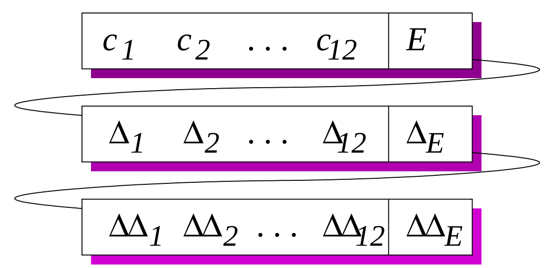
Centre for Vision Speech & Signal Processing
University of Surrey, Guildford GU2 7XH.

Feature extraction 2

Dr Philip Jackson

- Linear prediction
- Perceptual linear prediction
- Comparison of feature methods
- Gain and delta features



Linear prediction

Formulation

Linear prediction assumes that the speech signal $y(n)$ can be represented by an *auto-regressive* model:

$$y(n) = x(n) + \sum_{m=1}^{M} b_m \, y(n-m) \qquad (1)$$

where there are M linear prediction coefficients b_m, and $x(n)$ is some excitation signal. Its Z transform gives

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \sum_{m=1}^{M} b_m \, z^{-m}} \qquad (2)$$

and it is a predictor in the sense

$$\tilde{y}(n) = \sum_{m=1}^{M} b_m \, y(n-m) \qquad (3)$$

Linear prediction of speech

If we apply a window $w(n)$ to a speech signal $s(n)$,

$$y(n) = w(n)s(n) \tag{4}$$

we can express the instantaneous prediction error as

$$e(n) = y(n) - \sum_{m=1}^{M} b_m \, y(n-m) \tag{5}$$

and the total squared error over the window as

$$E = \sum_{n=0}^{N-1} e^2(n) = \sum_{n=0}^{N-1} \left(y(n) - \sum_{m=1}^{M} b_m \, y(n-m) \right)^2 \tag{6}$$

Taking partial derivatives with respect to each b gives us a set of M equations

$$\sum_{m=1}^{M} b_m \, \phi(n,m) = \phi(n,0) \qquad \text{for } n \in \{1, 2, \ldots, M\} \tag{7}$$

where $\phi(n,m)$ is correlation of signals delayed by n and by m samples respectively.

Calculation of LP coefficients

Using the autocorrelation $R(n-m)$ gives the equation

$$\begin{aligned} \mathbf{Rb} &= \mathbf{r} \\ \Rightarrow \quad \mathbf{b} &= \mathbf{R}^{-1}\mathbf{r} \end{aligned} \qquad (8)$$

where

$$\mathbf{R} = \begin{pmatrix} R(0) & R(1) & \cdots & R(M-1) \\ R(1) & R(0) & & R(M-2) \\ \vdots & & \ddots & \vdots \\ R(M-1) & R(M-2) & \cdots & R(0) \end{pmatrix}$$

and

$$\mathbf{r} = \begin{pmatrix} R(1) \\ R(2) \\ \vdots \\ R(M) \end{pmatrix}$$

which can be solved efficiently by the Levinson-Durbin recursion (Deller et al. 1993, Fig 5.8, p. 300).

Frequency response of cepstral coefficients

Recall from the definition of the ceptrum, described in Feature Extraction 1, that

$$c_s(m) = \mathcal{F}\{\ln|S(\omega)|\} \tag{9}$$

and the cepstral coefficients of the vocal-tract filter are modeled

$$c_y(m) = \mathcal{F}\{\ln|Y(\omega)|\} \tag{10}$$

using the low lag coefficients, $0 \leq m < M$. Hence, the model gives

$$\begin{aligned} |Y(\omega)| &= \exp\left(\mathcal{F}\{c_y(m)\}\right) \\ \Rightarrow \quad Y(z) &= \exp\left(\sum_{m=0}^{M-1} z^{-m} c_y(m)\right) \end{aligned} \tag{11}$$

where $z = \mathrm{e}^{j\omega}$ yields the filter's frequency response.

Parametric models of spectral envelope

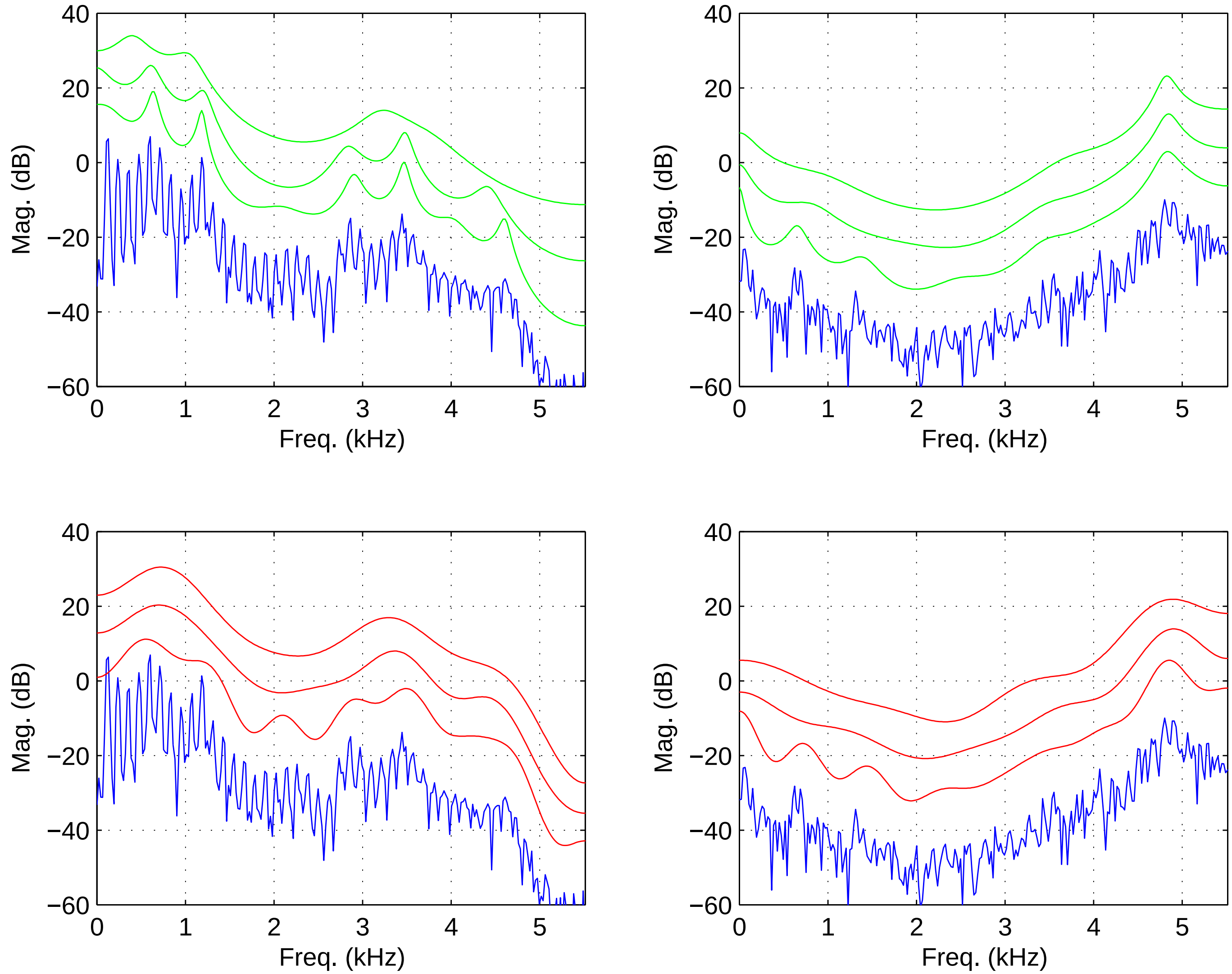
- **Linear prediction**: Auto-regressive (AR) model

$$H(z) = \frac{1}{1 - \sum_{m=0}^{M-1} b(m)\, z^{-m}}$$

- **Cepstral analysis**: Exponential model

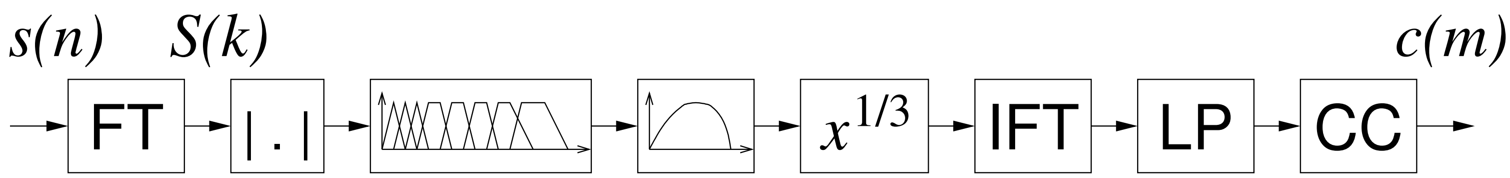
$$H(z) = \exp\left(\sum_{m=0}^{M-1} c(m)\, z^{-m}\right)$$

LP and CC features with model order



Perceptual linear prediction (PLP)

Similar to the Mel-warping of the frequency scale when merging spectral energy across bands in calculation of MFCC features, a perceptually-relevant linear prediction process can be used to compute PLP features:



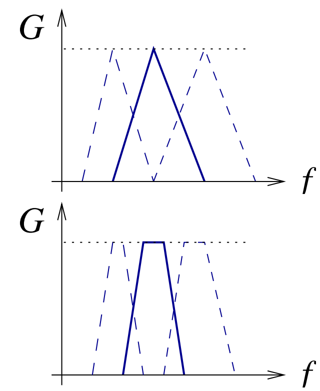
- Frequency bands based on critical bands (Bark scale)
- Loudness weighting and compression
- Conversion of LP coefficients into cepstra

Perceptually-weighted acoustic features

Feature extraction steps for MFCC and PLP:

1. **Short-term spectrum**
2. **Binning**
3. **Pre-emphasis**
4. **Compression**
5. **Inverse FT**
6. **Spectral smoothing**
7. **Decorrelation**
8. **Liftering**

1. **Short-term spectrum**
   - apply Hamming window
   - compute Fourier transform (as an FFT)
   - take magnitude of coefficients
2. **Binning**



   (a) Mel (pitch perception) & triangular
   (b) Bark (critical band) & trapezoidal
3. **Pre-emphasis**
   (a) A zero giving +6 dB/octave
   (b) Equal-loudness weighting
4. **Compression**
   (a) logarithm
   (b) cube root (intensity-loudness power law)

5. **Inverse FT** — info. concentration

   (a) IDCT

   (b) IDFT

6. **Spectral smoothing**

   (a) cepstral truncation

   (b) AR model (i.e., LP)

7. **Decorrelation**

   (a) none required

   (b) conversion to cesptra

8. **Liftering**

   - variance equalisation

Comparison of MFCC and PLP-CC features

Operation	MFCC	PLP-CC
Pre-emphasis	6 dB/octave of waveform	equal-loudness of spectrum
Filter bank	triangular mel filters	critical-band filters
Amplitude compression	logarithmic	cube-root
Spectral smoothing	Cepstral	LPC smoothing

Extensions to the feature set

Two important ways of extending the set of features:

1. Amplitude features

2. Dynamic features
   – Deltas
   – Accelerations

Amplitude feature – log energy

When using cepstral features, the zeroth coefficient provides a measure of the overall spectral magnitude for a given frame of speech.

An alternative way of representing this information uses the **log energy**:

$$E = \ln \sum_{n=1}^{N} s(n)^2 \tag{12}$$

The feature is appended to the other cepstral features to make up the feature vector for each frame.

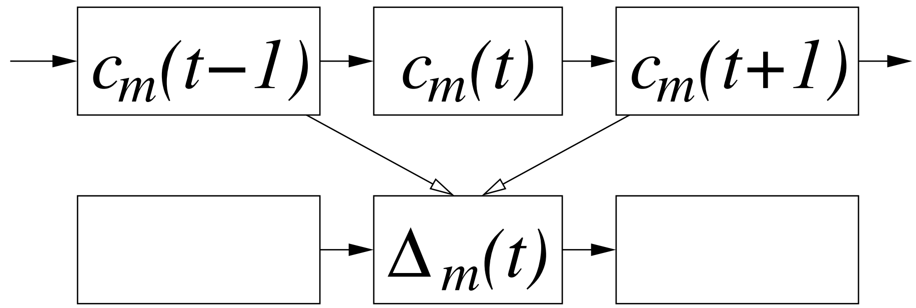
Dynamic features – velocity and acceleration

Speech dynamics can be incorporated into the set of features using their rate of change. This improves performance of speech recognition systems dramatically, especially for consonant sounds with rapid transitions.

For example, consider differences in cepstral coefficients $m \in \{1, \ldots, M\}$ for consecutive speech frames:

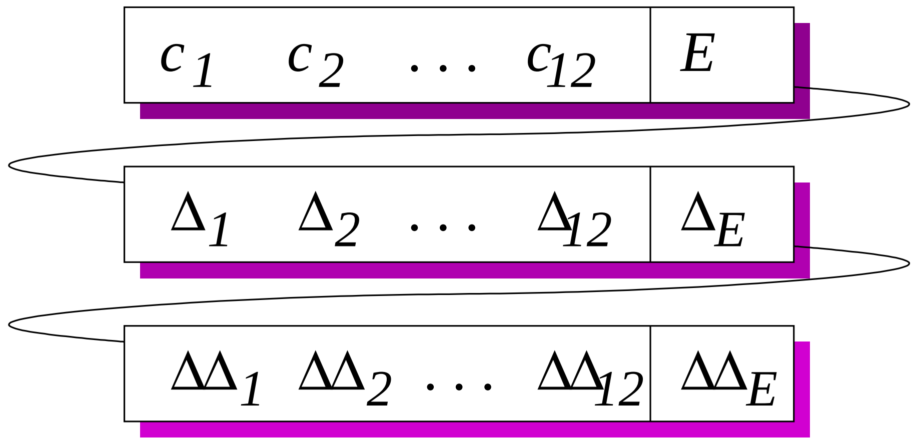
$$\Delta_m(t) = \frac{1}{2}\left[c_m(t+1) - c_m(t-1)\right] \tag{13}$$

The same operation can be applied to these **delta** features to yield the delta-delta or **acceleration** features.



A typical observation sequence of features, $\mathrm{o}(t)$

Many speech recognition systems use a standard 39-feature representation of the audio signal to form 39-D feature vectors for each speech frame at time t:



For a 2-second speech recording of one sentence, we would get 200 vectors with a 10-ms frame offset.

Feature extraction 2 summary

- Linear prediction features
- Perceptually-weighted PLP versus MFCC
- Extensions
  - Energy features
  - Deltas & accelerations