m_x$
  - Variance - $E[(x(n) - m_x)^2]$

White Noise

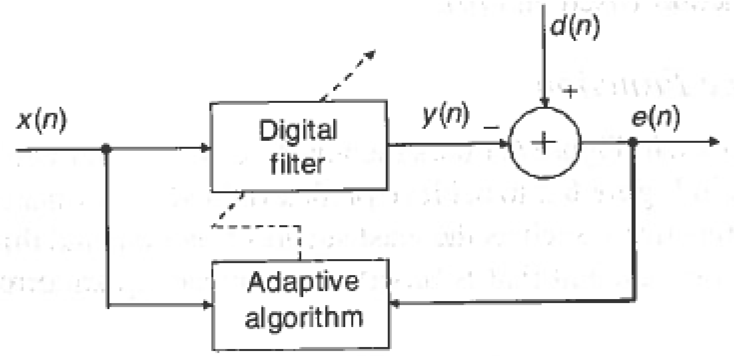
- Very popular random signal
  - Typical noise model
  - $v(n)$ with zero mean and variance σ_v^2
- Autocorrelation
  - $r_{vv}(k) = \sigma_v^2 \delta(k)$
  - Statistically uncorrelated except at zero time lag
- Power spectrum
  - $P_{vv}(\omega) = \sigma_v^2, \quad |\omega| \leq \pi$
  - Uniformly distributed over entire frequency range

Outline

- Random Processes
  - Autocorrelation, white noise, expectation
- Adaptive Signal Processing
  - Adaptive filtering, LMS, applications
- Speech Signal Processing
  - LPC, CELP, noise subtraction, recognition
- Audio Signal Processing
  - Masking, MDCT, coding systems, equalizers

General Adaptive Filter

- Signal characteristics in practical applications are time varying and/or unknown
  - Must modify filter coefficients adaptively in an automated fashion to meet objectives
- Two components
  - Digital filter – defined by coefficients
  - Adaptive algorithm – automatically update filter coefficients (weights)



- Adaption occurs by comparing filtered signal $y(n)$ with a desired (reference) signal $d(n)$
  - Minimize error $e(n)$ using a cost function (e.g. mean-square error)
  - Continually lower error and get $y(n)$ closer to $d(n)$

FIR Adaptive Filter

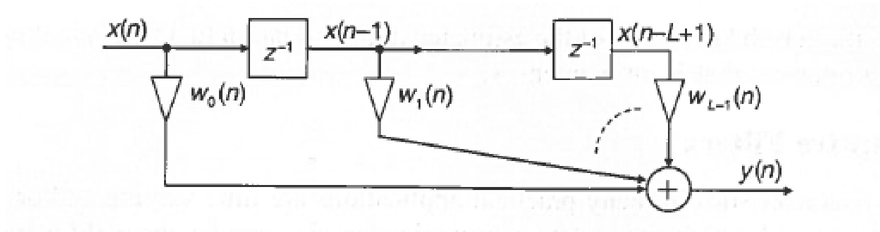


Figure 6.2 Block diagram of time-varying FIR filter for adaptive filtering

- $y(n) = \sum_{l=0}^{L-1} w_l(n)x(n-l)$
  - Notice time-varying weights
- In vector form
  - $y(n) = \boldsymbol{w}^T(n)\boldsymbol{x}(n) = \boldsymbol{x}^T(n)\boldsymbol{w}(n)$
  - $\boldsymbol{x}(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$
  - $\boldsymbol{w}(n) = [w_0(n), w_1(n), \dots, w_{L-1}(n)]^T$
- Error signal
  - $e(n) = d(n) - y(n) = d(n) - \boldsymbol{w}^T(n)\boldsymbol{x}(n)$

- Use mean-square error (MSE) cost function
- $\xi(n) = E[e^2(n)]$
- $\xi(n) = E[d^2(n)] - 2\boldsymbol{p}^T\boldsymbol{w}(n) + \boldsymbol{w}^T(n)\boldsymbol{R}\boldsymbol{w}(n)$
  - $\boldsymbol{p} = E[d(n)\boldsymbol{x}(n)] = [r_{dx}(0), r_{dx}(1), \dots, r_{dx}(L-1)]^T$
  - $\boldsymbol{R}$ – autocorrelation matrix
    - $\boldsymbol{R} = E[\boldsymbol{x}(n)\boldsymbol{x}^T(n)]$

$$= \begin{bmatrix} r_{xx}(0) & r_{xx}(1) & \dots & r_{xx}(L-1) \\ r_{xx}(1) & r_{xx}(0) & \dots & r_{xx}(L-2) \\ \vdots & \dots & \ddots & \vdots \\ r_{xx}(L-1) & r_{xx}(L-2) & \dots & r_{xx}(0) \end{bmatrix},$$

- Error function is quadratic surface
  - Can use gradient descent
  - $w(n+1) = w(n) - \frac{\mu}{2}\nabla\xi(n)$

LMS Algorithm

- Practical applications do not have knowledge of $d(n), x(n)$
  - Cannot directly compute MSE and gradient
  - Stochastic gradient algorithm
- Use instantaneous squared error to estimate MSE
  - $\hat{\xi}(n) = e^2(n)$
- Gradient estimate
  - $\nabla\hat{\xi}(n) = 2[\nabla e(n)]e(n)$
    - $e(n) = d(n) - w^T(n)x(n)$
  - $\nabla\hat{\xi}(n) = -2x(n)e(n)$
- Steepest descent algorithm
  - $w(n+1) = w(n) + \mu x(n)e(n)$

- LMS Steps

1. Set $L, \mu,$ and $\boldsymbol{w}(0)$
   - L – filter length
   - μ – step size (small e.g. 0.01)
   - $\boldsymbol{w}(0)$ – initial filter weights
2. Compute filter output
   - $y(n) = \boldsymbol{w}^T(n)\boldsymbol{x}(n)$
3. Compute error signal
   - $e(n) = d(n) - y(n)$
4. Update weight vector
   - $w_l(n+1) = w_l(n) + \mu x(n-l)e(n),$
     $l = 0,1, \dots L-1$

- Notice this requires a reference signal
- Must choose small μ for stability

Practical Applications

- Four classes of adaptive filtering applications
- System identification – determine unknown system coefficients

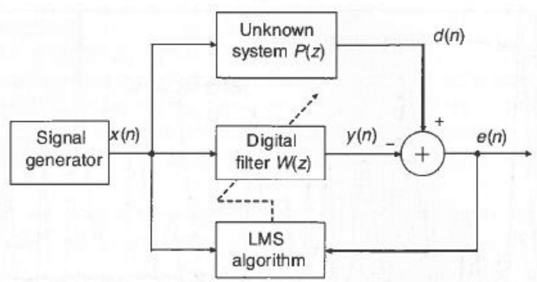


Figure 6.7 Adaptive system identification using the LMS algorithm

- Prediction – estimate future values

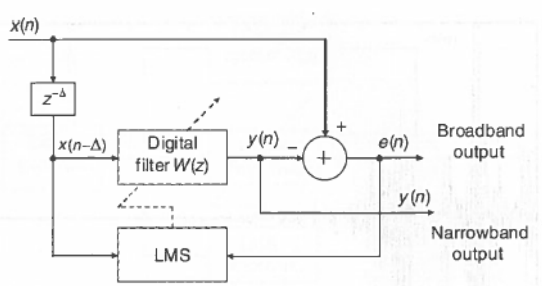


Figure 6.9 Adaptive predictor with the LMS algorithm

- Noise cancellation – remove embedded noise

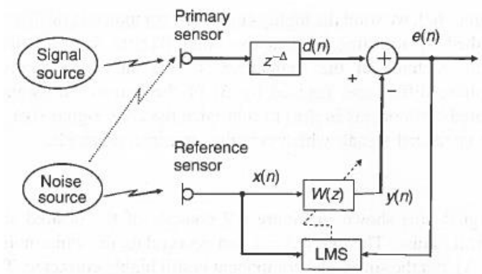


Figure 6.11 Basic concept of adaptive noise canceling

- Inverse modeling – estimate inverse of unknown system

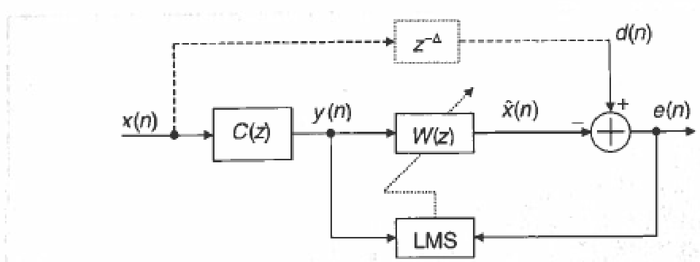


Figure 6.14 An adaptive channel equalizer as an example of inverse modeling

Outline

- Random Processes
  - Autocorrelation, white noise, expectation
- Adaptive Signal Processing
  - Adaptive filtering, LMS, applications
- Speech Signal Processing
  - LPC, CELP, noise subtraction, recognition
- Audio Signal Processing
  - Masking, MDCT, coding systems, equalizers

Linear Predictive Coding (LPC)

- Speech production model with excitation input, gain, and vocal-tract filter

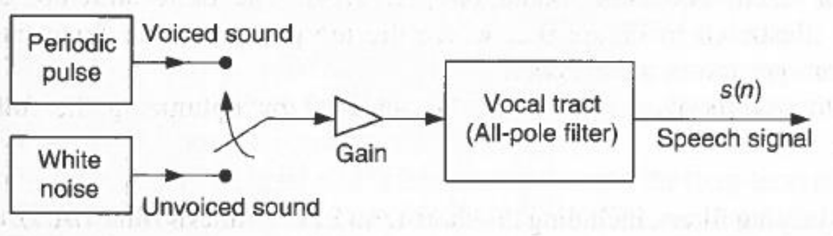


Figure 9.1 Speech production model

- Gain represents amount of air from lungs and voice loudness
- Unvoiced (e.g. "s", "sh", "f") – no vibration
  - Use white noise for excitation signal
- Voiced (e.g. vowels) – caused by vibration of vocal-cords with rate of vibration the pitch
  - Modeled with periodic pulse with fundamental (pitch) frequency
  - Generate periodic pulse train for excitation signal
- Vocal tract model
  - Vocal tract is a pipe from vocal cords to oral cavity
  - Modeled as all pole filter
    - Match formants
  - Most important part of LPC model (changes shape to make sounds)

Code-Exited Linear Prediction (CELP)

- Algorithms based on LPC approach using analysis by synthesis scheme
- Three main components:
- LPC vocal tract model $(1/A(z))$
  - Solve using Levinson-Durbin recursive algorithm with autocorrelation normal equations

$$\begin{bmatrix} r_m(0) & r_m(1) & \dots & r_m(p-1) \\ r_m(1) & r_m(0) & \dots & r_m(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_m(p-1) & r_m(p-2) & \dots & r_m(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r_m(1) \\ r_m(2) \\ \vdots \\ r_m(p) \end{bmatrix}.$$

  - More coefficients → better match to speech

- Perceptual-based minimization $(W(z))$
  - Control sensitivity of error calculation
  - Shape noise so it appears in regions where the ear cannot detect it
    - Place in louder regions of spectrum
- Voice activity detection
  - Critical for reduced coding

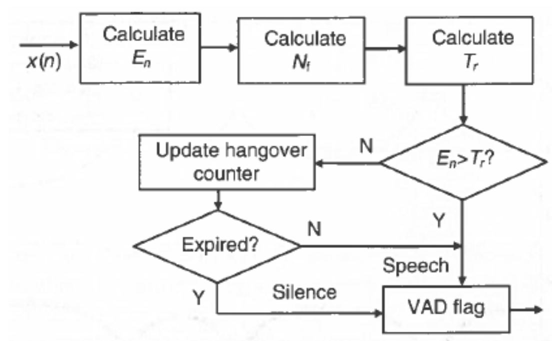


Figure 9.7 Block diagram of simple VAD algorithm

Noise Subtraction

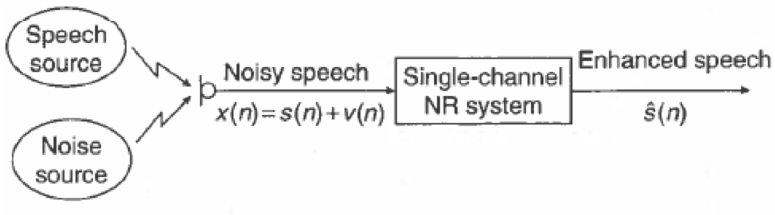


Figure 9.13 A single-channel speech enhancement system

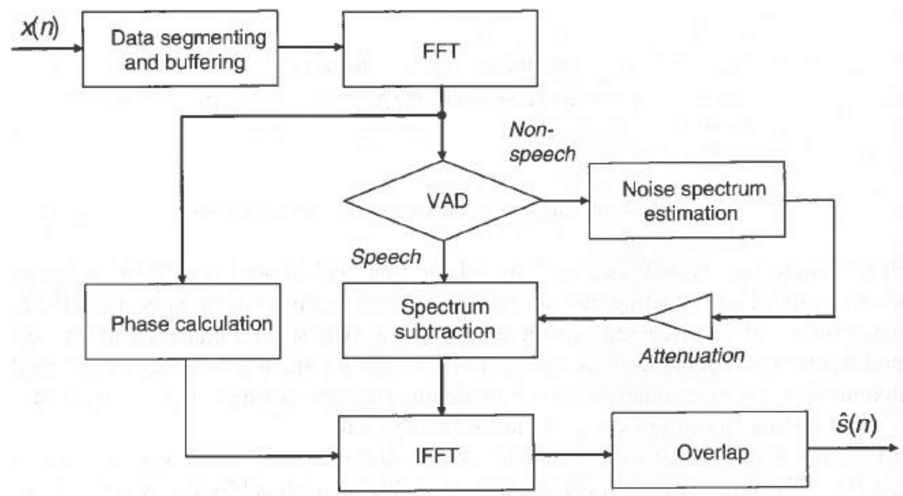
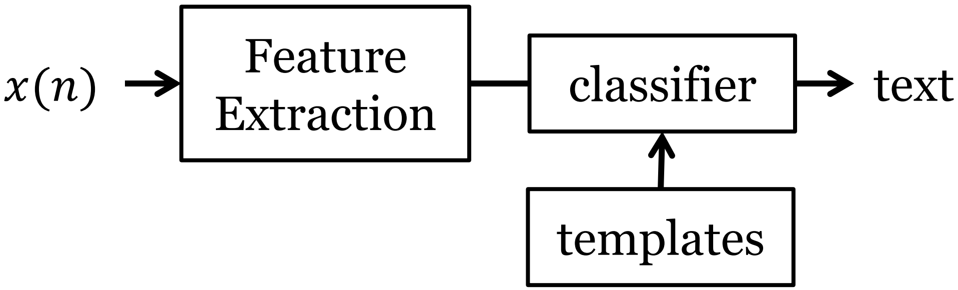


Figure 9.14 Block diagram of the spectral subtraction algorithm

- Input is noisy speech + stationary noise
  - Estimate noise characteristics during silent period between utterances with VAD system
- Spectral subtraction – implemented in frequency domain
  - Based on short-time magnitude spectra estimation
  - $S(k) = H(k)X(k)$
    - $H(k) = 1 - \frac{E|V(k)|}{|X(k)|}$
- Subtract estimated noise mag spectrum from input signal
- Reconstruct enhanced speech signal using IFFT
  - Coefficients are difference in mag and original phase

Speech Recognition



- Feature extraction
  - Represent speech content with mel-frequency cepstrum (MFCC) coefficients
    - $c[n] = \mathcal{F}^{-1}\{\log|X(e^{j\omega})|\}$
  - Rate of change in spectrum bands
  - MFCC use non-linear frequency bands to mimic human perception
- Recognizer system
  - Pattern recognition problem
  - Must design templates and method to meaningfully compare speech signals
  - Big issues: unequal length data
  - Two solutions:
    - Dynamic time warping (DTW) – optimal alignment technique for sequences
    - Hidden Markov model – probabilistic model of speech with phoneme state transitions

Outline

- Random Processes
  - Autocorrelation, white noise, expectation
- Adaptive Signal Processing
  - Adaptive filtering, LMS, applications
- Speech Signal Processing
  - LPC, CELP, noise subtraction, recognition
- Audio Signal Processing
  - Masking, MDCT, coding systems, equalizers

Audio Coding

- Techniques are required to enable high quality sound reproduction efficiently
- Differences with speech
  - Much wider bandwidth (not just 300-1000 Hz)
  - Uses multiple channels
  - Psychoacoustic principles can be utilized for coding
    - Do not code frequency components below hearing threshold
- Lossy compression used based on noise shaping
  - Noise below masking threshold is not audible
- Entropy coding applied
  - Large amount of data from high sampling rate and multi-channels

Audio Codec

- Codec = coder-decoder

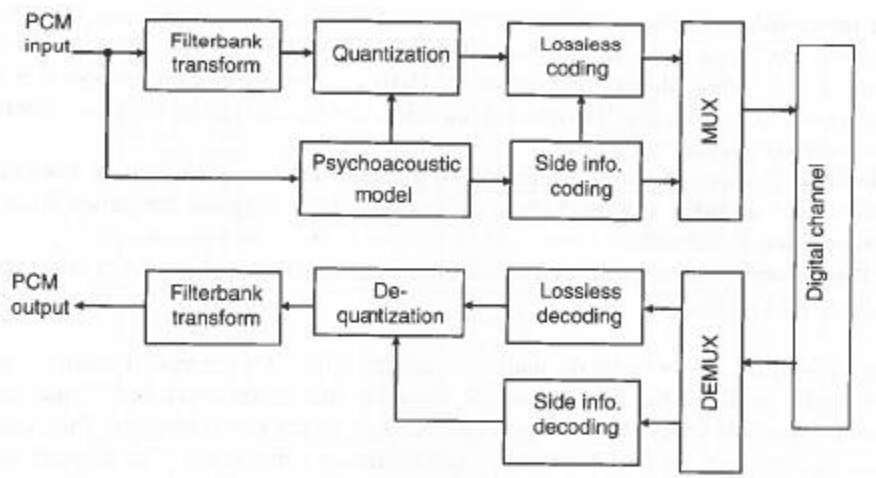


Figure 10.1 Basic structure of audio CODEC

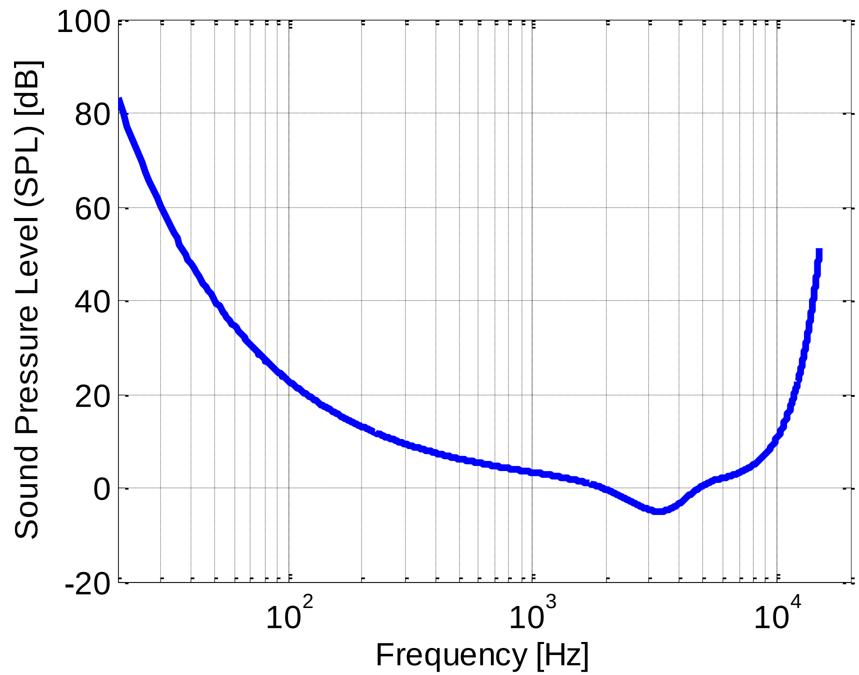
- Filterbank transform
  - Convert between full-band signal (all frequencies) into subbands (modified discrete cosine transform MDCT)
- Psychoacoustic model
  - Calculates thresholds according to human masking effects and used for quantization of MDCT
- Quantization
  - MDCT coefficient quantization of spectral coefficients
- Lossless coding
  - Use entropy coding to reduce redundancy of coded bitstream
- Side information coding
  - Bit allocation information
- Multiplexer
  - Pack all coded bits into bitstream

Auditory Masking Effects

- Psychoacoustic principle that a low-level signal (maskee) becomes inaudible when a louder signal (masker) occurs simultaneously
- Human hearing does not respond equally to all frequency components
- Auditory masking depends on the spectral distribution of masker and maskee
  - These will vary in time
- Will do noise shaping during encoding to exploit human hearing

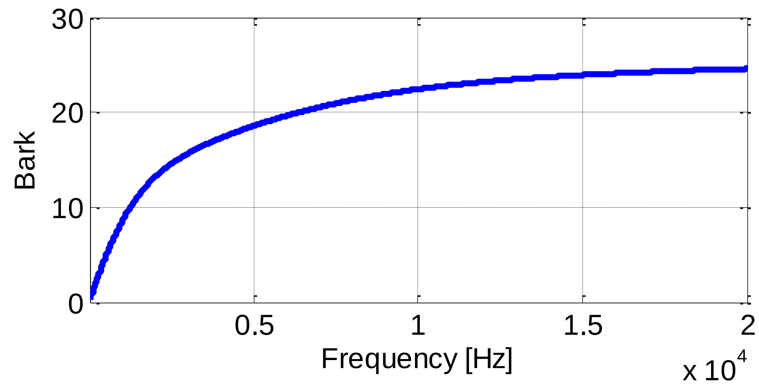
Quiet Threshold

- First step of perceptual coding
  - Shape coding distortion spectrum
- Represent a listener with acute hearing
  - No signal level below threshold will be perceived
- Quiet (absolute) threshold
  - $T_q(f) = 3.64\left(\frac{f}{1000}\right)^{-0.8} - 6.5e^{-0.6\left(\frac{f}{1000}-3.3.\right)^2} + 10^{-3}\left(\frac{f}{1000}\right)^4 \text{ dB}$
- Most humans cannot sense frequencies outside of 20-20k Hz
  - Range changes in time and narrows with age



Masking Threshold

- Threshold determined by stimuli at a given time
  - Time-varying threshold
- Human hearing non-linear response to frequency components
- Divide auditory system into 26 critical bands (barks)
  - $z(f) = 13\tan^{-1}(0.00076f) + 3.5\tan^{-1}[(f/7500)^2]$ bark
  - Higher bandwidth at higher frequencies
  - Difficult to distinguish frequencies within the same bark
- Simultaneous masking
  - Dominant frequency masks (overpowers) frequencies in same critical band
  - No need to code any other frequency components in bark
- Masking spread
  - Masking effect across adjacent critical bands
  - Use triangular spread function
    - +25 dB/bark lower frequencies
    - -10 dB/bark higher frequencies



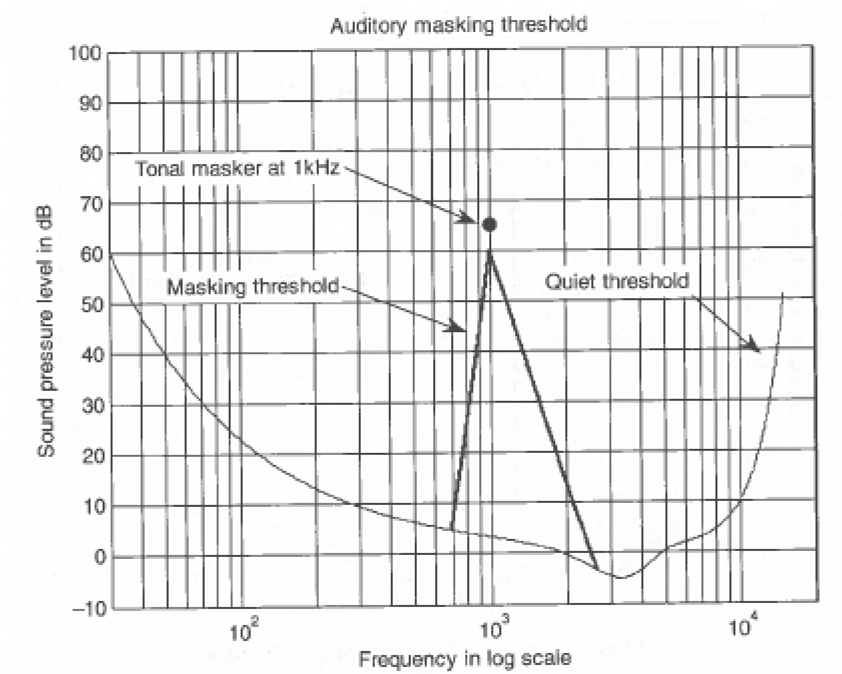


Figure 10.3 Auditory masking thresholds

Frequency Domain Coding

- Representation of frequency content of signal
- Modified discrete cosine transform (MDCT) widely used for audio
  - DCT energy compaction (lower # of coefficients)
  - Reduced block effects
- MDCT definition
  - $X(k) = \sum_{n=0}^{N-1} x(n) \cos\left[\left(n + \frac{N+2}{4}\right)\left(k + \frac{1}{2}\right)\frac{2\pi}{N}\right]$
  - $x(n) = \sum_{k=0}^{N/2-1} X(k) \cos\left[\left(n + \frac{N+2}{4}\right)\left(k + \frac{1}{2}\right)\frac{2\pi}{N}\right]$
    - $n = 0,1,\dots,N-1$
    - $k = 0,1,\dots,(N/2)-1$
  - Notice half coefficients for each window
    - Lapped transform (designed with overlapping windows built in)
- Like with FFT, windows are used but muse satisfy more conditions (Princen-Bradley condition)
  - Window applied both to analysis (MDCT) and synthesis (iMDCT) equations

Audio Coding

- Entropy (lossless) coding removes redundancy in coded data without loss in quality
- Pure entropy coding (lossless-only)
  - Huffman encoding – statistical coding
    - More often occurring symbols have shorter code words
    - Fast method using a lookup table
  - Cannot achieve very high compression
- Extended lossless coding
  - Lossy coder followed by entropy coding
  - 20% compression gain
    - MP3 – perceptual coding followed by entropy coding
- Scalable lossless coding
  - Can have perfect reproduction
  - Input first encoded, residual error is entropy coded
  - Results in two bit streams
    - Can choose lossy lowbit rate and combine for high quality lossless

Audio Equalizers

- Spectral equalization uses filtering techniques to reshape magnitude spectrum
  - Useful for recording and reproduction

- Example uses
  - Simple filters to adjust bass and treble
  - Correct response of microphone, instrument pick-ups, loudspeakers, and hall acoustics

- Parametric equalizers provide better frequency compensations but require more operator knowledge than graphic equalizers

Graphic Equalizers

- Use of several frequency bands to display and adjust the power of audio frequency components

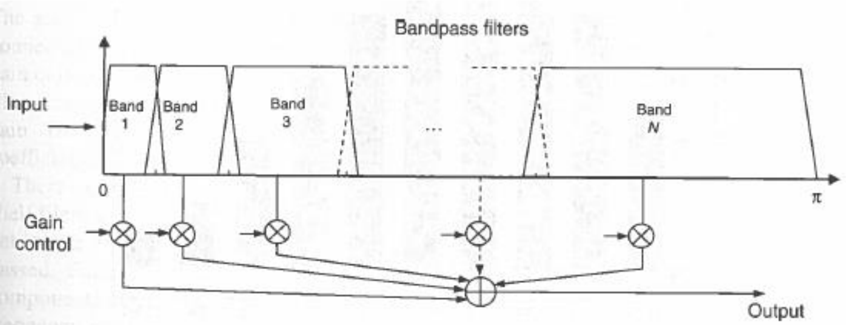
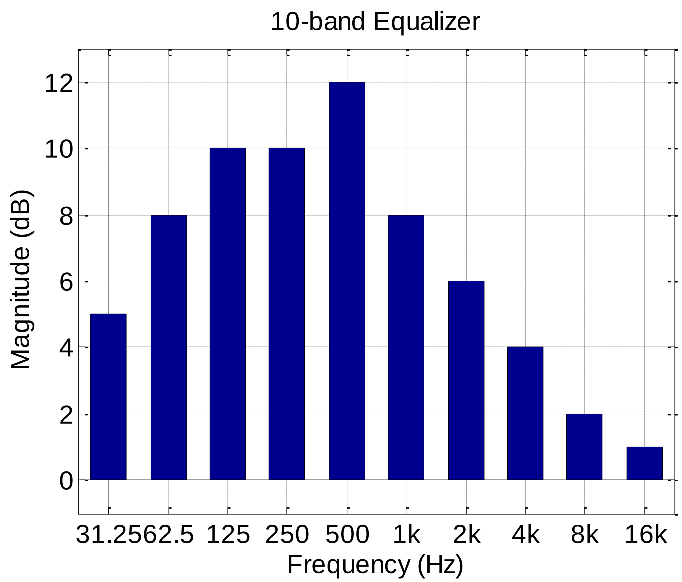


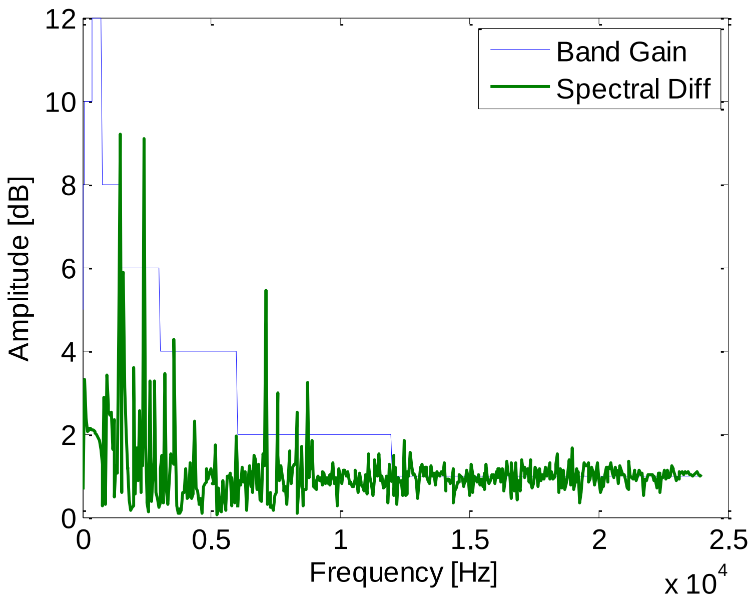
Figure 10.9 An example of an N-band graphic equalizer using octave scale

- Input signal decomposed with bank of parallel bandpass filters
- Separate gain control for each band
- Signal power in each band estimated and displayed graphically with a bar
- Divide spectrum using octave scale (doubling scale)
- Bandpass filters can be realized using IIR filter design techniques
- DFT bins of audio signal $X(k)$ need to be combined to form the equalizer frequency bands
  - Use octave scaling to combine

Example 10.4

- Graphic equalizer to adjust signal
- Select bands
  - Use octave scaling
  - `bandFreqs = {'31.25','62.5','125','250','500', '1k','2k','4k','8k','16k'};`





Parametric Equalizers

- Provides a set of filters connected in cascade that are tunable in terms of both spectral shape and filter gain
  - Not fixed bandwidth and center as in graphic
  - Use 2nd-order IIR filters

- Parameters:
  - f_s - sampling rate
  - f_c - cutoff frequency [center (peak) or midpoint (shelf)
  - Q – quality factor [resonance (peak) slope (shelf)]
  - $Gain$ – boost in dB (max $\pm$12 dB)

Shelf Filters

- Low-shelf
  - Boost frequencies below cuttoff and pass higher components
- High-shelf
  - Boost frequencies above cuttoff and pass rest
- See book for equations

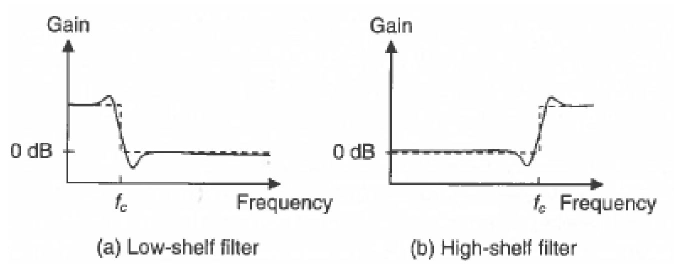
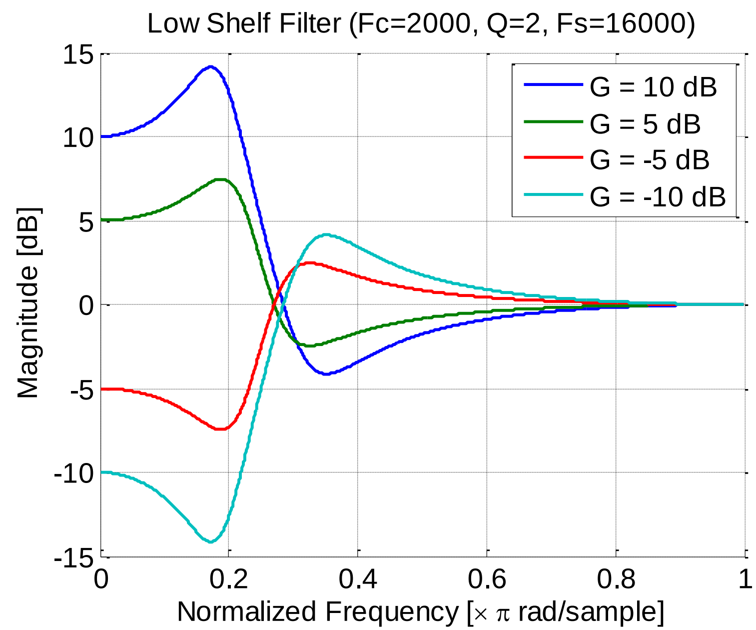
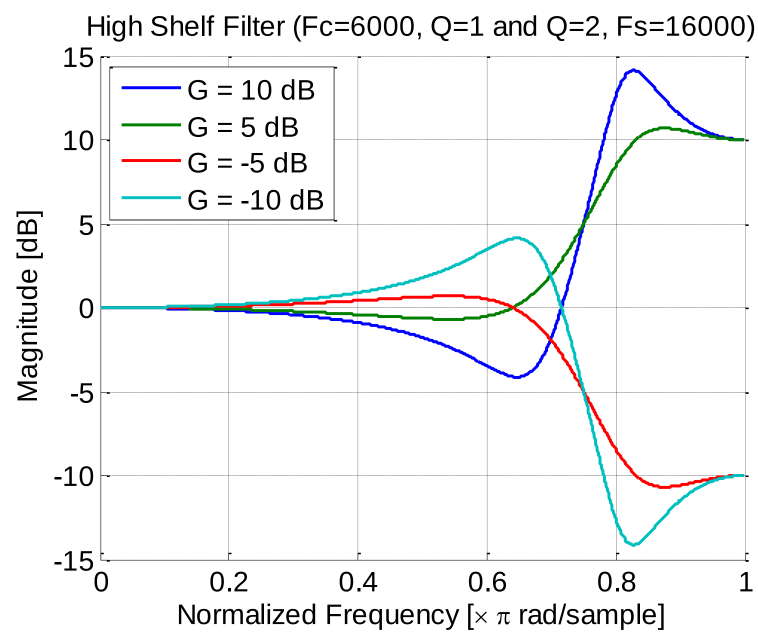


Figure 10.11 Magnitude responses of shelf filters

- Ex 10.6
  - Shape of shelf filter with different gain parameters

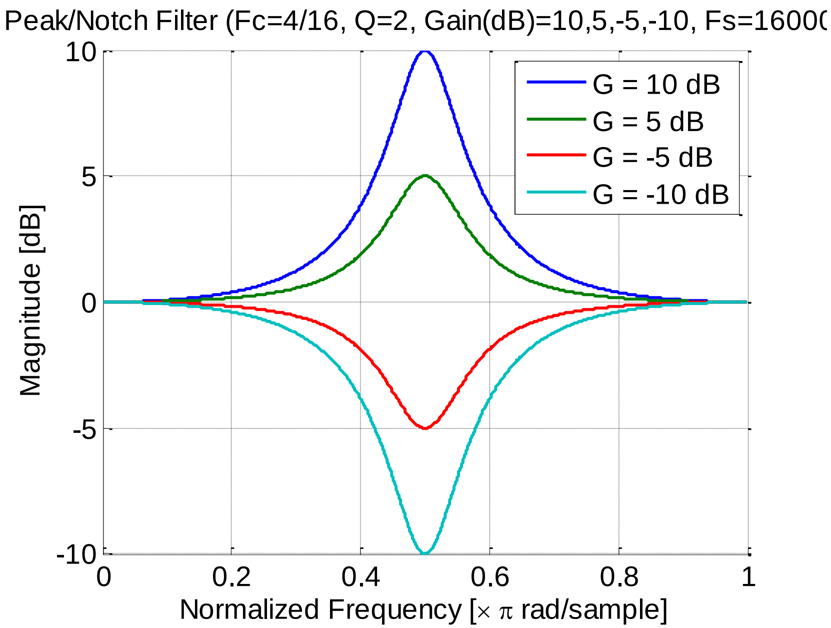




Peak Filter

- Peak filter – amplify certain narrow frequency bands
- Notch filter – attenuate certain narrow frequency bands
- E.g. loudness of certain frequency
- See book for equations

- Ex 10.5
  - Shape of peak filter for different parameters



Example 10.7

- Implement parametric equalizer
  - $f_s = 16{,}000$ Hz
- Cascade 3 filters:
  - Low-shelf filter
    - $f_c = 1000, Gain = -10\ dB, Q = 1.0$
  - High-shelf filter
    - $f_c = 4000, Gain = 10\ dB, Q = 1.0$
  - Peak filter
    - $f_c = 7000, Gain = 10\ dB, Q = 1.0$
- Play example file outside of powerpoint
  - Left channel – original signal
  - Right channel - filtered

Audio (Sound) Effects

- Use of filtering techniques to emphasize audio signal in “artistic” manner

- Will only mention and give examples of some common effects
  - Not an in-depth look

Sound Reverberation

- Reverberation is echo sound from reflected sounds
- The echoes are related to the physical properties of the space
  - Room size, configuration, furniture, etc.
- Use impulse response to measure

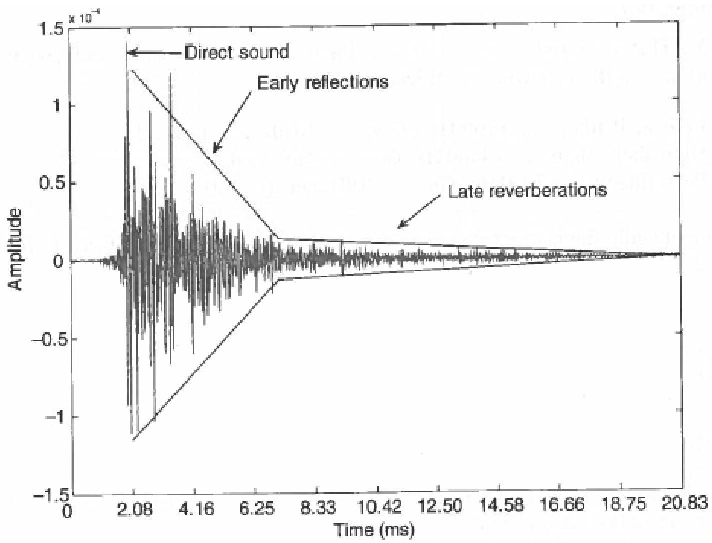


Figure 10.15 An example of a room impulse response

- Direct sound
  - First sound wave to reach ear
- Reflected sound
  - The echo waves that arrive after bouncing off a surface

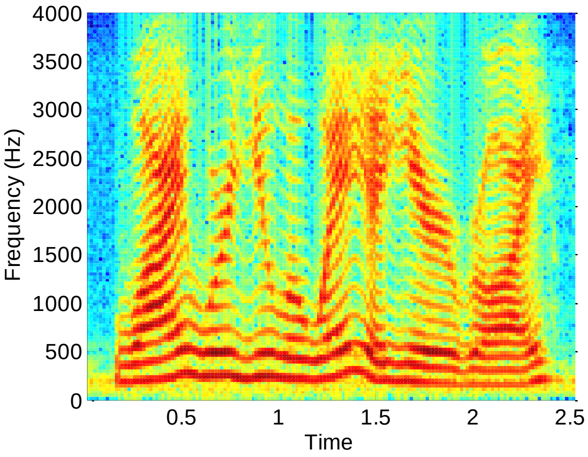
- Example 10.8
- Use hall impulse response to simulated reverberated sound

- Input

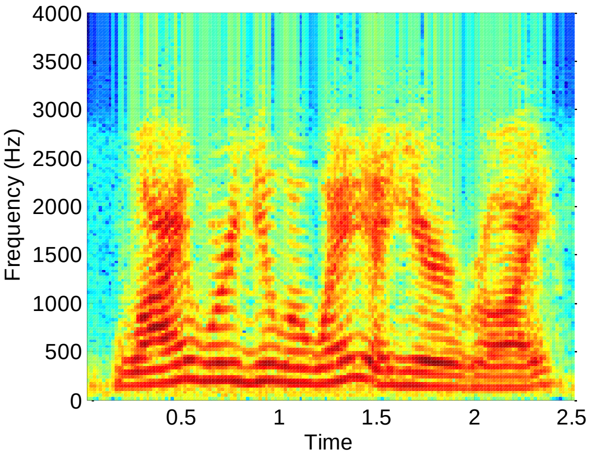
- Output

Pitch Shift

- Change speech pitch (fundamental frequency)
- All frequencies are adjusted over the entire signal
  - Chipmunk voice
- Example 10.9a
  - Adjust pitch
- See audio files



original

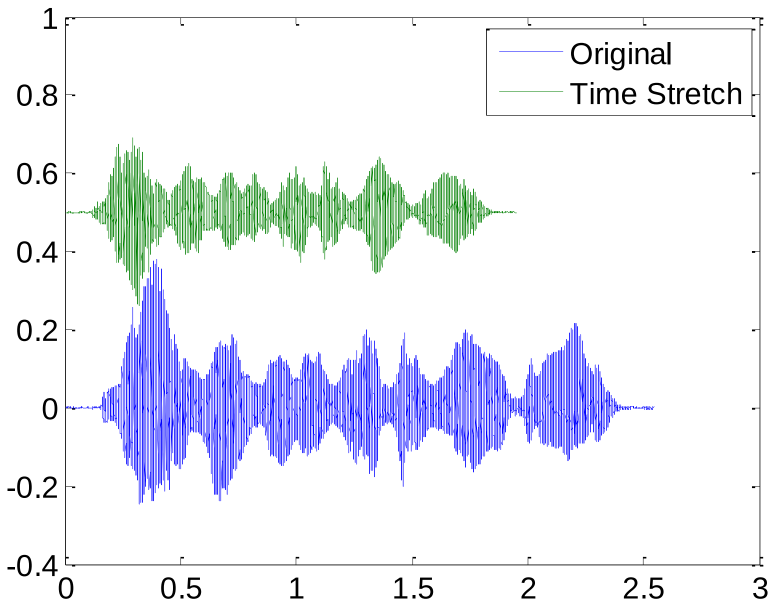


pitch shifted

Time Stretch

- Change speed of audio playback without affecting pitch
- Audio editing: adjust audio to fit a specific timeline

- Example 10.9b
  - Adjust play time
- See audio files



Tremolo

- Amplitude modulation of audio signal
  - $y(n) = [1 + AM(n)]x(n)$
    - A – max modulation amplitude
    - $M(n)$ – slow modulation oscillator

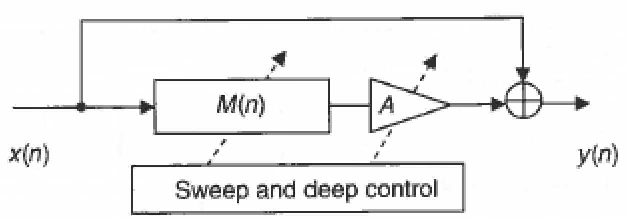
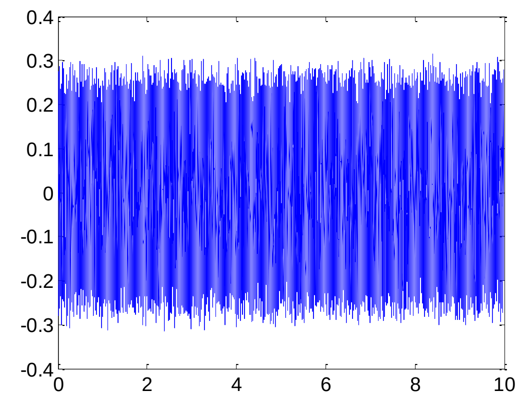
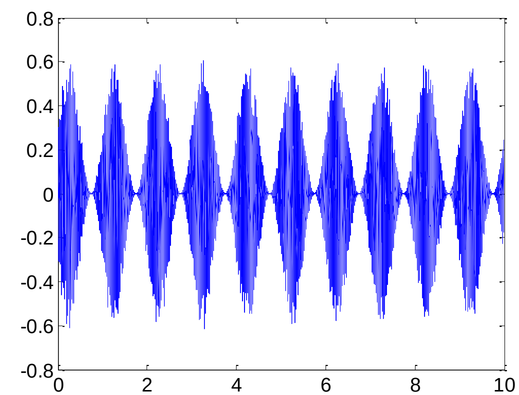


Figure 10.23 A block diagram of tremolo using low-frequency modulation

  - $M(n) = \sin(2\pi f_r nT)$
    - f_r - modulation rate

- Example 10.10
  - $A = 1, f_r = 1\ Hz$
  - White noise input at $f_s = 8000\ Hz$





Spatial Sounds

- Audio source localization determined by the way it is perceived by human ears
  - Time delay and intensity differences

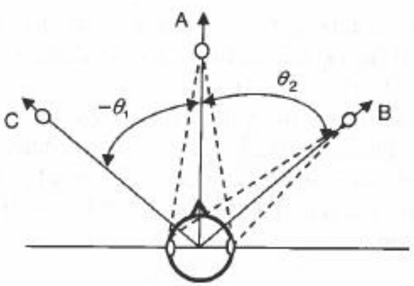


Figure 10.25 The sound source A directly in front of the head and the source B displaced at the θ_2^o azimuth

- Sounds in different positions arrive differently at ears
  - Interaural time difference (ITD) - delay between sounds reaching ear for localization
  - Iteraural intensity difference (IID) - loudness difference for localization

- Binaural audio demos
  - Great home fun
  - http://www.youtube.com/watch?v=IUDTlvagjJA
  - http://www.youtube.com/watch?v=3FwDa7TWHHc
  - http://www.qsound.com/demos/binaural-audio.htm
  - http://www.studio360.org/story/126833-adventures-3d-sound/