

Digital Speech Processing- Lecture 14A

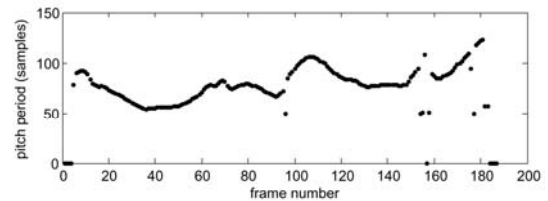
Algorithms for Speech Processing

Speech Processing Algorithms

- Speech/Non-speech detection
 - Rule-based method using log energy and zero crossing rate
 - Single speech interval in background noise
- Voiced/Unvoiced/Background classification
 - Bayesian approach using 5 speech parameters
 - Needs to be trained (mainly to establish statistics for background signals)
- Pitch detection
 - Estimation of pitch period (or pitch frequency) during regions of voiced speech
 - Implicitly needs classification of signal as voiced speech
 - Algorithms in time domain, frequency domain, cepstral domain, or using LPC-based processing methods
- Formant estimation
 - Estimation of the frequencies of the major resonances during voiced speech regions
 - Implicitly needs classification of signal as voiced speech
 - Need to handle birth and death processes as formants appear and disappear depending on spectral intensity

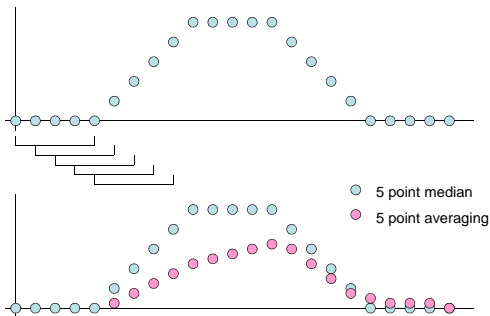
Median Smoothing and Speech Processing

Why Median Smoothing



Obvious pitch period discontinuities that need to be smoothed in a manner that preserves the character of the surrounding regions – using a median (rather than a linear filter) smoother.

Running Medians



Non-Linear Smoothing

- linear smoothers (filters) are not always appropriate for smoothing parameter estimates because of smearing and blurring discontinuities
- pitch period smoothing would emphasize errors and distort the contour
- use combination of non-linear smoother of running medians and linear smoothing
- linear smoothing => separation of signals based on non-overlapping frequency content
- non-linear smoothing => separating signals based on their character (smooth or noise-like)

$$x[n] = S(x[n]) + R(x[n]) \text{ - smooth + rough components}$$

$$y(x[n]) = \text{median}(x[n]) = M_L(x[n])$$

$$M_L(x[n]) = \text{median of } x[n] \dots x[n - L + 1]$$

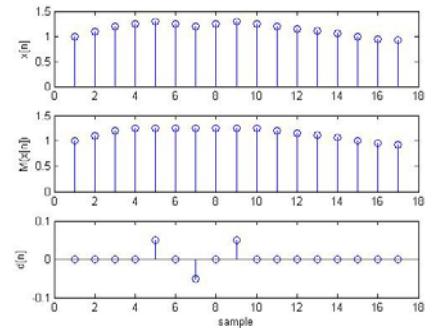
Properties of Running Medians

Running medians of length L :

1. $M_L(\alpha x[n]) = \alpha M_L(x[n])$
2. Medians will not smear out discontinuities (jumps) in the signal if there are no discontinuities within $L/2$ samples
3. $M_L(\alpha x_1[n] + \beta x_2[n]) \neq \alpha M_L(x_1[n]) + \beta M_L(x_2[n])$
4. Median smoothers generally preserve sharp discontinuities in signal, but fail to adequately smooth noise-like components

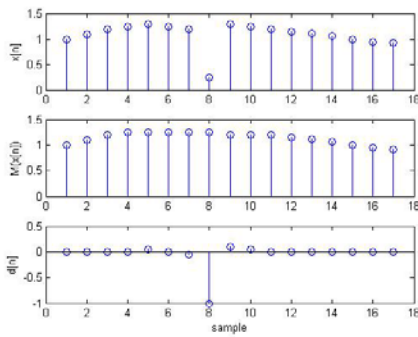
7

Median Smoothing



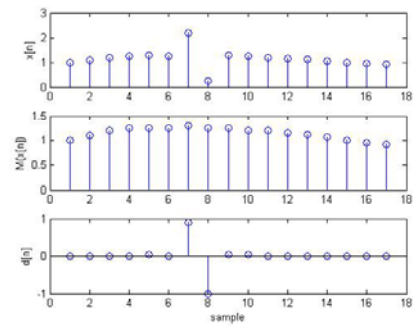
8

Median Smoothing



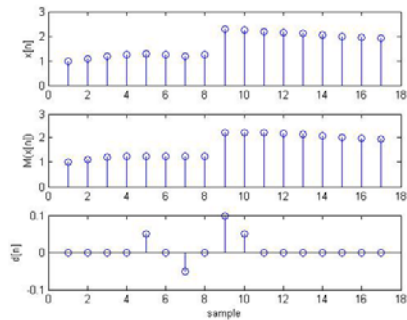
9

Median Smoothing



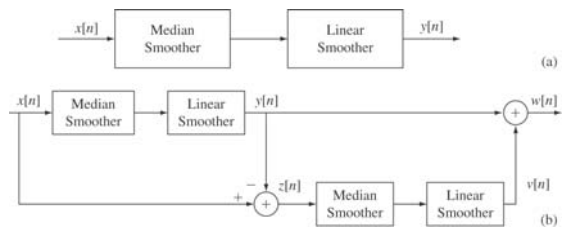
10

Median Smoothing



11

Nonlinear Smoother Based on Medians



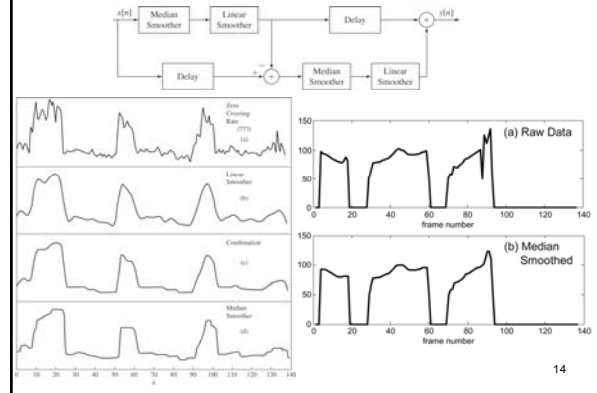
12

Nonlinear Smoother

- $y[n]$ is an approximation to the signal $S(x[n])$
- second pass of non-linear smoothing improves performance based on:
 - $y[n] = S(x[n])$
- the difference signal, $z[n]$, is formed as:
 - $z[n] = x[n] - y[n] = R(x[n])$
- second pass of nonlinear smoothing of $z[n]$ yields a correction term that is added to $y[n]$ to give $w[n]$, a refined approximation to $S(x[n])$
 - $w[n] = S(x[n]) + S[R(x[n])]$
- if $z[n] = R(x[n])$ exactly, i.e., the non-linear smoother was ideal, then $S[R(x[n])]$ would be identically zero and the correction term would be unnecessary

13

Nonlinear Smoother with Delay Compensation



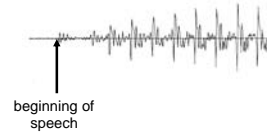
14

Algorithm #1

Speech/Non-Speech Detection Using Simple Rules

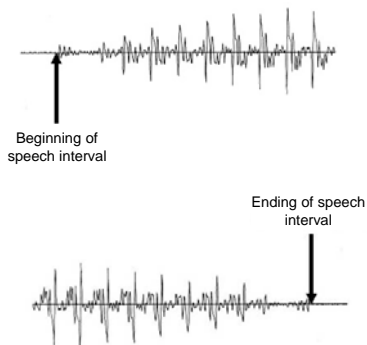
Speech Detection Issues

- key problem in speech processing is locating accurately the beginning and end of a speech utterance in noise/background signal

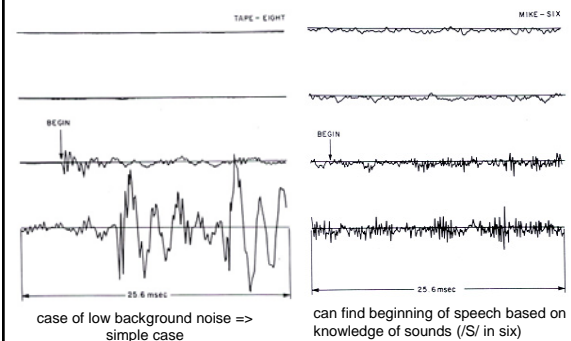


- need endpoint detection to enable:
 - computation reduction (don't have to process background signal)
 - better recognition performance (can't mistake background for speech)
- non-trivial problem except for high SNR recordings

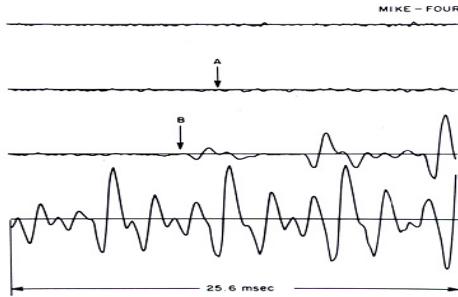
Ideal Speech/Non-Speech Detection



Speech Detection Examples



Speech Detection Examples



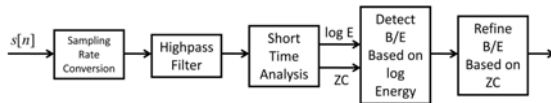
difficult case because of weak fricative sound, /f/, at beginning of speech

Problems for Reliable Speech Detection

- weak fricatives (/f/, /th/, /h/) at beginning or end of utterance
- weak plosive bursts for /p/, /t/, or /k/
- nasals at end of utterance (often devoiced and reduced levels)
- voiced fricatives which become devoiced at end of utterance
- trailing off of vowel sounds at end of utterance

the good news is that highly reliable endpoint detection is not required for most practical applications; also we will see how some applications can process background signal/silence in the same way that speech is processed, so endpoint detection becomes a moot issue

Speech/Non-Speech Detection

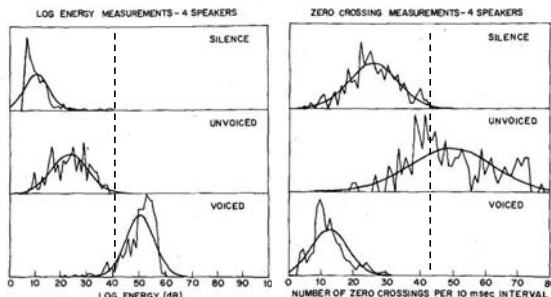


- sampling rate conversion to standard rate (10 kHz)
- highpass filtering to eliminate DC offset and hum, using a length 101 FIR equiripple highpass filter
- short-time analysis using frame size of 40 msec, with a frame shift of 10 msec; compute short-time log energy and short-time zero crossing rate
- detect putative beginning and ending frames based entirely on short-time log energy concentrations
- detect improved beginning and ending frames based on extensions to putative endpoints using short-time zero crossing concentrations

Speech/Non-Speech Detection – Algorithm #1

1. Detect **beginning** and **ending** of speech intervals using short-time energy and short-time zero crossings
2. Find **major concentration of signal** (guaranteed to be speech) using region of signal energy around maximum value of short-time energy => energy normalization
3. **Refine region of concentration** of speech using reasonably tight short-time energy thresholds that separate speech from backgrounds—but may fail to find weak fricatives, low level nasals, etc
4. **Refine endpoint estimates** using zero crossing information outside intervals identified from energy concentrations—based on zero crossing rates commensurate with unvoiced speech

Speech/Non-Speech Detection



Log energy separates Voiced from Unvoiced and Silence

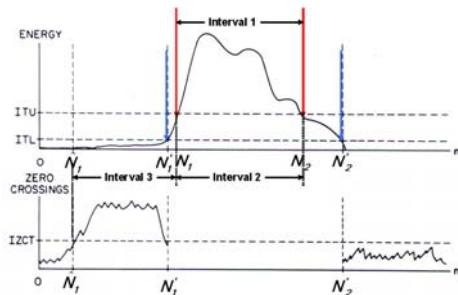
Zero crossings separate Unvoiced from Silence and Voiced

Rule-Based Short-Time Measurements of Speech

Algorithm for endpoint detection:

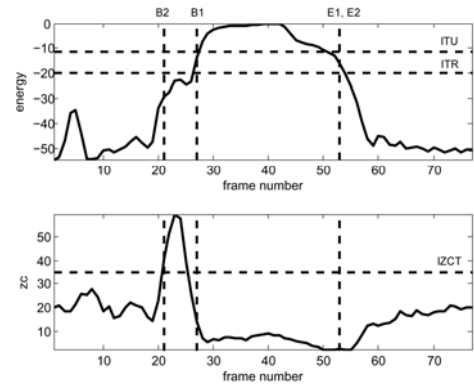
1. compute mean and σ of $\log E_n$ and Z_{100} for first 100 msec of signal (assuming no speech in this interval and assuming $F_s=10,000$ Hz).
2. determine maximum value of $\log E_n$ for entire recording => normalization.
3. compute $\log E_n$ thresholds based on results of steps 1 and 2—e.g., take some percentage of the peaks over the entire interval. Use threshold for zero crossings based on ZC distribution for unvoiced speech.
4. find an interval of $\log E_n$ that exceeds a high threshold ITU.
5. find a putative starting point (N_s) where $\log E_n$ crosses ITL from above; find a putative ending point (N_e) where $\log E_n$ crosses ITL from above.
6. move backwards from N_s by comparing Z_{100} to IZCT, and find the first point where Z_{100} exceeds IZCT; similarly move forward from N_e by comparing Z_{100} to IZCT and finding last point where Z_{100} exceeds IZCT.

Endpoint Detection Algorithm

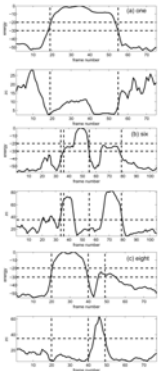


1. find heart of signal via conservative energy threshold => Interval 1
2. refine beginning and ending points using tighter threshold on energy => Interval 2
3. check outside the regions using zero crossing and unvoiced threshold => Interval 3

Endpoint Detection Algorithm



Isolated Digit Detection

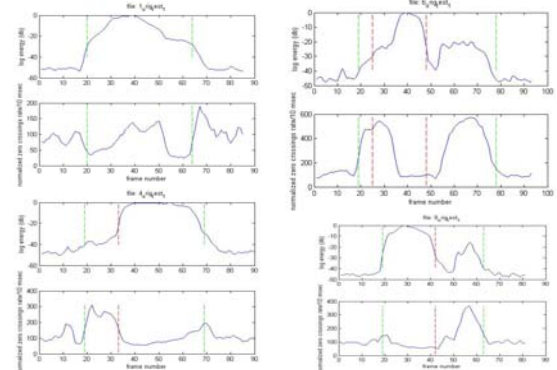


Panels 1 and 2: digit /one/
- both initial and final endpoint determined from short-time log energy

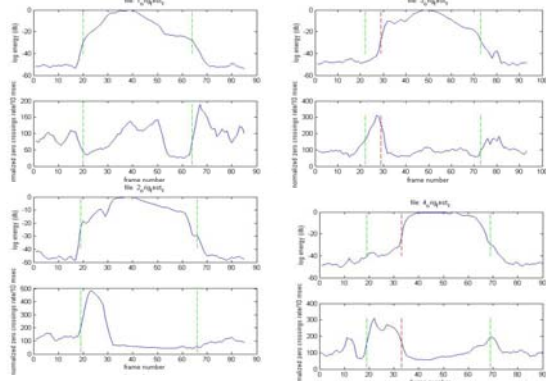
Panels 3 and 4: digit /six/
- both initial and final endpoints determined from both short-time log energy and short-time zero crossings

Panels 5 and 6: digit /eight/
- initial endpoint determined from short-time log energy; final endpoint determined from both short-time log energy and short-time zero crossings

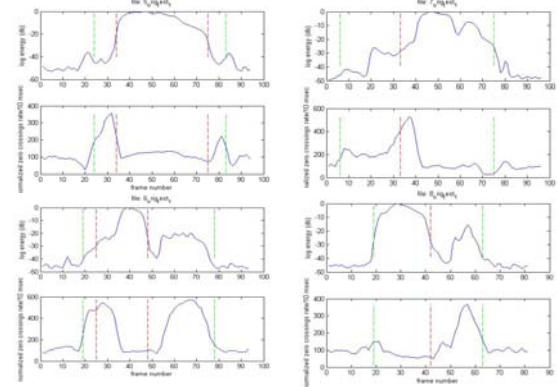
Isolated Digit Detection

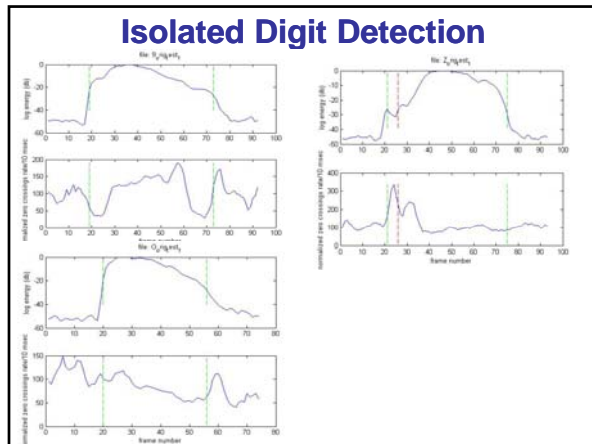


Isolated Digit Detection



Isolated Digit Detection





Algorithm #2

Voiced/Unvoiced/Background (Silence) Classification

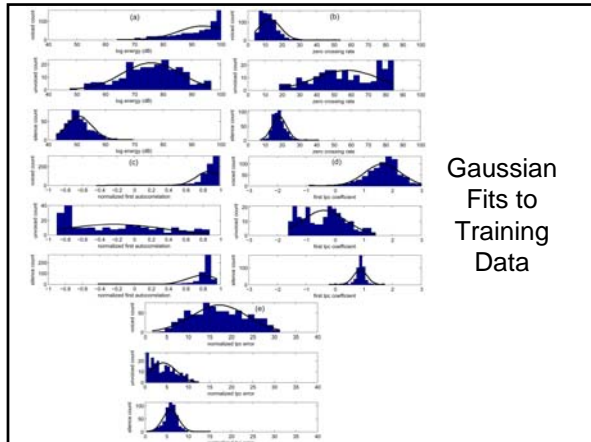
- ### Voiced/Unvoiced/Background Classification—Algorithm #2
- Utilize a Bayesian statistical approach to classification of frames as voiced speech, unvoiced speech or background signal (i.e., 3-class recognition/classification problem)
 - Use 5 short-time speech parameters as the basic feature set
 - Utilize a (hand) labeled training set to learn the statistics (means and variances for Gaussian model) of each of the 5 short-time speech parameters for each of the classes

Speech Parameters

$X = [x_1, x_2, x_3, x_4, x_5]$
 $x_1 = \log E_s$ -- short-time log energy of the signal
 $x_2 = Z_{100}$ -- short-time zero crossing rate of the signal for a 100-sample frame
 $x_3 = C_1$ -- short-time autocorrelation coefficient at unit sample delay
 $x_4 = \alpha_1$ -- first predictor coefficient of a p^{th} order linear predictor
 $x_5 = E_p$ -- normalized energy of the prediction error of a p^{th} order linear predictor

- ### Speech Parameter Signal Processing
- Frame-based measurements
 - Frame size of 10 msec
 - Frame shift of 10 msec
 - 200 Hz highpass filter used to eliminate any residual low frequency hum or dc offset in signal

- ### Manual Training
- Using a designated training set of sentences, each 10 msec interval is classified manually (based on waveform displays and plots of parameter values) as either:
 - Voiced speech – clear periodicity seen in waveform
 - Unvoiced speech – clear indication of frication or whisper
 - Background signal – lack of voicing or unvoicing traits
 - Unclassified – unclear as to whether low level voiced, low level unvoiced, or background signal (usually at speech beginnings and endings); not used as part of the training set
 - Each classified frame is used to train a single Gaussian model, for each speech parameter and for each pattern class; i.e., the mean and variance of each speech parameter is measured for each of the 3 classes



Bayesian Classifier

Class 1, $\omega_1, i = 1$, representing the background signal class
 Class 2, $\omega_2, i = 2$, representing the unvoiced class
 Class 3, $\omega_3, i = 3$, representing the voiced class

$\mathbf{m}_i = E[x]$ for all x in class ω_i
 $W_i = E[(x - \mathbf{m}_i)(x - \mathbf{m}_i)^T]$ for all x in class ω_i

Bayesian Classifier

Maximize the probability:

$$p(\omega_i | x) = \frac{p(x | \omega_i) \cdot P(\omega_i)}{p(x)}$$

where

$$p(x) = \sum_{i=1}^3 p(x | \omega_i) \cdot P(\omega_i)$$

$$p(x | \omega_i) = \frac{1}{(2\pi)^{5/2} |W_i|^{1/2}} e^{-1/2(x - \mathbf{m}_i)^T W_i^{-1} (x - \mathbf{m}_i)}$$

Bayesian Classifier

Maximize $p(\omega_i | x)$ using the monotonic discriminant function

$$g_i(x) = \ln p(\omega_i | x)$$

$$= \ln [p(x | \omega_i) \cdot P(\omega_i)] - \ln p(x)$$

$$= \ln p(x | \omega_i) + \ln P(\omega_i) - \ln p(x)$$

Disregard term $\ln p(x)$ since it is independent of class, ω_i , giving

$$g_i(x) = -\frac{1}{2}(x - \mathbf{m}_i)^T W_i^{-1} (x - \mathbf{m}_i) + \ln P(\omega_i) + c_i$$

$$c_i = -\frac{5}{2} \ln(2\pi) - \frac{1}{2} \ln |W_i|$$

Bayesian Classifier

- Ignore bias term, c_i , and apriori class probability, $\ln P_i$. Then we can convert maximization to a minimization by reversing the sign, giving the decision rule:

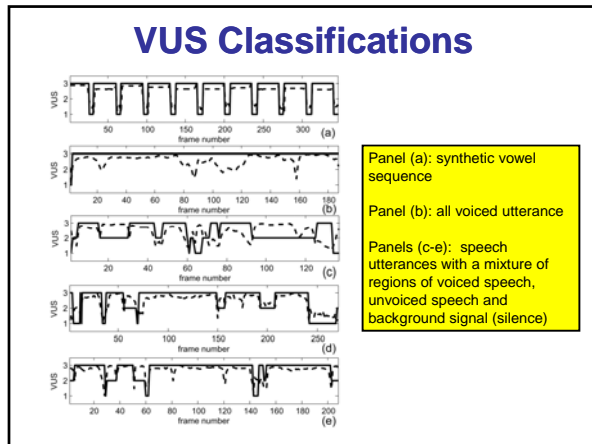
Decide class ω_i if and only if

$$d_i(x) = (x - \mathbf{m}_i)^T W_i^{-1} (x - \mathbf{m}_i) \leq d_j(x) \quad \forall j \neq i$$

- Utilize confidence measure, based on relative decision scores, to enable a no-decision output when no reliable class information is obtained.

Classification Performance

	Training Set	Count	Testing Set	Count
Background-Class 1	85.5%	76	96.8%	94
Unvoiced-Class 2	98.2%	57	85.4%	82
Voiced-Class 3	99%	313	98.9%	375



Algorithm #3

Pitch Detection (Pitch Period Estimation Methods)

- ### Pitch Period Estimation
- Essential component of general synthesis model for speech production
 - Major component of **excitation source** information (along with voiced-unvoiced decision, amplitude)
 - Pitch period estimation involves two problems, simultaneously; determination as to whether the speech is periodic, and, if so, the resulting pitch (period or frequency)
 - A range of pitch detection methods have been proposed including several time domain/frequency domain/cepstral domain/LPC domain methods

Fundamentals of Pitch Period Estimation

The Ideal Case of Perfectly Periodic Signals

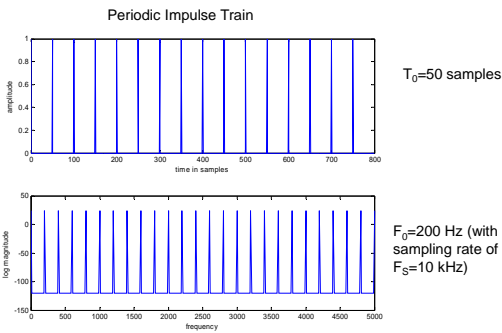
- ### Periodic Signals
- An analog signal $x(t)$ is periodic with period T_0 if:

$$x(t) = x(t + mT_0) \quad \forall t, m = \dots -1, 0, 1, \dots$$
 - The fundamental frequency is:

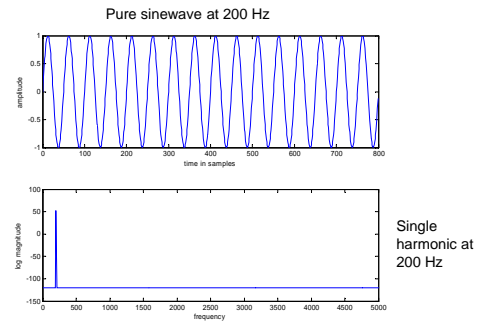
$$f_0 = \frac{1}{T_0}$$
 - A true periodic signal has a line spectrum, i.e., non-zero spectral values exist only at frequencies $f = kf_0$, where k is an integer
 - Speech is not precisely periodic, hence its spectrum is not strictly a line spectrum; further the period generally changes slowly with time

- ### The “Ideal” Pitch Detector
- To estimate pitch period reliably, the **ideal** input would be either:
 - a periodic impulse train at the pitch period
 - a pure sinusoid at the pitch frequency
 - In reality, we can't get either (although we use signal processing to either try to flatten the signal spectrum, or eliminate all harmonics but the fundamental)

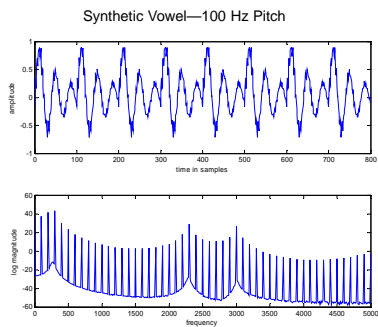
Ideal Input to Pitch Detector



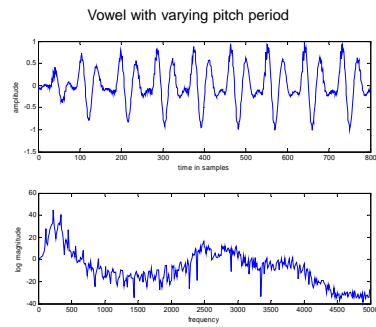
Ideal Input to Pitch Detector



Ideal Synthetic Signal Input



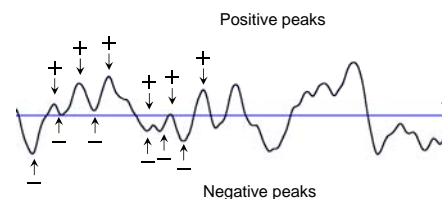
The “Real World”



Time Domain Pitch Detection (Pitch Period Estimation) Algorithm

1. Filter speech to 900 Hz region (adequate for all ranges of pitch—eliminates extraneous signal harmonics)
2. Find all positive and negative peaks in the waveform
3. At each positive peak:
 - determine peak amplitude pulse (positive pulses only)
 - determine peak-valley amplitude pulse (positive pulses only)
 - determine peak-previous peak amplitude pulse (positive pulses only)
4. At each negative peak:
 - determine peak amplitude pulse (negative pulses only)
 - determine peak-valley amplitude pulse (negative pulses only)
 - determine peak-previous peak amplitude pulse (negative pulses only)
5. Filter pulses with an exponential (peak detecting) window to eliminate false positives and negatives that are far too short to be pitch pulse estimates
6. Determine pitch period estimate as the time between remaining major pulses in each of the six elementary pitch period detectors
7. Vote for best pitch period estimate by combining the 3 most recent estimates for each of the 6 pitch period detectors
8. Clean up errors using some type of non-linear smoother

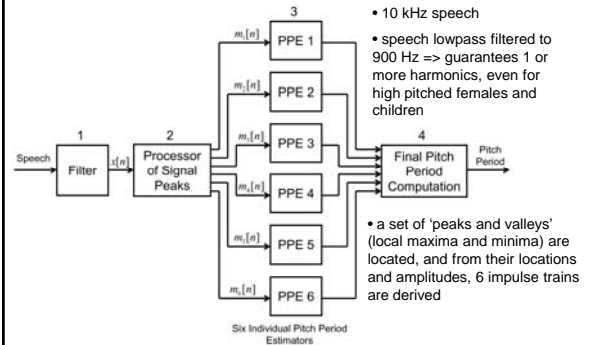
Time Domain Pitch Measurements



Basic Pitch Detection Principles

- use 6, semi-independent, parallel processors to create a number of impulse trains which (hopefully) retain the **periodicity** of the original signal and discard features which are irrelevant to the pitch detection process (e.g., amplitude variations, spectral shape, etc)
- **very simple pitch detectors** are used
- the 6 pitch estimates are **logically combined** to infer the best estimate of pitch period for the frame being analyzed
- the frame could be **classified** as unvoiced/silence, with zero pitch period

Parallel Processing Pitch Detector

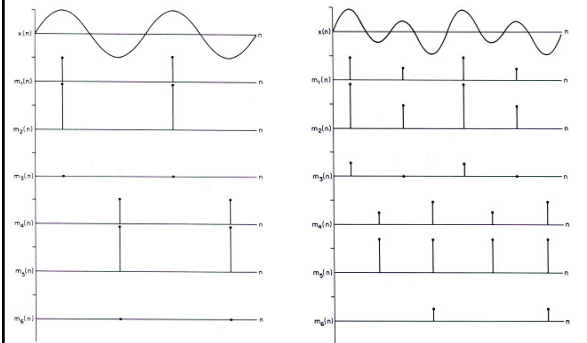


Pitch Detection Algorithm

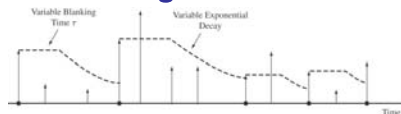
6 impulse trains:

1. $m_1(n)$: an impulse equal to the peak amplitude at the location of each peak
2. $m_2(n)$: an impulse equal to the difference between the peak amplitude and the preceding valley amplitude occurs at each peak
3. $m_3(n)$: an impulse equal to the difference between the peak amplitude and the preceding peak amplitude occurs at each peak (so long as it is positive)
4. $m_4(n)$: an impulse equal to the negative of the amplitude at a valley occurs at each valley
5. $m_5(n)$: an impulse equal to the negative of the amplitude at a valley plus the amplitude at the preceding peak occurs at each valley
6. $m_6(n)$: an impulse equal to the negative of the amplitude at a valley plus the amplitude at the preceding local minimum occurs at each valley (so long as it is positive)

Peak Detection for Sinusoids



Processing of Pulse Trains

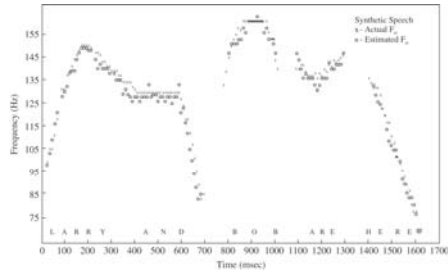


- each impulse train is processed by a time-varying non-linear system (called a peak detecting exponential window)
 - impulse of sufficient amplitude is detected => output is reset to value of impulse and "held" for a blanking interval, $Tau(n)$ during which no new pulses can be detected
 - after the blanking interval, the detector output decays exponentially with a rate of decay dependent on the most recent estimate of pitch period
 - the decay continues until an impulse that exceeds the level of the decay is detected
- output is a quasi-periodic sequence of pulses, and the duration between estimated pulses is an estimate of the pitch period
- pitch period estimated periodically, e.g., 100/sec

Final Processing for Pitch

- same detection applied to all 6 detectors => 6 estimates of pitch period every sampling interval
 - the 6 current estimates are combined with the two most recent estimates for each of the 6 detectors
 - the pitch period with the most occurrences (to within some tolerance) is declared the pitch period estimate at that time
- the algorithm works well for voiced speech
- there is a lack of pitch period consistency for unvoiced speech or background signal

Pitch Detector Performance



- using synthetic speech gives a measure of accuracy of the algorithm
- pitch period estimates generally within 2 samples of actual pitch period
- first 10-30 msec of voicing often classified as unvoiced since decision method needs about 3 pitch periods before consistency check works properly => delay of 2 pitch periods in detection

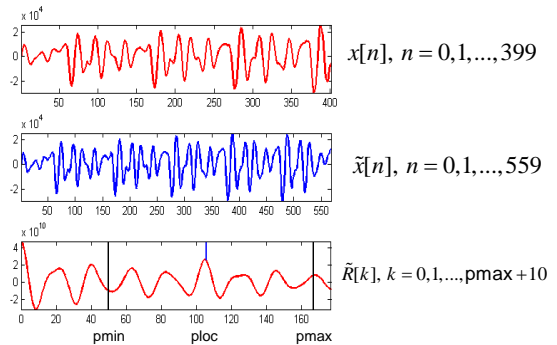
Yet Another Pitch Detector (YAPD)

Autocorrelation Method of Pitch Detection

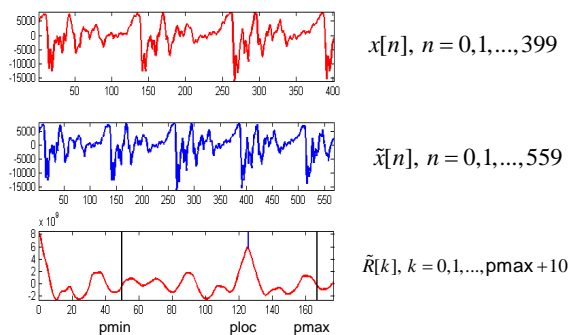
Autocorrelation Pitch Detection

- basic principle – a periodic function has a periodic autocorrelation – just find the correct peak
- basic problem – the autocorrelation representation of speech is just too rich
 - it contains information that enables you to estimate the vocal tract transfer function (from the first 10 or so values)
 - many peaks in autocorrelation in addition to pitch periodicity peaks
 - some peaks due to rapidly changing formants
 - some peaks due to window size interactions with the speech signal
- need some type of spectrum flattening so that the speech signal more closely approximates a periodic impulse train => center clipping spectrum flattener

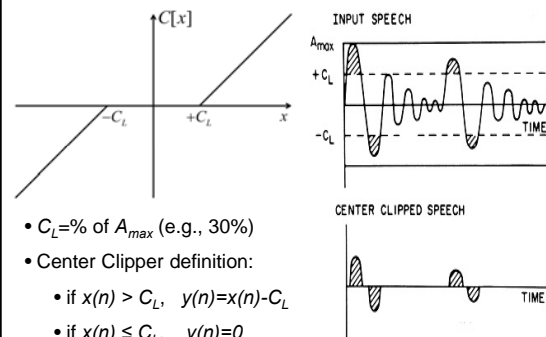
Autocorrelation of Voiced Speech Frame



Autocorrelation of Voiced Speech Frame

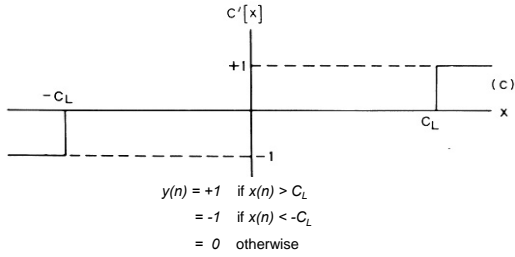


Center Clipping

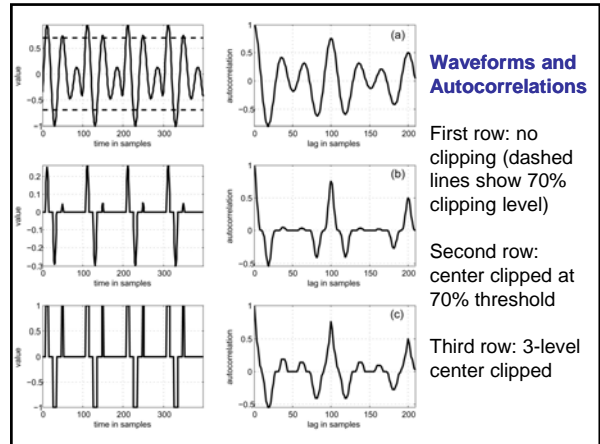


- $C_L = \% \text{ of } A_{max}$ (e.g., 30%)
- Center Clipper definition:
 - if $x(n) > C_L$, $y(n) = x(n) - C_L$
 - if $x(n) \leq C_L$, $y(n) = 0$

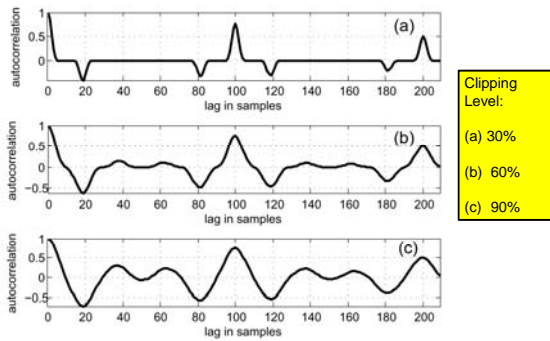
3-Level Center Clipper



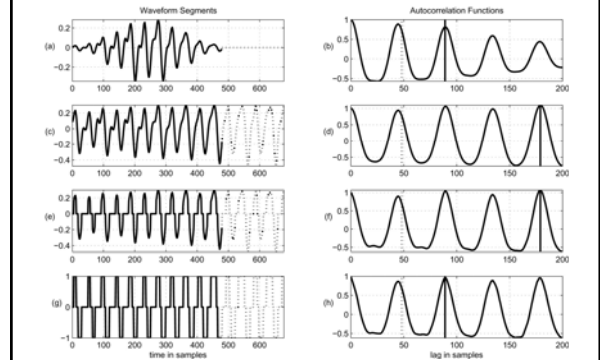
- significantly simplified computation (no multiplications)
- autocorrelation function is very similar to that from a conventional center clipper => most of the extraneous peaks are eliminated and a clear indication of periodicity is retained



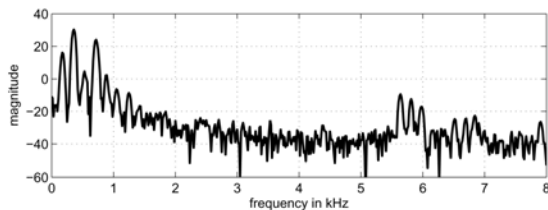
Autocorrelations of Center-Clipped Speech



Doubling Errors in Autocorrelation

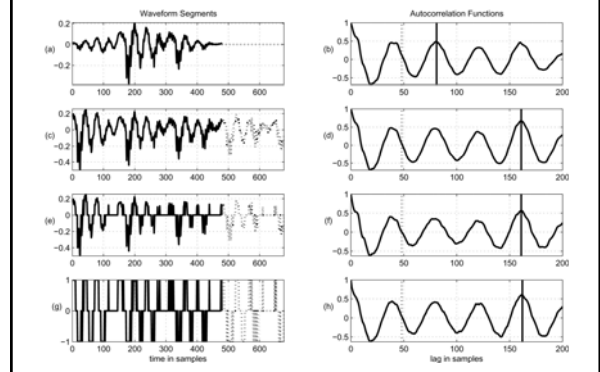


Doubling Errors in Autocorrelation

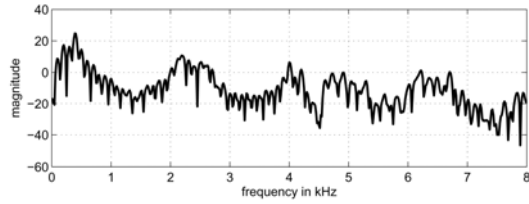


Second and fourth harmonics much stronger than first and third harmonics => potential doubling error in pitch detection.

Doubling Errors in Autocorrelation

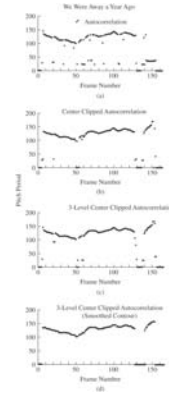


Doubling Errors in Autocorrelation



Second and fourth harmonics again much stronger than first and third harmonics => potential doubling error in pitch detection.

Autocorrelation Pitch Detector



- lots of errors with conventional autocorrelation—especially short lag estimates of pitch period
- center clipping eliminates most of the gross errors
- nonlinear smoothing fixes the remaining errors

Yet Another Pitch Detector (YAPD)

Log Harmonic Product Spectrum Pitch Detector

STFT for Pitch Detection

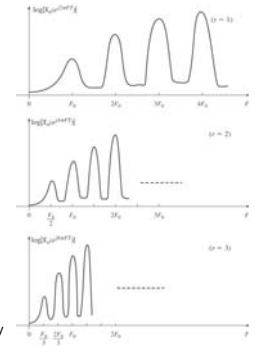
- from narrowband STFT's we see that the pitch period is manifested in sharp peaks at integer multiples of the fundamental frequency => good input for designing a pitch detection algorithm
- define a new measure, called the harmonic product spectrum, as

$$P_n(e^{j\omega}) = \prod_{r=1}^K |X_n(e^{j\omega r})|^2$$

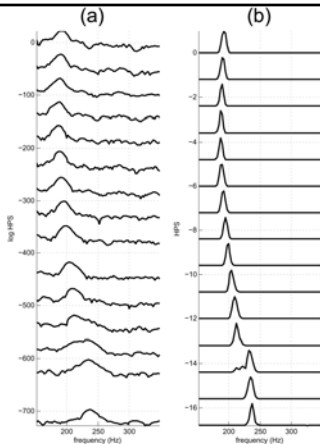
- the log harmonic product spectrum is thus

$$\hat{P}_n(e^{j\omega}) = 2 \sum_{r=1}^K \log |X_n(e^{j\omega r})|$$

- \hat{P} is a sum of K frequency compressed replicas of $\log |X_n(e^{j\omega r})|$ => for periodic voiced speech, the harmonics will all align at the fundamental frequency and reinforce each other



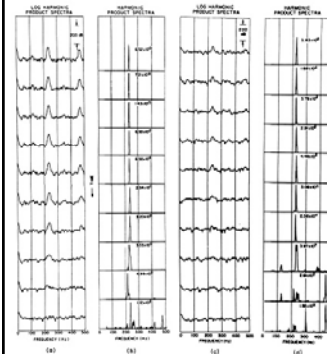
sharp peak at F_0



Column (a): sequence of log harmonic product spectra during a voiced region of speech

Column (b): sequence of harmonic product spectra during a voiced region of speech

STFT for Pitch Detection



- no problem with unvoiced speech—no strong peak is manifest in log harmonic product spectrum
- no problem if fundamental is missing (e.g., highpass filtered speech) as fundamental is found from higher order terms that line up at the fundamental but nowhere else
- no problem with additive noise or linear distortion (see plot at 0 dB SNR)

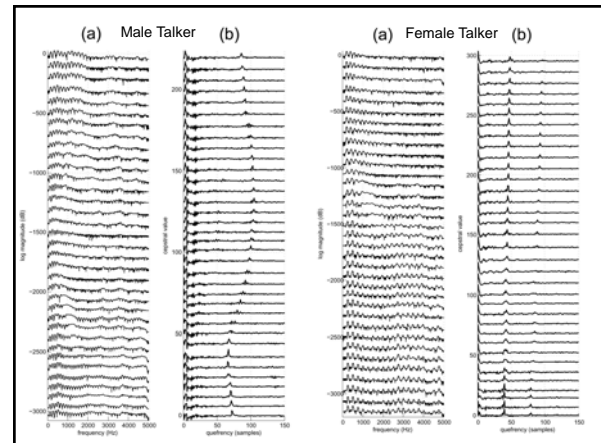
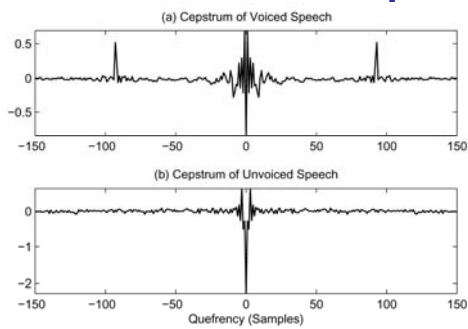
Yet Another Pitch Detector (YAPD)

Cepstral Pitch Detector

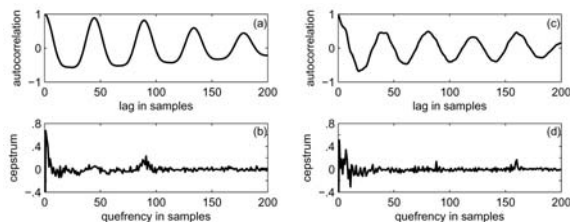
Cepstral Pitch Detection

- simple procedure for cepstral pitch detection
1. compute cepstrum every 10-20 msec
 2. search for periodicity peak in expected range of n
 3. if found and above threshold => voice, pitch=location of cepstral peak
 4. if not found => unvoiced

Cepstral Sequences for Voiced and Unvoiced Speech



Comparison of Cepstrum and ACF



Pitch doubling errors eliminated in cepstral display, but not in autocorrelation display. Weak cepstral peaks still stand out in cepstral display.

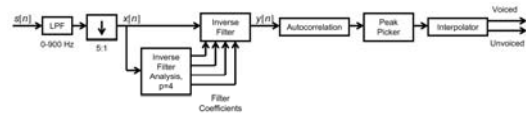
Issues in Cepstral Pitch Detection

1. strong peak in 3-20 msec range is strong indication of voiced speech-absense of such a peak does not guarantee unvoiced speech
 - cepstral peak depends on length of window, and formant structure
 - maximum height of pitch peak is 1 (RW, unchanging pitch, window contains exactly N periods); height varies dramatically with HW, changing pitch, window interactions with pitch period => need at least 2 full pitch periods in window to define pitch period well in cepstrum => need 40 msec window for low pitch male—but this is way too long for high pitch female
2. bandlimited speech makes finding pitch period harder
 - extreme case of single harmonic => single peak in log spectrum => no peak in cepstrum
 - this occurs during voiced stop sounds (b,d,g) where the spectrum is cut off above a few hundred Hz
3. need very low threshold-e.g., 0.1-on pitch period-with lots of secondary verifications of pitch period

Yet Another Pitch Detector (YAPD)

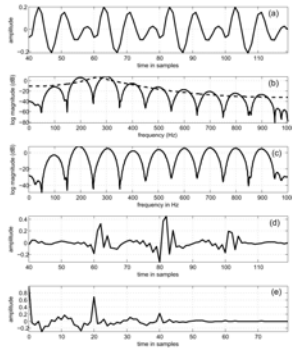
LPC-Based Pitch Detector

LPC Pitch Detection-SIFT



- sampling rate reduced from 10 kHz to 2 kHz
- $p=4$ analysis
- inverse filter signal to give spectrally flat result
- compute short time autocorrelation and find strongest peak in estimated pitch region

LPC Pitch Detection-SIFT



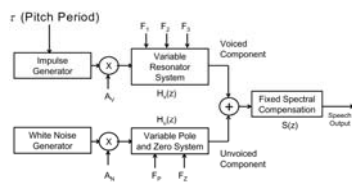
- **part a:** section of input waveform being analyzed
- **part b:** input spectrum and reciprocal of the inverse filter
- **part c:** spectrum of signal at output of the inverse filter
- **part d:** time waveform at output of the inverse filter
- **part e:** normalized autocorrelation of the signal at the output of the inverse filter
=> 8 msec pitch period found here

Algorithm #4 – Formant Estimation

Cepstral-Based Formant Estimation

Cepstral Formant Estimation

- the low-time cepstrum corresponds primarily to the combination of vocal tract, glottal pulse, and radiation, while the high time part corresponds primarily to excitation
=> use lowpass filtered cepstrum to give smoothed log spectra to estimate formants

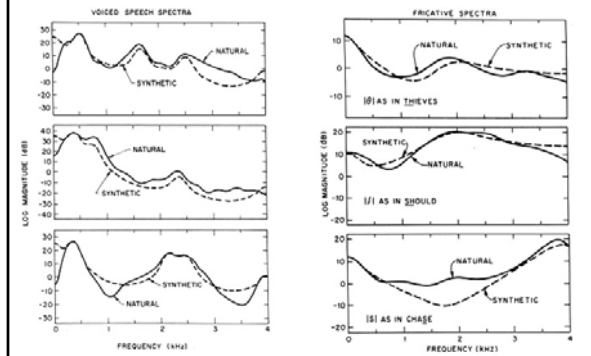


want to estimate time-varying model parameters every 10-20 msec

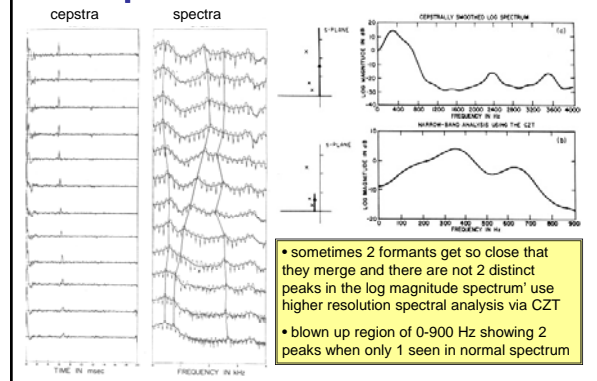
Cepstral Formant Estimation

1. fit peaks in cepstrum—decide if section of speech voiced or unvoiced
2. if voiced—estimate pitch period, lowpass filter cepstrum, match first 3 formant frequencies to smooth log magnitude spectrum
3. if unvoiced, set pole frequency to highest peak in smoothed log spectrum; choose zero to maximize fit to smoothed log spectrum

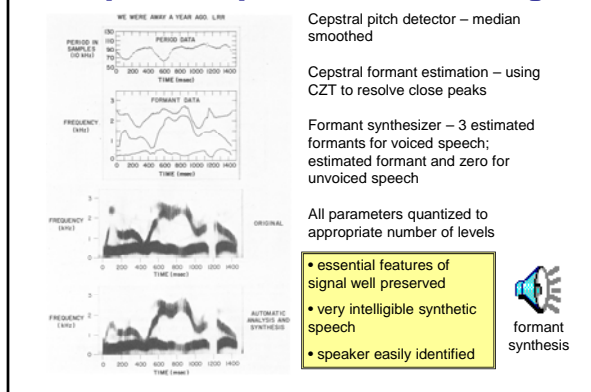
Cepstral Formant Estimation



Cepstral Formant Estimation

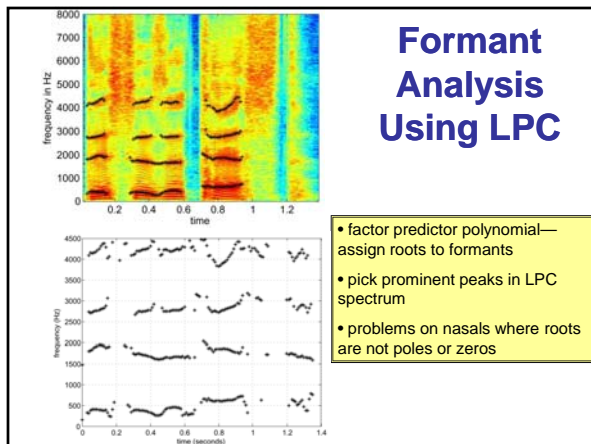


Cepstral Speech Processing



LPC-Based Formant Estimation

Formant Analysis Using LPC



Algorithm #5 – Speech Synthesis Methods

Speech Synthesis

- can use cepstrally (or LPC) estimated parameters to control speech synthesis model
- for voiced speech the vocal tract transfer function is modeled as

$$V(z) = \prod_{k=1}^4 \frac{1 - 2e^{-\alpha_k T} \cos(2\pi F_k T) + e^{-2\alpha_k T}}{1 - 2e^{-\beta_k T} \cos(2\pi F_k T) z^{-1} + e^{-2\beta_k T} z^{-2}}$$

- cascade of digital resonators ($F_1 - F_4$) with unity gain at $f = 0$
- estimate $F_1 - F_3$ using formant estimation methods, F_4 fixed at 4000 Hz
- formant bandwidths fixed ($\alpha_1 - \alpha_4$)
- fixed spectral compensation approximates glottal pulse shape and radiation

$$S(z) = \frac{(1 - e^{-aT})(1 + e^{-bT})}{(1 - e^{-aT} z^{-1})(1 + e^{-bT} z^{-1})}$$

$a = 400\pi, b = 5000\pi$

Speech Synthesis

- for unvoiced speech the model is a complex pole and zero of the form

$$V(z) = \frac{(1 - 2e^{-\beta T} \cos(2\pi F_p T) + e^{-2\beta T})(1 - 2e^{-\beta T} \cos(2\pi F_2 T) z^{-1} + e^{-2\beta T} z^{-2})}{(1 - 2e^{-\beta T} \cos(2\pi F_p T) z^{-1} + e^{-2\beta T} z^{-2})(1 - 2e^{-\beta T} \cos(2\pi F_2 T) + e^{-2\beta T})}$$

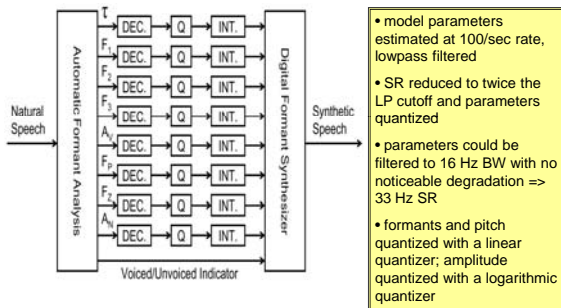
F_p = largest peak in smoothed spectrum above 1000 Hz

$$F_2 = (0.0065F_p + 4.5 - \Delta)(0.014F_p + 28)$$

$$\Delta = 20 \log_{10} |H(e^{j2\pi F_p T})| - 20 \log_{10} |H(e^{j0})|$$

- these formulas ensure spectral amplitudes are preserved

Quantization of Synthesizer Parameters



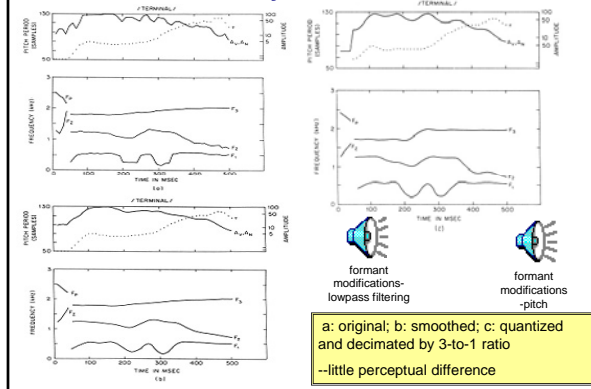
- model parameters estimated at 100/sec rate, lowpass filtered
- SR reduced to twice the LP cutoff and parameters quantized
- parameters could be filtered to 16 Hz BW with no noticeable degradation => 33 Hz SR
- formants and pitch quantized with a linear quantizer; amplitude quantized with a logarithmic quantizer

Quantization of Synthesizer Parameters

Parameter	Required Bits/Sample
Pitch Period (Tau)	6
First Formant (F1)	3
Second Formant (F2)	4
Third Formant (F3)	3
log-amplitude (AV)	2

600 bps total rate for voiced speech with 100 bps for V/UV decisions

Quantization of Synthesizer Parameters



a: original; b: smoothed; c: quantized and decimated by 3-to-1 ratio
--little perceptual difference

Algorithms for Speech Processing

- Based on the various representations of speech we can create algorithms for measuring features that characterize speech and estimating properties of the speech signal, e.g.,
 - presence or absence of speech (Speech/Non-Speech Discrimination)
 - classification of signal frame as Voiced/Unvoiced/Background signal
 - estimation of the pitch period (or pitch frequency) for a voiced speech frame
 - estimation of the formant frequencies (resonances and anti-resonances of the vocal tract) for both voiced and unvoiced speech frames
- Based on the model of speech production, we can build a speech synthesizer on the basis of speech parameters estimated by the above set of algorithms and synthesize intelligible speech