Signal Processing S2
Week 12: SFFT, Windowing, Feature Extraction

@btatmaja

Index

- Short-time Fourier Transform equation
- Analysis window

Short-time Fourier Transform

DFT: $$X[k] = x[n]e^{-j2\pi kn/N}$$

Splitting x[n] into windowed functions w[n] with frame and steps

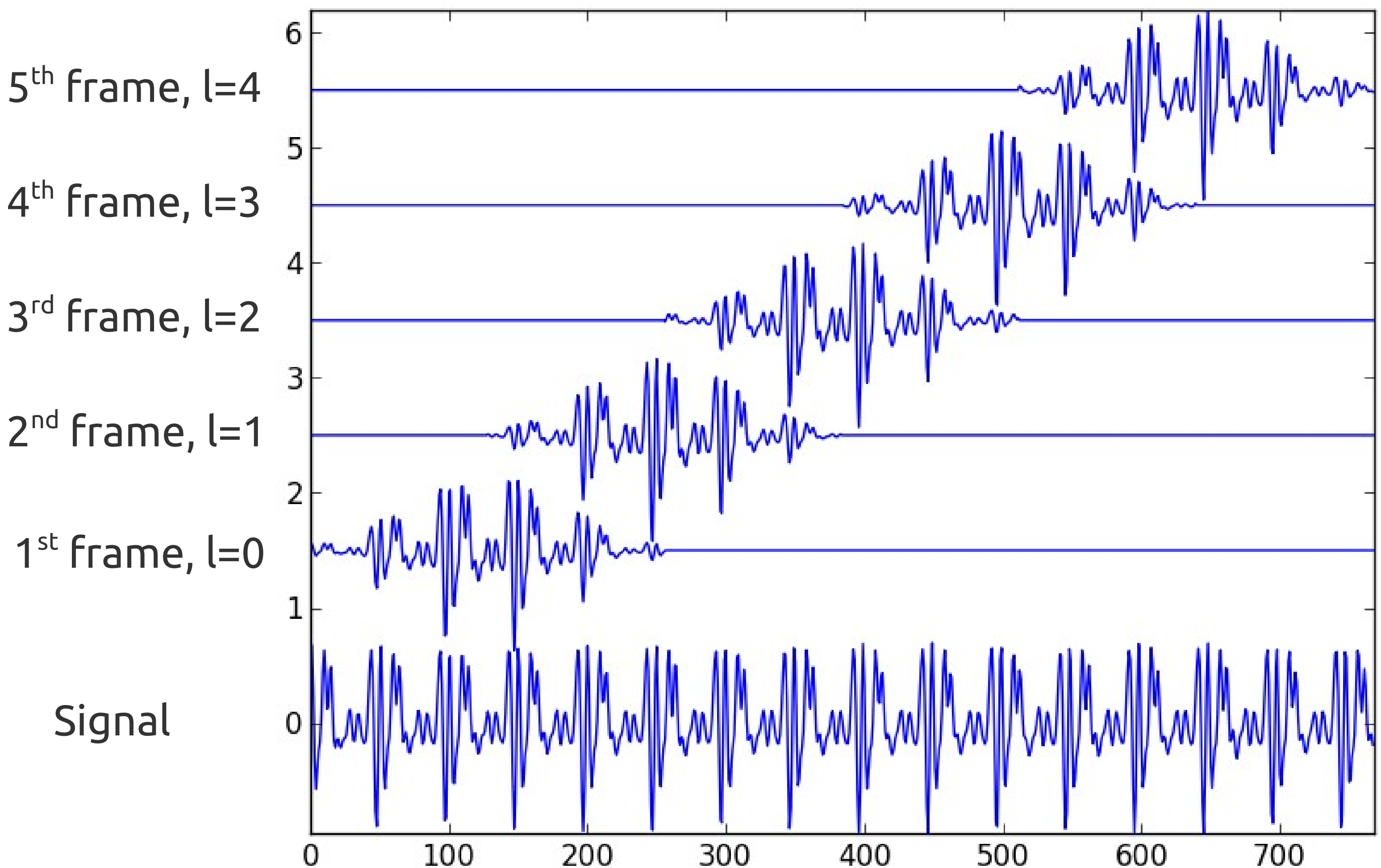
STFT: $$X_l[k] = \sum_{n=-N/2}^{N/2-1} w[n]x[n+lH]e^{-j2\pi kn/N} \qquad l = 0,1,\ldots,$$

w: analysis window

l: frame number

H: hop-size

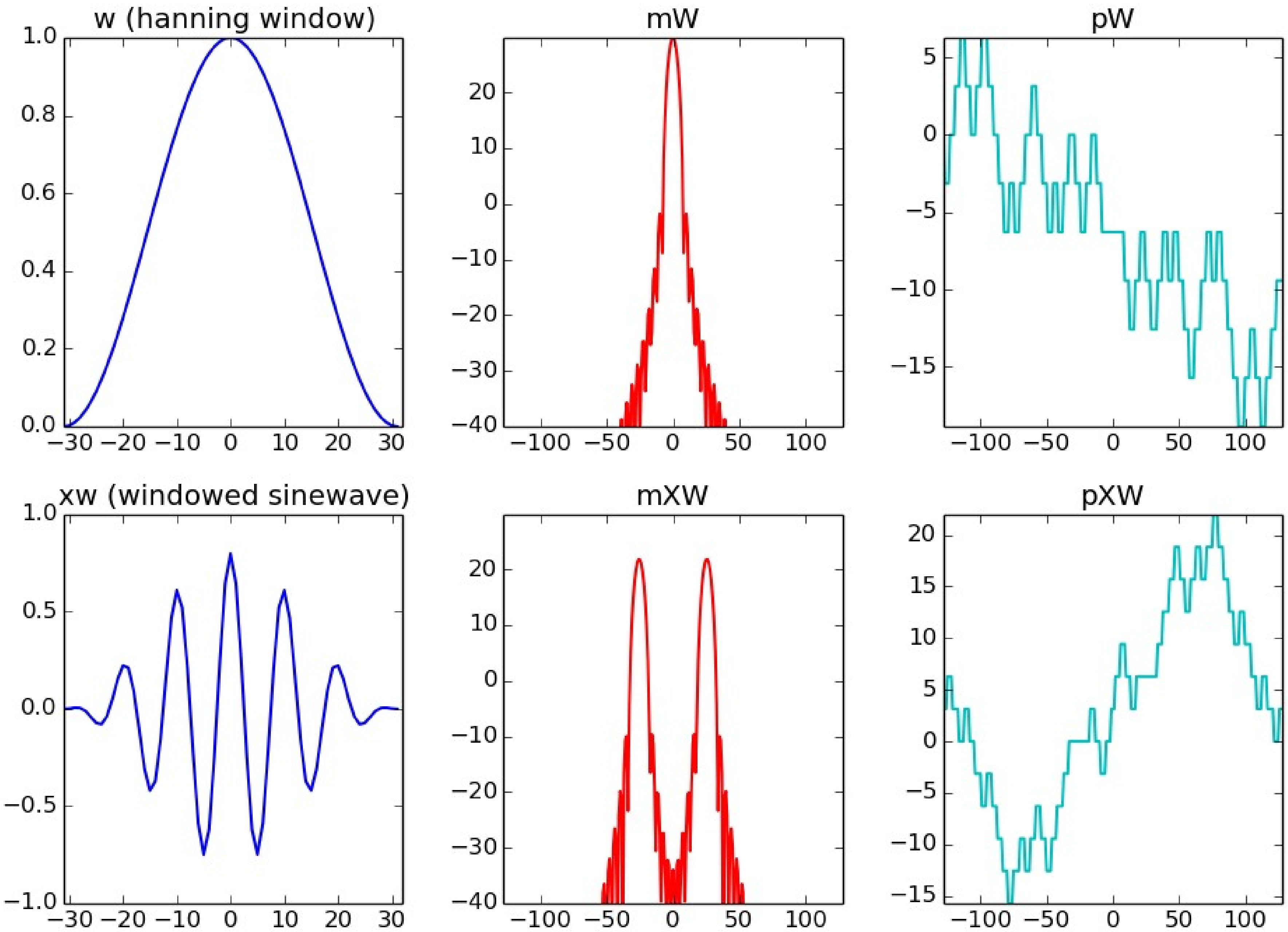
$$xw_l[n]=w[n]x[n+lH] \qquad l=0,1,\ldots,$$



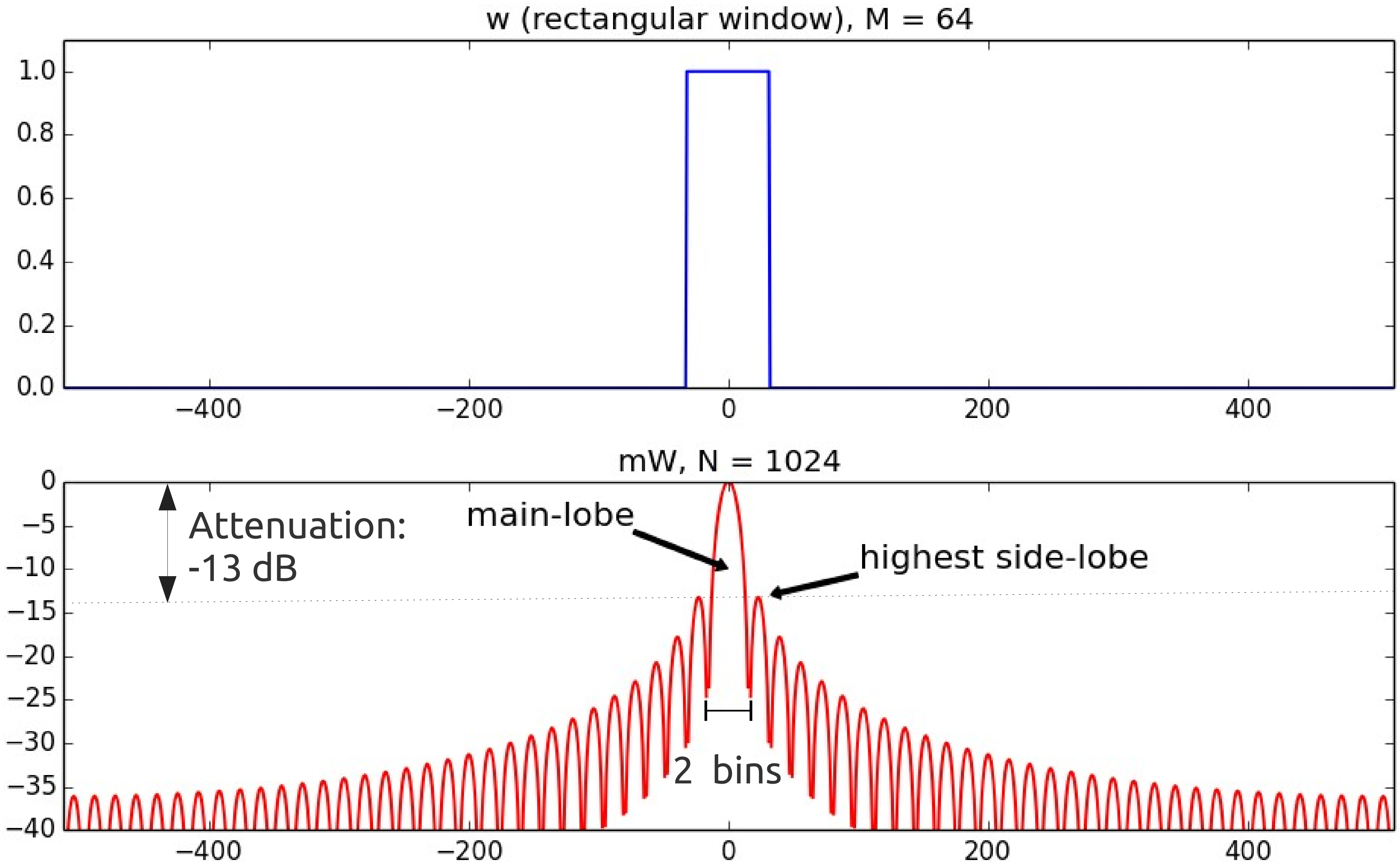
Transform of a windowed sinewave

$$x[n] = A_0 \cos\left(2\pi k_0 n/N\right) = \frac{A_0}{2} e^{j2\pi k_0 n/N} + \frac{A_0}{2} e^{-j2\pi k_0 n/N}$$

$$\begin{aligned}
X[k] &= \sum_{n=-N/2}^{N/2-1} w[n]\,x[n]\,e^{-j2\pi kn/N} \\
&= \sum_{n=-N/2}^{N/2-1} w[n]\left(\frac{A_0}{2} e^{j2\pi k_0 n/N} + \frac{A_0}{2} e^{-j2\pi k_0 n/N}\right) e^{-j2\pi kn/N} \\
&= \sum_{n=-N/2}^{N/2-1} w[n]\frac{A_0}{2} e^{j2\pi k_0 n/N} e^{-j2\pi kn/N} + \sum_{n=-N/2}^{N/2-1} w[n]\frac{A_0}{2} e^{-j2\pi k_0 n/N} e^{-j2\pi kn/N} \\
&= \frac{A_0}{2}\sum_{n=-N/2}^{N/2-1} w[n]\,e^{-j2\pi(k-k_0)n/N} + \frac{A_0}{2}\sum_{n=-N/2}^{N/2-1} w[n]\,e^{-j2\pi(k+k_0)n/N} \\
&= \frac{A_0}{2} W[k-k_0] + \frac{A_0}{2} W[k+k_0]
\end{aligned}$$


w (hanning window)
mW
pW
xw (windowed sinewave)
mXW
pXW
1.0
0.8
0.6
0.4
0.2
0.0
−30 −20 −10 0 10 20 30
20
10
0
−10
−20
−30
−40
−100 −50 0 50 100
5
−5
−15
0.5
−0.5
−1.0

Analysis window



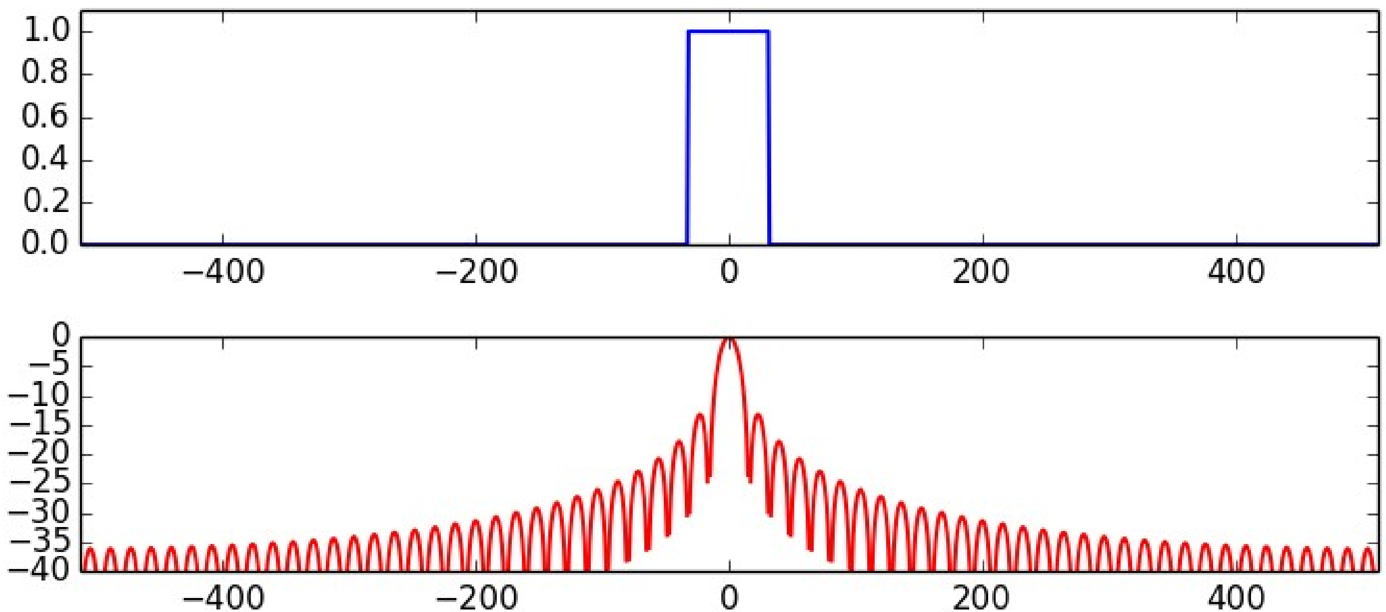
Window functions in Scipy

barthann (M[, sym])	Return a modified Bartlett-Hann window.
bartlett (M[, sym])	Return a Bartlett window.
blackman (M[, sym])	Return a Blackman window.
blackmanharris (M[, sym])	Return a minimum 4-term Blackman-Harris window.
bohman (M[, sym])	Return a Bohman window.
boxcar (M[, sym])	Return a boxcar or rectangular window.
chebwin (M, at[, sym])	Return a Dolph-Chebyshev window.
flattop (M[, sym])	Return a flat top window.
gaussian (M, std[, sym])	Return a Gaussian window.
general-gaussian (M, p, sig[, sym])	Return a window with a generalized Gaussian shape.
hamming (M[, sym])	Return a Hamming window.
hann (M[, sym])	Return a Hann window.
kaiser (M, beta[, sym])	Return a Kaiser window.
nuttall (M[, sym])	Return a minimum 4-term Blackman-Harris window according to Nuttall.
parzen (M[, sym])	Return a Parzen window.
slepian (M, width[, sym])	Return a digital Slepian window.
triang (M[, sym])	Return a triangular window.

Rectangular window

$$w[n]=1,\quad n=-M/2,\ldots,0,\ldots M/2$$
$$=0,\quad n=\text{elsewhere}$$

$$W[k]=\frac{\sin(\pi k)}{\sin(\pi k/M)}$$

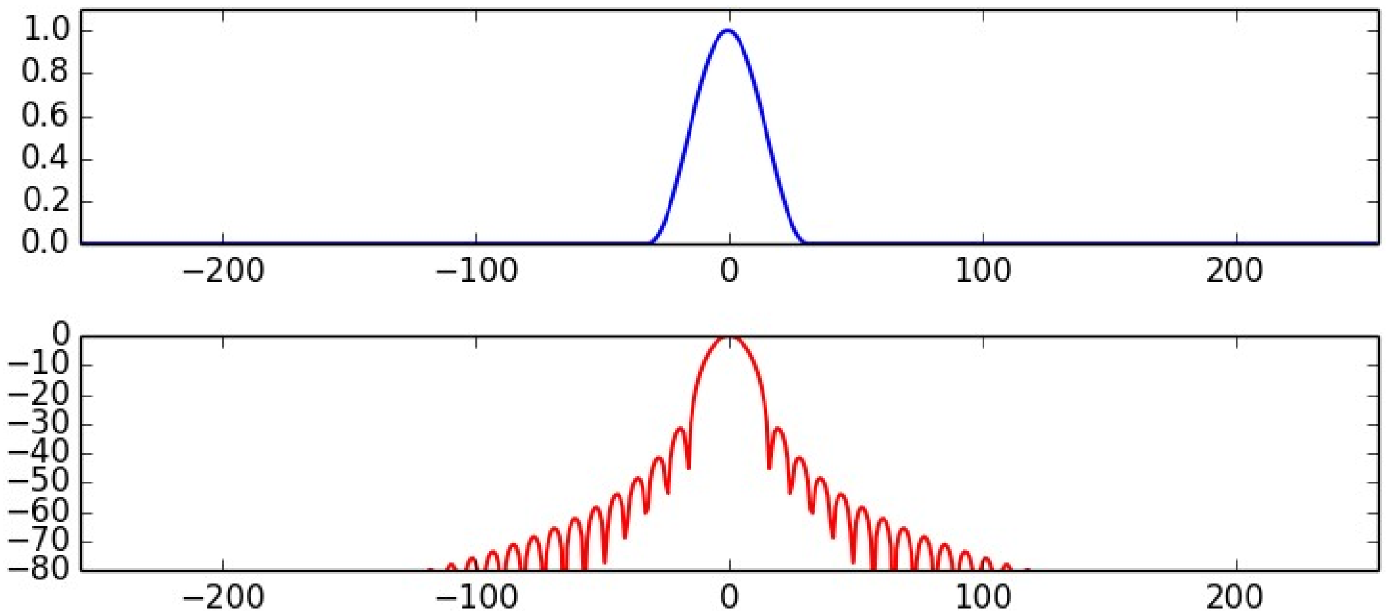


main-lobe width: 2 bins
side-lobe level: -13.3 dB

Hanning window

$$w[n] = .5 + .5\cos(2\pi n/M), \quad n = -M/2, \ldots, 0, \ldots M/2$$

$$W[k] = .5\,D[k] + .25\left(D[k-1] + D[k+1]\right) \quad \text{where}\ D[k] = \frac{\sin(\pi k)}{\sin(\pi k/M)}$$

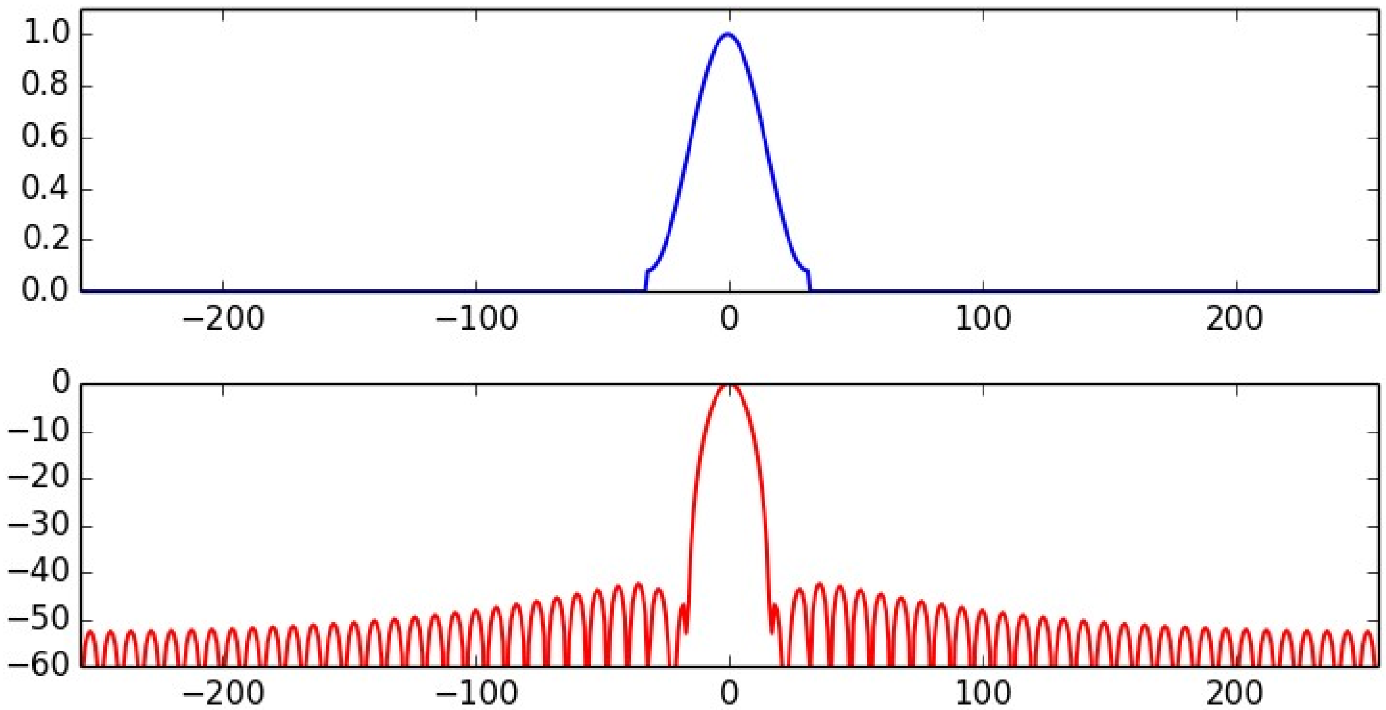


main-lobe width: 4 bins

side-lobe level: -31.5 dB

Hamming window

$$w[n] = .54 + .46\cos(2\pi n/M), \quad n = -M/2, \ldots, 0, \ldots M/2$$

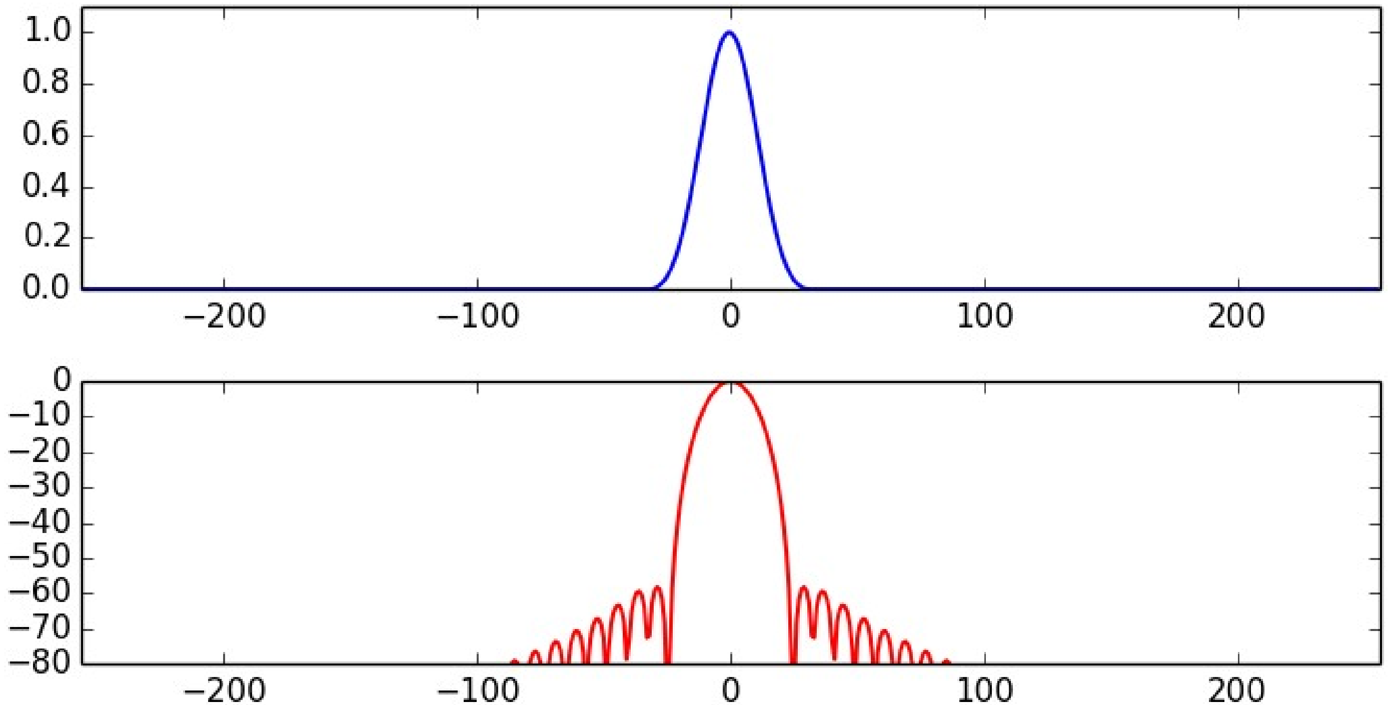


main-lobe width: 4 bins

side-lobe level: -42.7 dB

Blackman window

$$w[n] = 0.42 - 0.5\cos(2\pi n/M) + 0.08\cos(4\pi n/M)$$

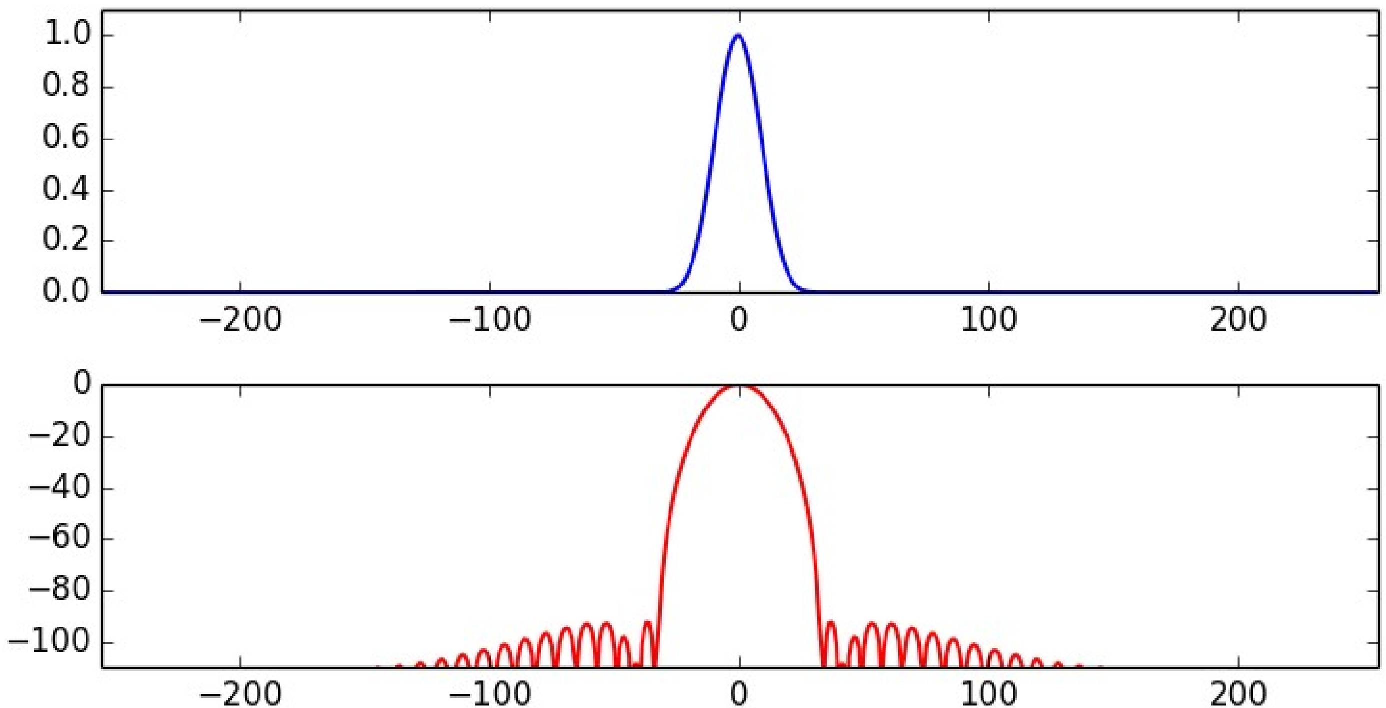


main-lobe width: 6 bins
side-lobe level: -58 dB

Blackman-Harris window

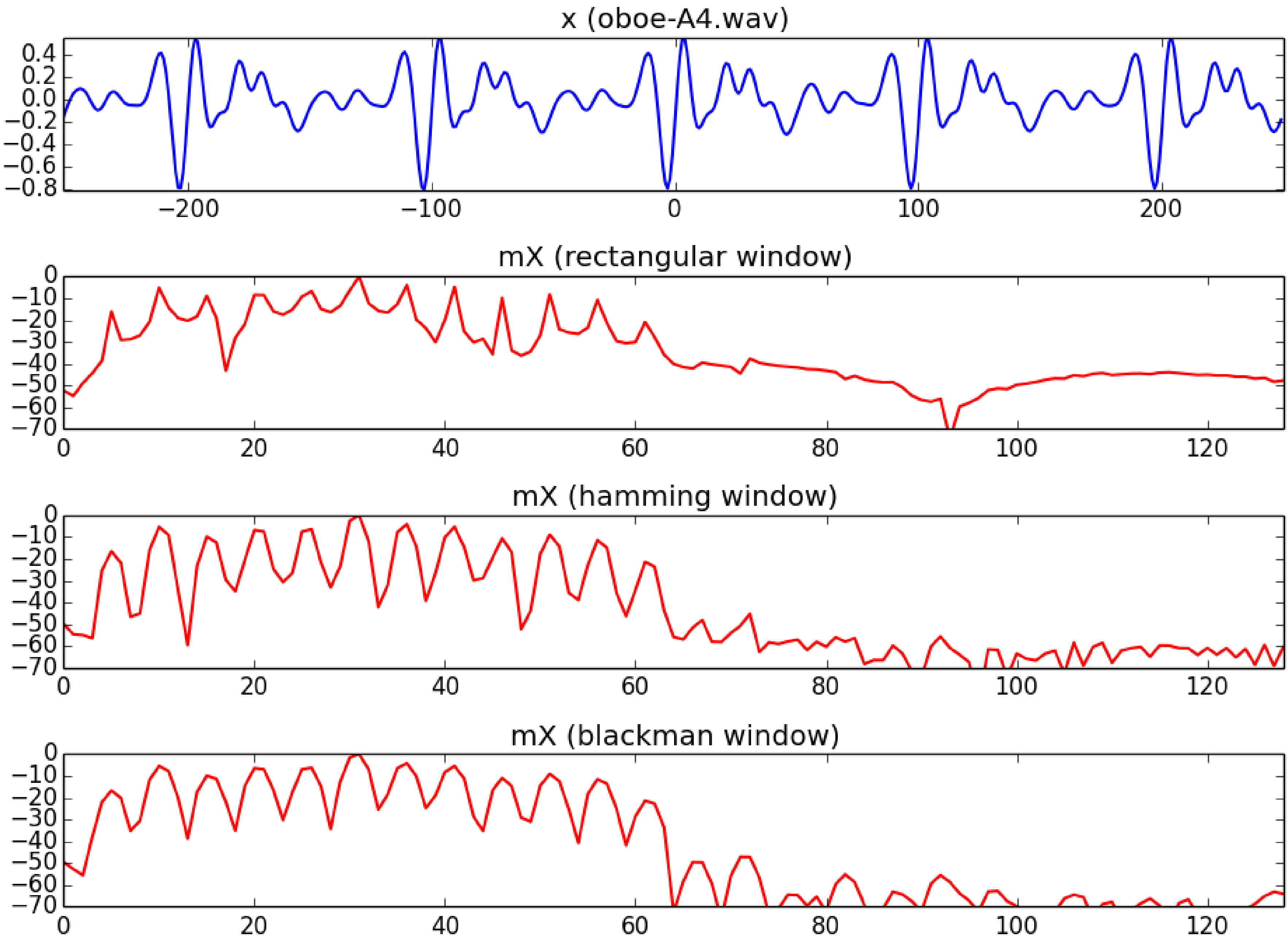
$$w(n)=\frac{1}{M}\sum_{l=0}^{3}\alpha_l\cos(2nl\pi/M),\quad n=-M/2,\ldots 0,\ldots M/2$$

where $\alpha_0=0.35875,\alpha_1=0.48829,\alpha_2=0.14128,\alpha_3=0.01168$



$\text{main lobe width}: 8\ \text{bins}$

$\text{side-lobe level}: -92\ dB$



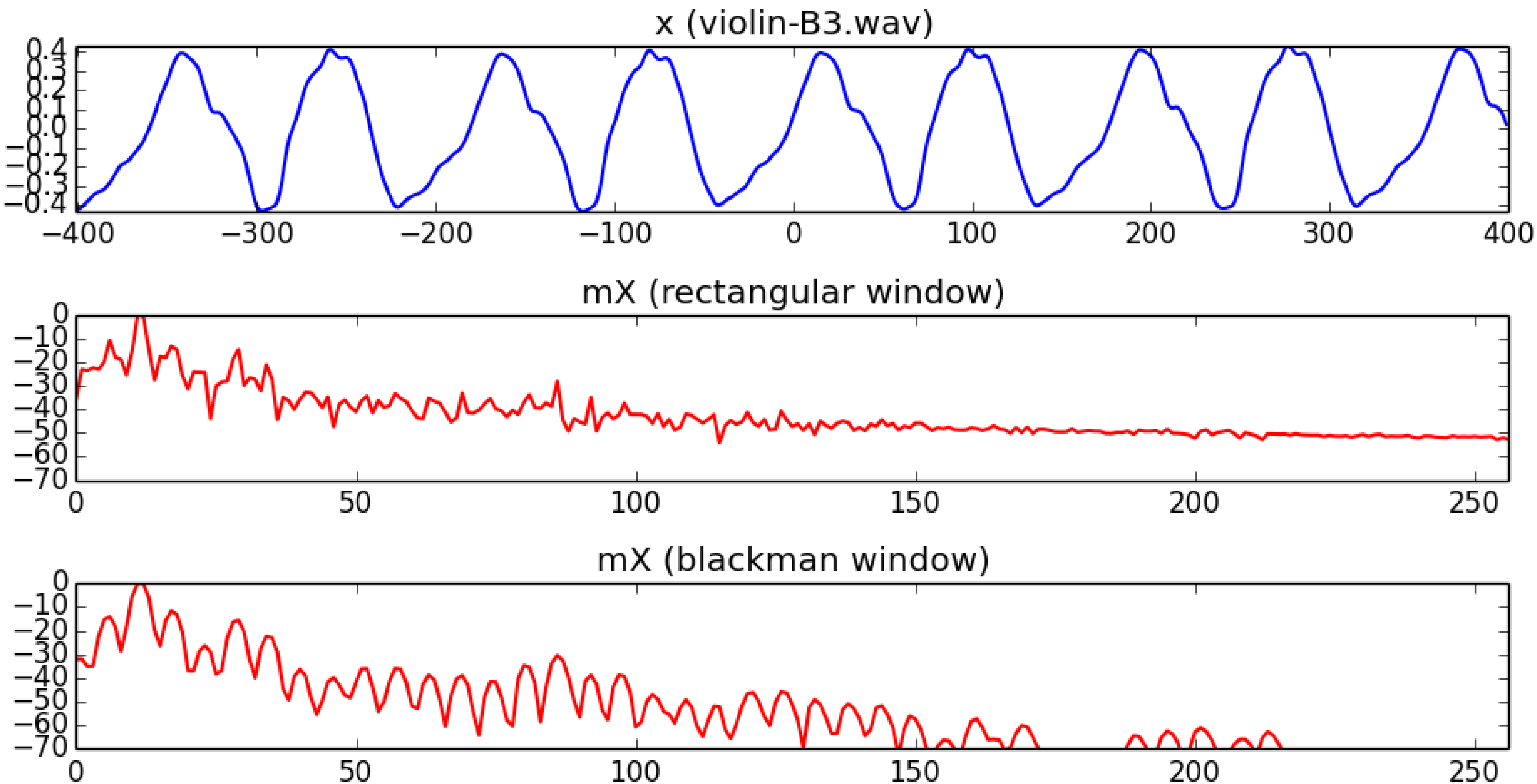
x (oboe-A4.wav)
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−200
−100
0
100
200
mX (rectangular window)
mX (hamming window)
mX (blackman window)
0
−10
−20
−30
−40
−50
−60
−70
0
20
40
60
80
100
120

Index

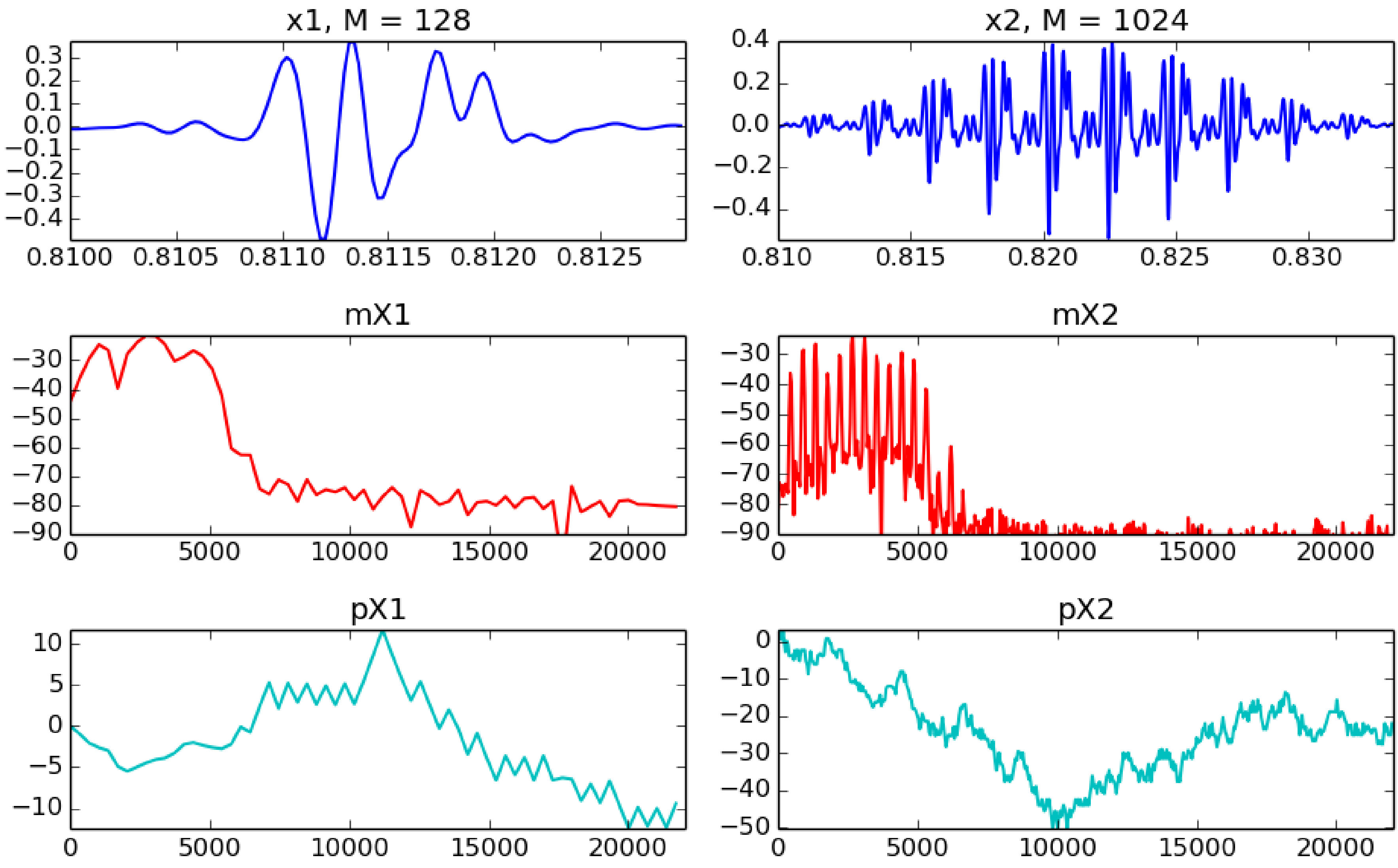
- STFT and analysis window
- Window size
- FFT size
- Hop size
- Time-frequency compromise
- Inverse STFT
- STFT system

STFT and analysis window

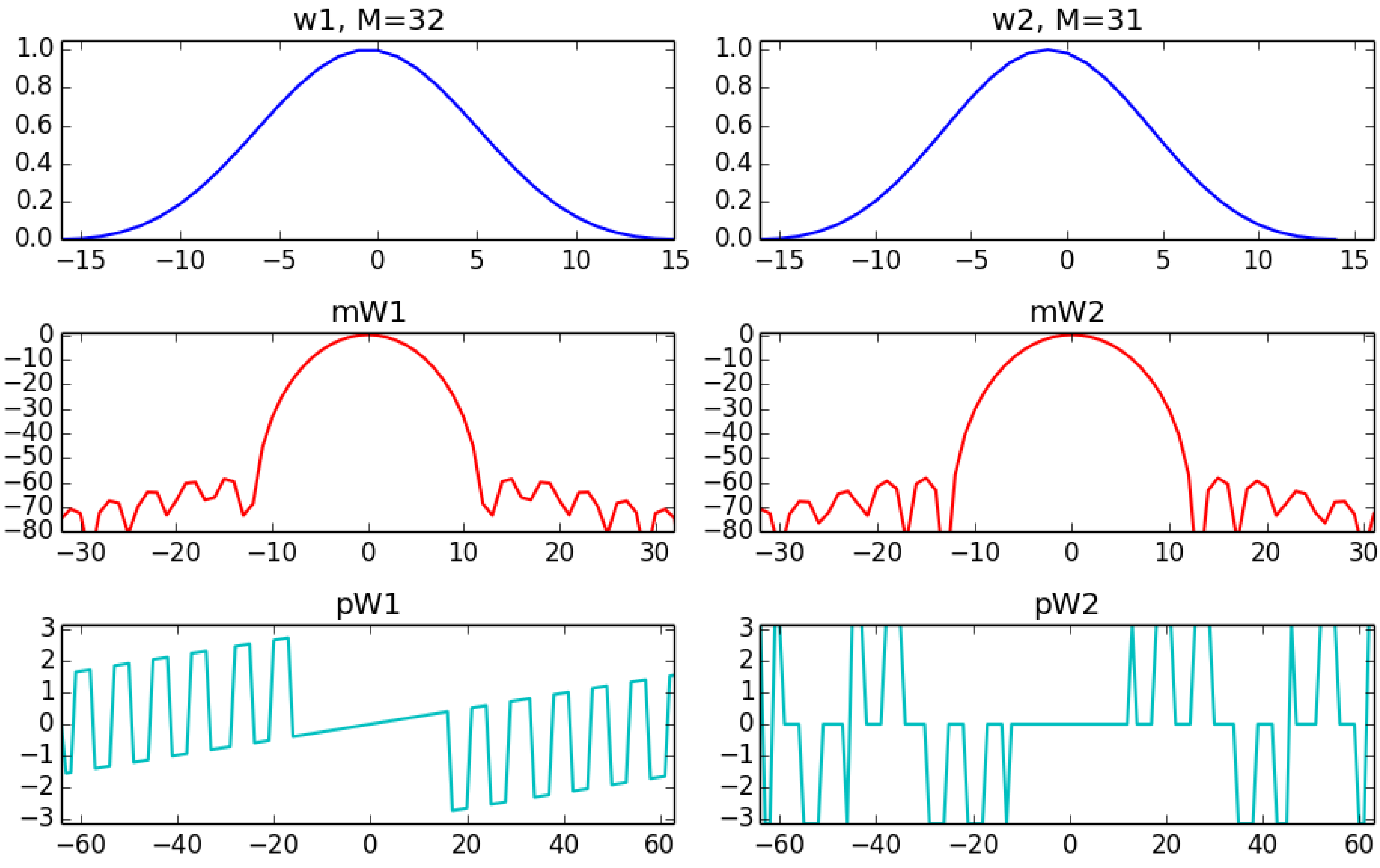
$$X_l[k] = \sum_{n=-N/2}^{N/2-1} w[n]\,x[n+lH]\,e^{-j2\pi kn/N} \qquad l = 0,1,\ldots,$$



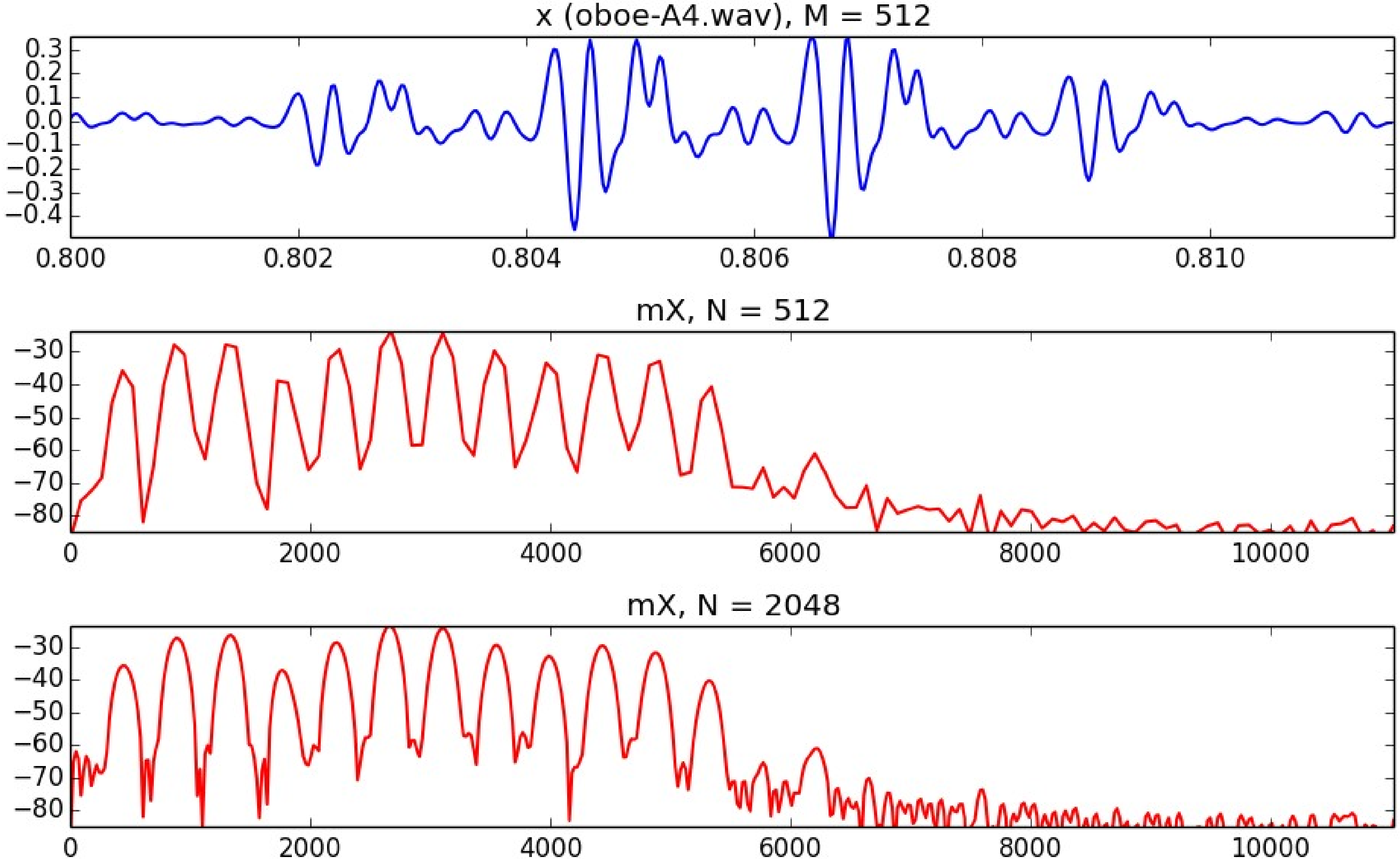
Window size



Even-odd size window

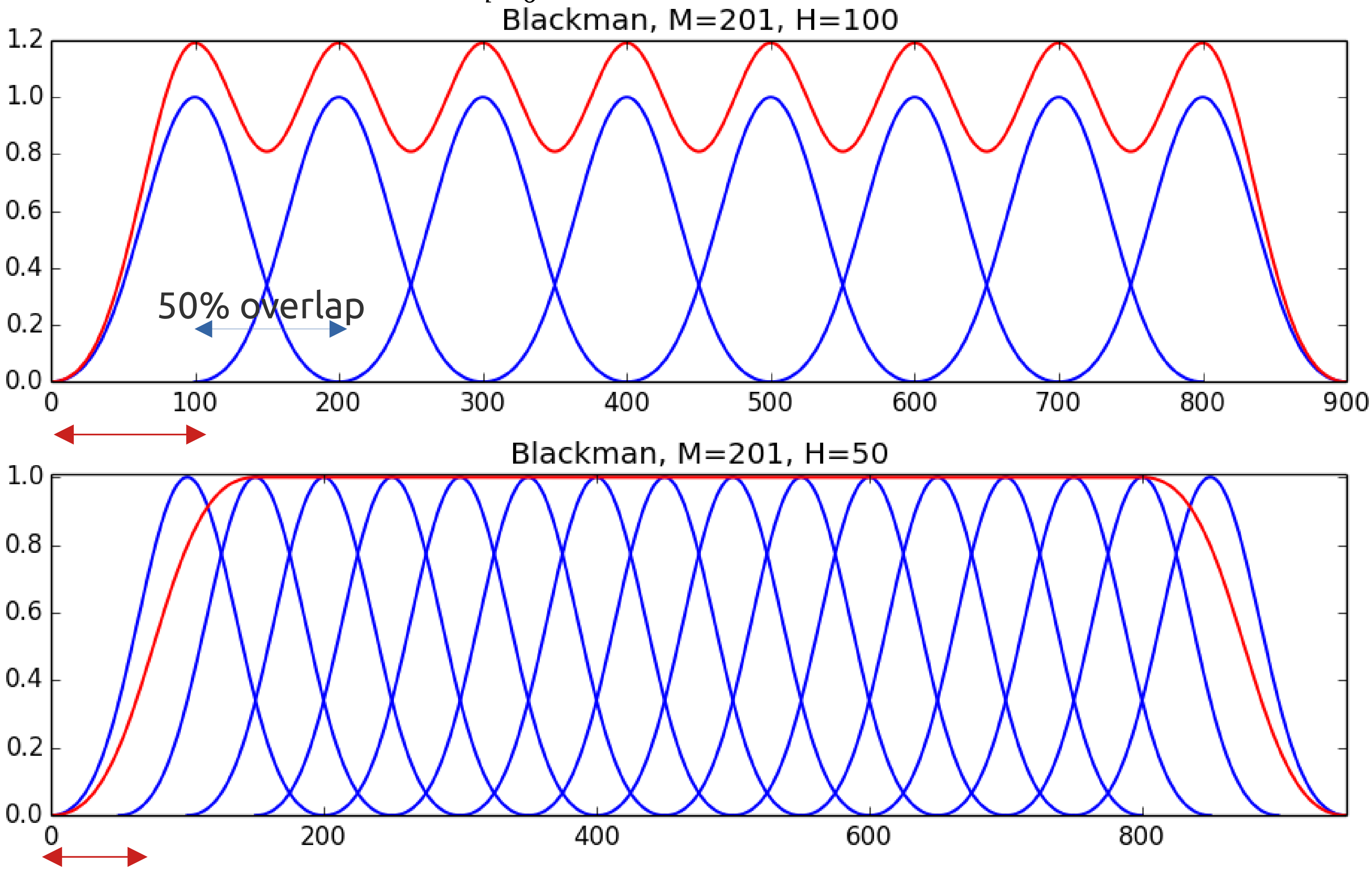


FFT size (n_fft)

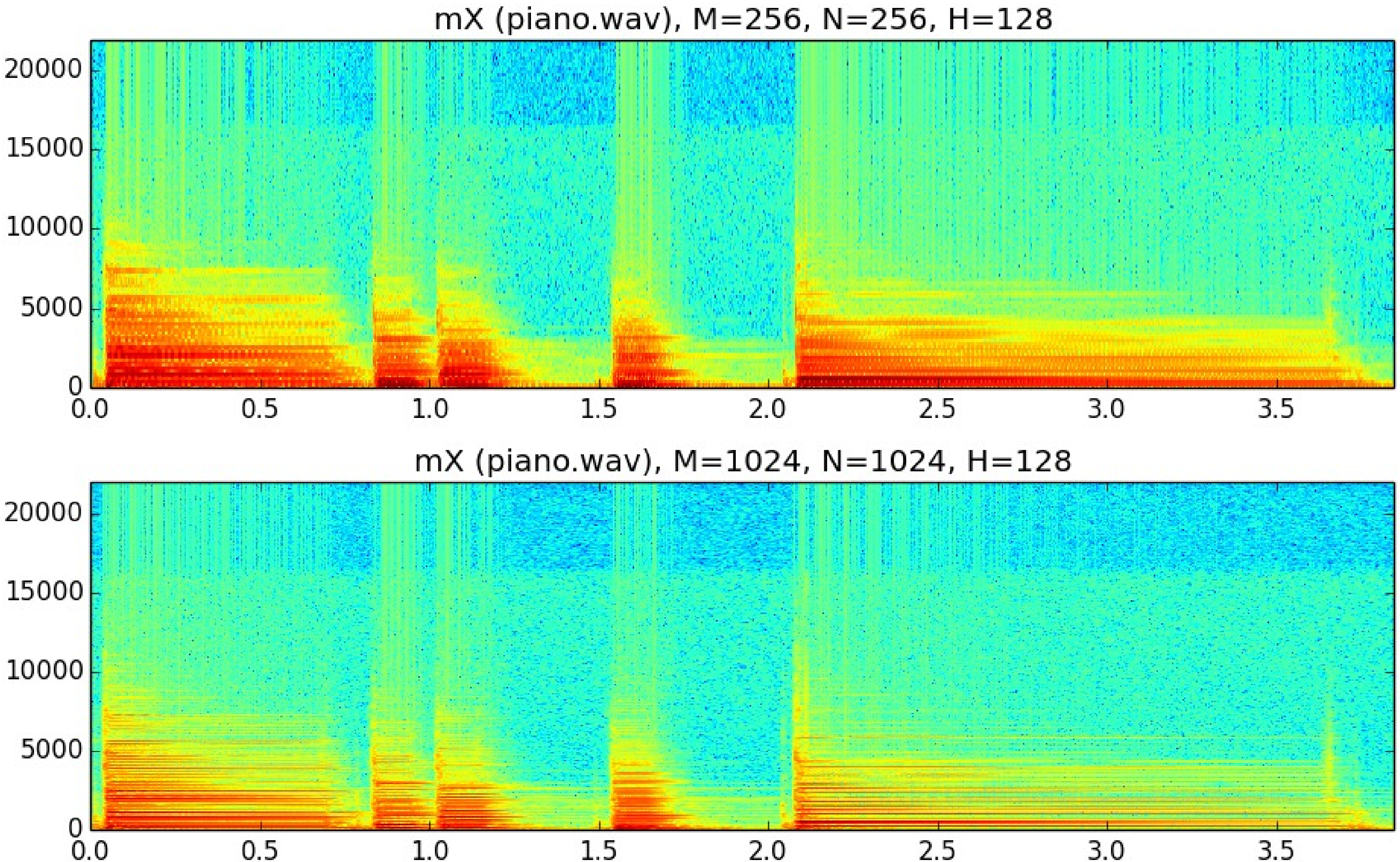


Hop size $A_w[n] = \sum_{l=0}^{L-1} w[n - lH] = c$

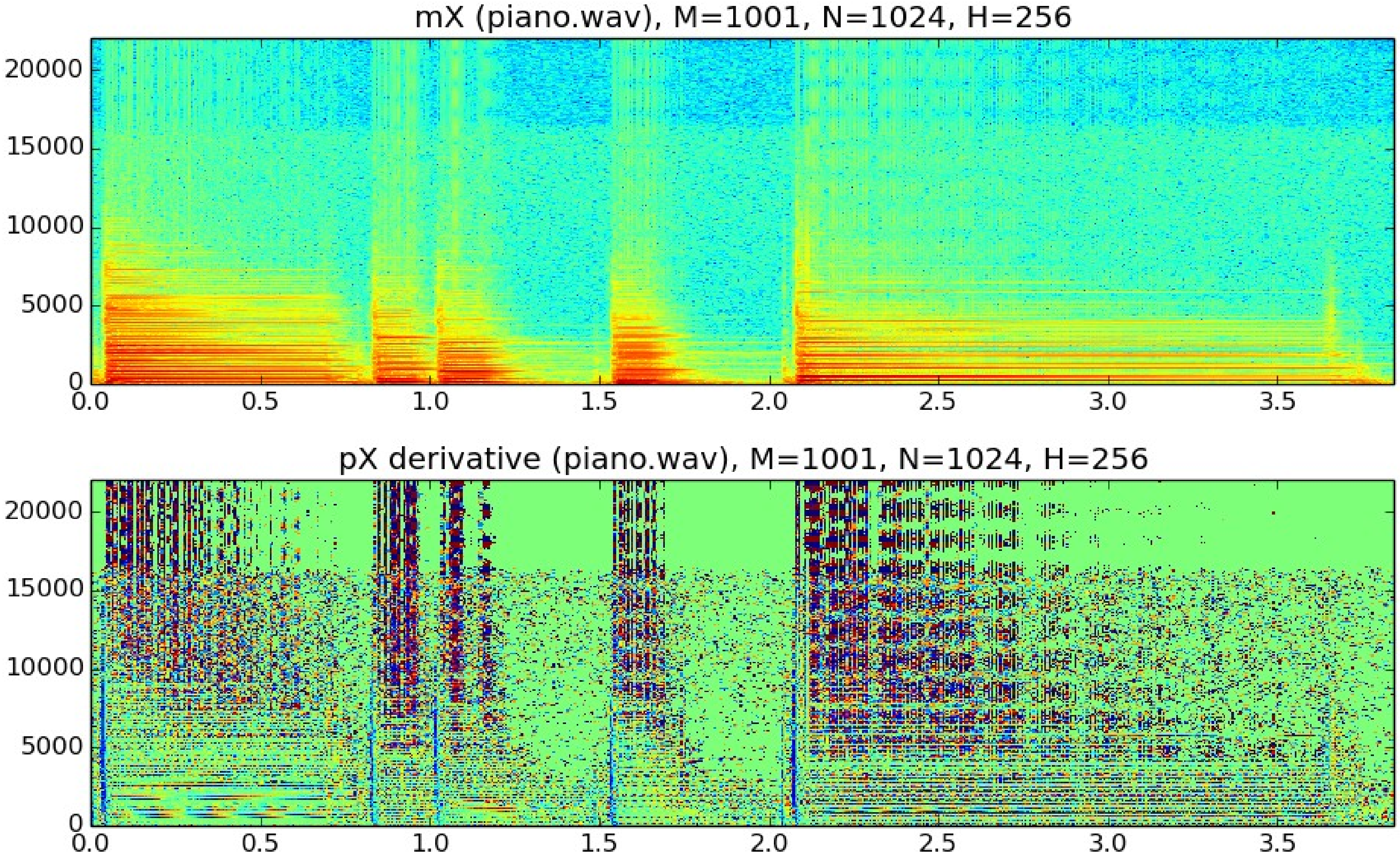
= step size



Time-frequency compromise



Amplitude and phase spectrogram



Inverse STFT

$$y[n]=\sum_{l=0}^{L-1} Shift_{lH,n}\left[\frac{1}{N}\sum_{k=-N/2}^{N/2-1} X_l[k]e^{j2\pi kn/N}\right]$$

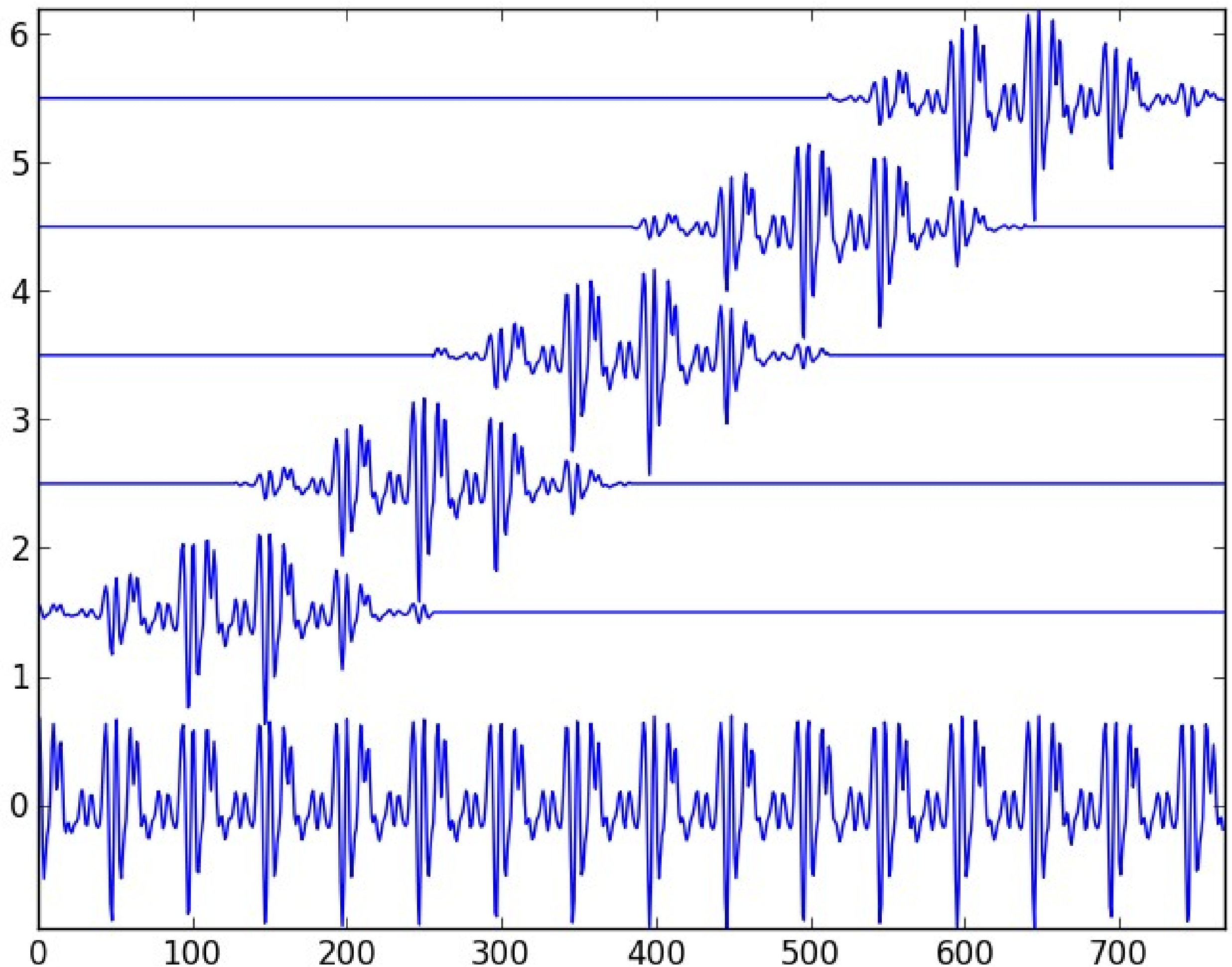
each output frame is:

$$yw_l[n]=x(n+lH)w[n]$$

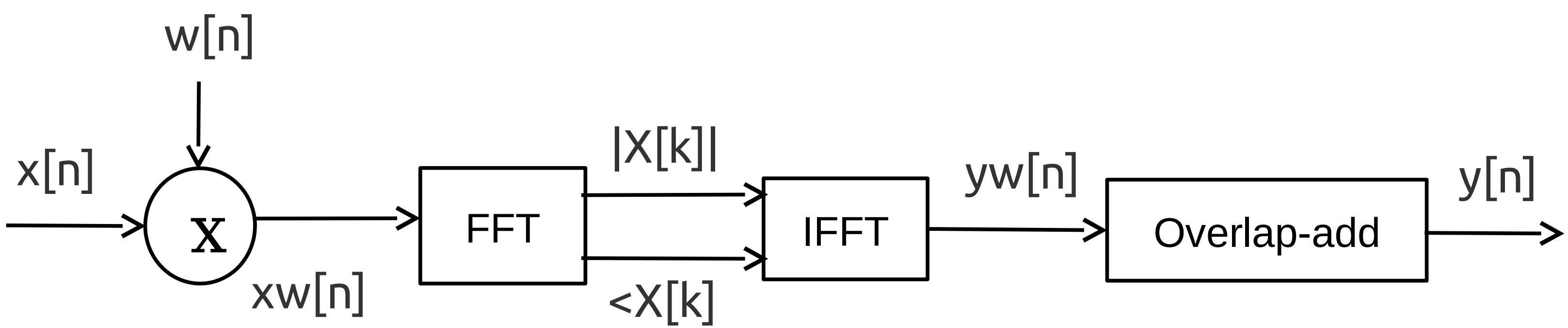
and the output sound is:

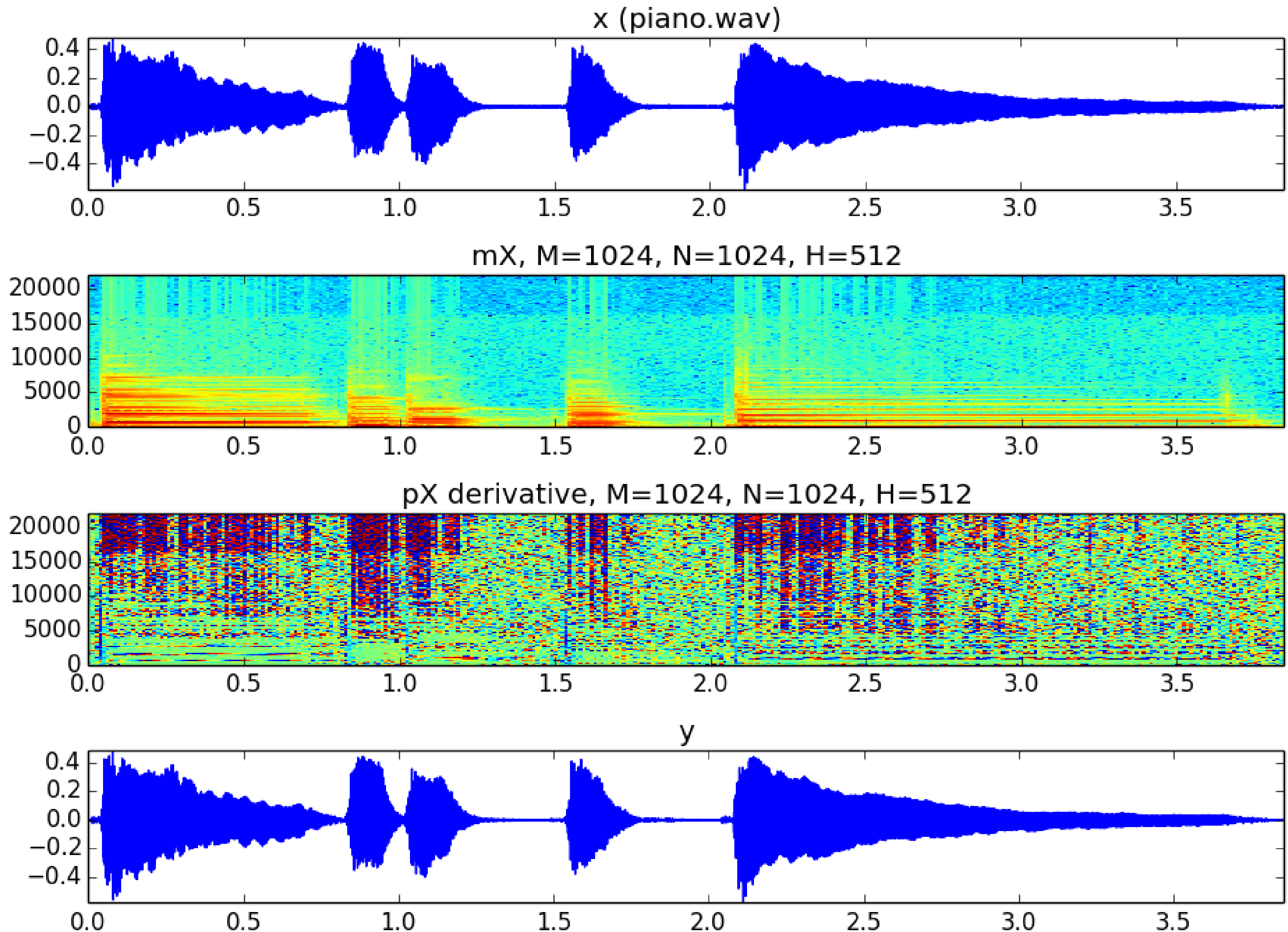
$$y[n]=\sum_{l=0}^{L-1} yw_l[n]=x[n]\sum_{l=0}^{L-1} w[n-lH]$$

$$yw_l[n]=w[n]\,x[n+lH] \qquad l=0,1,\ldots,$$



STFT system




x (piano.wav)
mX, M=1024, N=1024, H=512
pX derivative, M=1024, N=1024, H=512
y

Spectrogram Demo

Index

- Introduction: acoustic features
- Single-frame spectral features
- Multiple-frames spectral features

Single-frame spectral features

- Energy, RMS, Loudness
- Spectral centroid
- Mel-frequency cepstral coefficients (MFCC)
- Also known as segmental features and low-level descriptors (LLD)

Energy, RMS, Loudness

Energy:

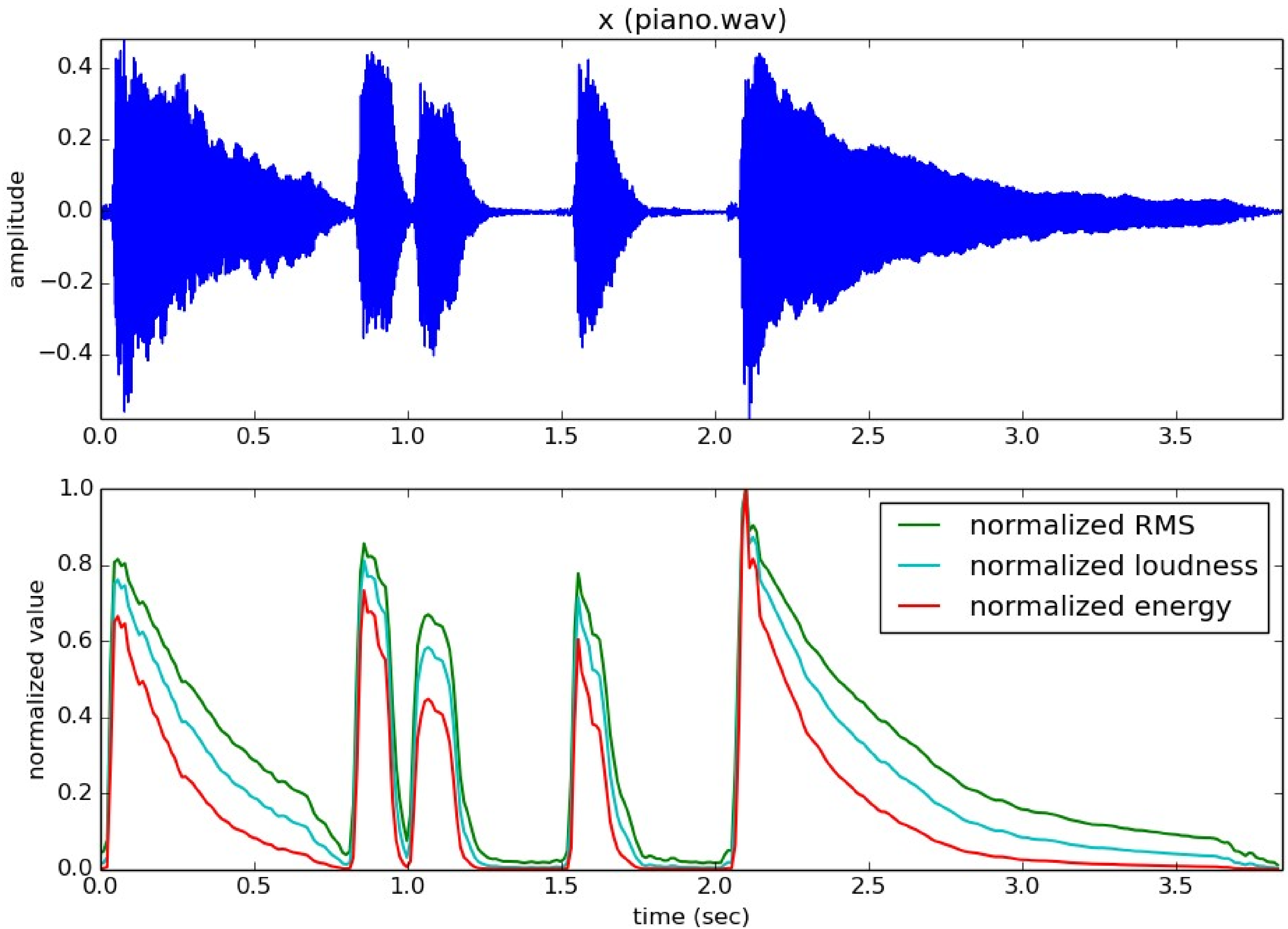
$$energy_l = \sum_{k=0}^{N-1} |X_l[k]|^2$$

Root mean square:

$$RMS_l = \sqrt{\frac{1}{N^2} \sum_{k=0}^{N-1} |X_l[k]|^2}$$

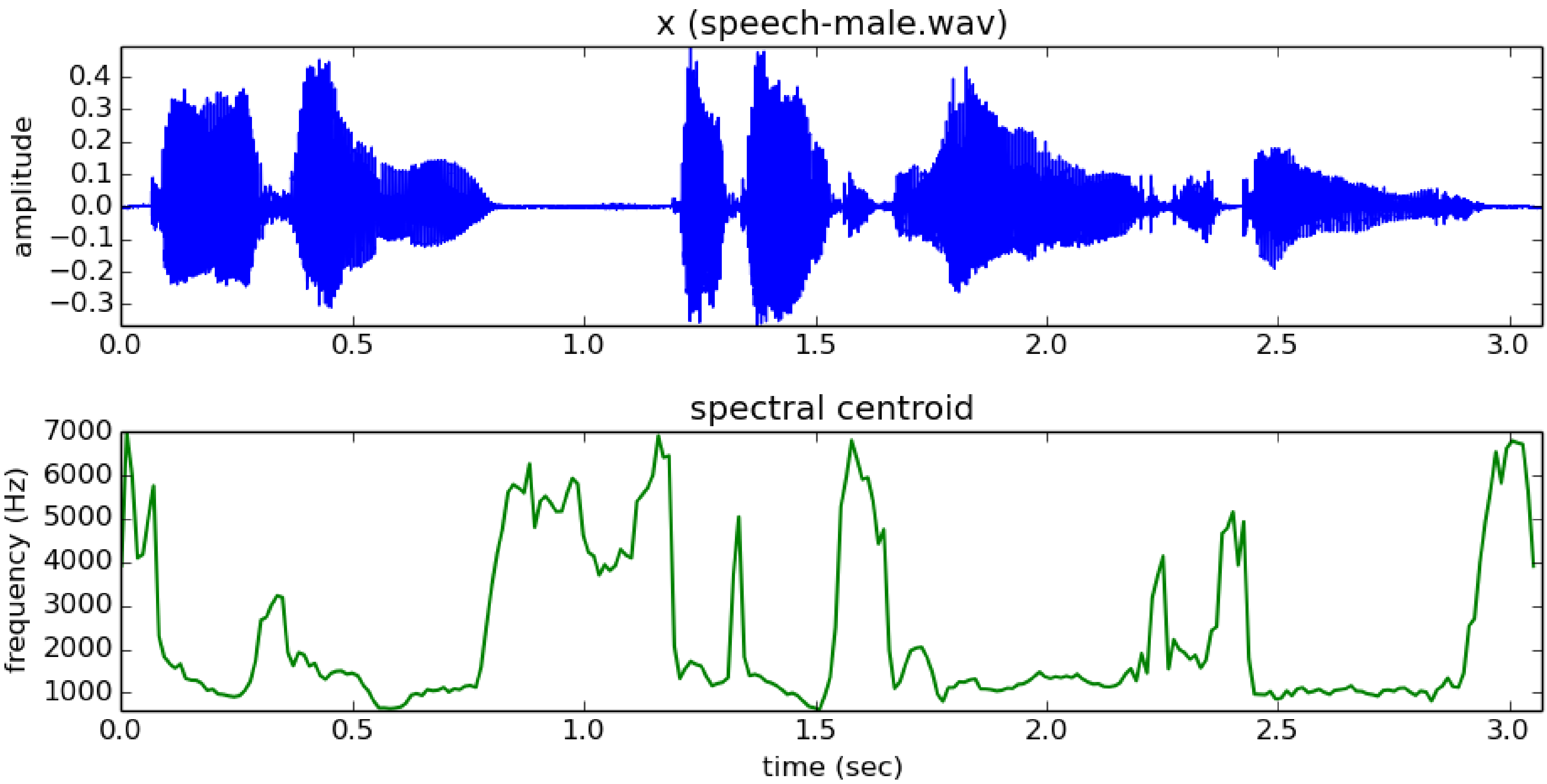
Steven's power law:

$$loudness_l = \left(\sum_{k=0}^{N-1} |X_l[k]|^2\right)^{0.67}$$


x (piano.wav)
amplitude
0.4
0.2
0.0
−0.2
−0.4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
normalized value
1.0
0.8
0.6
0.4
0.2
0.0
normalized RMS
normalized loudness
normalized energy
time (sec)

Spectral centroid

$$centroid_l = \frac{\sum_{k=0}^{N/2} k |X_l[k]|}{\sum_{k=0}^{N/2} |X_l[k]|}$$



Mel frequency cepstral coefficients (MFCC)

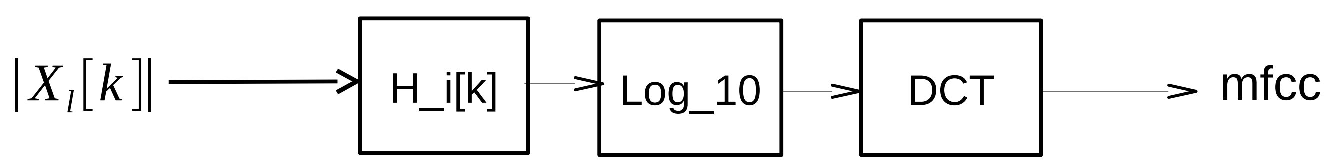
$$mfcc_l = DCT\left(\log_{10}\left(\sum_{k=0}^{N/2} |X_l[k]| H_i[k]\right)\right)$$

where

$|X[k]|$ is the positive magnitude spectrum

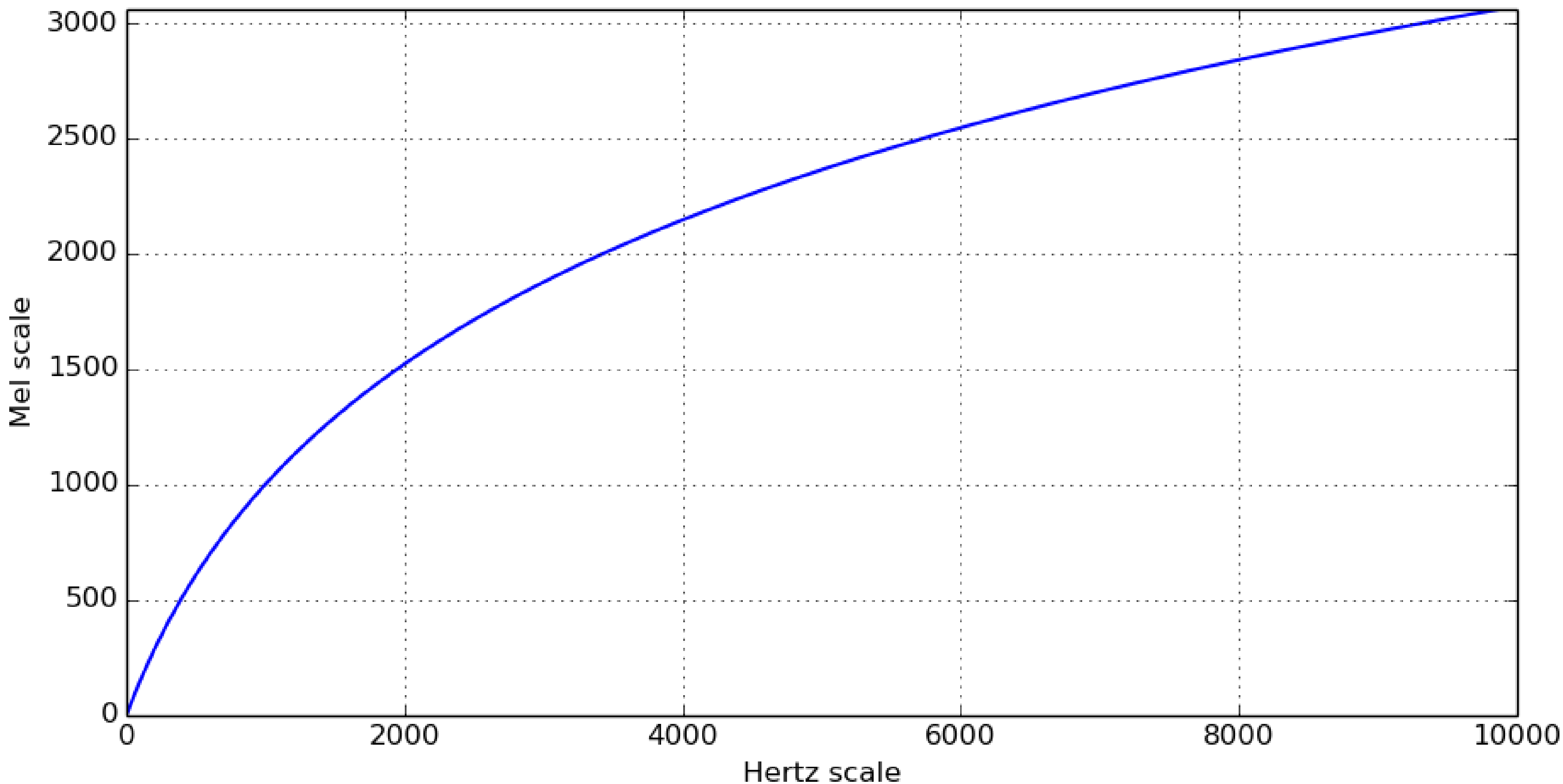
$H_i[k]$ is the mel scale filter bank for each filter i

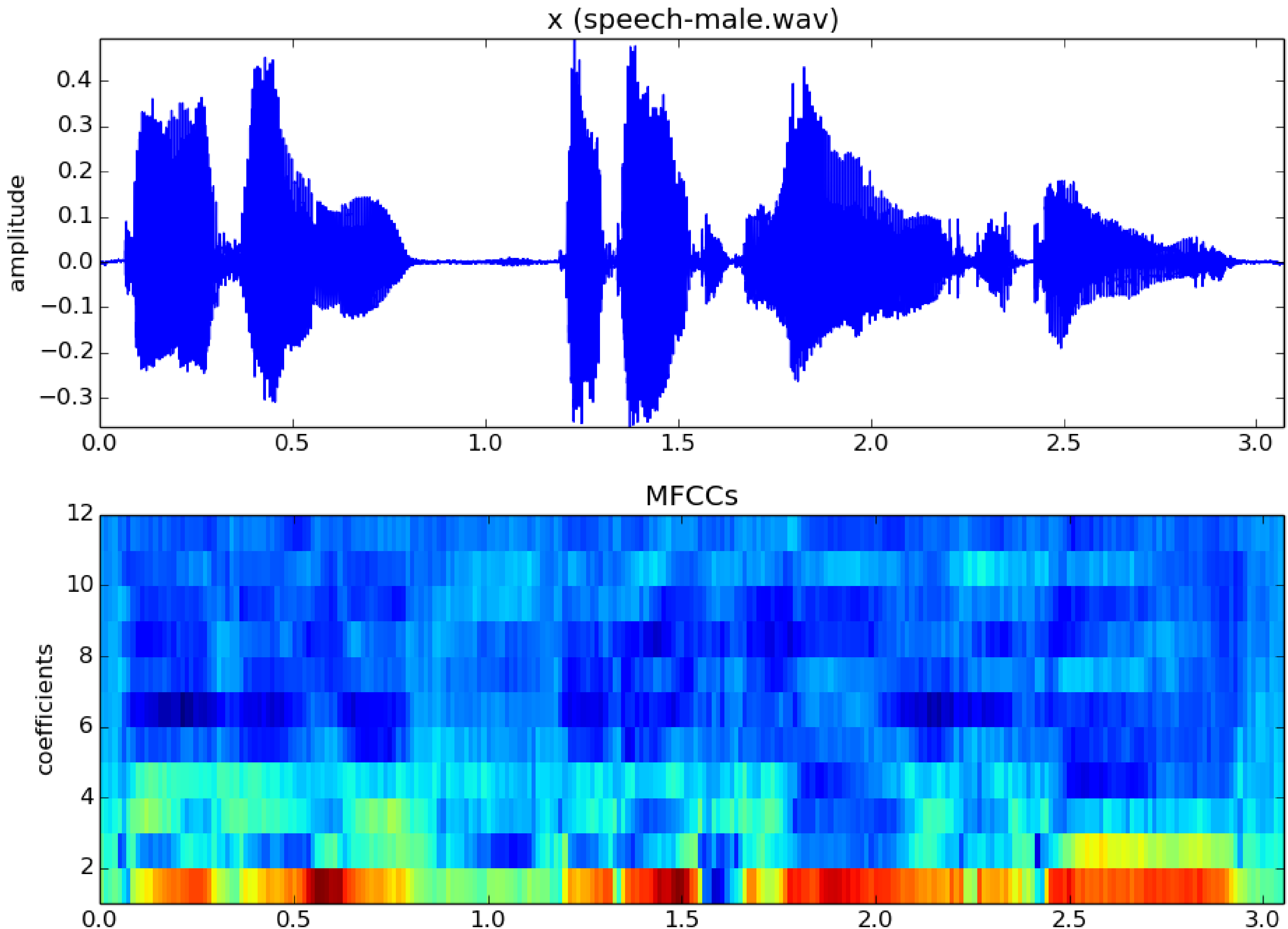
$$DCT[m] (\text{Discrete Cosine Transform}) = \sum_{n=0}^{N-1} f[n] \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)m\right)$$



MFCC: Mel scale

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





x (speech-male.wav)
amplitude
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0.0
0.5
1.0
1.5
2.0
2.5
3.0
MFCCs
coefficients
12
10
8
6
4
2
0.0
0.5
1.0
1.5
2.0
2.5
3.0

Multiple-frames spectral features

- Event segmentation, onsets
- Predominant pitch
- Statistics of single-frame features

Event segmentation, onsets

- Spectral flux (used in segmentation)

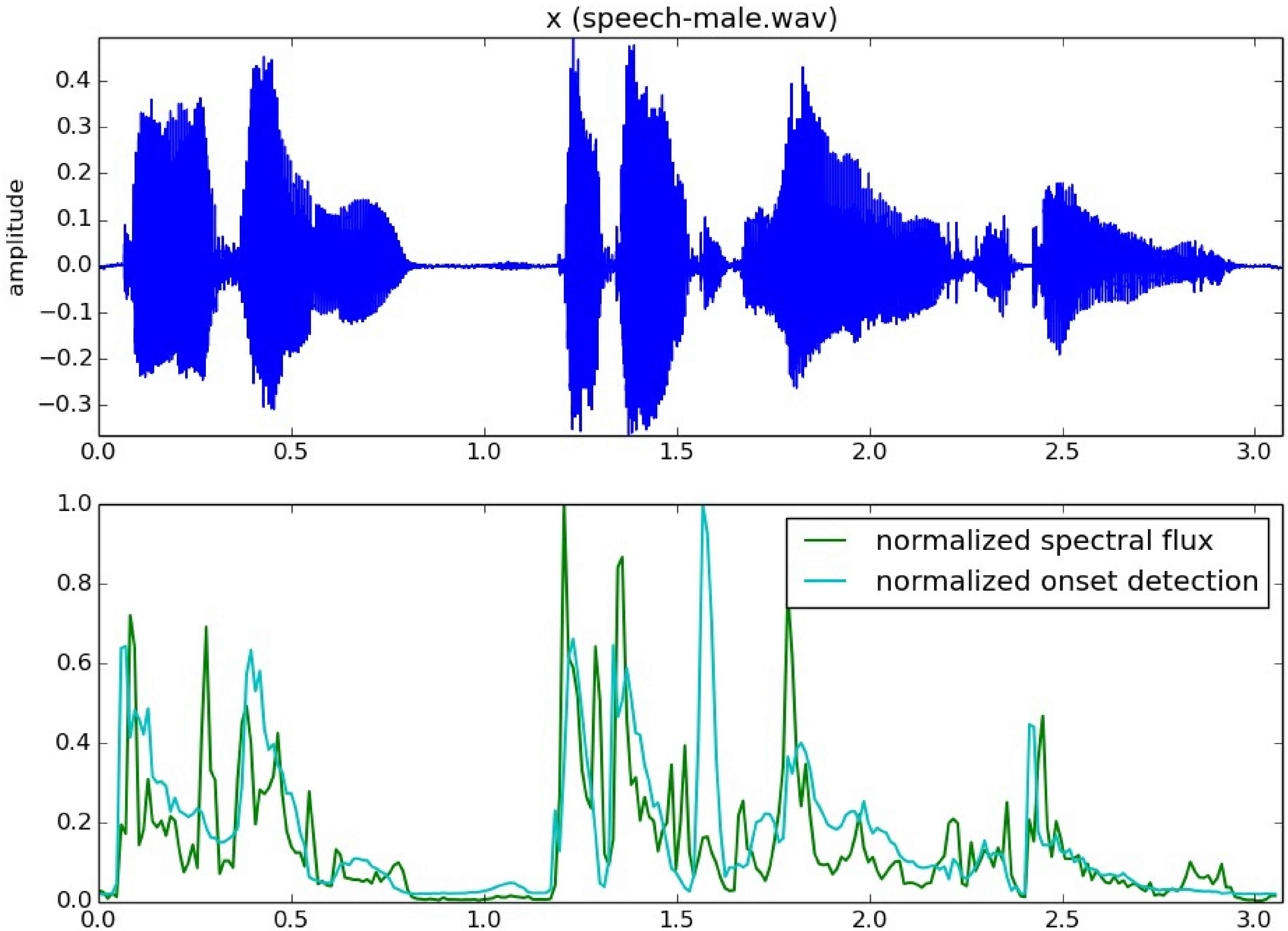
$$SF_l = \sum_{k=0}^{N/2} H\left(|X_l[k]| - |X_{(l-1)}[k]|\right)$$

$$\text{where } H(x) = \frac{x + |x|}{2}$$

- Onset detection based on high-frequency content

$$\text{Onset detection function} = HFC_l - HFC_{(l-1)}$$

$$\text{where} \quad HFC_l = \sum_{k=1}^{N/2} |X_l[k]| k^2$$


x (speech-male.wav)
amplitude
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0.0
0.5
1.0
1.5
2.0
2.5
3.0
normalized spectral flux
normalized onset detection
1.0
0.8
0.6
0.4
0.2
0.0

Statistics of single frame features

- Arithmetic mean (first moment)

$$mean = \frac{1}{N} \sum_{i=0}^{N-1} y[i]$$

- Variance (second moment)

$$variance = \frac{1}{N} \sum_{i=0}^{N-1} \left(y[i] - mean\right)^2$$

- Skewness (third moment)

$$skewness = \frac{\frac{1}{N} \sum_{i=0}^{N-1} \left(y[i] - mean\right)^3}{\left[\frac{1}{N-1} \sum_{i=0}^{N-1} \left(y[i] - mean\right)^2\right]^{3/2}}$$

Task: download the following

- Paper
  - Independent component analysis: algorithms and applications, A. Hyva¨rinen, E. Oja
  - On The Differences Between Song and Speech Emotion Recognition: Effect of Feature Sets, Feature Types, and Classifiers, B.T. Atmaja, M. Akagi
- Tools:
  - Tensorflow==2.5.0, tensorflow-io==0.18.0
  - Scikit-learn
  - Python Numerical Tours: https://nbviewer.jupyter.org/github/gpeyre/numerical-tours/blob/master/python/audio_2_separation.ipynb