Envelope Detection Methods

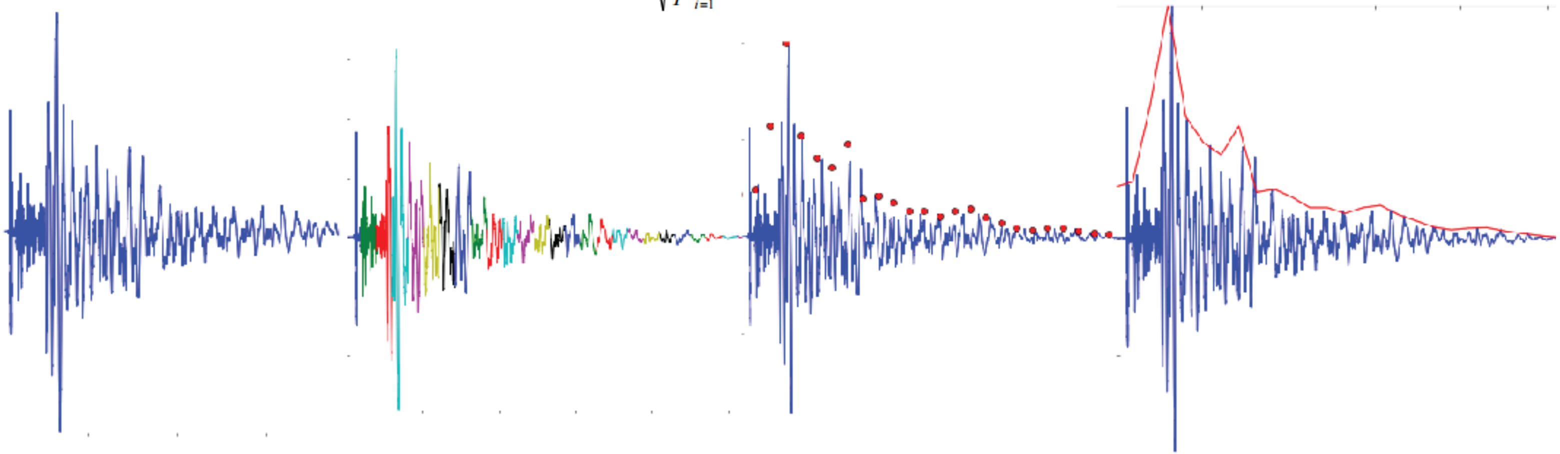
Logan Thomas

Root-Mean Square (RMS) Energy

- Very Popular
- Normally used to calculate a measure of energy for a wave
- Easily adapted to calculate an approximate of the amplitude envelope by applying it to a sliding window

RMS visually

$$RMS(t) = \sqrt{\frac{1}{T}\sum_{i=1}^{T} w_i(t)x_i^2(t)}$$



RMS Pros and Cons

Pros

- easy to impliment
- gives a direct measure of the “power”of the wave (physical significance)
- unambiguous

Cons

- window selection is a win-lose situation.
    - big: lag behind abrubt changes
    - small: noise overpowering
- heavily weights outliers

How to make it better

- apply some sory of smoothing/line fit to the points, polynomial fit etc.
- use a low pass filter to reduce noise
- optimize window size each time by measuring how noisy the result is, if it is too noisy, increasing window size

Frequency Domain Linear Prediction

- Tradional Linear Prediction (TDLP) for envelopes was done from the time domain to predict a spectral envelope.
- This method applies the Linear Prediction methods to the Discrete Cosine Transform in order to retreive an estimation of the time domain envelope.
- This works because of the time-frequency duality
- It is fast and relies on one parameter, order (number of arguments in the Linear Prediction)

Discrete Cosine Transform

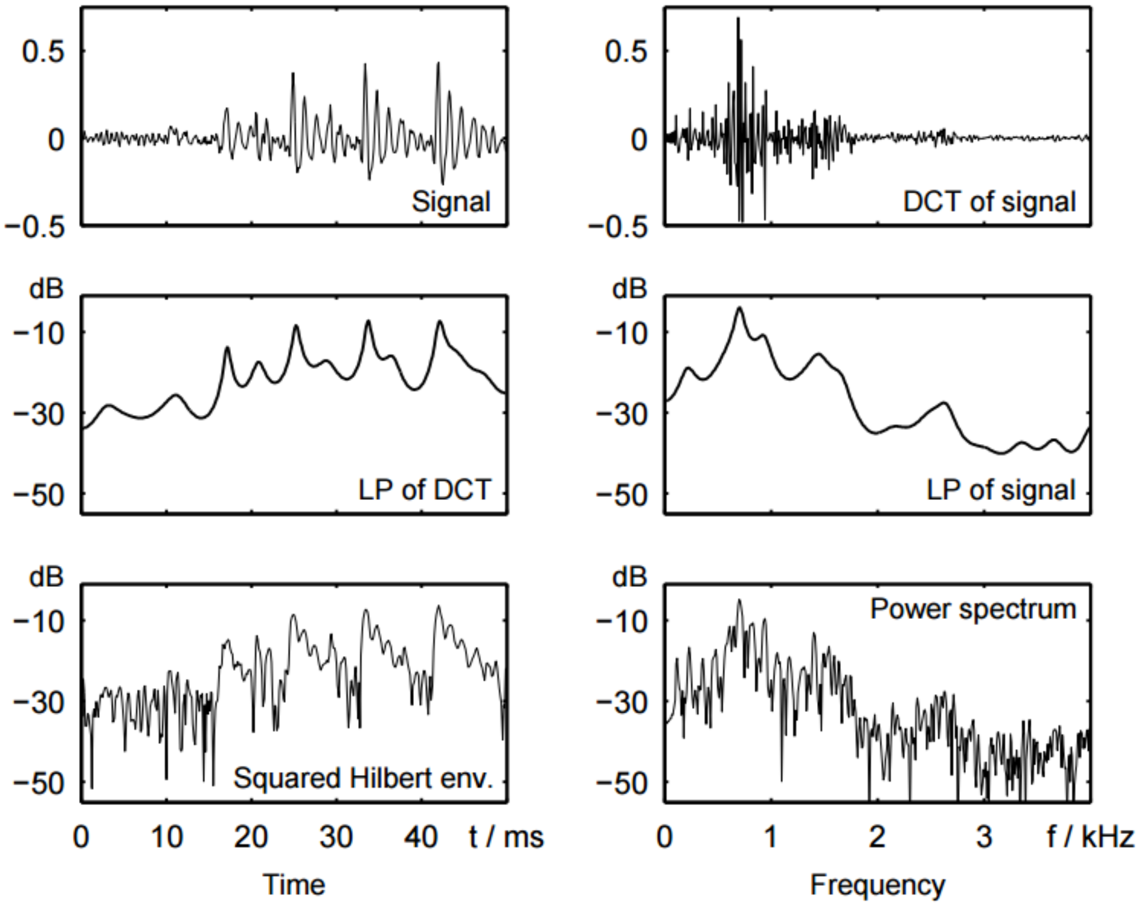
- Used because it is real valued and allows the linear prediction (which acts on real values) to be effective
- Equivalent to the Discrete Fourier Transform of a symmetric version of the wave
- Half the resolution of the normal Fourier Transform

$$\hat{X}(k) = \sum_{n=0}^{N-1} x(n) \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right)$$

Linear Predictive Coding

- LPC methods prove extremley accurate estimates efficiently
- Basic idea is that a current sample can be closely approximated by a linear combination of past samples:
- These coeffficents are found by minimizing the squared distance between the actual and the predicted wave.

$$s(n) = \sum_{k=1}^{p} \alpha_k \, s(n-k)$$



0.5
0
−0.5
Signal
DCT of signal
dB
−10
−30
−50
LP of DCT
LP of signal
Squared Hilbert env.
Power spectrum
0 10 20 30 40 t / ms
Time
0 1 2 3 f / kHz
Frequency

FDLP Steps

1. take DCT of signal
2. filter DCT if necessary
3. apply Linear Prediction to DCT, returns order+1 LPC coefficients
4. Envelope Prediction = Gain./DFT(zero-padded LPC coefficients)

Model Gain is calculated by comparing the Energy of the original signal with the linearly predicted model

FDLP Pros and Cons

Pros

- Used widely through other fields because it is fast and accurately portrays “important” features
- easy to impliment frequency filters because DCT is an intermediate step

Cons

- Does take some optimization of order (number of poles)

How to improve Implimentation

- Attempt a sliding window FDLP and see how effective/accurate it is
- Filter more frequencies
- add and order optimization loop that changes the order and attempts to minimize error

True Amplitde Envelope

- TAE aims to make a set of points that outline the waveform following its general shape without representing the harmonic structure
- At each point takes the maximum of the cepstral smoothing and the actual rectified wave

Cepstral Smoothing

- Defined as the inverse fourier transform of the log of the amplitude spectrum
- using a low pass filter on this transform results in a smoother wave that follows the amplitude spectrum

$$\hat{x}(n) = \sum_{k=0}^{K-1} \log\left|X(k)\right| \exp\left(\frac{j2\pi kn}{N}\right)$$

True Amplitude Envelope Steps

1. set A_0 = log|x(n)|
2. take the first smoothed Cespstral spectrum of A_0, C_0
3. iteratively set A_i = max(log|x(n)|,C_(i-1)) until within a threshold of the actual wave
4. return A_i

True Amplitude Envelope

Pros

- Gives a very accurate waveform that is very close to the actual wave without any of the harmonic details

Cons

- Expensive threshold loops
- Taking inverse and forward Fourier transforms each iteration

How to improve Implimentation

- smooth between the segments more, by overlapping windows and averaging the points that get doubled
-

RMS Method

```
n_windows = n_samples/sample_size
for(int i=0;i<n_windws;i++){
    window = audio[i:(i+window_size)]
    rms[i] = rms_energy(window)
}
```

References

- IMPROVED ESTIMATION OF THE AMPLITUDE ENVELOPE OF TIME-DOMAIN SIGNALS USING TRUE ENVELOPE CEPSTRAL SMOOTHING, Marcels,C., Rodet, X.
- FREQUENCY DOMAIN LINEAR PREDICTION BASED ON TEMPORAL ANALYSIS, Shenoy, R., Seelamantula, C.
- Linear Prediction: A tutorial review, Makhoul, J
- LP-TRAP: Linear Predictive temporal patterns, Athineos, M., Hermansky, H., Ellis, D.
- FREQUENCY-DOMAIN LINEAR PREDICTION FOR TEMPORAL FEATURES, Athineos, M., Ellis, D.