Speech Signal Analysis

Peter Bell

Automatic Speech Recognition— ASR Lectures 4&5
23 January 2019
Overview

Speech Signal Analysis for ASR

- Features for ASR
- Spectral analysis
- Cepstral analysis
- Standard features for ASR: FBANK, MFCCs and PLP analysis
- Dynamic features

Reading:
- Jurafsky & Martin, sec 9.3
- P Taylor, *Text-to-Speech Synthesis*, chapter 12, signal processing background chapter 10
Speech signal analysis for ASR

- Recorded Speech
- Signal Analysis
- Training Data
- Acoustic Model
- Language Model
- Search Space
- Decoded Text (Transcription)
Speech production model

Vocal Organs & Vocal Tract

Vocal Organs & Vocal Tract

(V0 : fundamental frequency)
A/D conversion — Sampling

Convert analogue signals in digital form

Microphone

Sound pressure wave

Conversion from impulse train to discrete-time sequence

$X_c(t_c) \xrightarrow{\times} X_s(t_c) \xrightarrow{\uparrow \uparrow \uparrow \uparrow 1} \frac{1}{T_S} \xrightarrow{\times} x[t_d]$
Things to know:

- **Sampling Frequency** \( (F_s = 1/T_s) \)

<table>
<thead>
<tr>
<th>Speech</th>
<th>Sufficient ( F_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microphone voice (&lt;10kHz)</td>
<td>20 kHz</td>
</tr>
<tr>
<td>Telephone voice (&lt;4kHz)</td>
<td>8 kHz</td>
</tr>
</tbody>
</table>

- Analogue low-pass filtering to avoid 'aliasing'
- NB: the cut-off frequency should be less than the **Nyquist frequency** \( (= F_s/2) \)
Acoustic Features for ASR

Speech signal analysis to produce a sequence of acoustic feature vectors
Desirable characteristics of acoustic features used for ASR:

- Features should contain sufficient information to distinguish between phones
  - good time resolution (10ms)
  - good frequency resolution (20 ∼ 40 channels)
- Be separated from $F_0$ and its harmonics
- Be robust against speaker variation
- Be robust against noise or channel distortions
- Have good “pattern recognition characteristics”
  - low feature dimension
  - features are independent of each other (NB: this applies to GMMs, but not required for NN-based systems)
MFCC-based front end for ASR

A/D conversion \( x(t) \rightarrow x[t_d] \)

Preemphasis \( x'[t_d] \)

Window \( x_t[n] \)

Energy \( e_t \)

Mel filterbank \( Y_t[m] \)

Logarithm \( \log(Y_t[m]) \)

DFT \( [X_t[k]^2] \)

Feature Transform \( y_t[j], e_t \)

Dynamic features \( y_t[j] \)

IDFT

Acoustic Model \( o_t[i] \)

\( \Delta y_t[j], \Delta e_t \)

\( \Delta \Delta y_t[j], \Delta \Delta e_t \)
Pre-emphasis increases the magnitude of higher frequencies in the speech signal compared with lower frequencies.

**Spectral Tilt**
- The speech signal has more energy at low frequencies (for voiced speech).
- This is due to the glottal source (see the figure).

Pre-emphasis (first-order) filter boosts higher frequencies:

\[
x'[t_d] = x[t_d] - \alpha x[t_d - 1], \quad 0.95 < \alpha < 0.99
\]
Speech production model

Vocal Organs & Vocal Tract

(F₀ : fundamental frequency)
Pre-emphasis and spectral tilt

- Pre-emphasis increases the magnitude of higher frequencies in the speech signal compared with lower frequencies
- **Spectral Tilt**
  - The speech signal has more energy at low frequencies (for voiced speech)
  - This is due to the glottal source (see the figure)
- Pre-emphasis (first-order) filter boosts higher frequencies:

\[
x'[t_d] = x[t_d] - \alpha x[t_d - 1] \quad 0.95 < \alpha < 0.99
\]
Pre-emphasis: example

Vowel /aa/ - time slice of the spectrum

(Jurafsky & Martin, fig. 9.9)
The speech signal is constantly changing (non-stationary)
Signal processing algorithms usually assume that the signal is stationary
Piecewise stationarity: model speech signal as a sequence of frames (each assumed to be stationary)

**Windowing:** multiply the full waveform $s[n]$ by a window $w[n]$ (in time domain):

$$x[n] = w[n] s[n] \quad (x_t[n] = w[n] x'[t_d+n])$$

Simply cutting out a short segment (frame) from $s[n]$ is a rectangular window — causes discontinuities at the edges of the segment

Instead, a tapered window is usually used
e.g. *Hamming* ($\alpha = 0.46164$) or *Hanning* ($\alpha = 0.5$) window

$$w[n] = (1 - \alpha) - \alpha \cos \left( \frac{2\pi n}{L-1} \right) \quad L : \text{window width}$$
**Window** the signal $x'[t_d]$ into frames $x_t[n]$ and apply Fourier Transform to each segment.

- Short frame width: *wide-band*, high time resolution, low frequency resolution
- Long frame width: *narrow-band*, low time resolution, high frequency resolution

For ASR:
- frame width $\sim 25ms$
- frame shift $\sim 10ms$
Window the signal $x'[t_d]$ into frames $x_t[n]$ and apply Fourier Transform to each segment.

- Short frame width: *wide-band*, high time resolution, low frequency resolution
- Long frame width: *narrow-band*, low time resolution, high frequency resolution

For ASR:
- frame width $\sim 25ms$
- frame shift $\sim 10ms$
Short-time spectral analysis

- Windowing
- Shift
- Frame

Short-time power spectrum
Fourier Transform
Discrete Fourier Transform

Intensity
Frequency
Short-time power spectrum

Frequency
Time (frame)
Discrete Fourier Transform (DFT)

- **Purpose:** extracts spectral information from a windowed signal (i.e. how much energy at each frequency band)
- **Input:** windowed signal $x[0], \ldots, x[L-1]$ (time domain)
- **Output:** a complex number $X[k]$ for each of $N$ frequency bands representing magnitude and phase for the $k$th frequency component (frequency domain)
- **Discrete Fourier Transform (DFT):**

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-j \frac{2\pi}{N} kn\right)$$

NB: $\exp(j\theta) = e^{j\theta} = \cos(\theta) + j\sin(\theta)$

- **Fast Fourier Transform (FFT) — efficient algorithm for computing DFT** when $N$ is a power of 2, and $N \geq L$. 
25ms Hamming window of vowel /iy/ and its spectrum computed by DFT

(Jurafsky and Martin, fig 9.12)
Windowing and spectral analysis

*Window* the signal $x'[t_d]$ into frames $x_t[n]$ and apply Fourier Transform to each segment.

- **Short frame width:** *wide-band*, high time resolution, low frequency resolution
- **Long frame width:** *narrow-band*, low time resolution, high frequency resolution

- For ASR:
  - frame width $\sim 25\, ms$
  - frame shift $\sim 10\, ms$
Window the signal $x'[t_d]$ into frames $x_t[n]$ and apply Fourier Transform to each segment.

- Short frame width: *wide-band*, high time resolution, low frequency resolution
- Long frame width: *narrow-band*, low time resolution, high frequency resolution

For ASR:
- frame width $\sim 25ms$
- frame shift $\sim 10ms$
Wide-band and narrow-band spectrograms

Figure 12.8  Wide band spectrogram  window width = 2.5ms

Figure 12.9  Narrow band spectrogram  window width = 25ms

(Taylor, figs 12.8, 12.9)
Window the signal $x'[t_d]$ into frames $x_t[n]$ and apply Fourier Transform to each segment.

- Short frame width: *wide-band*, high time resolution, low frequency resolution
- Long frame width: *narrow-band*, low time resolution, high frequency resolution

For ASR:
- frame width $\sim 25ms$
- frame shift $\sim 10ms$
Windowing and spectral analysis

- **Window** the signal $x'[t_d]$ into frames $x_t[n]$ and apply Fourier Transform to each segment.
  - Short frame width: *wide-band*, high time resolution, low frequency resolution
  - Long frame width: *narrow-band*, low time resolution, high frequency resolution
- For ASR:
  - frame width $\sim 25\, ms$
  - frame shift $\sim 10\, ms$
Effect of windowing — time domain

Rectangular  Hamming  Hanning

(a) Rectangular window  (b) Hanning window  (c) Hamming window

(Taylor, fig 12.1)
Effect of windowing — frequency domain

\[
x(t) = 0.15 \sin(2\pi f_1 t) + 0.85 \sin(2\pi f_2 t + 0.3)
\]
\[
f_1 = 0.13, \quad f_2 = 0.22
\]
Effect of windowing — frequency domain

\[ x(t) = 0.15 \sin(2\pi f_1 t) + 0.85 \sin(2\pi f_2 t + 0.3) \]

\[ f_1 = 0.13, \quad f_2 = 0.22 \]
MFCC-based front end for ASR

\[ x(t) \rightarrow A/D \text{ conversion} \rightarrow \text{Preemphasize} \rightarrow \text{Window} \rightarrow \text{DFT} \rightarrow \text{Dynamic features} \rightarrow \text{IDFT} \rightarrow \text{Acoustic Model} \]

- \[ x[n] \]
- Energy \[ e_t \]
- Mel filterbank \[ Y_t[m] \]
- Logarithm \[ \log(Y_t[m]) \]
- Logarithm of energy \[ \log(e_t) \]
- \[ |X_t[k]|^2 \]
- \[ y_t[j], e_t \]
- \[ \Delta y_t[j], \Delta e_t \]
- \[ \Delta \Delta y_t[j], \Delta \Delta e_t \]
Equally-spaced frequency bands — but human hearing less sensitive at higher frequencies (above \( \sim 1000\text{Hz} \))

The estimated power spectrum contains harmonics of F0, which makes it difficult to estimate the envelope of the spectrum

Frequency bins of STFT are highly correlated each other, i.e. power spectrum representation is highly redundant
### Human hearing

<table>
<thead>
<tr>
<th>Physical quality</th>
<th>Perceptual quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intensity</td>
<td>Loudness</td>
</tr>
<tr>
<td>Fundamental frequency</td>
<td>Pitch</td>
</tr>
<tr>
<td>Spectral shape</td>
<td>Timbre</td>
</tr>
<tr>
<td>Onset/offset time</td>
<td>Timing</td>
</tr>
<tr>
<td>Phase difference in binaural hearing</td>
<td>Location</td>
</tr>
</tbody>
</table>

**Technical terms**

- equal-loudness contours
- masking
- auditory filters (critical-band filters)
- critical bandwidth
Equal loudness contour
Human hearing is less sensitive to higher frequencies — thus human perception of frequency is nonlinear

**Mel scale**

\[ M(f) = 1127 \ln(1 + f/700) \]

**Bark scale**

\[ b(f) = 13 \arctan(0.00076f) + 3.5 \arctan((f/7500)^2) \]
Mel-Filter Bank

- Apply a mel-scale filter bank to DFT power spectrum to obtain mel-scale power spectrum.
- Each filter collects energy from a number of frequency bands in the DFT.
- Linearly spaced < 1000 Hz, logarithmically spaced > 1000 Hz.

DFT (STFT) power spectrum $|X[k]|^2$

Triangular band-pass filters

Mel-scale power spectrum $Y[m]$

Frequency bins
Mel-Filter Bank

- Apply a mel-scale filter bank to DFT power spectrum to obtain mel-scale power spectrum
- Each filter collects energy from a number of frequency bands in the DFT
- Linearly spaced < 1000 Hz, logarithmically spaced > 1000 Hz

DFT(STFT) power spectrum $|X[k]|^2$

Triangular band-pass filters

Mel-scale power spectrum $Y[m]$
Apply a mel-scale filter bank to DFT power spectrum to obtain mel-scale power spectrum.

Each filter collects energy from a number of frequency bands in the DFT.

Linearly spaced $< 1000$ Hz, logarithmically spaced $> 1000$ Hz.
Apply a mel-scale filter bank to DFT power spectrum to obtain mel-scale power spectrum.

Each filter collects energy from a number of frequency bands in the DFT.

Linearly spaced $< 1000$ Hz, logarithmically spaced $> 1000$ Hz.
Apply a mel-scale filter bank to DFT power spectrum to obtain mel-scale power spectrum.

Each filter collects energy from a number of frequency bands in the DFT.

Linearly spaced < 1000 Hz, logarithmically spaced > 1000 Hz.
Mel-Filter Bank (cont.)

\[
Y_t[m] = \sum_{k=1}^{N} W_m[k]|X_t[k]|^2
\]

where \( k \) : DFT bin number \((1, \ldots, N)\)

\( m \) : mel-filter bank number \((1, \ldots, M)\).

- How many number of mel-filter channels?
  
  \[ \approx 20 \quad \text{for GMM-HMM based ASR} \]
  
  \[ 20 \sim 40 \quad \text{for DNN (+HMM) based ASR} \]
MFCC-based front end for ASR

A/D conversion → Preemphasis → Window → DFT

Energy $e_t$

Mel filterbank

IDFT

Feature Transform

Dynamic features

Acoustic Model

|y_t[j], e_t
Δy_t[j], Δe_t
ΔΔy_t[j], ΔΔe_t

Output $o_t[i]$
Log Mel Power Spectrum

- Compute the log magnitude squared of each mel-filter bank output: $\log Y[m]$
  - Taking the log compresses the dynamic range
  - Human sensitivity to signal energy is logarithmic — i.e. humans are less sensitive to small changes in energy at high energy than small changes at low energy
  - Log makes features less variable to acoustic coupling variations
  - Removes phase information — not important for speech recognition (not everyone agrees with this)

- Aka “log mel-filter bank outputs” or “FBANK features”, which are widely used in recent DNN-HMM based ASR systems
MFCC-based front end for ASR

- A/D conversion
- Preemphasis
- Window
- DFT
  - Mel filterbank
  - Log

Feature Transform
- $y_t[j]$, $e_t$
- $\Delta y_t[j]$, $\Delta e_t$
- $\Delta\Delta y_t[j]$, $\Delta\Delta e_t$

Dynamic features
- $o_t[i]$

IDFT

Acoustic Model
DFT Spectrum Features for ASR

- Equally-spaced frequency bands — but human hearing less sensitive at higher frequencies (above $\sim 1000\text{Hz}$)

- The estimated power spectrum contains harmonics of F0, which makes it difficult to estimate the envelope of the spectrum

- Frequency bins of STFT are highly correlated each other, i.e. power spectrum representation is highly redundant
Cepstral Analysis

- **Source-Filter model of speech production**
  - **Source**: Vocal cord vibrations create a glottal source waveform
  - **Filter**: Source waveform is passed through the vocal tract: position of tongue, jaw, etc. give it a particular shape and hence a particular filtering characteristic

- Source characteristics ($F_0$, dynamics of glottal pulse) do not help to discriminate between phones
- The filter specifies the position of the articulators
- ... and hence is directly related to phone discrimination
- Cepstral analysis enables us to separate source and filter
Speech production model

Vocal Organs & Vocal Tract

\[ |X(\Omega)| \]

\[ |H(\Omega)| \]

\[ |V(\Omega)| \]

\[ F_0 = \frac{1}{T_0} \]

\[ v(t) \]

\[ +6\text{dB/oct.} \]

\[ -12\text{dB/oct.} \]

\[ F_1, F_2, F_3 \] (formants)
Cepstral Analysis

- **Source-Filter model of speech production**
  - **Source**: Vocal cord vibrations create a glottal source waveform
  - **Filter**: Source waveform is passed through the vocal tract: position of tongue, jaw, etc. give it a particular shape and hence a particular filtering characteristic

- Source characteristics ($F_0$, dynamics of glottal pulse) do not help to discriminate between phones

- The filter specifies the position of the articulators

- ... and hence is directly related to phone discrimination

- Cepstral analysis enables us to separate source and filter
Cepstral Analysis

Split power spectrum into spectral envelope and $F_0$ harmonics.

Log spectrum (freq domain)

\[ \downarrow \text{Inverse Fourier Transform} \]

Cepstrum (time domain) *(quefrency)*

\[ \downarrow \text{Liftering to get low/high part} \]

*(lifter: filter used in cepstral domain)*

\[ \downarrow \text{Fourier Transform} \]

Smoothed log spectrum (freq domain)

\[ [\text{low-part of cepstrum}] + \]

Fine structure

\[ [\text{high-part of cepstrum}] \]
The Cepstrum

- Cepstrum obtained by applying inverse DFT to log magnitude spectrum (may be mel-scaled)
- Cepstrum is time-domain (we talk about quefrency)
- Inverse DFT:

\[ x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \exp \left( j \frac{2\pi}{N} nk \right) \]

- Since log power spectrum is real and symmetric the inverse DFT is equivalent to a discrete cosine transform (DCT)

\[ y_t[n] = \sum_{m=0}^{M-1} \log(Y_t[m]) \cos \left( n(m+0.5) \frac{\pi}{M} \right), \quad n = 0, \ldots, J \]
MFCCs

- Smoothed spectrum: transform to cepstral domain, truncate, transform back to spectral domain
- Mel-frequency cepstral coefficients (MFCCs): use the cepstral coefficients directly
  - Widely used as acoustic features in HMM-based ASR
  - First 12 MFCCs are often used as the feature vector (removes F0 information)
  - Less correlated than spectral features — easier to model than spectral features
  - Very compact representation — 12 features describe a 20ms frame of data
  - For standard HMM-based systems, MFCCs result in better ASR performance than filter bank or spectrogram features
  - MFCCs are not robust against noise
MFCC-based front end for ASR

A/D conversion → Preemphasis → Window → DFT → Mel filterbank → Log( ) → IDFT

- $x(t)$ → $x[t_d]$ → $x'[t_d]$ → $x_t[n]$ → $|X_t[k]|^2$ → $Y_t[m]$ → $\log(Y_t[m])$
- Energy $e_t$

Feature Transform
- $y_t[j]$, $e_t$
- $\Delta y_t[j]$, $\Delta e_t$
- $\Delta \Delta y_t[j]$, $\Delta \Delta e_t$

Dynamic features → IDFT

Acoustic Model
- $o_t[i]$
PLP — Perceptual Linear Prediction

- PLP (Hermansky, JASA 1990)
- Uses equal loudness pre-emphasis and cube-root compression (motivated by perceptual results) rather than log compression
- Uses linear predictive auto-regressive modelling to obtain cepstral coefficients
- PLP has been shown to lead to
  - slightly better ASR accuracy
  - slightly better noise robustness

\[ \hat{y}[n] = \sum_{k=1}^{P} a_k y_t[n - k] \]

\[ \hat{y}[n] = \sum_{k=1}^{P} a_k y_t[n - k] \]
Dynamic features

- Speech is not constant frame-to-frame, so we can add features to do with how the cepstral coefficients change over time.
- $\Delta^*, \Delta^2*$ are delta features (dynamic features / time derivatives).
- Simple calculation of delta features $d(t)$ at time $t$ for cepstral feature $c(t)$ (e.g. $y_t[j]$):
  \[
d(t) = \frac{c(t + 1) - c(t - 1)}{2}\]
- More sophisticated approach estimates the temporal derivative by using regression to estimate the slope (typically using 4 frames each side).
- “Standard” ASR features (for GMM-based systems) are 39 dimensions:
  - 12 MFCCs, and energy
  - 12 $\Delta$MFCCs, $\Delta$energy
  - 12 $\Delta^2$MFCCs, $\Delta^2$energy
Estimating dynamic features

\[ c(t) \]

\[ c'(t_0) \]

\[ t_0 \]
Dynamic features

- Speech is not constant frame-to-frame, so we can add features to do with how the cepstral coefficients change over time.
- $\Delta^*, \Delta^2*$ are delta features (dynamic features / time derivatives).
- Simple calculation of delta features $d(t)$ at time $t$ for cepstral feature $c(t)$ (e.g. $y_t[j]$):
  \[
  d(t) = \frac{c(t + 1) - c(t - 1)}{2}
  \]
- More sophisticated approach estimates the temporal derivative by using regression to estimate the slope (typically using 4 frames each side).
- “Standard” ASR features (for GMM-based systems) are 39 dimensions:
  - 12 MFCCs, and energy
  - 12 $\Delta$MFCCs, $\Delta$energy
  - 12 $\Delta^2$MFCCs, $\Delta^2$energy
MFCC-based front end for ASR

A/D conversion → Preemphasis → Window → DFT

Energy: $e_t$

Mel filterbank:

$Y_t[m]$

log(Y_t[m])

IDFT

Feature Transform

$y_t[j], e_t$

Dynamic features

$\Delta y_t[j], \Delta e_t$

$\Delta \Delta y_t[j], \Delta \Delta e_t$

Acoustic Model
Feature Transforms

- Orthogonal transformation (orthogonal bases)
  - **DCT** (discrete cosine transform)
  - **PCA** (principal component analysis)
- Transformation based on the bases that maximises the separability between classes.
  - **LDA** (linear discriminant analysis) / Fisher’s linear discriminant
  - **HLDA** (heteroscedastic linear discriminant analysis)
Feature Normalisation

- **Basic Idea:** Transform the features to reduce mismatch between training and test.

- **Cepstral Mean Normalisation (CMN):** Subtract the average feature value from each feature, so each feature has a mean value of 0. Makes features robust to some linear filtering of the signal (channel variation).

- **Cepstral Variance Normalisation (CVN):** Divide feature vector by standard deviation of feature vectors, so each feature vector element has a variance of 1.

- Cepstral mean and variance normalisation, CMN/CVN:
  \[
  \hat{y}_t[j] = \frac{y_t[j] - \mu(y[j])}{\sigma(y[j])}
  \]

- Compute mean and variance statistics over longest available segments with the same speaker/channel.

- Real time normalisation: compute a moving average.
Acoustic features in state-of-the-art ASR systems

See Tables 1, 2, and 3 in

Jinyu Li, Dong Yu, Jui-Ting Huang, and Yifan Gong,
"Improving Wideband Speech Recognition Using Mixed-Bandwidth Training Data In CD-DNN-HMM",
https://doi.org/10.1109/SLT.2012.6424210
Table 1: Comparison of different input features for DNN. All the input features are mean-normalized and with dynamic features. Relative WER reduction in parentheses.

<table>
<thead>
<tr>
<th>Setup</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD-GMM-HMM (MFCC, fMPE+BMMI)</td>
<td>34.66 (baseline)</td>
</tr>
<tr>
<td>CD-DNN-HMM (MFCC)</td>
<td>31.63 (-8.7%)</td>
</tr>
<tr>
<td>CD-DNN-HMM (24 log filter-banks)</td>
<td>30.11 (-13.1%)</td>
</tr>
<tr>
<td>CD-DNN-HMM (29 log filter-banks)</td>
<td>30.11 (-13.1%)</td>
</tr>
<tr>
<td>CD-DNN-HMM (40 log filter-banks)</td>
<td>29.86 (-13.8%)</td>
</tr>
<tr>
<td>CD-DNN-HMM (256 log FFT bins)</td>
<td>32.26 (-6.9%)</td>
</tr>
</tbody>
</table>
Table 2: Comparison of DNNs with and without dynamic features. All the input features are mean normalized.

<table>
<thead>
<tr>
<th>CD-DNN-HMM (40 log filter-banks)</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>static+$\Delta$+$\Delta\Delta$ (11-frame)</td>
<td>29.86</td>
</tr>
<tr>
<td>static only (11-frame)</td>
<td>31.11</td>
</tr>
<tr>
<td>static only (19-frame)</td>
<td>30.48</td>
</tr>
</tbody>
</table>
Table 3: Comparison of features with and without mean normalization. Dynamic features are used.

<table>
<thead>
<tr>
<th>CD-DNN-HMM (29 log filter banks)</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>With mean normalization</td>
<td>30.11</td>
</tr>
<tr>
<td>Without mean normalization</td>
<td>29.96</td>
</tr>
</tbody>
</table>
Summary: Speech Signal Analysis for ASR

- Good characteristics of ASR features
- **FBANK features**
  - Short-time DFT analysis
  - Mel-filter bank
  - Log magnitude squared
  - Widely used for DNN ASR ($M \approx 40$)
- **MFCCs - mel frequency cepstral coefficients**
  - FBANK features
  - Inverse DFT (DCT)
  - Use first few (12) coefficients
  - Widely used for GMM-HMM ASR
- **Delta features (dynamic features)**
- **39-dimension feature vector (for GMM-HMM ASR):**
  - MFCC-12 + energy; + Deltas; + Delta-Deltas
