520.445/645 – Audio Signal Processing
☞ Topic 6 ☜

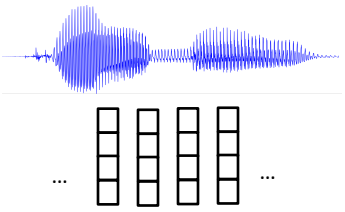
Speech Analysis

Speech analysis

- Speech analysis describes **methods** for mapping the acoustic signal onto a new representation that is more amenable to further processing or information extraction.

Speech analysis

- How to extract features from the signal
- Transform signal into another representation
  - Another signal
  - Another set of signals
  - Vector of parameters

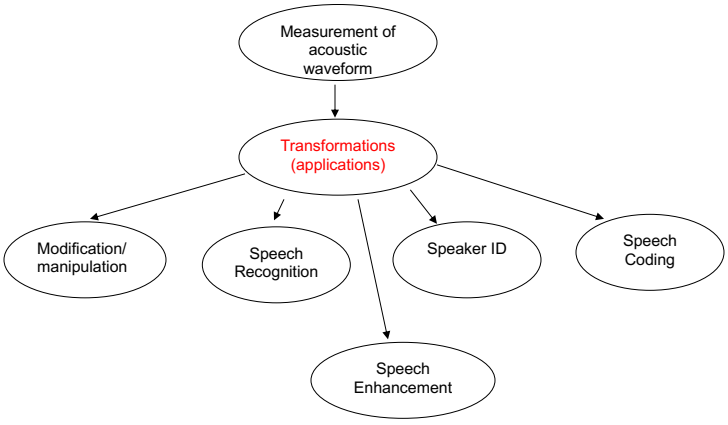


- Objective:
  - simplification
  - Data reduction
  - Reduce redundancy
  - Example: map ~10sec speech signal (100,000's bits) into 20-30 words (100's bits)

3

3

Speech analysis



4

4

Speech analysis

- Different applications require different mappings
  - **Speech recognition**: mapping into text
    - Features invariant to speakers
    - Features enhance variations across phonemes
    - syllable/phoneme boundaries are important
  - **Speech synthesis**: map text into a signal
    - Continuity of parameters over time is crucial
      - not frame-by-frame analysis
  - **Sound compression**: map audio signal into another audio signal (or codebook)
    - Eliminate redundancies

5

5

Speech analysis

- LPC
- Cepstral analysis
- Time-frequency (Spectrogram) analysis

6

6

Linear predictive Coding (LPC)

- Takes advantage of what we know about speech production
- LPC derivation:
  - LPC provides an analysis-by-synthesis system for speech
    - Analyze by knowing how synthesis works

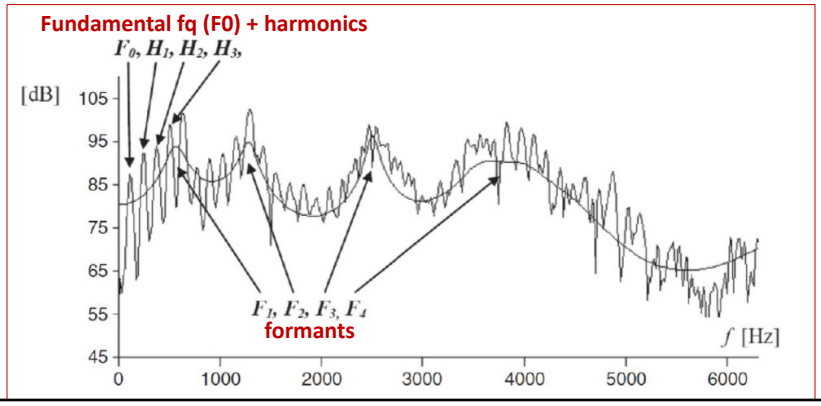
7

7

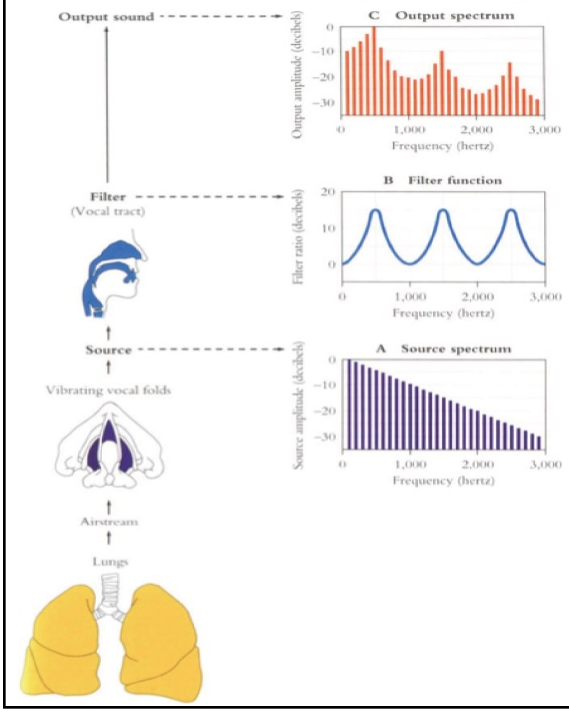
Synthesis Model

- Recall:

  source-filter model uncouples glottis and vocal tract:
  - production approximated by excitation source and spectral shaping filter



8



Speech $\mathbf{X}(\boldsymbol{\omega})$

Filter $\boldsymbol{H}(\boldsymbol{\omega})$

Source $\boldsymbol{U}(\boldsymbol{\omega})$

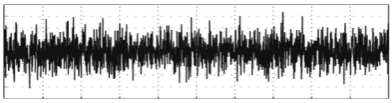
9

Source $\boldsymbol{U}(\boldsymbol{\omega})$

- Recall speech sounds can be:
  - **Voiced:** source is a uniform periodic train
    - periodic at pitch period P

Can vary in complexity from simple impulse train to more realistic glottal waveform

  - **Unvoiced:** source assumed to be flat spectrum noise

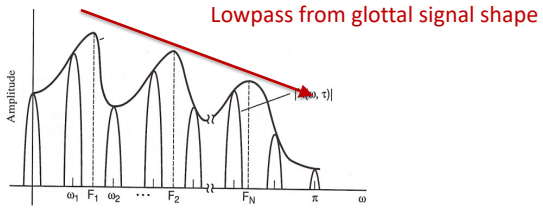


10

10

Source $U(\omega)$

- Spectrum of the source is technically not flat



  - But assumed to be flat (for simplicity).
    - Can incorporate its shape with vocal tract filter

11

11

Vocal tract filter $H(\omega)$

- The filter assumes the system is stationary over a window length N
  - filter has fixed coefficients
  - Filter changes frame-by-frame (window-by-window)

These assumptions can be relaxed

- Filter models formant parameters
- Simplest model assumes all-pole filter

$$H(z) = \frac{1}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_P z^{-P}}$$

$$= \frac{1}{(1 - \alpha_1 z^{-1})(1 - \alpha_2 z^{-1}) \ldots (1 - \alpha_p z^{-1})}$$

Factorized form showing poles

$$= \frac{z^p}{(z - \alpha_1)(z - \alpha_2) \ldots (z - \alpha_p)}$$

12

12

Produced speech X(ω)

- Combining model of source U(ω) and filter H(ω) gives produced speech X(ω)

$$X(z) = U(z).H(z)$$
$$= G.U(z).\frac{1}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_P z^{-P}}$$

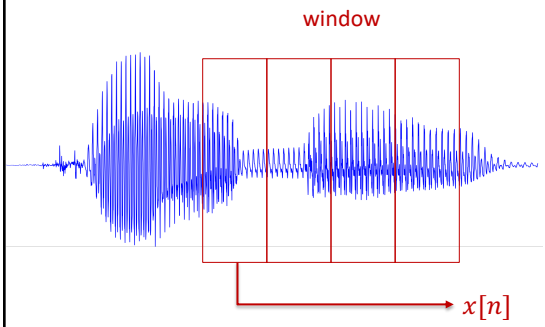
Gain factor

constant (allows speech to have prosody across frames)

13

13

Finding model parameters



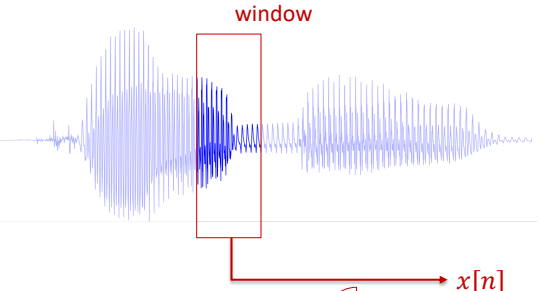
Can we estimate
- Source $u[n]/U(\omega)$
- Filter h$[n]/H(\omega)$

that generated this segment

14

14

Finding model parameters



Z-transform

$$X(z) = G.U(z).\frac{1}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_P z^{-P}}$$

expand

$$X(z) = a_1 z^{-1} X(z) + \cdots + a_p z^{-p} X(z) + G.U(z)$$

Back to time domain

$$\boldsymbol{x[n] = \sum_{k=1}^{p} a_k x[n-k] + Gu[n]}$$

15

15

Finding model parameters

- Define the system

$$x[n] = \sum_{k=1}^{p} a_k x[n-k]$$

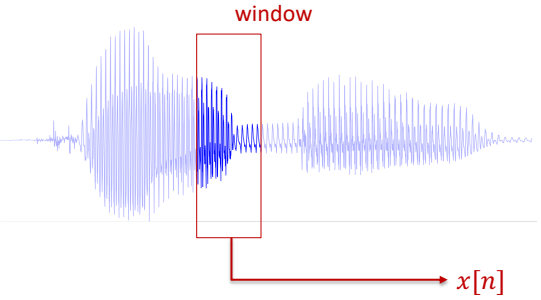
  - Called **autoregressive** (AR) model
  - Outputs are regressively added together
    - Each sample at time n can be **predicted** from a **linear** average of previous samples at times $n-1, n-2, \ldots, n-p$

→ **Linear Predictive Coding (LPC)**

16

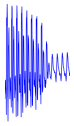
16

Intuition behind LPC



Find coefficients:

- **Source** (either unvoiced –noise- or voiced –periodic signal with period P-
- **Filter** (find gain G and p coefficients a_k)

100's bits  Reduced to  **few bits**

17

17

Intuition behind LPC

- Provides economical and accurate representation of speech parameters (only a_k coefficients)
  - Reduce transmission rates
    - Most common low-bit rate speech coding technique
  - Increase accuracy and reduce calculation in ASR
  - Efficient speech synthesis
- Find small # of parameters that represent the configuration of vocal tract
- In the simplest form, it assumes the vocal tract is an all-pole system.

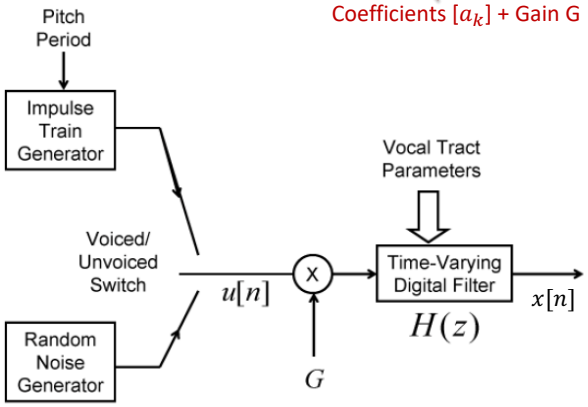
18

18

Linear Predictive Coding

- The goal of LPC is to find $H(z)$ such that:

Coefficients $[a_k]$ + Gain G



19

19

Intuition behind LPC

- LPC is based on this idea;
  - i.e., estimates each sample as linear combination of p pervious samples
  - i.e. estimate of x[n] as $\hat{x}[n] = \sum_{k=1}^{p} a_k x[n-k]$
- That's why it is called LINEAR - PREDICTION
  - The larger p, the more accurate the model is
  - The linear weights are called LPC coefficients (a_k)
  - Once a_k's are known, can be stored as templates for automatic speech processing (e.g. recognition)
  - Or, can be used for efficient coding

20

20

Problems/Drawbacks of LPC

- Speech signal is assumed to come from all-pole source (p **past** samples)
  - Reduces complexity ~approximation
  - Not good model for nasals for example (H(z) should have zeros because of frequency damping through nasal tract)
- Assumption of linear model
- Assumption of stationary system

21

21

How to find the filter $H(z)$; or a_k

- Define the error as:

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

$$= x[n] - \sum_{k=1}^{p} a_k x[n-k]$$

- As the model of H(z) approaches the 'real' vocal tract model, $e[n]$ becomes small

22

22

How to find the filter $H(z)$; or a_k

- If we reproduce the exact model derived earlier:

$$x[n] = \sum_{k=1}^{p} a_k x[n-k] + Gu[n]$$

  - Then, the error term $e[n] = Gu[n]$
  - So, the gain term G can be derived from the RMS (root mean square) of the error between $x[n]$ and $\hat{x}[n]$

23

23

How to find the filter $H(z)$; or a_k

- Knowns:
  - $x[n], x[n-1], x[n-2], \ldots$
- Unknowns:
  - a_k
  - G
- Equation:
  - Minimize: $e[n] = x[n] - \sum_{k=1}^{p} a_k x[n-k]$

24

24

Least-square autocorrelation Method

- Error energy:

$$E = \sum_{n=-\infty}^{\infty} e^2[n] = \sum_{n=-\infty}^{\infty} \left(x[n] - \sum_{k=1}^{p} a_k x[n-k] \right)^2$$

  - ⇨ Find a_k that minimize E
  - ⇨ $\frac{\partial E}{\partial a_k} = 0$, for $k = 1,2,\dots,p$
  - ⇨ Yields p linear equations (with p unknowns a_k):

$$\sum_{n=-\infty}^{\infty} x[n]x[n-i] = \sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} x[n-i]x[n-k], i = 1,2,\dots,p$$

25

25

Least-square autocorrelation Method

- Therefore, the equation:

$$\underbrace{\sum_{n=-\infty}^{\infty} x[n]x[n-i]}_{R[i]} = \sum_{k=1}^{p} a_k \underbrace{\sum_{n=-\infty}^{\infty} x[n-i]x[n-k]}_{R[i-k]}, i = 1,2,\dots,p$$

reduces to:

$$R[i] = \sum_{k=1}^{p} a_k R[i-k], i = 1,2,\dots,p$$

⇨ Solving the equations above yields a_k's.

26

26

Least-square autocorrelation Method

- Equation

$$R[i] = \sum_{k=1}^{p} a_k R[i-k], i = 1,2, \dots, p$$

- Can be written in matrix form as:

$$\underline{r} = \underline{R}a$$

27

27

Least-square autocorrelation Method

- $$\underline{R}a = \underline{r}$$
  - $\underline{R}$ is a pxp matrix; with elements $\underline{R}[i,k] = R(|i-k|), (0 \leq (i,k) \leq p)$
  - $\underline{r}$ is a column vector $\{R[1], R[2], \dots R[p]\}^T$
  - a is a column vector of LPC coefficients

$$\begin{bmatrix} R[0] & R[1] & \dots & R[p-1] \\ R[1] & R[0] & \dots & R[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ R[p-1] & R[p-2] & \dots & R[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R[1] \\ R[2] \\ \vdots \\ R[p] \end{bmatrix}$$

28

28

Least-square autocorrelation Method

$$\begin{bmatrix} R[0] & R[1] & \dots & R[p-1] \\ R[1] & R[0] & \dots & R[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ R[p-1] & R[p-2] & \dots & R[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R[1] \\ R[2] \\ \vdots \\ R[p] \end{bmatrix}$$

- Solving equations requires inverting matrix R
  - ⇨ Not efficient (order p^3 operations)
- Matrix R has some properties which ease computation
  - It is symmetric $R(i,k) = R(k,i)$
  - It is Toeplitz (all elements along a given diagonal are equal)
- Algorithm called levinson-durbin (MATLAB: levinson)
  - Recursive algorithm

29

29

Intuition behind Levinson-Durbin

- Build an "order recursion" in which solution for iteration $i+1$ can be found form solution of the i-th order iteration
- Each recursion solves the i-th order prediction problem (i.e. i-th pole modeling)

1-pole ⇨ $a_1^{(1)}, E^{(1)}$

2-pole ⇨ $a_1^{(2)}, a_2^{(2)}, E^{(2)}$

⋮

p-pole ⇨ $a_1^{(p)}, a_2^{(p)}, \dots, a_p^{(p)}, E^{(p)}$

30

30

Durbin's Recursion

- Each recursion solves the i-th pole (or i-th order prediction) problem:

0. Initialize prediction error $E^{(0)} = R(0)$

1. Derive a constant for i-th pole model:

$$k_i = \frac{R(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)}{E^{(i-1)}}$$

k_i: called reflection coefficients of partial correlation coefficients

2. Update coefficients for the i-th pole model:

$$a_i^{(i)} = k_i \quad , \quad a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \quad , \qquad 1 \le j \le i-1$$

3. Form prediction error for the i-th pole model:

$$E^{(i)} = (1 - k_i^2)\, E^{(i-1)}$$

4. Repeat steps 1-3 for $i = 1, 2, \ldots, p$

Example Durbin's recursion (Case: p=2)

- Original matrix equation

$$\begin{bmatrix} R(0) & R(1) \\ R(1) & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \end{bmatrix}$$

  - $i = 0$,

$$E^{(0)} = R(0)$$

  - $i = 1$,

$$k_1 = \frac{[R(1) - \sum_{j=1}^{0} X]}{E^{(0)}} = \frac{R(1)}{R(0)}$$

$$a_1^{(1)} = k_1$$

$$\begin{aligned} E^{(1)} &= (1 - k_1^2) E^{(0)} \\ &= \left(1 - \frac{R^2(1)}{R^2(0)}\right) R(0) \\ &= \frac{R^2(0) - R^2(1)}{R(0)} \end{aligned}$$

Example Durbin's recursion (Case: p=2)

- $i = 2$,

$$k_2 = \frac{[R(2) - \sum_{j=1}^{1} a_j^{(1)} R(i-j)]}{E^{(1)}} = \frac{[R(2) - a_1^{(1)} R(1)]}{E^{(1)}} = \frac{R(0)R(2) - R^2(1)}{R^2(0) - R^2(1)}$$

$$a_2^{(2)} = k_2, \qquad a_1^{(2)} = a_1^{(1)} - k_2 a_1^{(1)}$$

$$E^{(2)} = (1 - k_2^{\,2}) E^{(1)}$$

33

33

Least-square autocorrelation Method

- Overall, the autocorrelation method uses two distinct steps:
  1. Computation of a matrix of correlation values
  2. Efficient solution of a set of linear equations
- Note: there are similar approach for LPC estimation based on other measures (e.g. covariance matrix)

34

34

LPC coefficients

- LPC algorithms derive the values of $\{a_k\}$ for a chosen filter order p.
- The gain can be derived as the RMS of error signal $e[n]$
- The source will be either a periodic signal (voiced) or noise (unvoiced)
  - Sophisticated models make this decision less binary

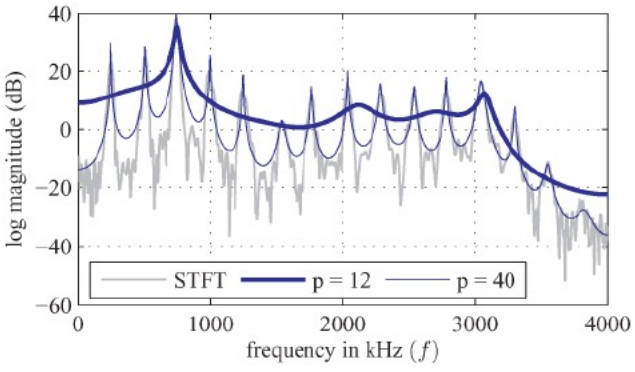
35

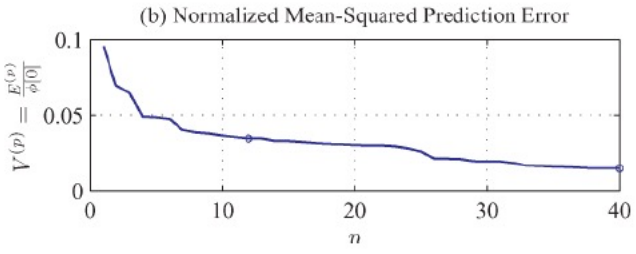
35

Effect of filter order (p)

original

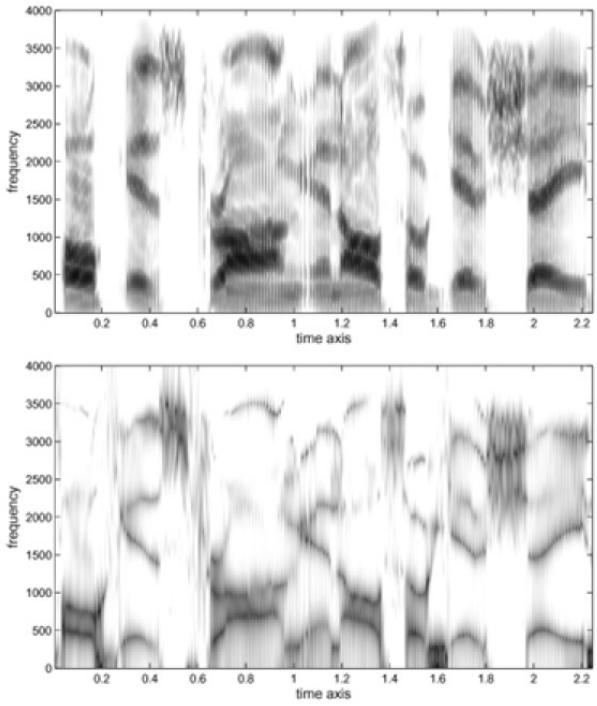
LPC version





36

Spectrogram using LPC



37

Extending LPC model

- Recall, the model is defined as:

$$X(z) = G.H(z).U(z)$$

where $$H(z) = \frac{1}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_P z^{-P}}$$ All-pole model

- In its general form

$$H(z) = \frac{1 - b_1 z^{-1} - b_2 z^{-2} - \cdots - b_Q z^{-Q}}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_P z^{-P}}$$ Pole-zero model

38

38

General model

- From the general model,

$$H(z) = \frac{1 - b_1 z^{-1} - b_2 z^{-2} - \cdots - b_Q z^{-Q}}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_P z^{-P}}$$

- We get a difference equation

$$x[n] = \sum_{k=1}^{p} a_k x[n-k] + G \sum_{l=0}^{q} b_l u[n-l]$$

Current sample — P past samples — Glottal input

39

39

General model

- General model:

$$\hat{x}[n] = \sum_{k=1}^{p} a_k x[n-k] + G \sum_{l=0}^{q} b_l u[n-l]$$

- An **all-zero** model would correspond to:

$$\hat{x}[n] = G \sum_{l=0}^{q} b_l u[n-l]$$

  - ⇨ $p = 0$, no poles
  - ⇨ It is called a **MA (Moving-average)** model
  - ⇨ Output is weighted average of q prior inputs

40

40

General model

- General model:

$$\hat{x}[n] = \sum_{k=1}^{p} a_k x[n-k] + G \sum_{l=0}^{q} b_l u[n-l]$$

- An **all-pole** model would correspond to:

$$\hat{x}[n] = \sum_{k=1}^{p} a_k x[n-k]$$

  - $q = 0$, no zeros
  - It is called an **AR (autoregressive)** model
  - Outputs are regressively added together

41

41

General model

- General model:

$$\hat{x}[n] = \sum_{k=1}^{p} a_k x[n-k] + G \sum_{l=0}^{q} b_l u[n-l]$$

  - $p > 0, q > 0$
  - It is called an **ARMA (autoregressive, moving average)** model

42

42

LPC – Final thoughts

- LPC operates on all-pole model
  - Deals with p linear equations
- ARMA model (pole-zero)
  - Would mean solving nonlinear equations
  - Improves performance *only slightly*
  - Can do 2-step procedure:
    - All pole estimation first, then residual signal (contains effect of zeros) estimated with a second AR model

43

43

Pre-emphasis

- In voiced sounds, vocal tract spectrum is lowpass (rolloff -6dB/octave)
  - So, high frequencies are very weak
  - Use pre-emphasis to compensate for falloff

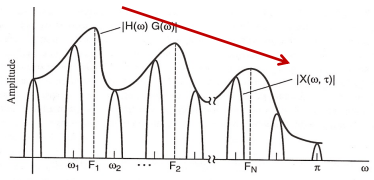


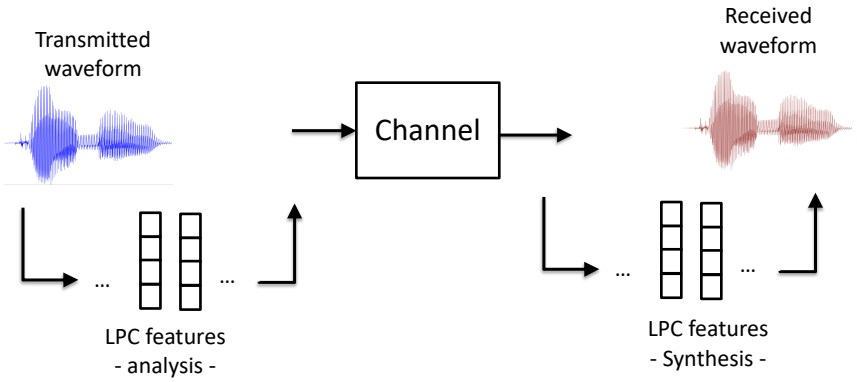
Figure 3.11 Illustration of relation of glottal source harmonics $\omega_1, \omega_2, \ldots \omega_N$, vocal tract formants $F_1, F_2, \ldots F_M$, and the spectral envelope $|H(\omega)G(\omega)|$.

- Solution:
  - Filtering signal with (first-order) high-pass filter with $H(z) = 1 - Az^{-1}$ with $A \in [0,1[$.
  - The closer A is to 1, the greater the pre-emphasis (A is a zero of H(z))
  - The pre-emphasis filter is applied on the input signal before windowing.
  - After reconstruction, sometime use de-emphasis filter to undo effect of pre-emphasis
  - Ideally, pre-emphasis should be done ONLY for voiced speech (with freq rolloff)
  - However, pre-emphasizing unvoiced speech sometimes doesn't affect analysis much
  - Or choose a compromise value of A (not too much pre-emphasis, ~A=0.9)

44

44

LPC in a practical transmission channel
* Project 2 *



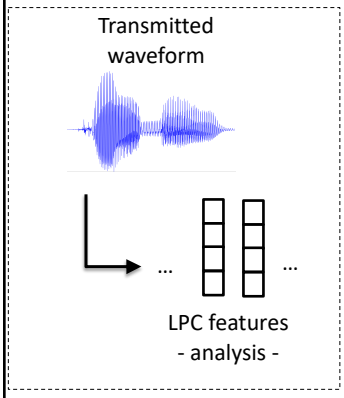
Balance:

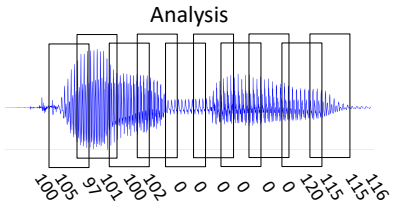
- Low bitrate (most compression)
- Good quality synthesis

45

45

LPC in a practical transmission channel
* Project 2 *





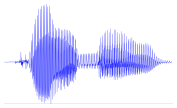
- Choose analysis frames (length, overlap ..)
- For each frame, extract LPC coefficients
  - ✓ Decide how many poles (p) to you use
  - ✓ Use MATLAB function *lpc* which gives you the coefficients a_k and variance of the error E which gives you the gain G
- For each frame, you want to also store information about voicing
  - ✓ If invoiced -> 0
  - ✓ If voiced -> pitch frequency

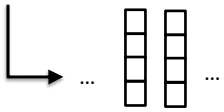
46

46

LPC in a practical transmission channel
* Project 2 *

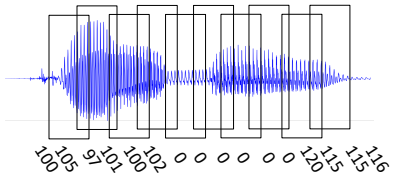
Transmitted waveform



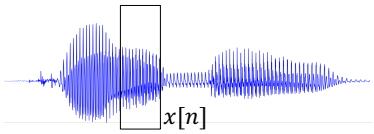


LPC features
- analysis -

Analysis



- Recall: goal of LPC

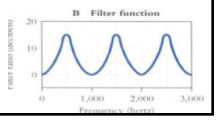


For m[th] frame: $x[n]$; find the best p coeff. a_k such that

$$x[n] = \sum_{k=1}^{p} a_k x[n-k]$$

i.e. → All pole filter? →

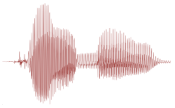
What is filter envelope?



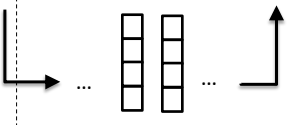
47

LPC in a practical transmission channel
* Project 2 *

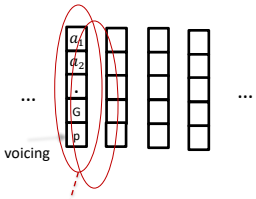
Received waveform

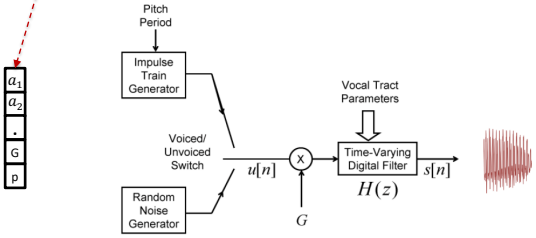


Channel



LPC features
- Synthesis -





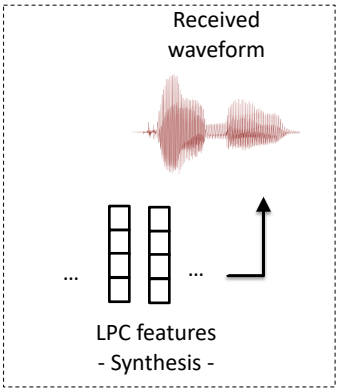
48

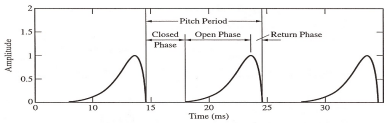
48

LPC in a practical transmission channel
* Project 2 *

- Caution/extensions:
  - ✓ Alignment of frames during synthesis

  - ✓ You can use different # poles for different frames but you need to tell receiver how many poles to expect
  - ✓ Choice of frame length
    - Short-enough to assume stationary vocal tract
    - Long-enough to get a reasonable estimate
  - ✓ You can extend impulse train to more realistic glottal waveform
  - ✓ Explore error term, can you encode it as well? Does it add anything to synthesized quality at expense of bit rate?
  - ✓ .....

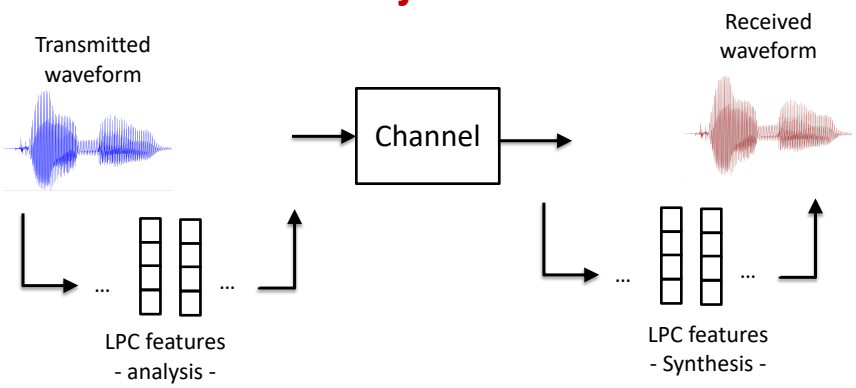




49

49

LPC in a practical transmission channel
* Project 2 *



Balance:

- **Low bitrate (most compression)**
- **Good quality synthesis**

50

50

Homomorphic Signal processing / Cepstral analysis

- Speech analysis => estimate parameters based on production system
- Production system (commonly used):
  - Excitation source: random noise or quasi-periodic pulses
  - Convolved with linear, time-varying system (vocal tract)

$$x[n] = u[n] * h[n]$$

⇨Estimating these parameters

⇨Try deconvolve the 2 components

51

51

Homomorphic Filtering

- Generally, deconvolving two signals is difficult
- Homomorphic filtering is a method of **deconvolving** two components
  - GENRAL FORM: apply a nonlinear mapping to a different domain
  - Deconvolution in old domain maps to a linear operation in the new domain
  - followed by mapping back to the original domain

52

52

Homomorphic Filtering for speech

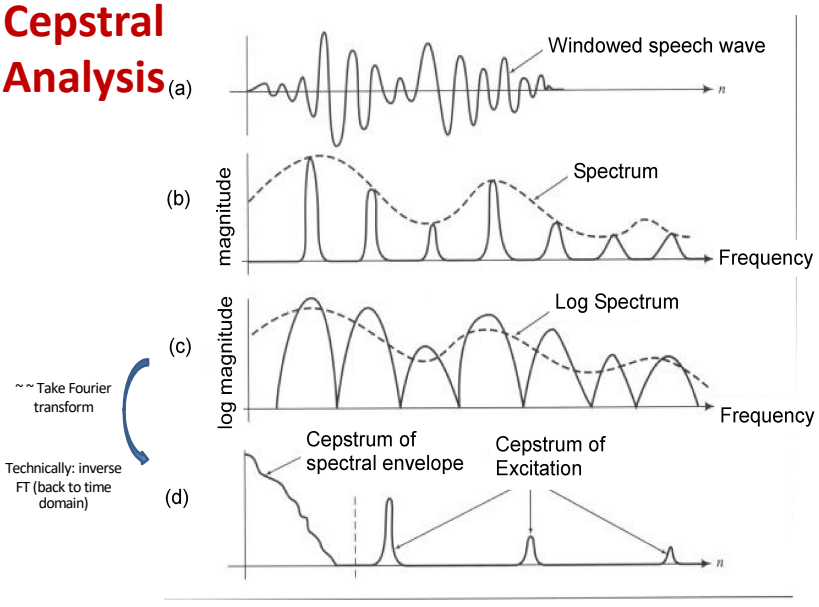
- Speech signals
  - Time domain: $x[n] = u[n] * h[n]$
  - Transform domain $X(z) = U(z)H(z)$
- Deconvolution of speech is feasible because
  - $U(z)$ and $H(z)$ have different spectra
    - $U(z)$: Excitation harmonics or noise varies much faster
    - $H(z)$: vocal tract response (formant structure) varies very slowly in frequency
  - The contribution of $U(z)$ and $H(z)$ could be separated if they are in a linear domain
  - Use trick

$$X(z) = U(z)H(z)$$
$$logX(z) = logU(z) + logH(z)$$

53

53

Cepstral Analysis



54

54

Derivation – Real cepstrum

- The spectral magnitude of a speech signal can be written as:

$$|X(z)| = |U(z)||H(z)|$$
$$log|X(z)| = log|U(z)| + log|H(z)|$$

  - Want to pass $log|X(z)|$ through a filter that would separate contribution of slow envelope components ($H(z)$) from excitation source ($U(z)$)
    - If $log|X(z)|$ was a time signal, could take low-pass or high-pass => spectral analysis
    - Instead, we are now in the CEPSTRUM domain => cepstral analysis
    - Similar concept: slow and fast components are separated and can be filtered out to separate them from each other

55

55

Real cepstrum

- $$c[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} log|X(\omega)|e^{j\omega n}d\omega$$
- $c[n]$: called n^{th} cepstral coefficients
- Not invertible [phase is missing]
- Note:
  - Can use DFT, FT or Z-transform

56

56

Complex cepstrum

- Gain more insight by using full complex-valued spectrum $X(\omega)$ instead of $|X(\omega)|$
- The complex cepstrum is defined as:

$$\hat{x}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} log X(\omega) e^{j\omega n} d\omega$$

  - General form of cepstrum (use mag + phase)
  - Use complex log, defined as: $\log|X(\omega)| + j < X(\omega)$
    - Complex cepstrum of a real sequence is also a real sequence

57

57

Complex cepstrum

- The complex cepstrum exists if the complex log has a convergent power series representation

$$\hat{X}(z) = \log[X(z)] = \sum_{n=-\infty}^{\infty} \hat{x}[n] z^{-n}$$

  - So $\log[X(z)]$ must have the properties of the z-transform of a stable sequence.

58

58

Terminology

- *Spectrum* – Fourier transform of signal
- *Cepstrum* – Inverse Fourier transform of log spectrum
- *Filtering* – operations on spectrum
- *Liftering* – operations on cepstrum
- *Frequency* – basic unit of spectrum
- *Quefrency* – basic unit of cepstrum
- *Analysis* – operation on spectrum
- *Alanysis* – operation on cepstrum

59

59

Summary

- To the extent that $\hat{X}(z) = \log[X(z)]$ is valid,

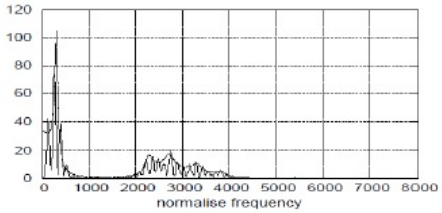
$$\begin{aligned} \hat{x}[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{X}(\omega)e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} log\ X(\omega)e^{j\omega n}d\omega \\ c[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} log\ |X(\omega)|e^{j\omega n}d\omega \end{aligned}$$

- There is a relationship between $c[n]$ and $\hat{x}[n]$ ($c[n]$ even part of $\hat{x}[n]$)

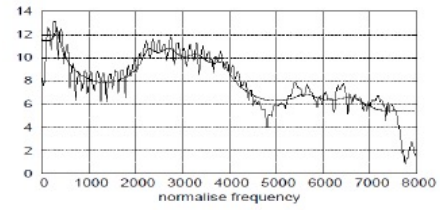
60

60

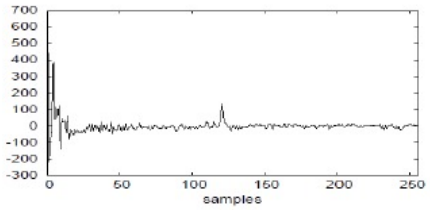
$\mathcal{F}\{x[n]\}$



$\log|\mathcal{F}\{x[n]\}|$



$\mathcal{F}^{-1}\{\log|\mathcal{F}\{x[n]\}|\}$



61

61

Side-notes

Cepstral proof

- If we look at speech signal $x[n]$ in z-domain

$$X(z) = G\frac{\prod_{k=1}^{M}(1-q_k z^{-1})}{\prod_{k=1}^{N}(1-p_k z^{-1})}$$ (Rational function)

  - What happens to the poles and zeros of the signal during cepstral analysis?

62

62

Side-notes

Appendix: Poles & zeros

- Taking the log: $$X(z) = G\frac{\prod_{k=1}^{M}(1-q_k z^{-1})}{\prod_{k=1}^{N}(1-p_k z^{-1})}$$
- We get:

$$\begin{aligned} logX(z) &= \log(G) + \log(1-q_1 z^{-1}) + \cdots + \log(1-q_M z^{-1}) - \log(1-p_1 z^{-1}) - \cdots \log(1-p_N z^{-1}) \\ &= \log(G) + \sum_{k=1}^{M} \log(1-q_k z^{-1}) - \sum_{k=1}^{N} log(1-p_k z^{-1}) \\ &= \log(G) - \sum_{k=1}^{M} \log\left(\frac{1}{1-q_k z^{-1}}\right) + \sum_{k=1}^{N} log(\frac{1}{1-p_k z^{-1}}) \end{aligned}$$

63

63

Side-notes

Appendix: Poles & zeros

- How does log(.) affect the poles and zeros?
  - Taylor series expansion:

$$\log\left(\frac{1}{1-x}\right) = x + \frac{x^2}{2} + \frac{x^3}{3} + \cdots + \frac{x^k}{k} + \cdots \qquad |x| < 1$$

  - We have 3 terms for $logX(z)$:

$$\log(G) - \sum_{k=1}^{M} \log\left(\frac{1}{1-q_k z^{-1}}\right) + \sum_{k=1}^{N} log(\frac{1}{1-p_k z^{-1}})$$

Split analysis for poles/zeros inside unit circle so that |.|<1

64

64

Side-notes

Appendix: Poles & zeros

- For poles and zeros inside unit circle:

$$\log\left(\frac{1}{1-p_k z^{-1}}\right) = \sum_{n=1}^{\infty} \frac{{p_k}^n}{n} z^{-n} \quad , |z| > |p_k|$$

$$\log\left(\frac{1}{1-q_k z^{-1}}\right) = \sum_{n=1}^{\infty} \frac{{q_k}^n}{n} z^{-n} \quad , |z| > |q_k|$$

65

65

Appendix: Poles & zeros

Corresponding time signal: $\frac{p_k^n}{n} u[n-1]$

- $$logX(z) = \log(G) - \sum_{k=1}^{M}\sum_{n=1}^{\infty} \frac{{q_k}^n}{n} z^{-n} + \sum_{k=1}^{N}\sum_{n=1}^{\infty} \frac{{p_k}^n}{n} z^{-n}$$

$$\hat{x}[n] = \begin{cases} \log(G) & , \quad n = 0 \\ \sum_{k=1}^{N} \frac{{p_k}^n}{n} - \sum_{k=1}^{M} \frac{{q_k}^n}{n} & , \quad n = 1,2,\ldots \end{cases}$$ ← *u[n-1]*

- Each stable pole contributes a positive decaying exponential (weighted by $\frac{1}{n}$) to the complex cepstrum (for $n > 0$)
- Each zero contributes a negative weighted exponential of the same type

66

66

Side-notes

Appendix: Poles & zeros outside unit circle

- For poles and zeros outside unit circle, rewrite equation as:

$$\log\left(\frac{1}{1-p_k z^{-1}}\right) = \log\left(\frac{{p_k}^{-1} z}{{p_k}^{-1} z - 1}\right)$$
$$= \log\left(\frac{-z}{p_k}\right) + \log\left(\frac{1}{1-{p_k}^{-1} z}\right)$$
$$= \log\left(\frac{-z}{p_k}\right) + \sum_{n=1}^{\infty} \frac{{p_k}^{-n}}{n} z^n$$
$$, |z| < |{p_k}^{-1}|$$

- Rewrite:

$$\sum_{n=1}^{\infty} \frac{{p_k}^{-n}}{n} z^n = -\sum_{n=-\infty}^{-1} \frac{{p_k}^{n}}{n} z^{-n}$$

Anticausal components

- Poles and zeros outsisde unit circle contribute anticausal exponentials to the complex cesptrum (n<0), negative for poles and positive for zeros

67

67

Poles & Zeros of the cepstrum

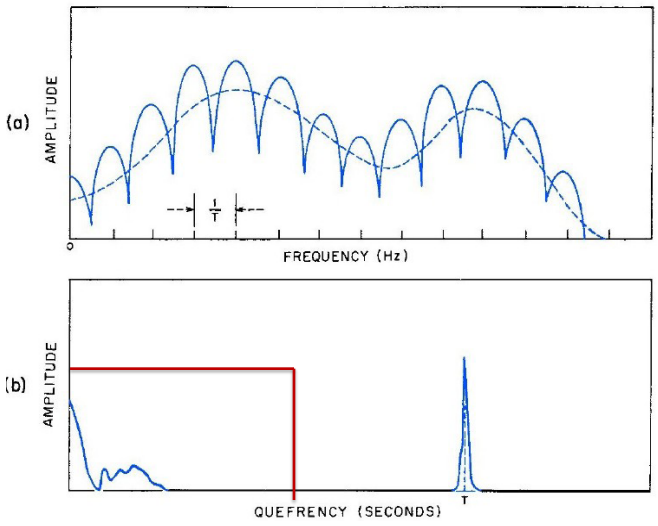
- Poles contribute decaying exponentials with a $1/n$ factor:
  - → $\hat{h}[n]$ Decays rapidly within a few milliseconds
- Excitation $u[n]$ (voiced speech) retains same form as $\hat{u}[n]$
  - Excitation is a uniform train of impulses with freq spacing $2\pi/P$
  - Taking the log doesn't change freq spacing, just amplitude (area of impulses)
  - So, complex cepstrum $\hat{u}[n]$ retains same form as $u[n]$ (an impulse train with period P)

- Can separate the two functions using a window
- Window should be about shortest possible pitch period

68

68

Example



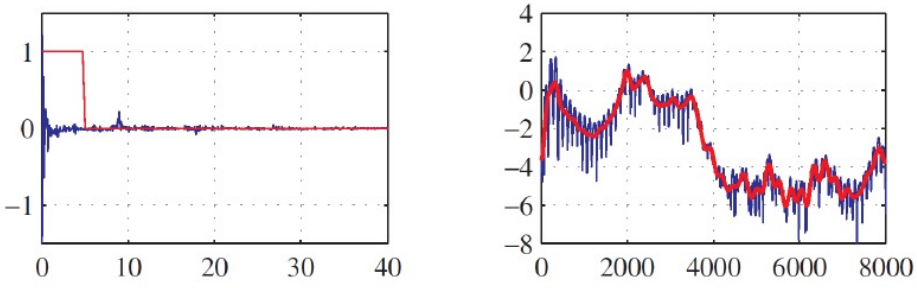
Mag spectrum

cepstrum

69

69

Estimation of filter



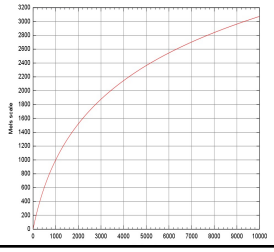
70

70

Mel Cepstrum

- In many applications, use different frequency axis (nonlinear)
  - Approximate biological freq analysis
- Recall
  - Basilar membrane is a frequency axis which is not linear
  - **mel scale** (comes from melody scale) related to critical band and perceptual scales of pitch judgments by human listeners
  - Related to absolute frequency as:

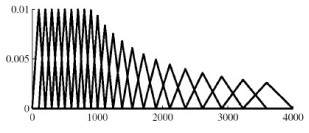
$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$



71

Mel Cepstrum

- Many applications use cepstral analysis on a mel scale
  - Leads to **M**el-**F**requency **C**epstral **C**oefficients (MFCC)
- Convert linear frequency to mel frequency
  - Achieved by integrating spectrum over rectangular windows
  - These windows are non-uniformly spaced in frequency
  - Correspond to different critical bands
  - Typically use 20-40 filters



72

72

Mel Cepstrum

- Many applications use cepstral analysis on a mel scale
  - Leads to **M**el-**F**requency **C**epstral **C**oefficients (MFCC)
- MFCC operations
  1. FT (or DFT) of signal
  2. Magnitude or power spectrum (square) in case of real cepstrum
  3. Map frequency axis into mel-frequency via Mel filterbank
  4. Take log of spectrum
  5. Take inverse FT (or iDFT) to get to cepstrum domain
  6. Keep first 12/13 coefficients (corresponding to parameters of vocal tract filter $\sim h[n]$)
     - Initial coefficient c_0 is average energy (DC term) – could be used to normalize energy

In practice, we replace inverse FT with Discrete Cosine Transform (DCT)

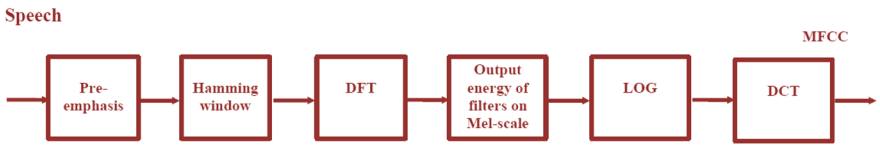
✓ Fourier mapping leads to highly correlated coefficients

✓ DCT decorrelates the coefficients which is useful for machine learning

73

73

Mel cepstrum

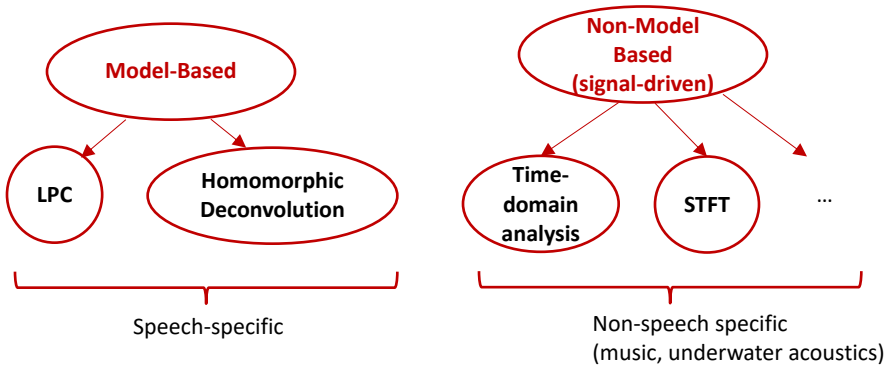


⇨MFCC is the most widely used mapping of speech signals in many technologies

74

74

Analysis/Synthesis – Overview of techniques



75

75

Time-domain parameters

- Example of time-domain parameters
  - Energy (time) can be used to segment signals (for ASR)
  - Voicing and F0 can be estimated in time

76

76

Short-term Energy/Magnitude

- Measure of **energy** in signal

$$E[n] = \sum_{m=-\infty}^{\infty} x^2[m]w[n-m]$$

  - Emphasizes high amplitudes in signal (because of $x^2[.]$)
- Measure of **amplitude** in signal

$$M[n] = \sum_{m=-\infty}^{\infty} |x[m]|w[n-m]$$

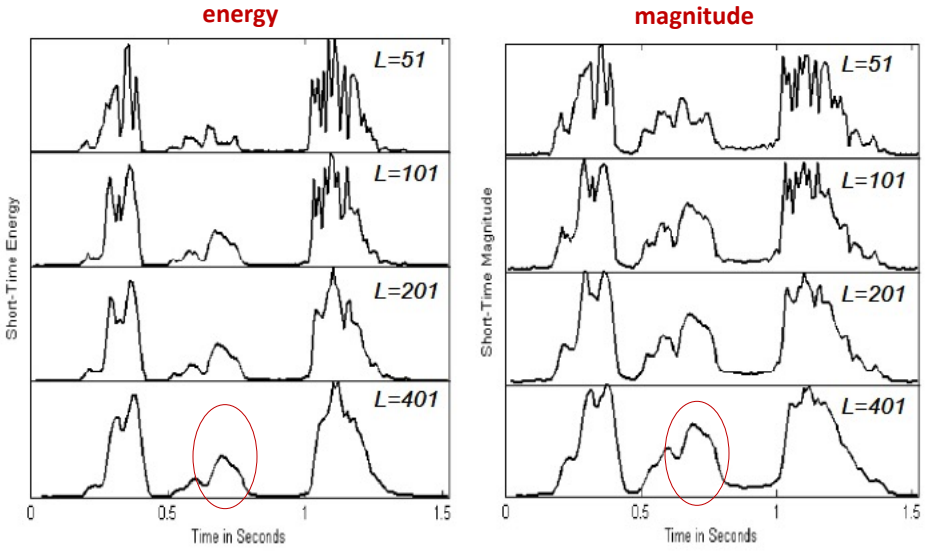
  - No such emphasis high amplitudes

- Useful measures to segment signal
  - syllabic/phonetic boundaries

77

77

Example: /what she said/



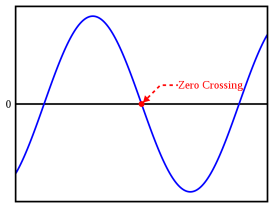
Differences between E[n] and M[n] noticeable especially in unvoiced regions

78

78

Short-time average zero-crossing rate (ZCR)

- Measuring zero-crossing (ZCR) in time-domain
  - Correlates with spectral information (frequency of variation in the signal over time)
  - Correlates with frequency with strongest concentration of energy
- ZCR is number of times $x[n] = 0$
- Defined as:



$$ZCR[n] = \frac{1}{2}|sgn\{x[n]\} - sgn\{x[n-1]\}|$$

Where: $$sgn\{x[n]\} = \begin{cases} +1 & x[n] \geq 0 \\ -1 & x[n] < 0 \end{cases}$$

79

79

Zero-crossings

Voiced speech:

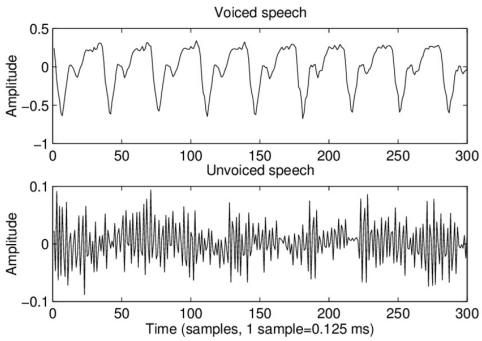
- Spectrum of voiced speech tends to have a lowpass-like shape
  - Vocal tract long (no/mild constriction)
- Major energy (low freq range) is F1
  - ZCR corresponds mostly to F1

➡ Low ZCR

Unvoiced sounds,

- Broadband noise excitation
  - Excites mostly high frequencies
    - Vocal tract shorter (frication at constriction)
  - Unvoiced speech tends to have high frequencies

➡ High ZCR

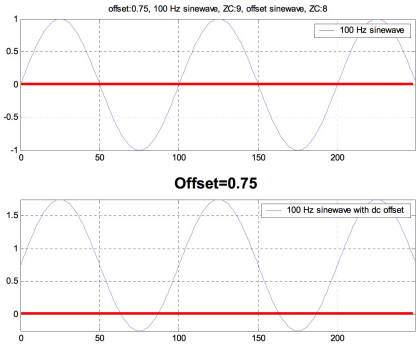


80

80

Problems with ZCR

- But, could be problematic for voiced fricatives
  - Low freq from voiced part /High freq from unvoiced (noise) energy
- Very dependent on signal normalization and pre-emphasis



- ZCR affected greatly with noise
  - If recording in environment ~60Hz

81

81

Autocorrelation

- Autocorrelation measures similarity of signal with shifted version of itself

$$R[k] = \sum_{n=-\infty}^{\infty} x[n]x[n-k]$$

- $R[0]$ is the signal energy
- Autocorrelation of speech can be (and is often) measured over short windows

$$R_m[k] = \sum_{n=-\infty}^{\infty} (x[n]w[m-n])(x[n-k]w[m-n+k])$$

82

82

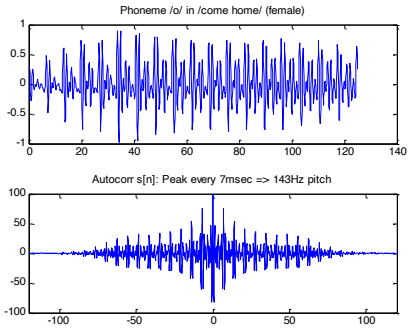
Autocorrelation

- Autocorrelation has many nice properties
  - $R_m[k]$ is an even function
    $$R_m[k] = R_m[-k]$$
  - It has zero phase (does not preserve the phase)
  - Its Fourier transform is the energy spectrum $|X(\omega)|^2$
  - In voiced speech segments, autocorrelation yields peaks corresponding to periodicity in signal

83

Cross-correlation/Autocorrelation Function

- For a **periodic signal** $x[n]$ with period P, the auto-correlation function has peaks (maxima) at $0, +/-P. +/-2P$, etc



- For short-term auto-correlation, the window has to be chosen to include more than one pitch period

84

Frequency-domain paramaters

- Frequency domain can be more informative than time-domain
  - Human ear pays much attention to spectra (tonotopy)
    - E.g. short-term Fourier transform (STFT)
      - Basically a Fourier transform operating on short-term windows

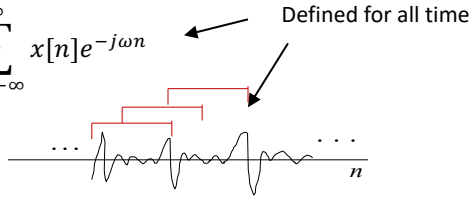
85

85

Short Term Fourier Transform

- **Discrete Time Fourier Transform**

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$



- **Define: Discrete Time Short-term Fourier Transform (Set of Fourier Transforms)**

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega n}$$

Now, we modify this notation in 2 ways

- Discretize frequency axis ω
- Decimate time-axis n

86

86

Discrete Short-Time Fourier Transform

➢ **Discretize the STFT**

- ~ DFT (discrete FT) $X(k) = X(\omega)\Big|_{k=\frac{2\pi k}{N}}$
- Likewise:

$$X(n,k) = X(n,\omega)\Big|_{k=\frac{2\pi k}{N}}$$

$$\Longrightarrow \quad X(n,k) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\frac{2\pi k}{N}m}$$

87

87

Time Decimation

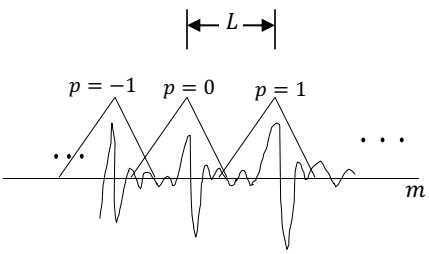
- Next: Add decimation (window step) in time domain

$$X(pL,k) = X(n,\omega)\Big|_{n=pL}$$

$$X(pL,k) = \sum_{m=-\infty}^{\infty} x[m]w[pL-m]e^{-j\frac{2\pi k}{N}m}$$

$n = pL$

$p = \cdots, -1, 0, 1, \ldots$

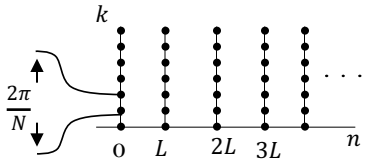


88

88

Discrete Short-Time Fourier Transform

-
- **Time Frequency Plot**

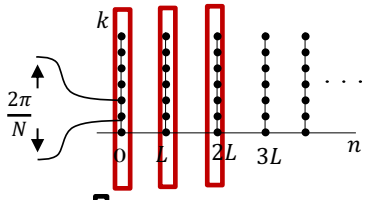


- Two interpretations:
  1. Fourier transform (i.e. STFT as a set of DFTs over many frames)
  2. Filter-bank view

89

89

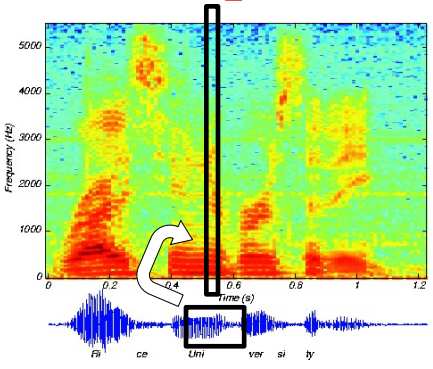
Discrete STFT



- **View 1:** Fourier transform
  - (column-wise)
  - Each column is the DFT of a corresponding frame

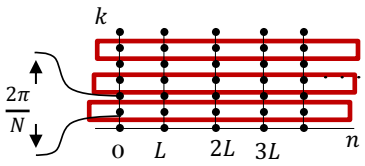
⇩

Familiar interpretation

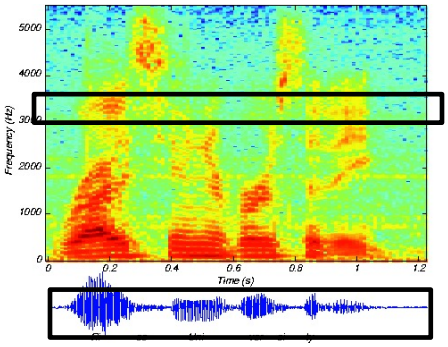


90

Discrete STFT



- **View 2:** Filterbank
  - (row-wise)
  - Entire signal is represented in each row



91

Filter Bank View (view 2)

- For a given frequency (Fix ω at ω_0 - pick one channel/row):

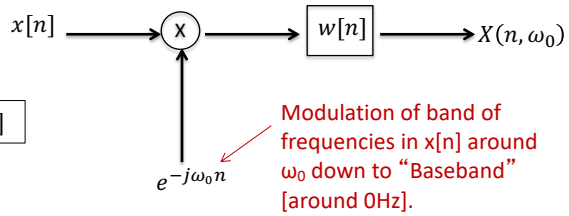
$$X(n,\omega_0) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega_0 m}$$
$$= \sum_{m=-\infty}^{\infty} \left[x[m]e^{-j\omega_0 m}\right]w[n-m]$$
$$= \left[x[n]e^{-j\omega_0 n}\right] * w[n]$$

⇨Analysis window $w[n]$ plays role of filter impulse response => "analysis filter"

92

92

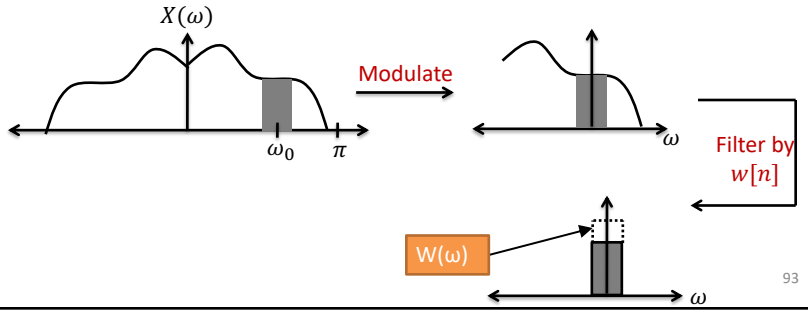
Filter Bank View



$$X(n, \omega_0) = \left[x[n]e^{-j\omega_0 n}\right] * w[n]$$

Modulation of band of frequencies in x[n] around ω_0 down to "Baseband" [around 0Hz].

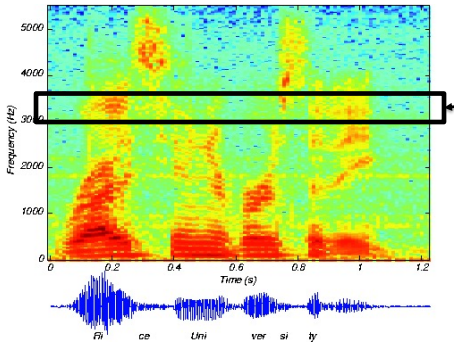
⇨ Graphically:



93

93

Filter Bank View



$$X(n, \omega_0) = \left[x[n]e^{-j\omega_0 n}\right] * w[n]$$

The time signal is modulated then filtered

94

94

Filter Bank view - variation

- Variation:

$$X(n,\omega_0) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega_0 m}$$

$$= \sum_{m'=-\infty}^{\infty} x[n-m']w[m']e^{-j\omega_0(n-m')}$$

$m' = n - m$

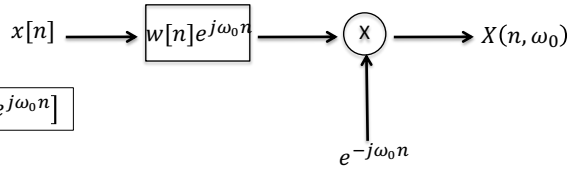
$$= e^{-j\omega_0 n}\sum_{m'=-\infty}^{\infty} x[n-m']w[m']e^{-j\omega_0 m'}$$

$$= e^{-j\omega_0 n}\left[x[n] * w[n]e^{j\omega_0 n}\right]$$

95

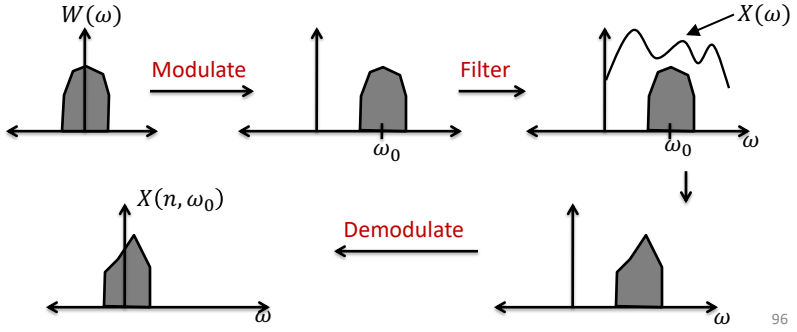
95

Filter Bank view - variation



$X(n,\omega_0) = e^{-j\omega_0 n}\left[x[n] * w[n]e^{j\omega_0 n}\right]$

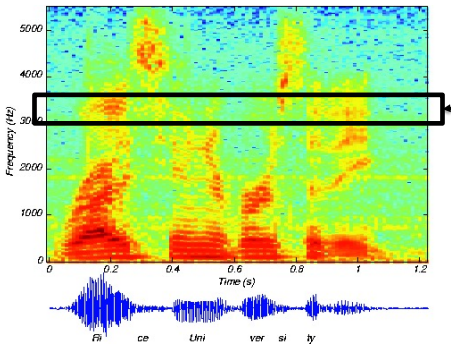
⇨ Graphically:



96

96

Filter Bank View



$$X(n,\omega_0) = e^{-j\omega_0 n}\left[x[n] * w[n]e^{j\omega_0 n}\right]$$

The window is modulated, used to filter the time signal, and the output is modulated-down

97

97

Filter Bank view – Discrete representation

- The earlier analysis deals with continuous-frequency ω
- The same interpretation holds if we discretize the frequency axis

$$X(n,k) = \left[x[n]e^{-j\frac{2\pi.k}{N}n}\right] * w[n]$$

or

$$X(n,k) = e^{-j\frac{2\pi k}{N}n}\left[x[n] * w[n]e^{j\frac{2\pi k}{N}n}\right]$$

$$k = 0,1,\dots,N-1$$

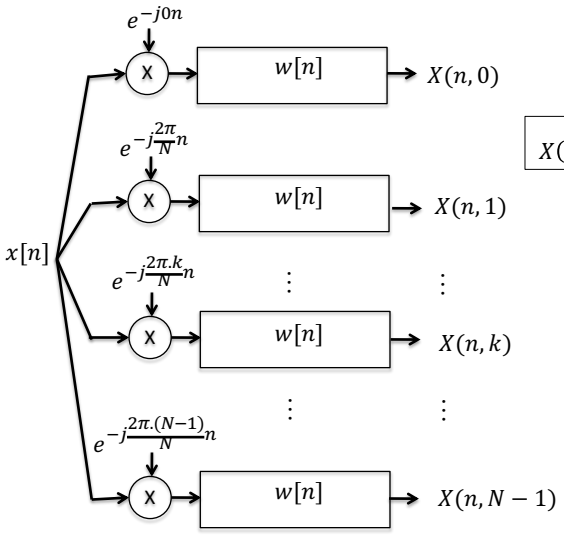
⇨Each filter acts as a bandpass filter centered around its selected frequency

Note: Time decimation can be thought of as sampling output of each filter

98

98

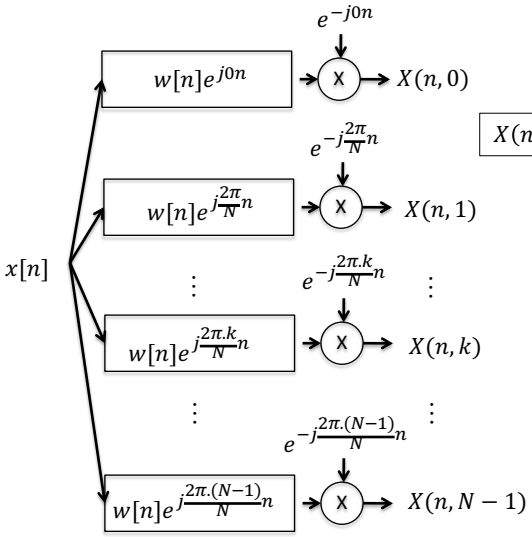
Discrete STFT as a bandpass signal filtered through a lowpass filter



$$X(n,k) = \left[x[n]e^{-j\frac{2\pi.k}{N}n}\right] * w[n]$$

99

Discrete STFT as output of a filter bank of bandpass filters



$$X(n,\omega_0) = e^{-j\omega_0 n}\left[x[n] * w[n]e^{j\omega_0 n}\right]$$

100

100

Properties of discrete STFT

- **Fourier Transform view**
  - $X(n,\omega)$ is a Fourier transform for each fixed n (short-time section)
  - For each n, $X(n,\omega)$ has all properties of a Fourier transform
  - For sampled $\omega = \frac{2\pi}{N}k$, $X(n,k)$ has properties of DFT
- **Filter-Bank view**
  - $X(n,\omega)$ is a filter output for each fixed ω
  - For each fixed ω, $X(n,\omega)$ has all properties of a filtered sequence
  - For decimated $n = pL$, $X(pL,\omega)$ has properties of a time-decimated sequence

Discrete STFT

101

101

Properties of discrete STFT

TABLE 6.1 PROPERTIES OF THE DISCRETE-TIME STFT BASED ON THE FOURIER TRANSFORM INTERPRETATION

Property 1:	$X(n,\omega) = X(n,\omega + 2\pi)$
Property 2:	$x(n)$ real $\longleftrightarrow X(n,\omega) = X^*(n,-\omega)$
Property 3:	$x(n)$ real $\longleftrightarrow \lvert X(n,\omega)\rvert = \lvert X(n,-\omega)\rvert$
Property 4:	$x(n)$ real $\longleftrightarrow \arg[X(n,\omega)] = -\arg[X(n,-\omega)]$
Property 5:	$x(n-n_0) \longleftrightarrow e^{-j\omega n_0}X(n-n_0,\omega)$

TABLE 6.2 PROPERTIES OF THE DISCRETE STFT

Property 1:	$X(n,k)$ is zero outside $0 \le k < N$
Property 2:	$x(n)$ real $\longleftrightarrow X(n,k) = X^*(n,N-k)$ for $0 < k < N$
Property 3:	$x(n)$ real $\longleftrightarrow \lvert X(n,k)\rvert = \lvert X(n,N-k)\rvert$ for $0 < k < N$
Property 4:	$x(n)$ real $\longleftrightarrow \arg[X(n,k)] = -\arg[X(n,N-k)]$ for $0 < k < N$
Property 5:	$x(n-n_0) \longleftrightarrow e^{-j(2\pi n_0 k/N)}X(n-n_0,k)$

102

102

Properties of discrete STFT

TABLE 6.3 PROPERTIES OF DISCRETE-TIME STFT BASED ON THE FILTERING INTERPRETATION

Property 1:	$X(n, 0) = x(n) * w(n)$
Property 2:	$x(n)$ length N, $w(n)$ length M, $X(n, \omega)$ length $N + M - 1$ along n
Property 3:	Bandwidth of sequence $X(n, \omega_0) \leq$ Bandwidth of $w(n)$
Property 4:	The sequence $X(n, \omega_0)$ has spectrum centered at the origin
Property 5:	$x(n)$ causal, $w(n)$ causal, $X(n, \omega)$ causal in time

103

103

INVERSE STFT

104

104

Short-term Fourier Transform

- Can we recover $x[n]$ from $X(n,\omega)$ or $X(pL,k)$
  - i.e. inverse STFT or inverse discrete STFT
- $X(n,\omega)$ – STFT
  - Always invertible [by Fourier transform view]
  - ~ analogy with Fourier transform
- $X(pL,k)$ – discrete STFT
  - Not always invertible

105

105

Invertibility of STFT: always invertible

- STFT: $$X(n,\omega) = \sum_{m=-\infty}^{\infty} \underbrace{x[m]w[n-m]e^{-j\omega m}}_{\text{Call this signal } f_n[m]}$$
- For each n, we can compute inverse FT

$$f_n[m] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(n,\omega)\, e^{j\omega m} d\omega = x[m]w[n-m]$$

- Set $m = n$, then $x[m]w[n-m] = x[n]w[0]$

$$x[n] = \frac{1}{2\pi w[0]}\int_{-\pi}^{\pi} X(n,\omega)\, e^{j\omega n} d\omega$$

← "Synthesis Equation"

106

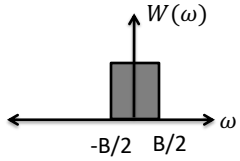
106

Invertibility of Discrete STFT

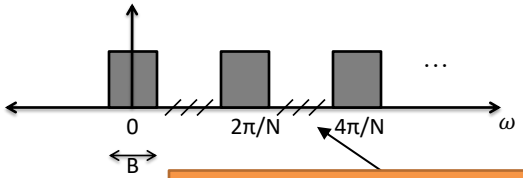
- Discrete STFT - $X(pL, k)$
  - Not always invertible
  - Can be seen both in frequency & time domain

Invertibility of Discrete STFT

- Frequency Domain (Frequency decimation)
  - Suppose $w[n]$ is an ideal low-pass filter with bandwidth B:



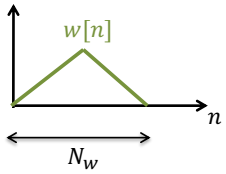
- Consider case where frequency–decimation interval > Bandwidth: $2\pi/N > B$



Lost Information: Frequencies of $x[n]$do not pass through any filters.

Invertibility of Discrete STFT

- Time Domain (Time decimation)
  - Suppose $w[n]$ is a finite-length window with duration N_w.



  - Consider case where time decimation interval > window length $L > N_w$

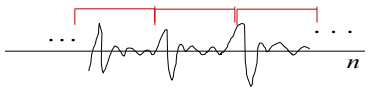
Lost Information: samples of $x[n]$ not included in any short time section of discrete STFT.

109

109

Conditions for invertibility of discrete STFT

- If temporal decimation factor L ≤ analysis window length N_w
  - ⇨ Discrete STFT invertible - provided we constrain frequency sampling
- Consider **upper cutoff L= N_w**
  - ⇨ i.e. DFT of adjacent but non-overlapping short-time segments
  - ⇨ Guarantees invertibility of time-decimated STFT



  - ❖ Then, we must require each N_W-point section to be recoverable from its DFT(i.e $N>N_W$) – otherwise (we get aliasing)

$$\frac{2\pi}{N} < \frac{2\pi}{N_w}$$

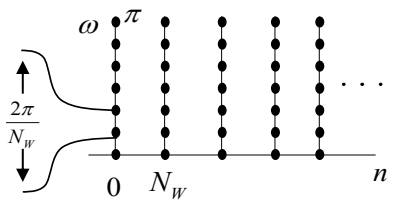
Guarantees invertibility of Frequency-decimated STFT

110

110

To sum up: Invertability of discrete STFT

- To recover a time signal from discrete STFT, we need to satisfy 2 constraints:
  - sampling of time sequences +
  - discrete Fourier transform



⇩

Time decimation (i.e. step between adjacent windows) to be no more than window length N_w

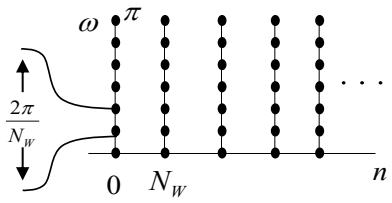
+

At least N_w samples in freq domain

111

111

Conditions for invertibility of discrete STFT



- The minimum requirements are time decimation of N_w and frequency sampling of $\frac{2\pi}{N_w}$
- We can establish relaxed bounds to recover the signal back (discussed in appendix slides)

112

Analysis

- Window selection …… compromise between:
  - Long window ⇨ good frequency resolution

vowel 

  - Short window ⇨ Good time resolution

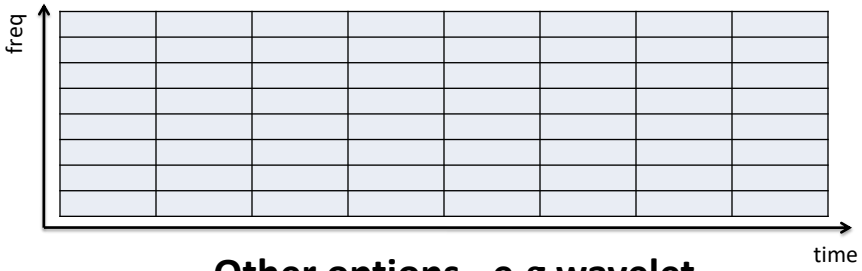
plosive 

113

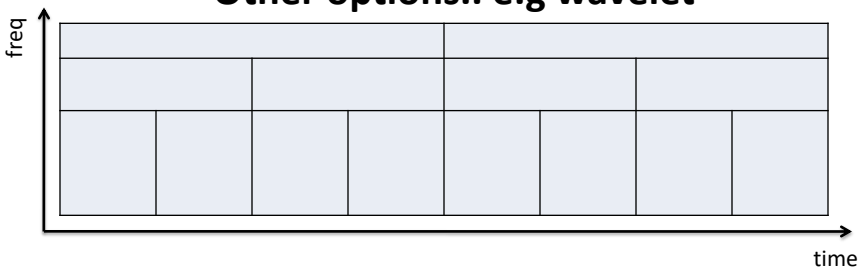
113

A Final Note

STFT



Other options.. e.g wavelet



114

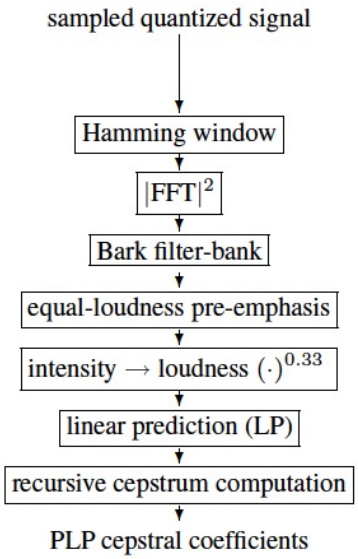
Other Time-frequency analysis models

- Knowledge from auditory physiology/perception gave rise to many popular models
  - Extend Fourier transform theory to incorporate key elements
    - Non-linear frequency scale (e.g. mel scale)
    - Critical band integration
    - Match loudness perception
    - Longer-integration time constants

115

115

Perceptual Linear Prediction (PLP)



116

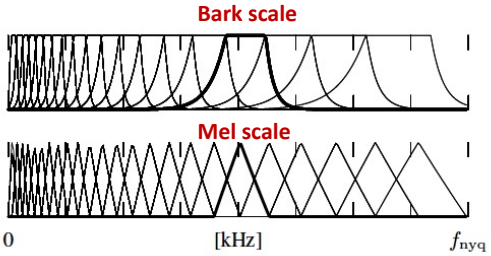
116

1/ PLP features – Freq Warping

- Warps frequency axis to a Bark scale
  - ~ similar to a mel scale
  - Closely mimics critical bands (auditory filters along the basilar membrane in the cochlea – each filter is referred as a Bark)

$$C_k(\omega) = \begin{cases} 10^{1.0(\Omega-\Omega_k+0.5)} & , \Omega \leq \Omega_k - 0.5 \\ 1 & , \Omega > \Omega_k - 0.5 \\ & , \Omega < \Omega_k + 0.5 \\ 10^{-2.5(\Omega-\Omega_k-0.5)} & , \Omega \geq \Omega_k + 0.5 \end{cases}$$

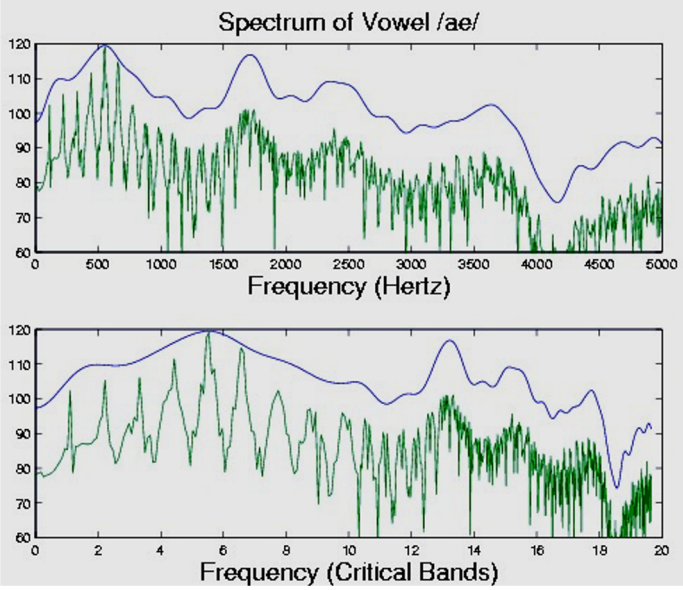
$C_k(\omega)$ is a weight of the k filter at frequency ω
Ω_k is a centre frequency of the filter k
$k = 1, 2, \ldots, K$.



Critical Band Formulas
(typically 24 bands)

117

Bark-Scale Warped Spectrum

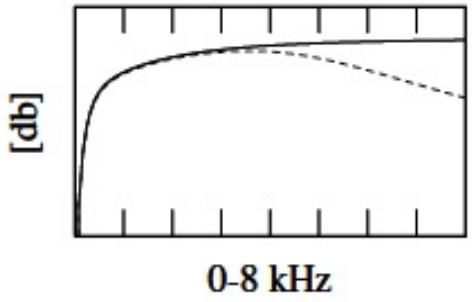


118

2/ Equal-loudness curve

- Scale spectrum to approximate nonlinear sensitivity to sound intensities at different frequencies

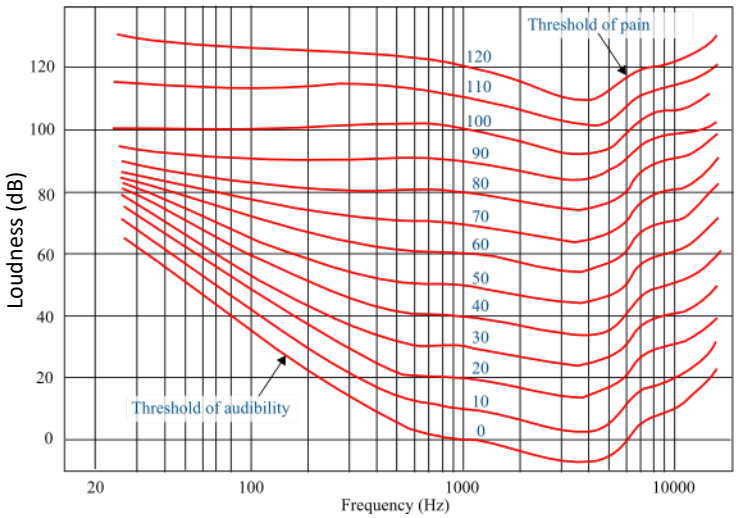
Equal-loudness curve
Applied in PLP



119

119

Equal Loudness Curves/Contours



Each contour represents equally-perceived tones

120

120

3/ Intensity Power law

- PLP applies a cubic-root to the power spectrum
  - Cubic root replaces the logarithm used in MFCC
  - Loudness of a tone is proportional to cubic root of its power

$$Y(\omega) = P(\omega)^{0.33}$$

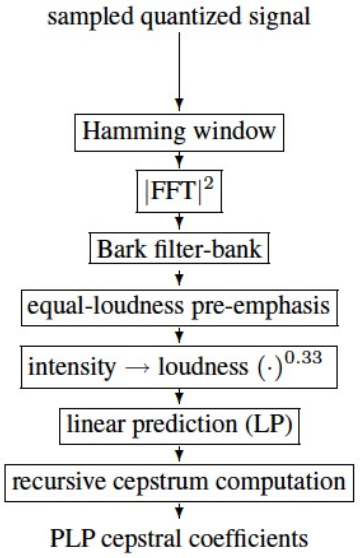
- This operation effectively decreases the dynamic variability and flattens the peaks of $P(\omega)$

121

121

4/ Linear prediction

- Finally, the resulting spectrum is modeled by a 5th order all pole filter (using LPC techniques)
  - produces a spectrum with at most 2 spectral peaks



122

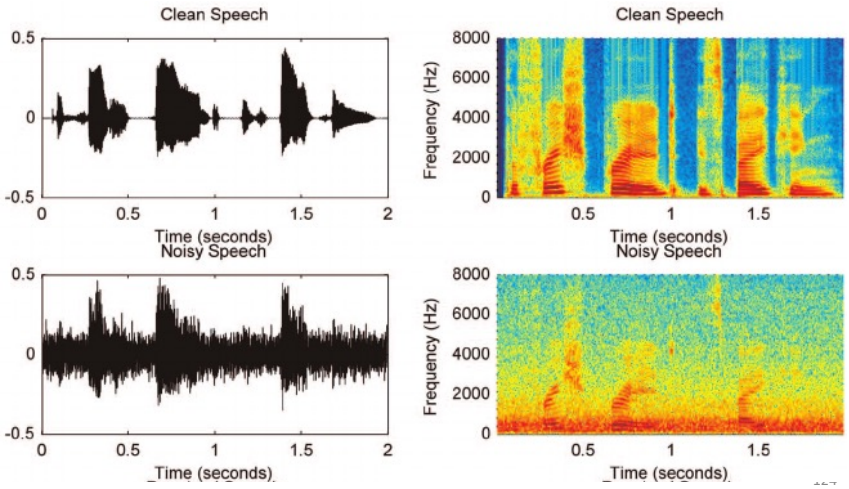
Choice of model

- Which model to use (LPC, PLP, MFCC, STFT) depends on the system and also conditions
  - Speech vs. speaker recognition
  - Clean (controlled quiet) vs. noisy environment
  - How stable are features?

123

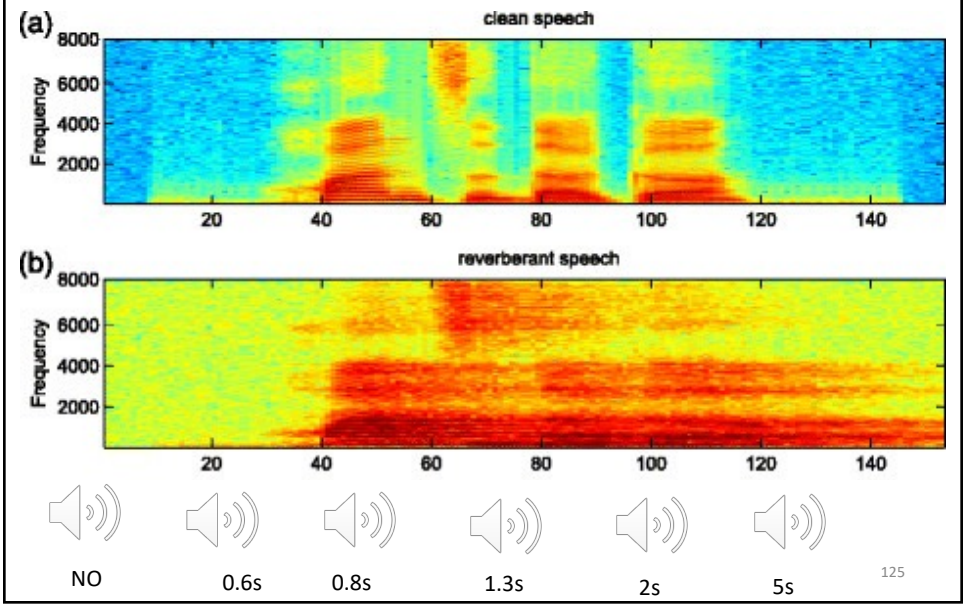
123

Noisy speech



124

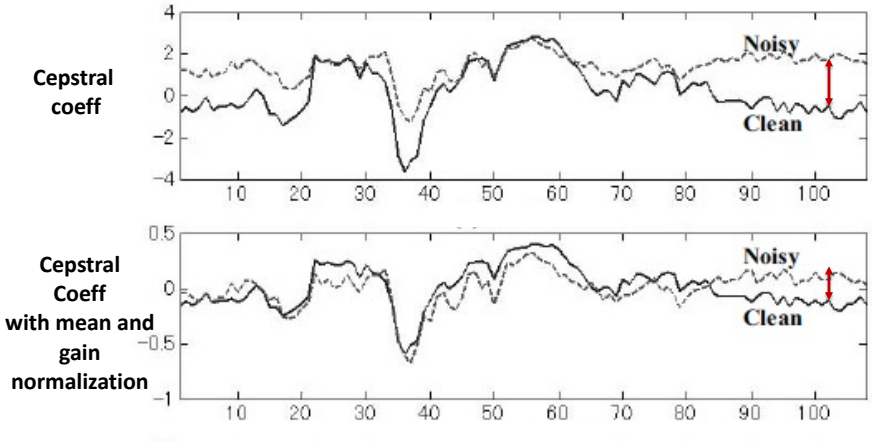
Reverberant speech



125

125

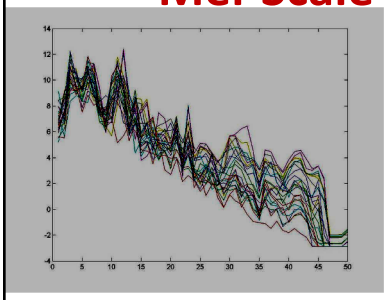
Noisy speech - MFCC



126

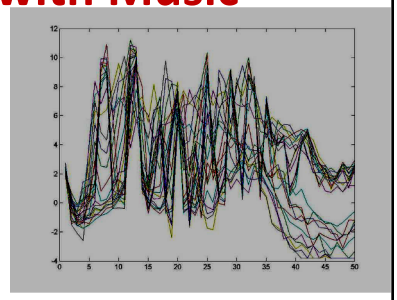
126

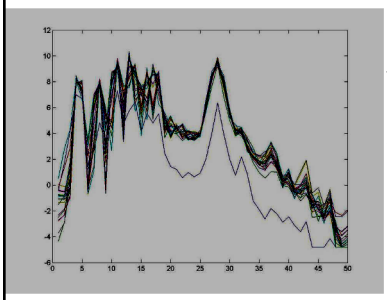
Feature stability – Mel-Scale Spectra with Music



Piano

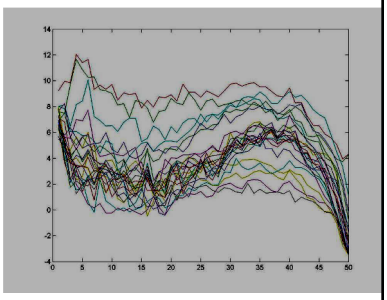
Saxophone





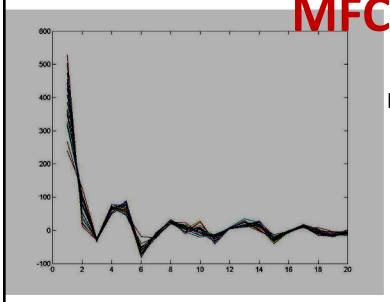
Tenor Opera Singer

Drums



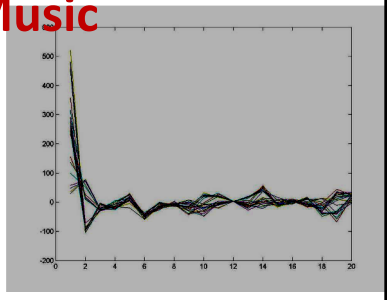
127

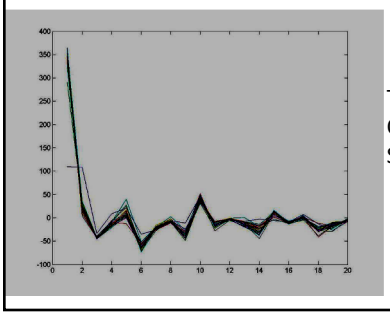
Feature stability – MFCC with Music



Piano

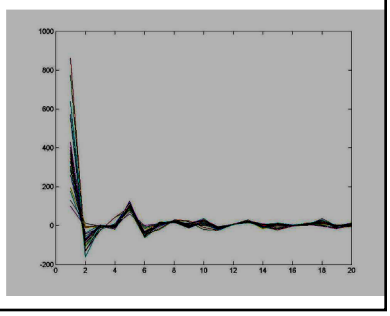
Saxophone





Tenor Opera Singer

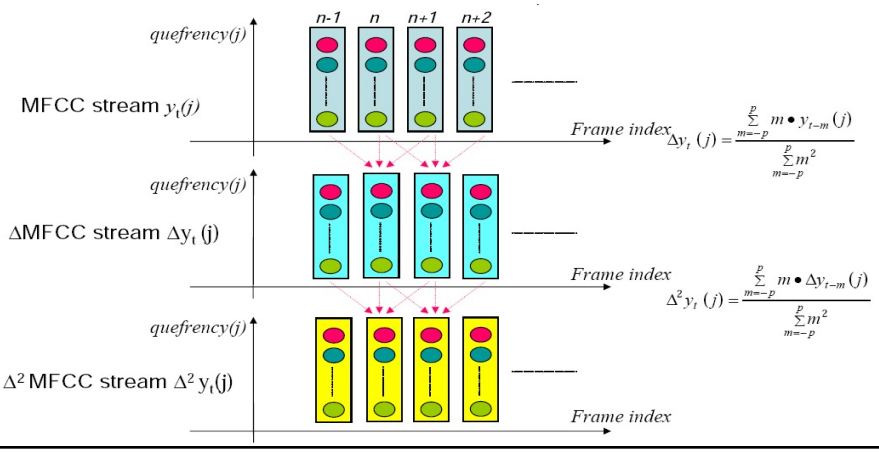
Drums



128

Dynamics in low-D features

- Current use of features like MFCC also introduces dynamics
  - Called delta and delta-delta features



129

Appendix:
Derivation of constraints for STFT

130

130

Resynthesis from discrete STFT

- Goal:
  - synthesize $x[n]$ from $X(pL, K)$
- Approach:
  - Start with synthesis equation $x[n] = g[X(n,\omega)]$

    i.e. $x[n] = \frac{1}{2\pi w[0]} \int_{-\pi}^{\pi} X(n,\omega) e^{j\omega n} d\omega$
  - Propose a discretized version
  - Derive conditions under which $x[n]$ can be recovered

- 2 **synthesis** approaches:
  - Filter-bank summation method(FBS)
  - Overlap and add method(OLA)

131

131

[1] Filterbank Summation Method

- Synthesis Equation: basis for the FBS method

$$x[n] = \frac{1}{2\pi w[0]} \int_{-\pi}^{\pi} X(n,\omega) e^{j\omega n} d\omega$$

Saw this earlier

$w[0] \neq 0$

⇨Proposed Discretized Synthesis Equation

$$\boldsymbol{y[n]} = \frac{1}{Nw[0]} \sum_{k=0}^{N-1} X(n,k) e^{j\frac{2\pi k}{N} n}$$

132

132

[1] Filterbank Summation Method

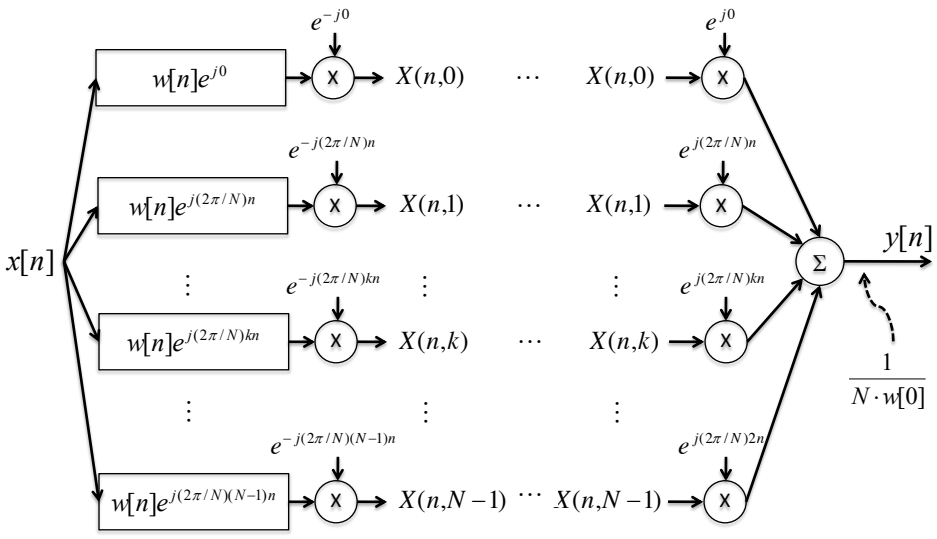
- $$y[n] = \frac{1}{Nw[0]} \sum_{k=0}^{N-1} X(n,k) e^{j\frac{2\pi k}{N}n}$$

  - Want to find conditions on $w[n]$ under which $x[n] = y[n]$
  - Can conjecture conditions by viewing discrete equation via filter-bank point-of-view.

133

133

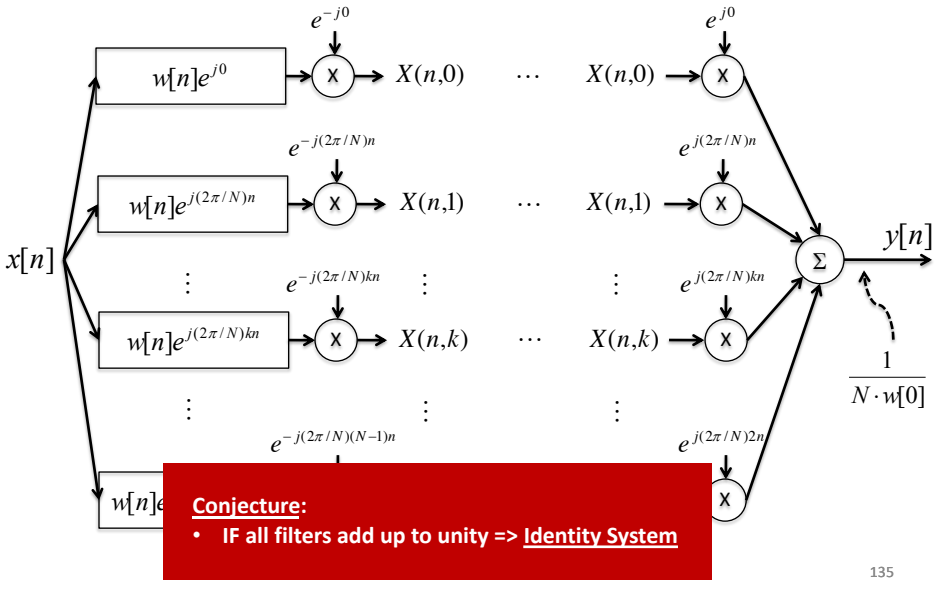
Discrete STFT as Filterbank Output



134

134

Discrete STFT as Filterbank Output



135

135

Conditions for Invertibility

- Substituting the discrete STFT into the inversion:

$$y[n]=\left[\frac{1}{N\cdot w[0]}\right]\sum_{k=0}^{N-1}\left[\sum_{m=-\infty}^{\infty}x[m]w[n-m]e^{-j\frac{2\pi}{N}km}\right]e^{j\frac{2\pi}{N}kn}$$

Discrete STFT: $X(n,k)$

$$=\left[\frac{1}{N\cdot w[0]}\right]\sum_{m=-\infty}^{\infty}x[m]\sum_{k=0}^{N-1}w[n-m]e^{-j\frac{2\pi}{N}km}e^{j\frac{2\pi}{N}kn}$$

$$=\left[\frac{1}{N\cdot w[0]}\right]\sum_{m=-\infty}^{\infty}x[m]\sum_{k=0}^{N-1}w[n-m]e^{j\frac{2\pi}{N}k(n-m)}$$

function of $(n\text{-}m)$

$$\Rightarrow\quad y[n]=\left[\frac{1}{N\cdot w[0]}\right]\left[x[n]*\sum_{k=0}^{N-1}w[n]e^{j\frac{2\pi}{N}kn}\right]$$

136

136

Aside: a note on the sum of exponentials

- Consider an infinite pulse train $p[n]$:

$$p[n] = \sum_{r=-\infty}^{\infty} \delta(n - rN)$$

- Taking the DFT (discrete Fourier transform):

$$p[n] = \frac{1}{N}\sum_{k=0}^{N-1} c[k] e^{j\frac{2\pi k}{N}n}, \quad 0 \le n \le N-1 \text{ (one period)}$$

$$c[k] = \sum_{n=0}^{N-1} p[n] e^{-j\frac{2\pi k}{N}n}, \quad 0 \le k \le N-1$$

- Since $p[n] = \delta(n)\ [n = 0]$ over one period then $c[k] = 1$
- So:

$$p[n] = \sum_{r=-\infty}^{\infty} \delta(n - rN) = \frac{1}{N}\sum_{k=0}^{N-1} e^{j\frac{2\pi k}{N}n}$$

137

137

Conditions for Invertibility

- From before:

$$y[n] = \left[\frac{1}{N \cdot w[0]}\right]\left[x[n] * \sum_{k=0}^{N-1} w[n] e^{j\frac{2\pi}{N}kn}\right]$$

$$y[n] = \left[\frac{1}{N \cdot w[0]}\right]\left\{x[n] * \left[w[n]\sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}kn}\right]\right\}$$

$$= \left[\frac{1}{N \cdot w[0]}\right]\left\{x[n] * \left[w[n] \cdot N \sum_{r=-\infty}^{\infty} \delta(n - rN)\right]\right\}$$

- For $y(n) = x(n)$, then:

$$w[n] \cdot N \sum_{r=-\infty}^{\infty} \delta(n - rN) = N \cdot w[0] \cdot \delta(n)$$

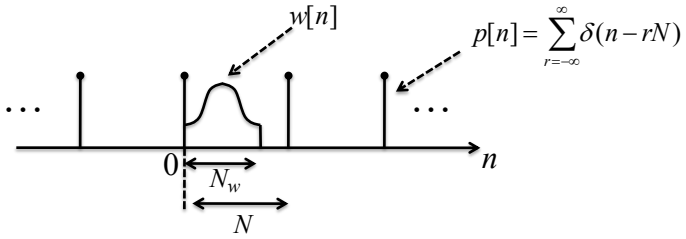
FBS Constraint

138

138

Conditions for Invertibility

1. If $N_w < N$
   - ➔Guaranteed invertibility
   - ➔FBS Condition always holds
   - ➔No aliasing from DFT point of view



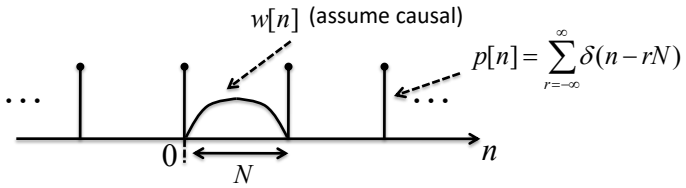
139

139

Conditions for Invertibility

2. If $N_w \geq N$
   - ➔Invertibility still possible
   - ➔Why? Recall FBS constraint

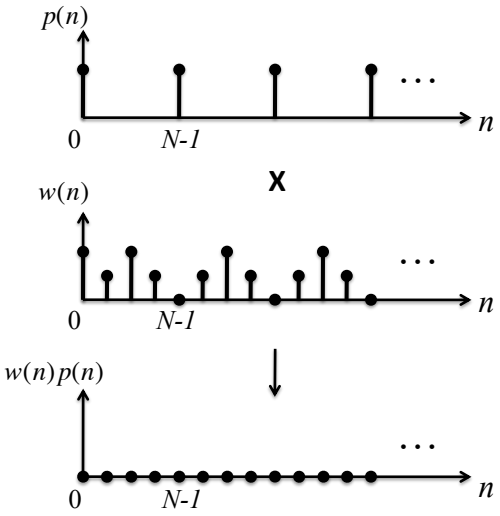
$$w[n] \cdot N \sum_{r=-\infty}^{\infty} \delta(n - rN) = N \cdot w[0] \cdot \delta(n) \quad \Longrightarrow \quad \text{need } w[n] = 0 \text{ at every } N\text{th sample}$$



140

140

In the time domain…



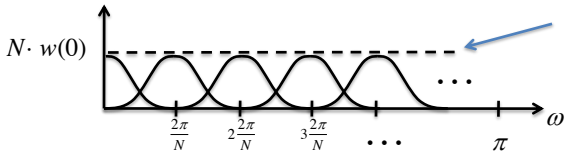
In the frequency domain…

- Taking the Fourier transform of both sides of the FBS constraint

$$F\left\{w[n]\cdot N\sum_{r=-\infty}^{\infty}\delta(n-rN)\right\}=F\{N\cdot w[0]\cdot\delta(n)\}$$

$$W(\omega)*\sum_{k=0}^{N-1}\delta\left(\omega-\frac{2\pi k}{N}\right)=N\cdot w[0]$$

or $$\sum_{k=0}^{N-1}W\left(\omega-\frac{2\pi k}{N}\right)=N\cdot w[0]$$



Frequency responses of analysis filters must sum to a constant

In the frequency domain…

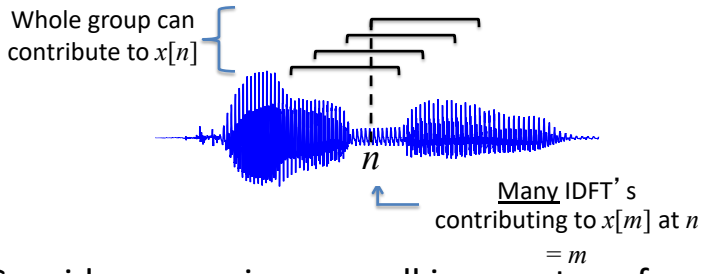
- Note:
  - For $N_w < N$ ➔ Perfect Filterbank
  - Not surprising since the IDFT of $\frac{1}{Nw[0]} X(n,k)$ must give $x[n]$
    - ⇨The DFT is longer than the sequence

143

143

[2] Overlap-Add Method (OLA)

- Uses redundancy in the overlapping windows, thus eliminating the effect of windowing



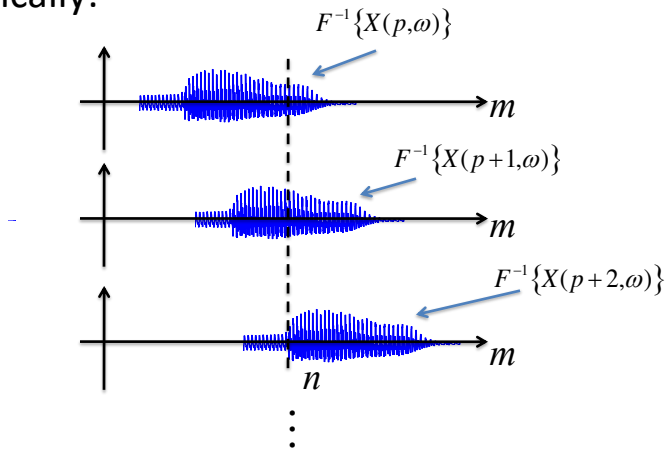
- Consider averaging over all inverse transforms to obtain value at time n
- Averaging effectively eliminates the window from inversion

144

144

OLA Motivated by the Synthesis Equation

- Graphically:



145

145

OLA Motivated by the Synthesis Equation

- OLA method is a <u>discretized version of the synthesis equation</u>:
  - Relationship between sequence $x[n]$ and its discrete-time STFT

$$x[n] = \frac{1}{2\pi W[0]} \int_{-\pi}^{\pi} \sum_{p=-\infty}^{\infty} X(p,\omega) e^{j\omega n} d\omega$$

$$W(0) = \sum_{n=-\infty}^{\infty} w(n)$$

  - So, a version discretized (in frequency)

$$y[n] = \frac{1}{NW[0]} \sum_{p=-\infty}^{\infty} \sum_{k=0}^{N-1} X(p,k) e^{j\frac{2\pi k}{N} n}$$

$$= \frac{1}{W[0]} \sum_{p=-\infty}^{\infty} \frac{1}{N} \sum_{k=0}^{N-1} X(p,k) e^{j\frac{2\pi k}{N} n}$$

Inverse DFT

$x[n]w[p-n]$

146

146

OLA Motivated by the Synthesis Equation

- $$y[n] = \frac{1}{W[0]} \sum_{p=-\infty}^{\infty} \frac{1}{N} \sum_{k=0}^{N-1} X(p,k) e^{j\frac{2\pi k}{N} n}$$

  ⇨ $y[n] = \frac{1}{W[0]} \sum_{p=-\infty}^{\infty} x[n] w[p-n]$

  ⇨Or: $y[n] = x[n] \frac{1}{W[0]} \sum_{p=-\infty}^{\infty} w[p-n]$

  ⇨Therefore: for $y[n] = x[n]$:

Sum of all windows shifted add up a to a constant

$$\boxed{\sum_{p=-\infty}^{\infty} w[p-n] = W[0]}$$ ***OLA Constraint***

147

147

OLA Motivated by the Synthesis Equation

- Note that the previous OLA constraint is defined for time decimation factor of $L = 1$
- Similarly, for $L > 1$, we get the OLA method

$$x(n) = \frac{L}{W(0)} \sum_{p=-\infty}^{\infty} \left[\frac{1}{N} \sum_{k=0}^{N-1} X(pL,k) e^{j\frac{2\pi}{N} kn} \right]$$

under the condition

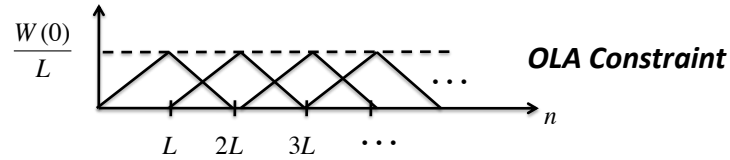
$$\sum_{p=-\infty}^{\infty} w(pL-n) = \frac{W(0)}{L}$$

148

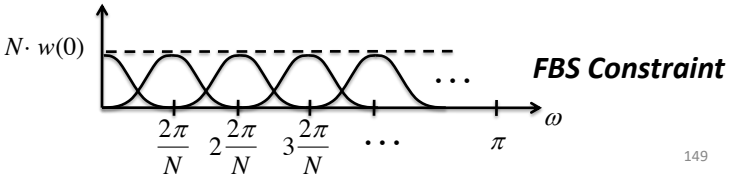
148

OLA Motivated by the Synthesis Equation

- Note: OLA Constraint **does not** hold for any arbitrary window
  - Constraint requires that sum of all analysis windows (by sliding $w(n)$ with L points) add up to a constant



- And remember the FBS Constraint

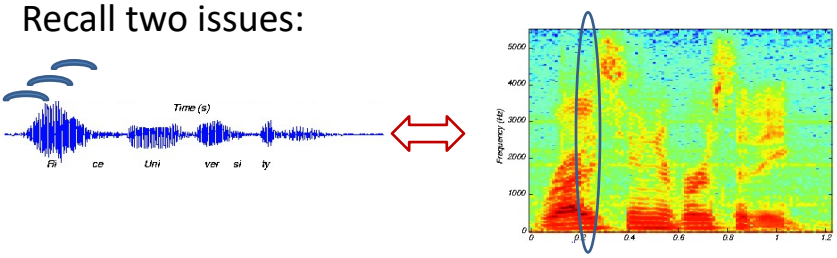


149

149

Analysis/Synthesis

- Recall two issues:



1. Analysis:
   - How to select window ("filter")
   - Tailor to speech signals (time-frequency resolution)
2. Synthesis:
   - Different methods for synthesizing a signal from (discrete) STFT
   - Sampling (decimation)

150

150

Analysis

- Window selection …… compromise between:
  - Long window ⇨ good frequency resolution

vowel 

  - Short window ⇨ Good time resolution

plosive 

151

151

Analysis
The math behind Time-Freq resolution

- Define windowed signal:

$$f_n[m] = x[m]w[n-m]$$

$w[-(m-n)]$

DTFT

$X(\omega)$ $\quad$ $W(-\omega)e^{-j\omega n}$

- Circular convolution:

$$X(n,\omega) = \frac{1}{2\pi} X(\omega) \bigstar W(-\omega)e^{-j\omega n}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\beta)W(-(\omega-\beta))e^{-j(\omega-\beta)n}d\beta$$

If $w[n]$ is even; $W(\omega)$ is even

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\beta)W(\omega-\beta)e^{-j(\omega-\beta)n}d\beta$$

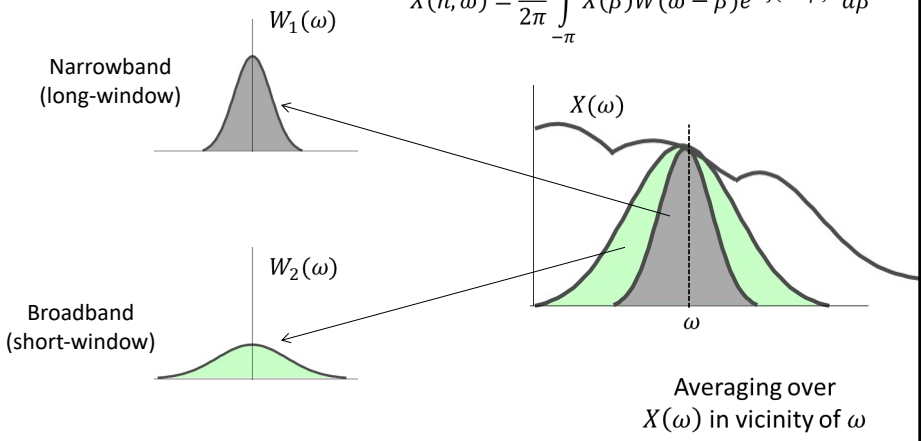
At time n, $X(n,\omega)$ is a smoothed version of $X(\beta)$ [Fourier transform of $x[n]$]

152

152

Analysis
The math behind Time-Freq resolution

$$X(n,\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\beta)W(\omega-\beta)e^{-j(\omega-\beta)n}d\beta$$



Averaging over
$X(\omega)$ in vicinity of ω

153

153

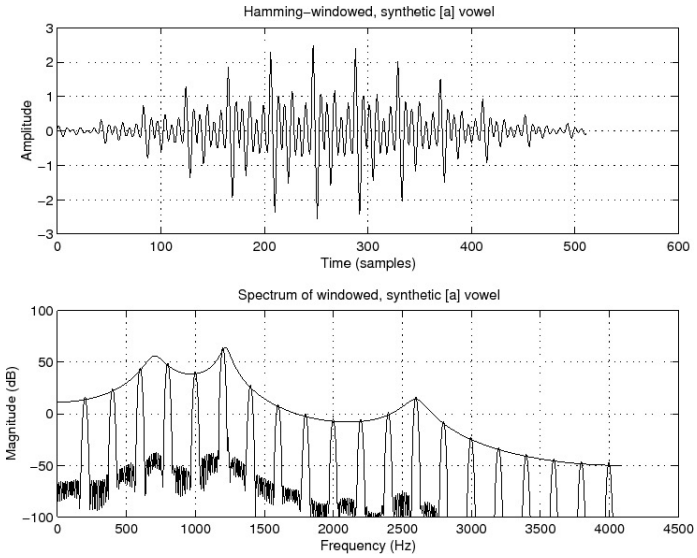
Analysis
The math behind Time-Freq resolution

- To faithfully reproduce $X(\omega)$, want $W(\omega)$ to be an impulse
  - ⇨ Valid if $x[n]$ not changing (stationary; ∞-long vowel)
    - make window very long
    - Good frequency resolution
  - ⇨BUT: if signal changing,
    - Need good time resolution

154

154

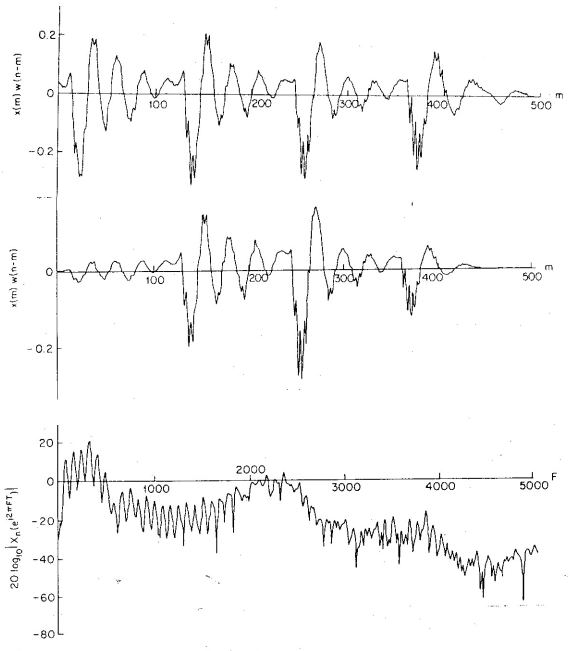
Formants & Harmonics



155

155

Voiced sounds



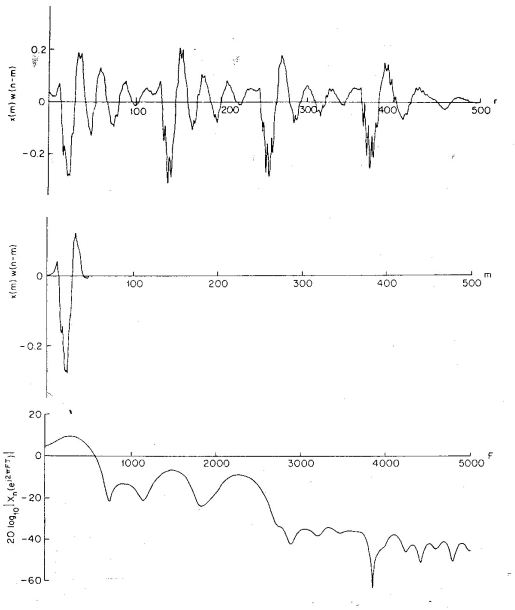
Periodicity

⇩

Long window

156

156



Voiced sounds

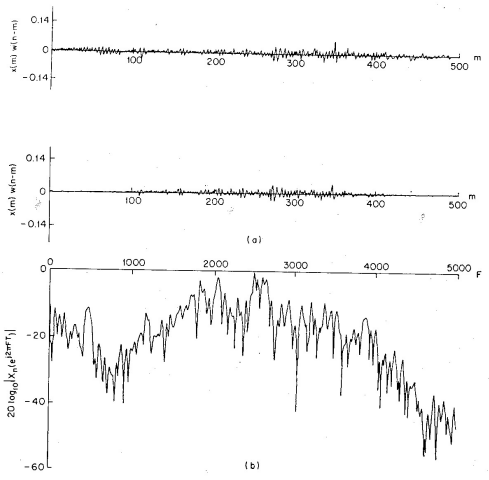
Changing formant structure

⇩

Short window

157

157

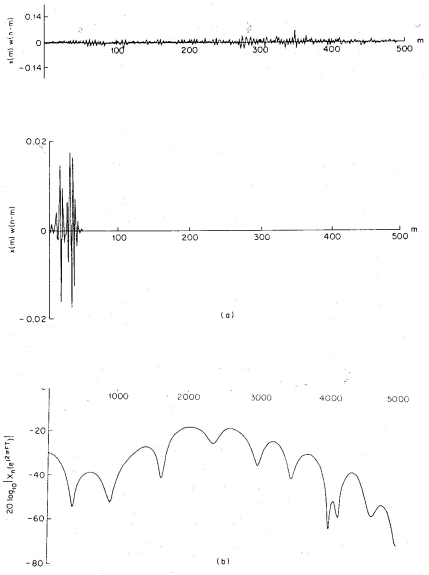


Unvoiced sounds

Long window

158

158



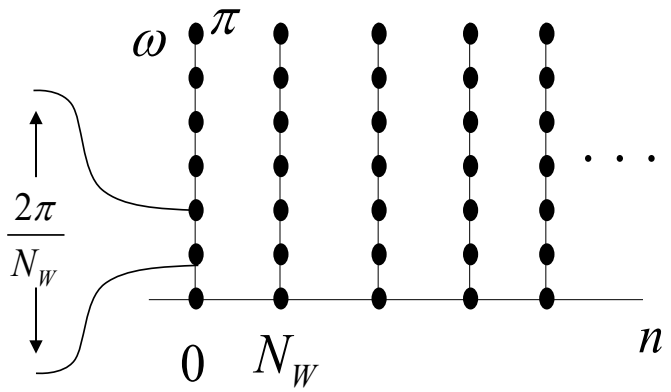
unvoiced sounds

Short window

159

159

STFT



160

160