Computational Perception

15-485/785

February 7, 2008

Auditory Coding 2

Review of last lecture: linear coding methods

Goal is to describe the data to desired precision.
Code signal by linear superposition of basis functions:

$$\begin{aligned} \mathbf{x} &= \vec{a}_1 s_1 + \vec{a}_2 s_2 + \cdots + \vec{a}_L s_L + \vec{\epsilon} \\ &= \mathbf{A}\mathbf{s} + \boldsymbol{\epsilon} \end{aligned}$$

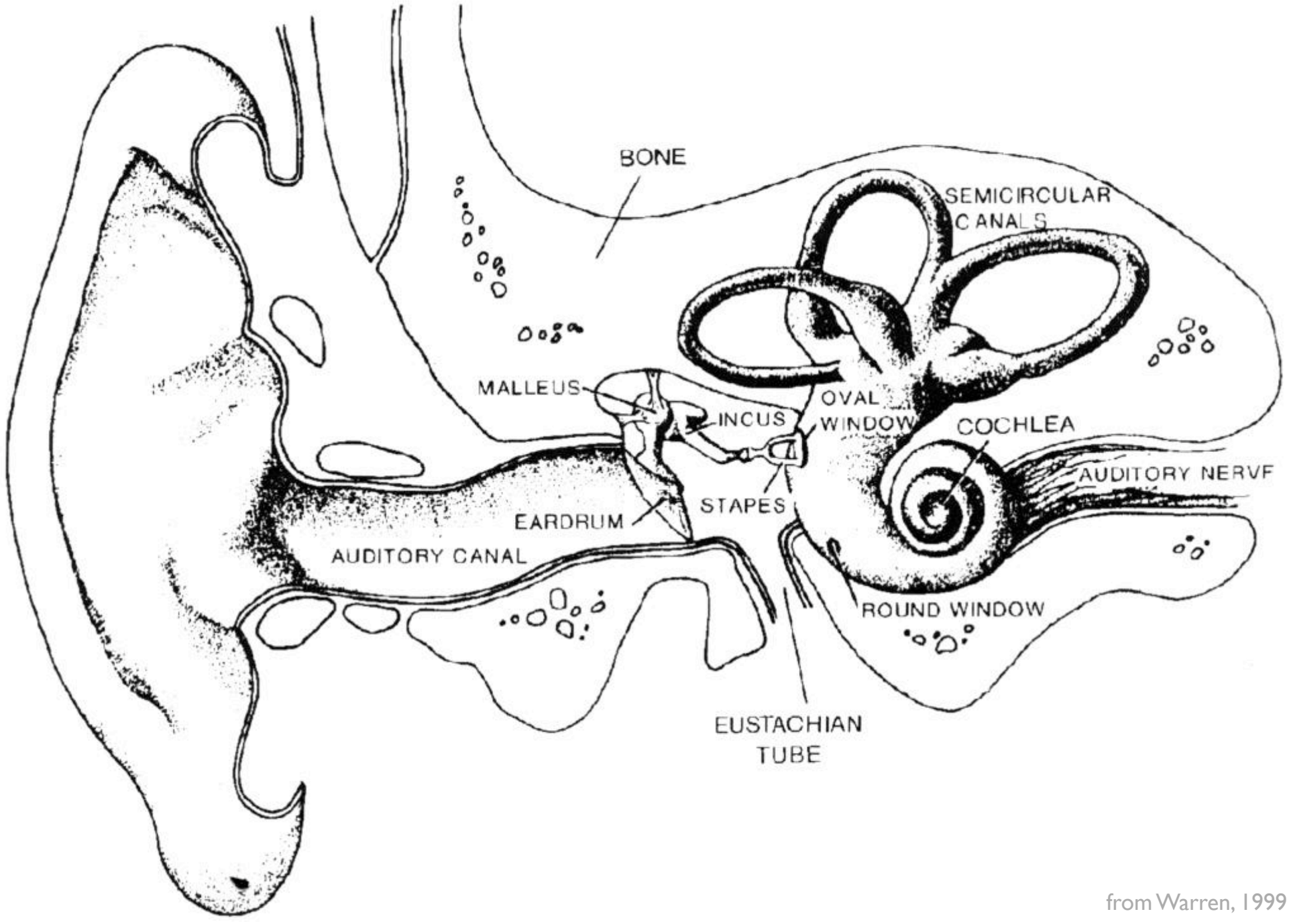
- x(t) is represented by a vector $\mathbf{x}$
- $\vec{a}_i$ are the *basis vectors*
- $\mathbf{A}$ is the *basis* (could be Fourier, wavelet, etc.)
- s_i are the *coefficients*

Can solve for ŝ in the no noise case

$$\hat{\mathbf{s}} = \mathbf{A}^{-1}\mathbf{x}$$

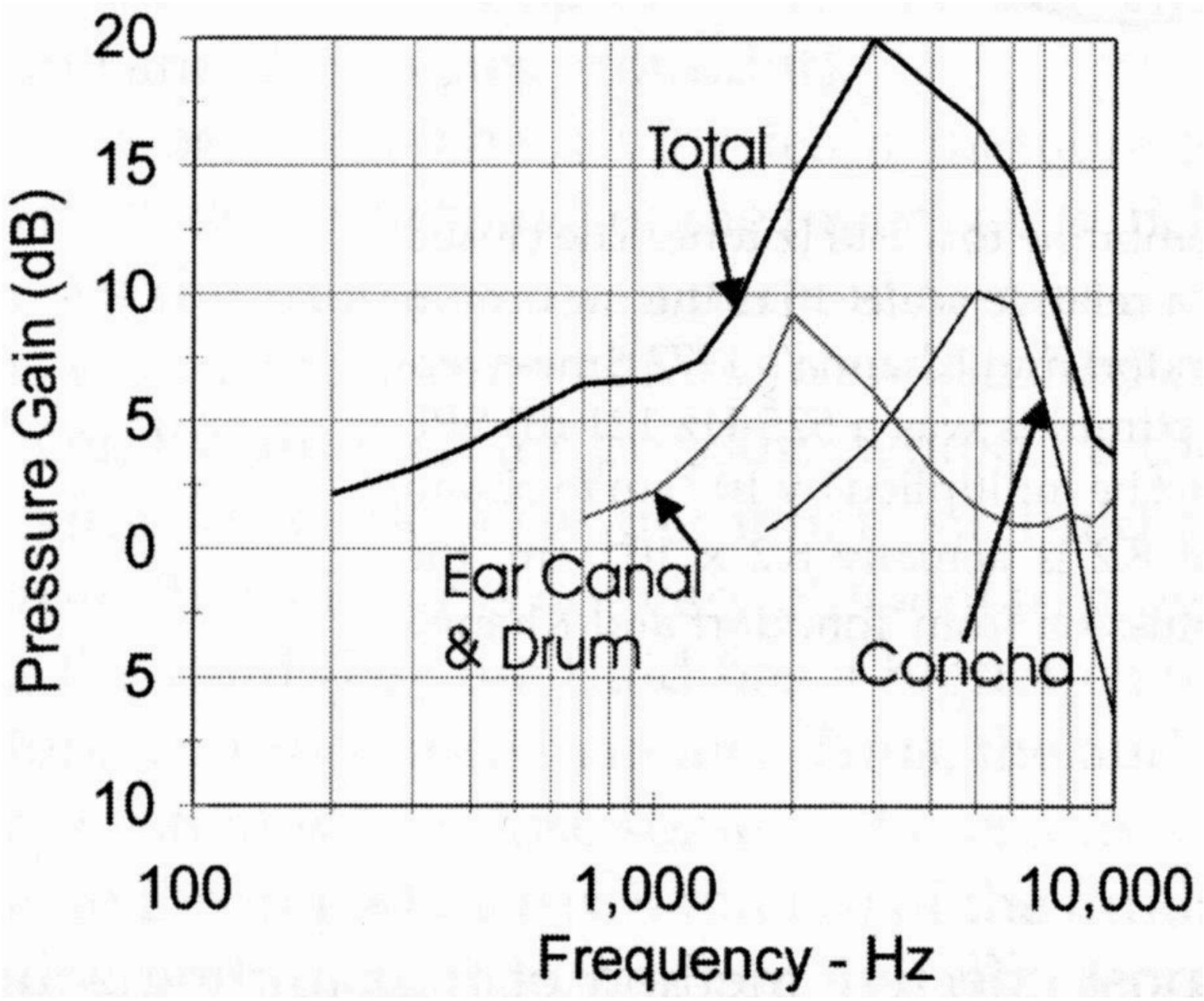
Is the peripheral auditory system a linear system?

Need a functional description of the peripheral auditory system



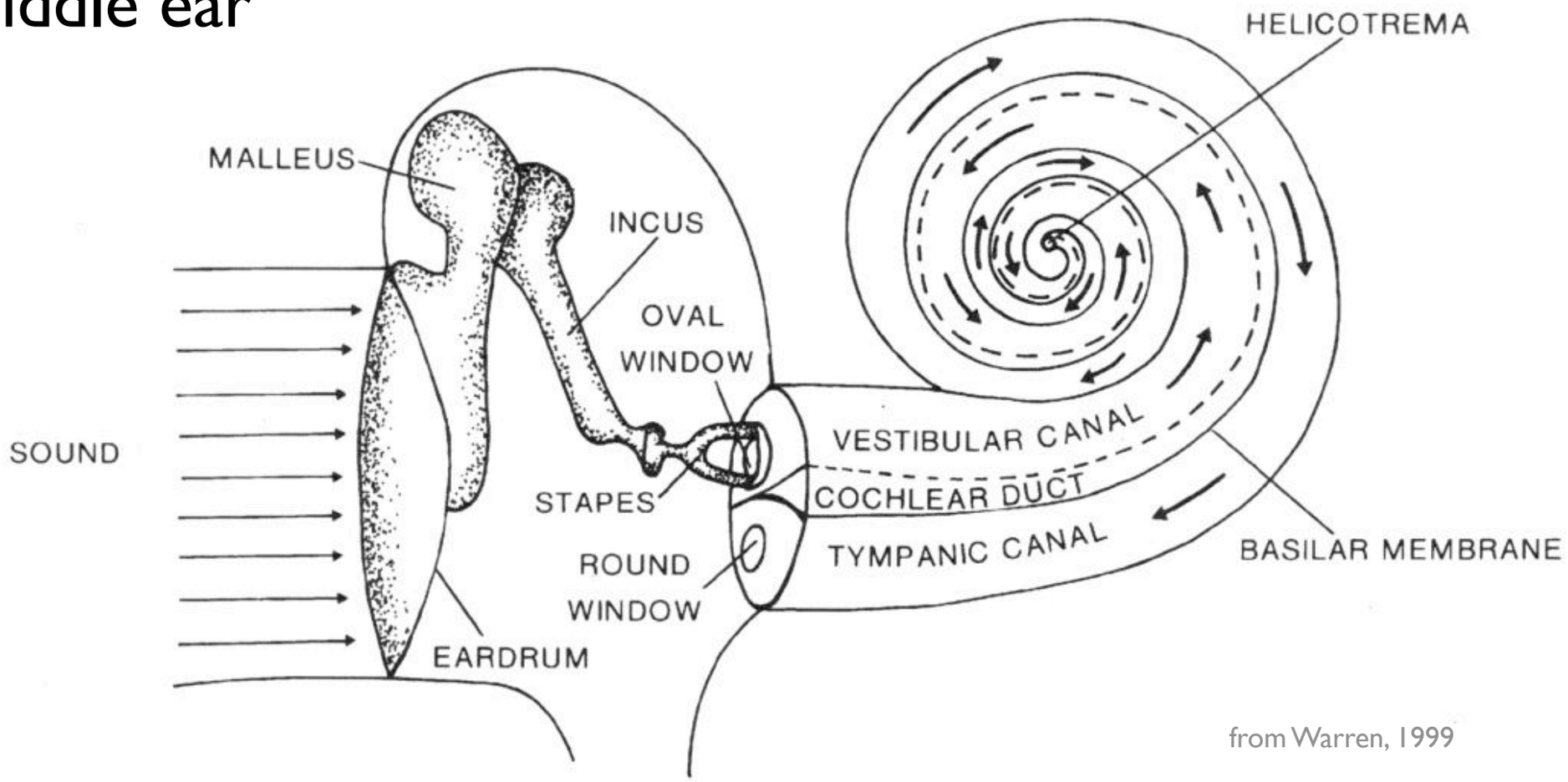
from Warren, 1999

The outer ear behaves like a series of resonance cavities



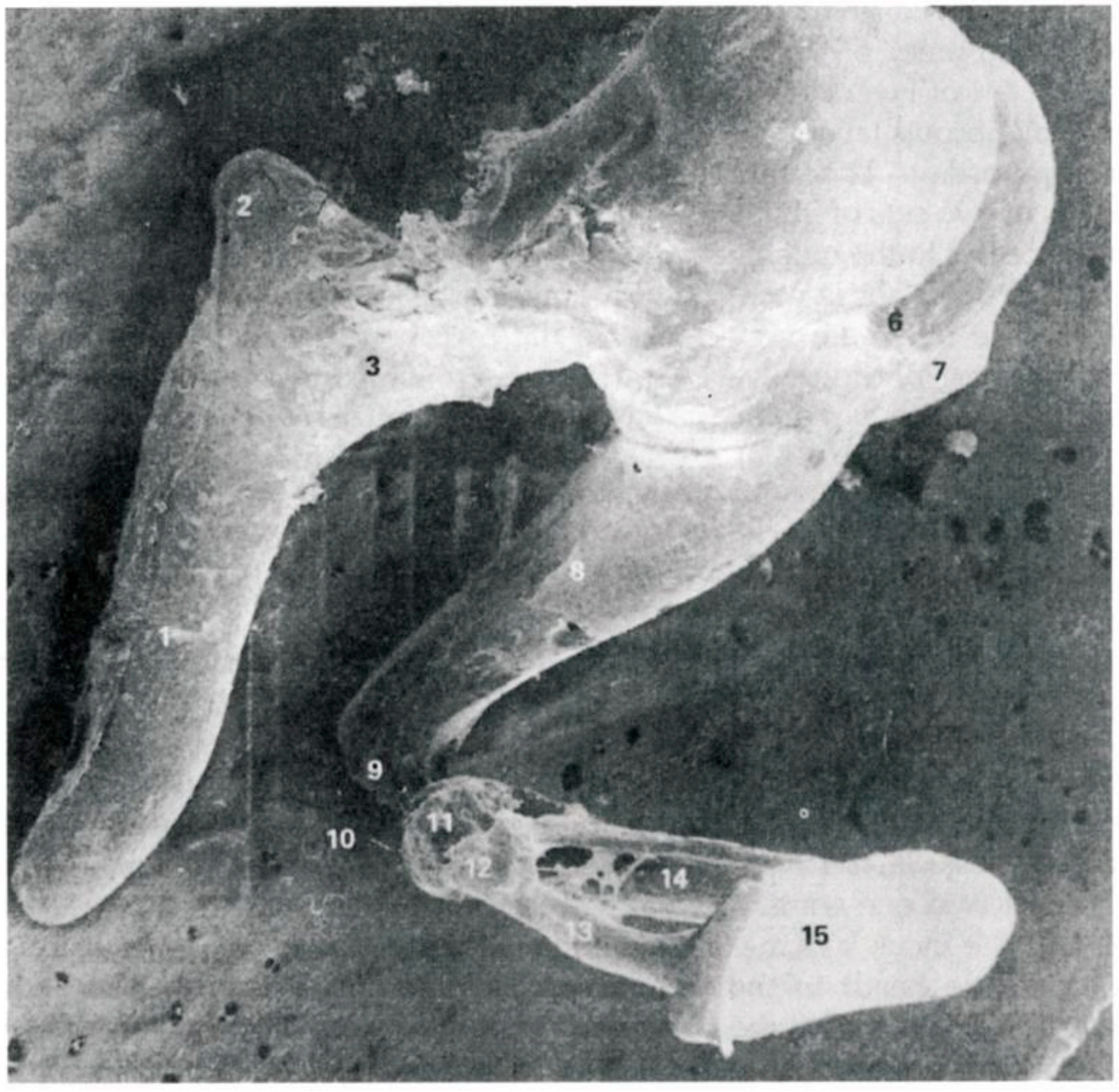
from Yost, 2000

The middle ear



from Warren, 1999

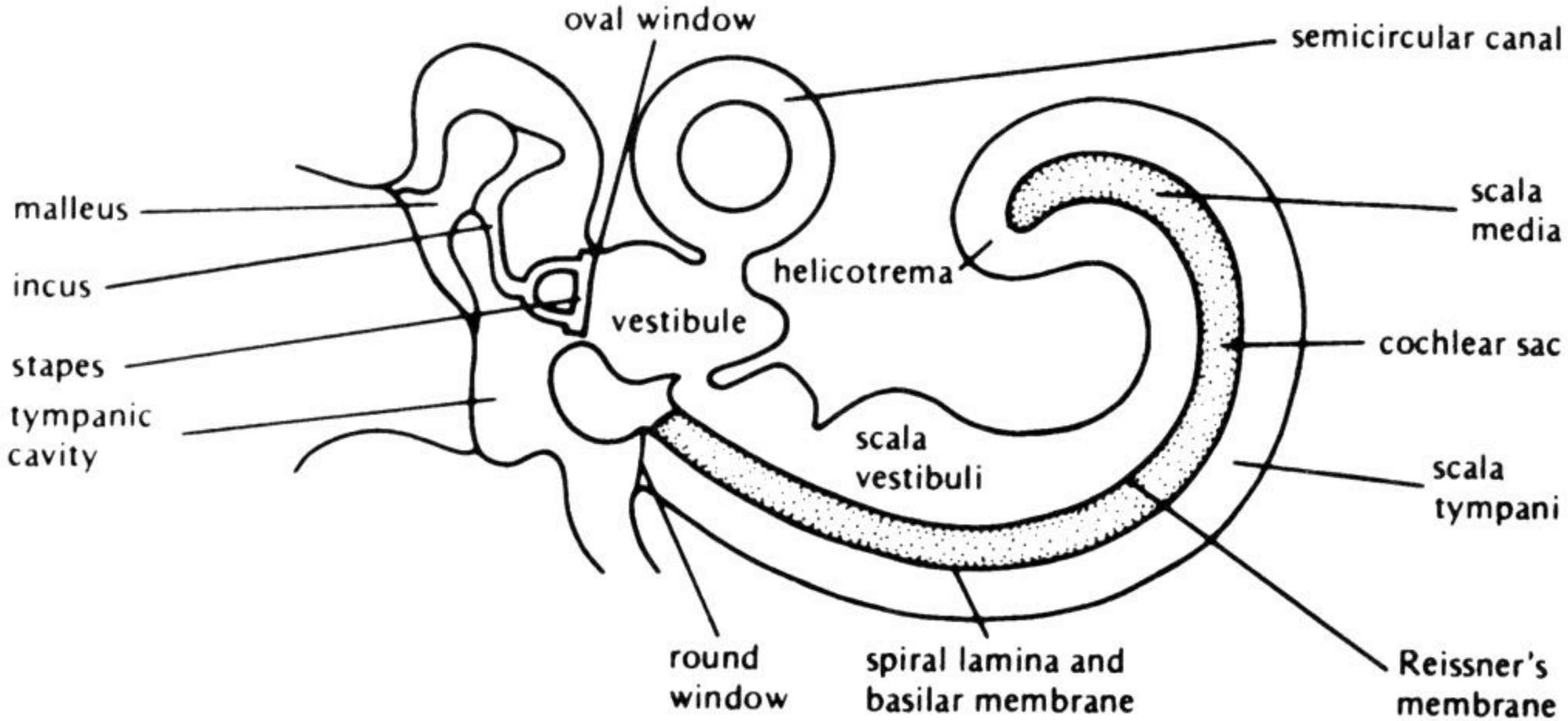
The bones of the middle ear (ossicles)



Middle ear function and properties:

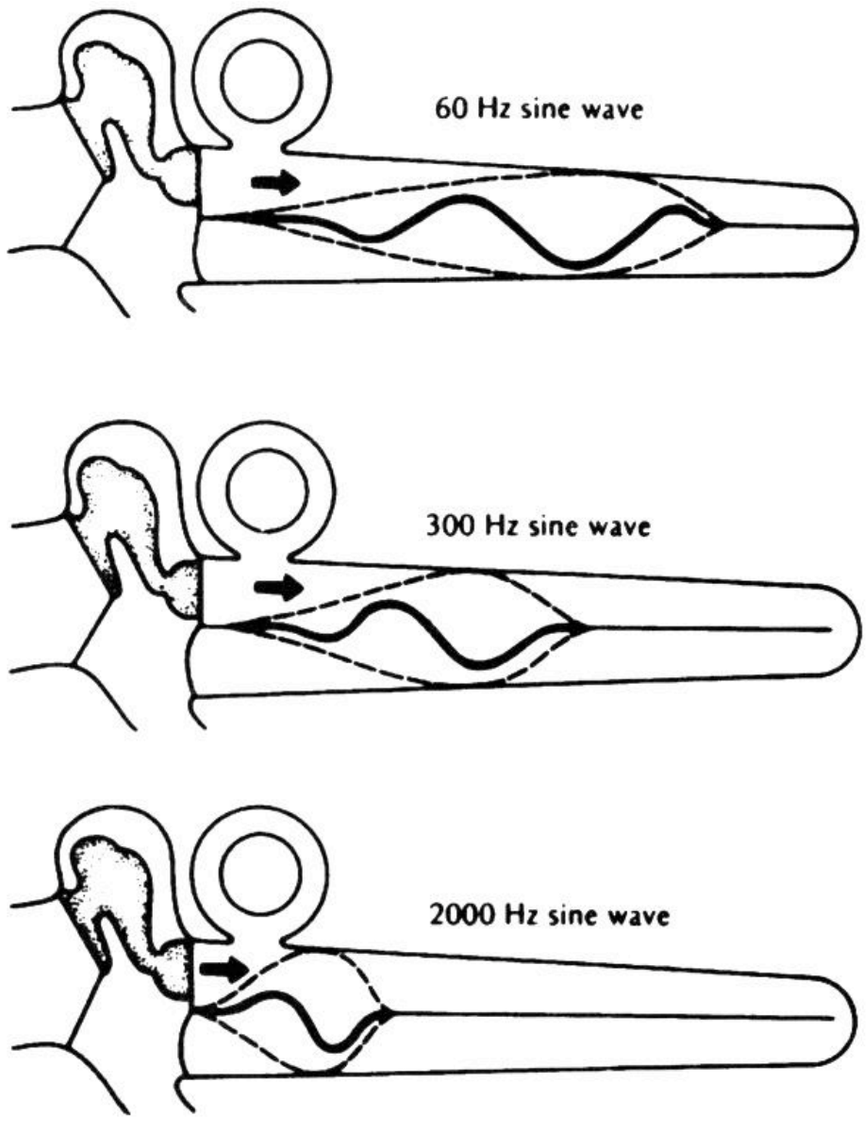
- ossicles are adapted to (apparently) to minimize non-linearities
- have a balanced center of gravity
- gain of middle ear is frequency dependent (maximum of 30 dB, 700-800 Hz)
- also has muscles that can attenuate loud sounds (by as much as 30 dB for low frequency sounds)
  - trigger is sound (> 80 dB) or chewing
  - can only protect against slow onset, low frequency sounds
  - takes 10 msecs (not fast enough for gunshots)

The inner ear



from Yost, 2000

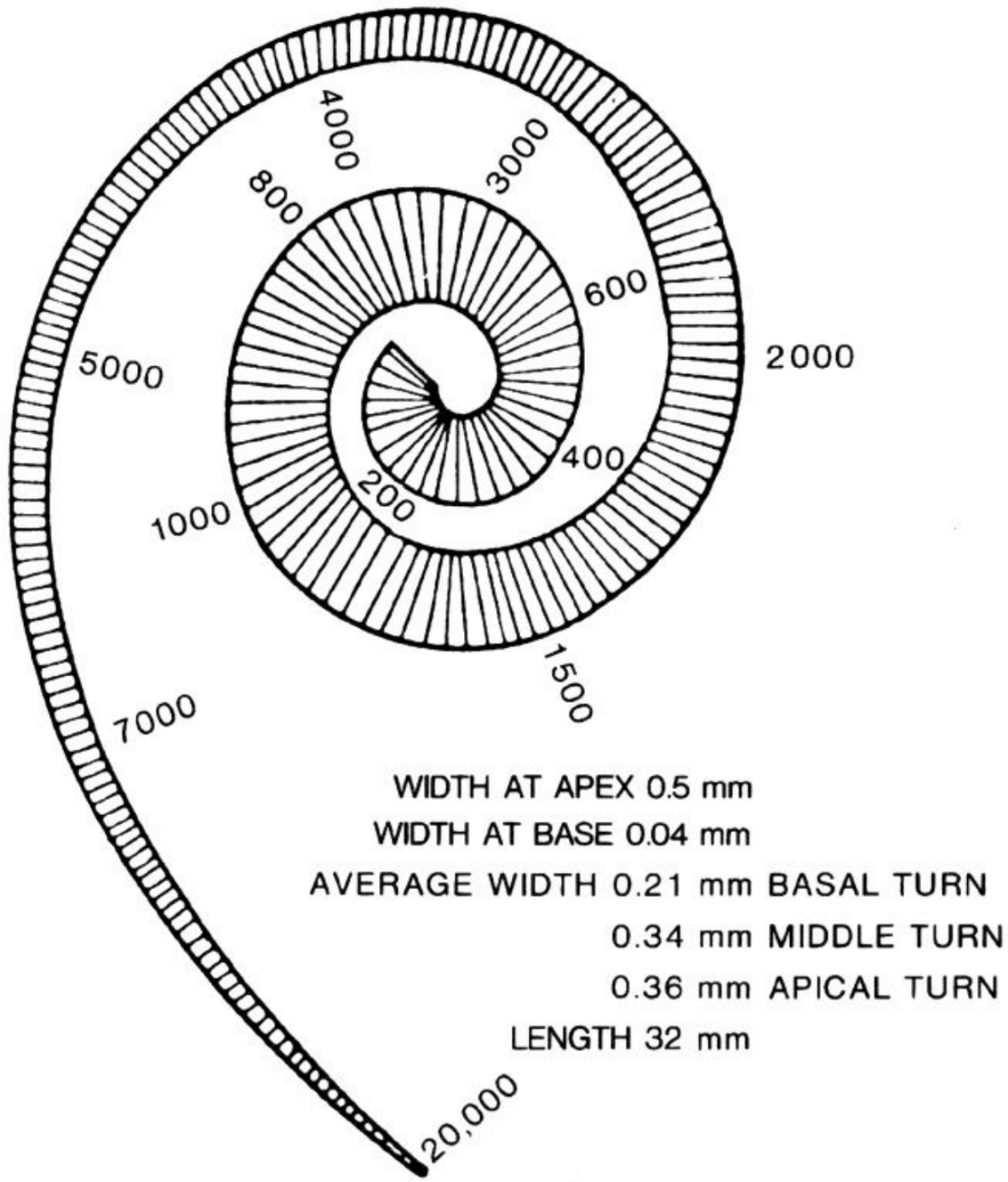
Traveling waves at different frequencies



from Yost, 2000

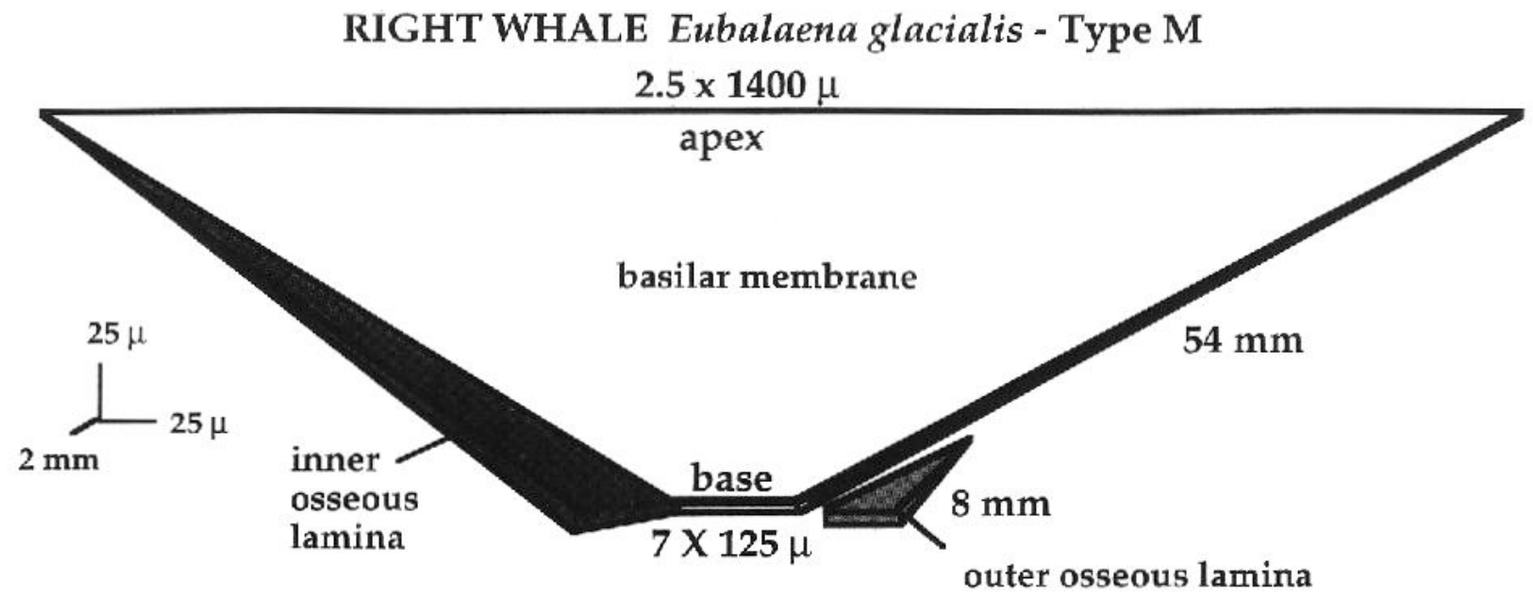
- sound waves are roughly converted into frequency by traveling waves
- maximal vibration occurs at different locations depending on frequency (low frequencies travel farther)
- basilar membrane is highly specialized:
  - thick and stiff at basal end
  - flexible at apical end
  - traveling waves slow down by a factor of 100, velocity is an exponential function of distance.

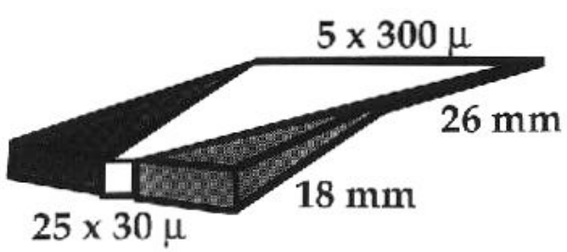
Shape of basilar membrane and its frequency map



Mechanical properties of BM determine hearing range

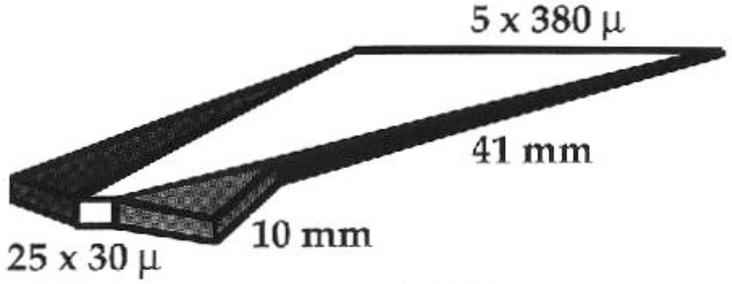
10 Hz - 22 kHz
(estimated)





HARBOR PORPOISE
Phocoena phocoena - Type I

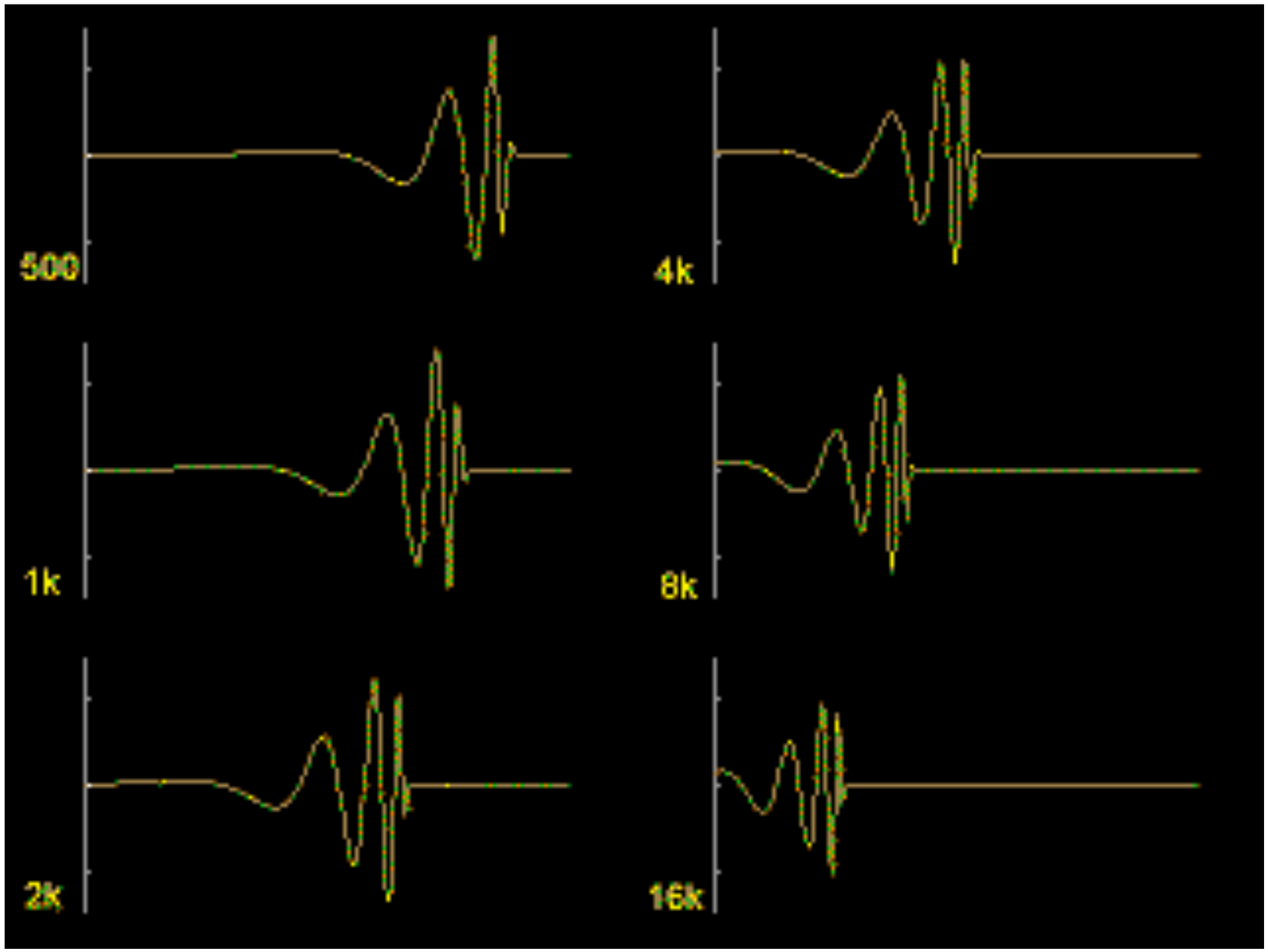
1 kHz - 150 kHz
dyn. range: 140 dB



BOTTLENOSED DOLPHIN
Tursiops truncatus - Type II

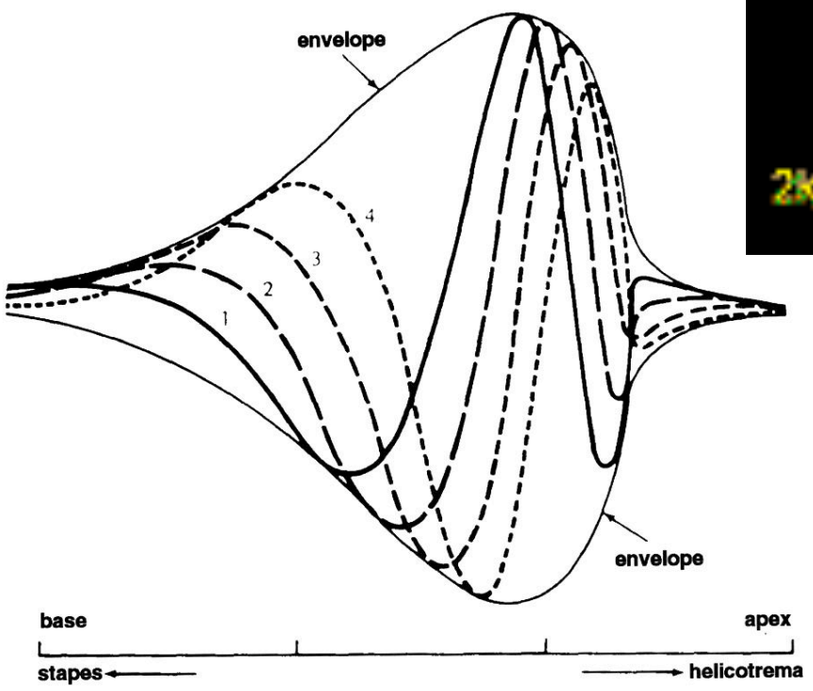
75 Hz - 150 kHz
dyn. range: 195 dB

Traveling Waves

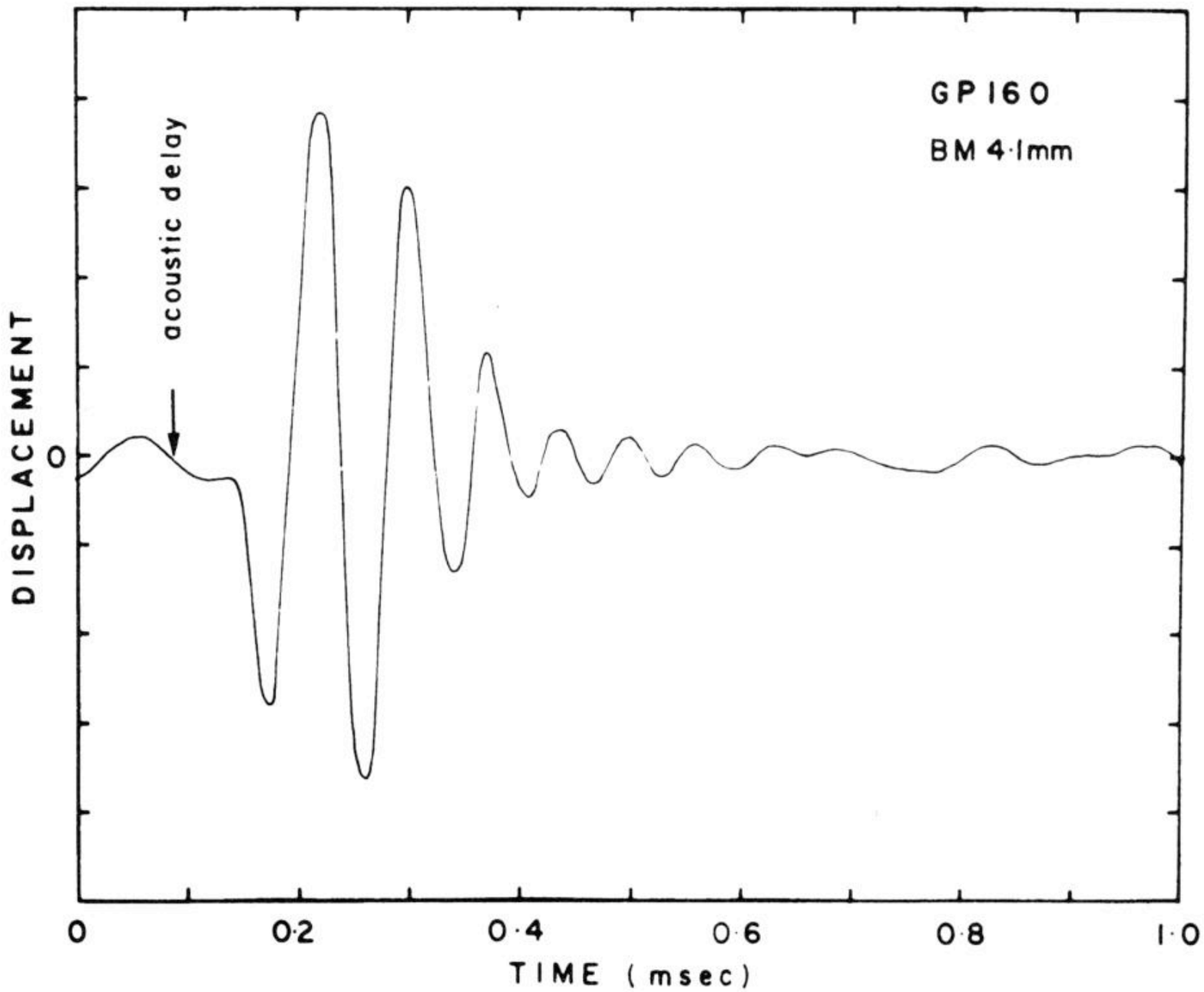


animation from Geisler

from Yost, 2000

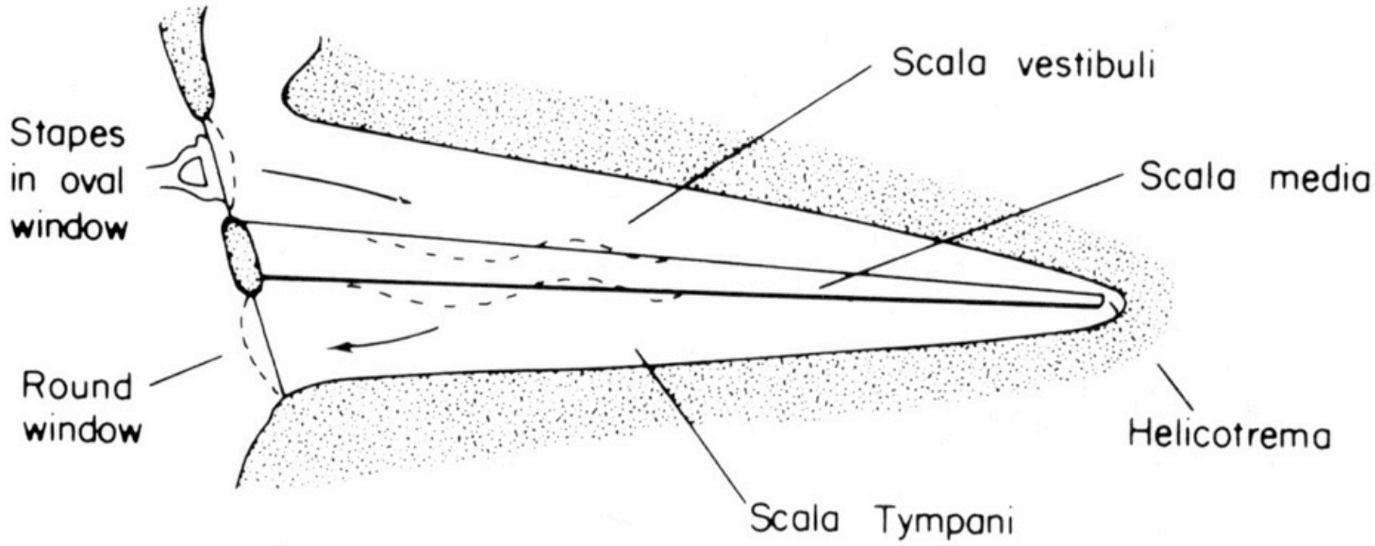


Basilar Membrane Impulse Response



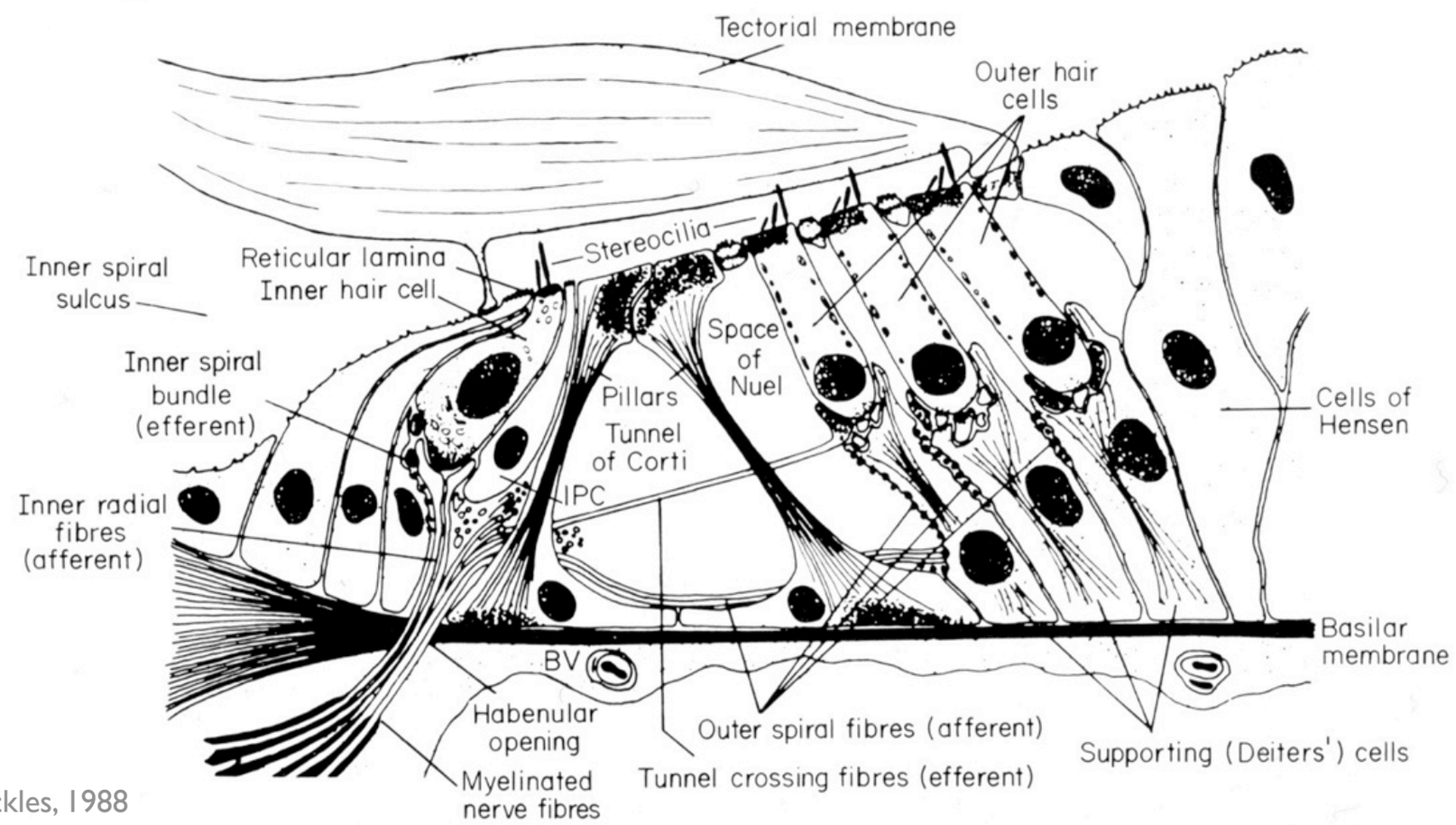
from Moore, 1997

The structure of the cochlea



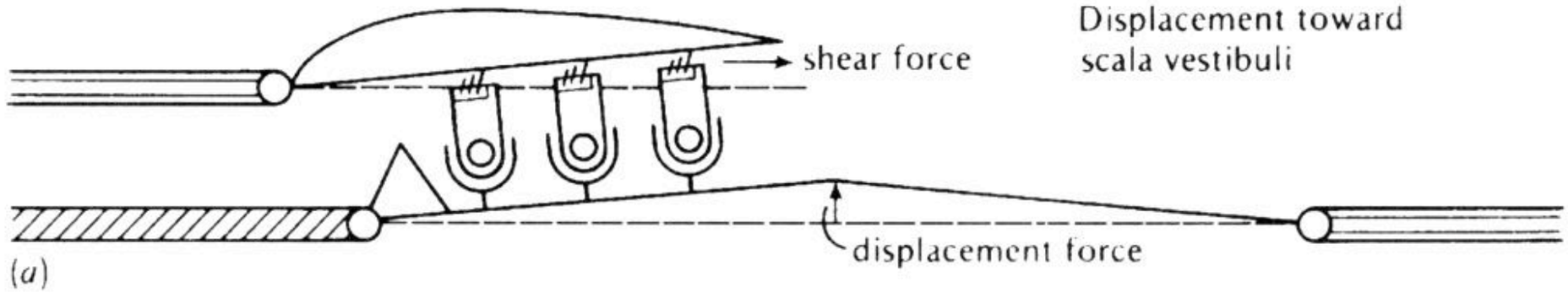
D

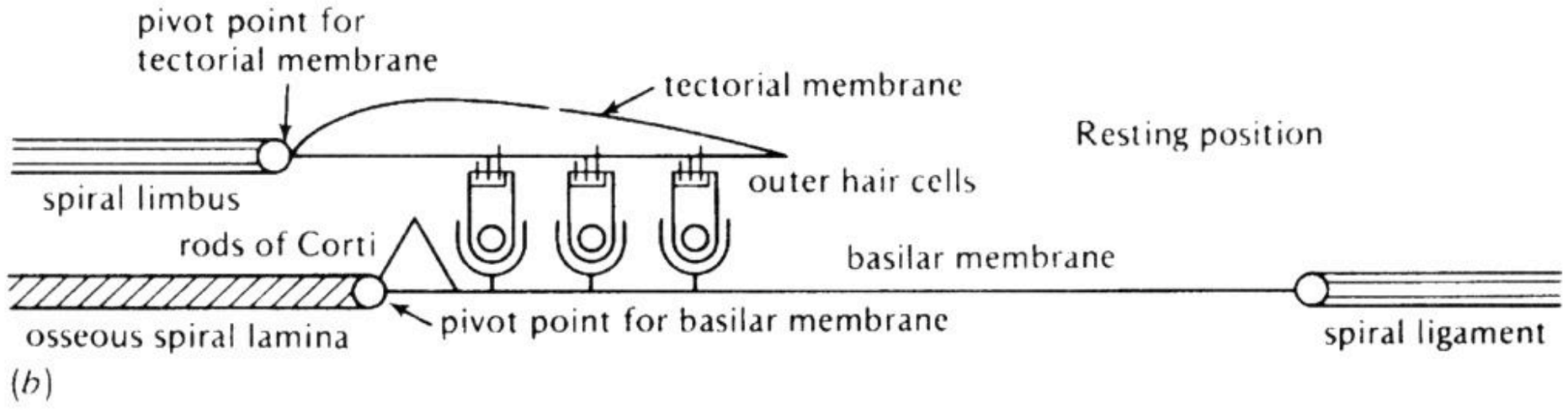
cross-section

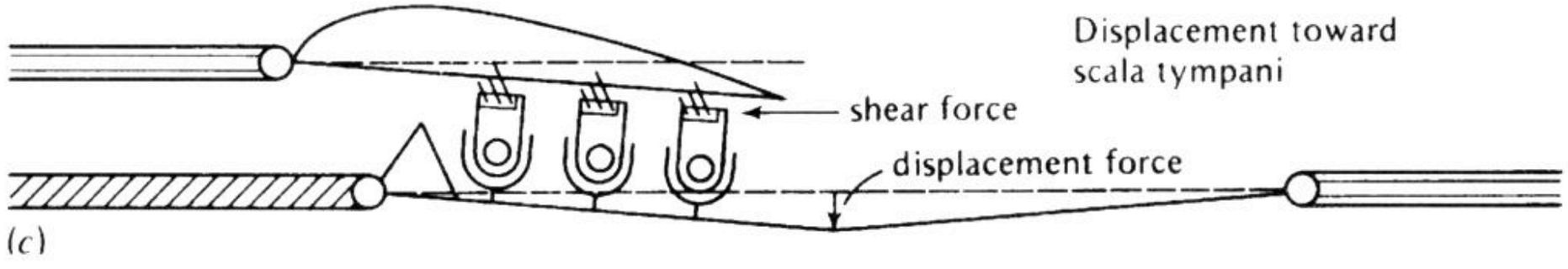


from Pickles, 1988

Sensing basilar membrane motion with hair cells

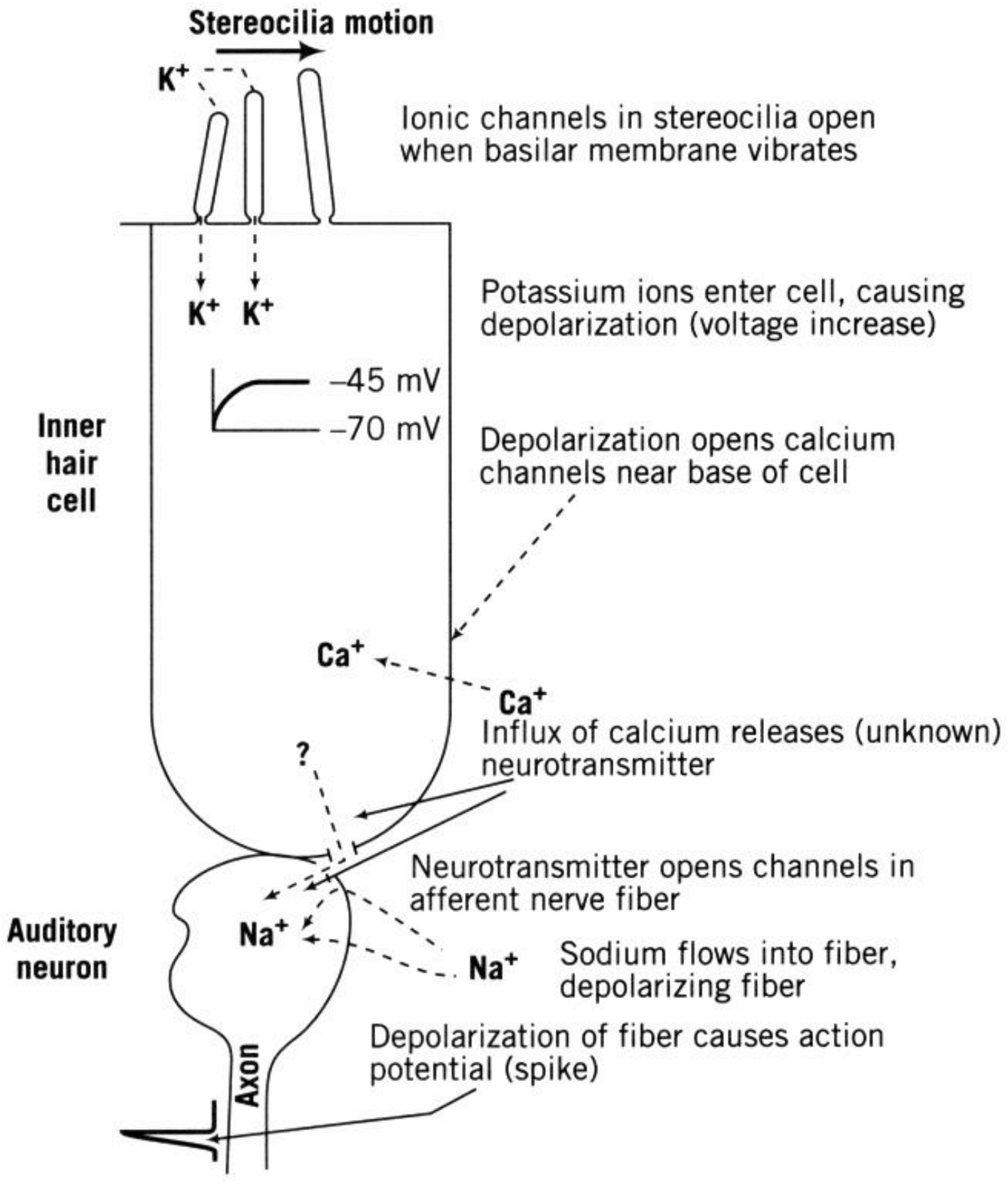






from Yost, 2000

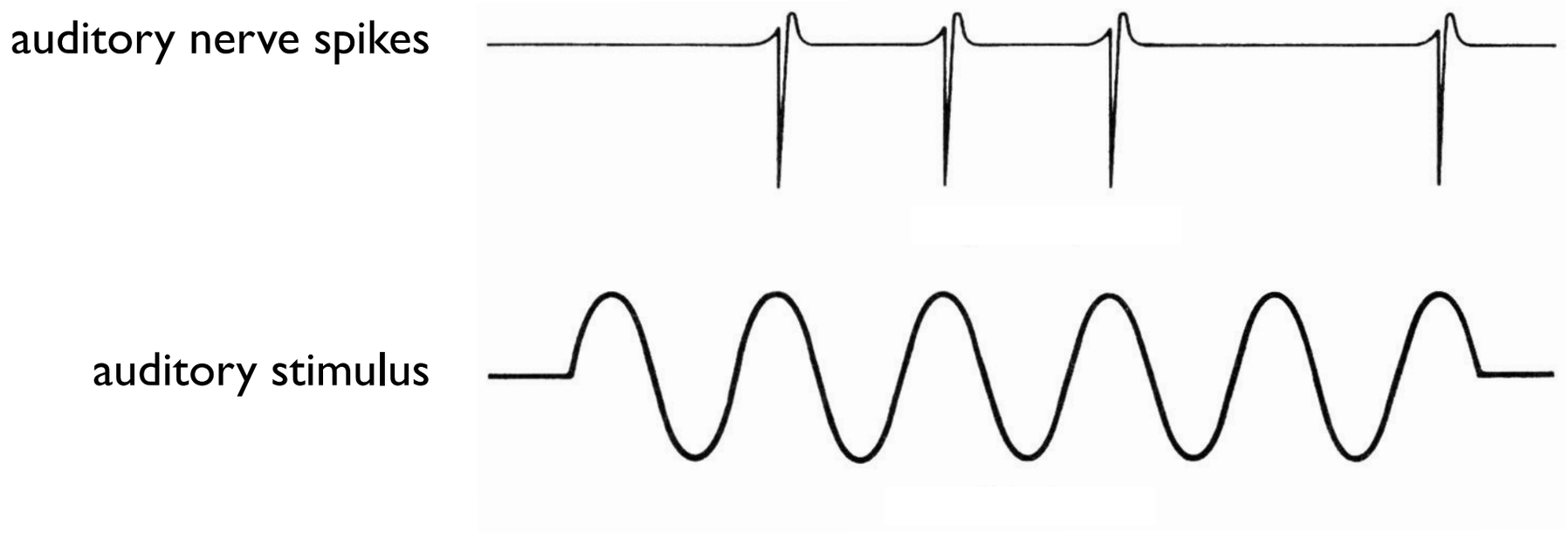
Hair-cell transduction



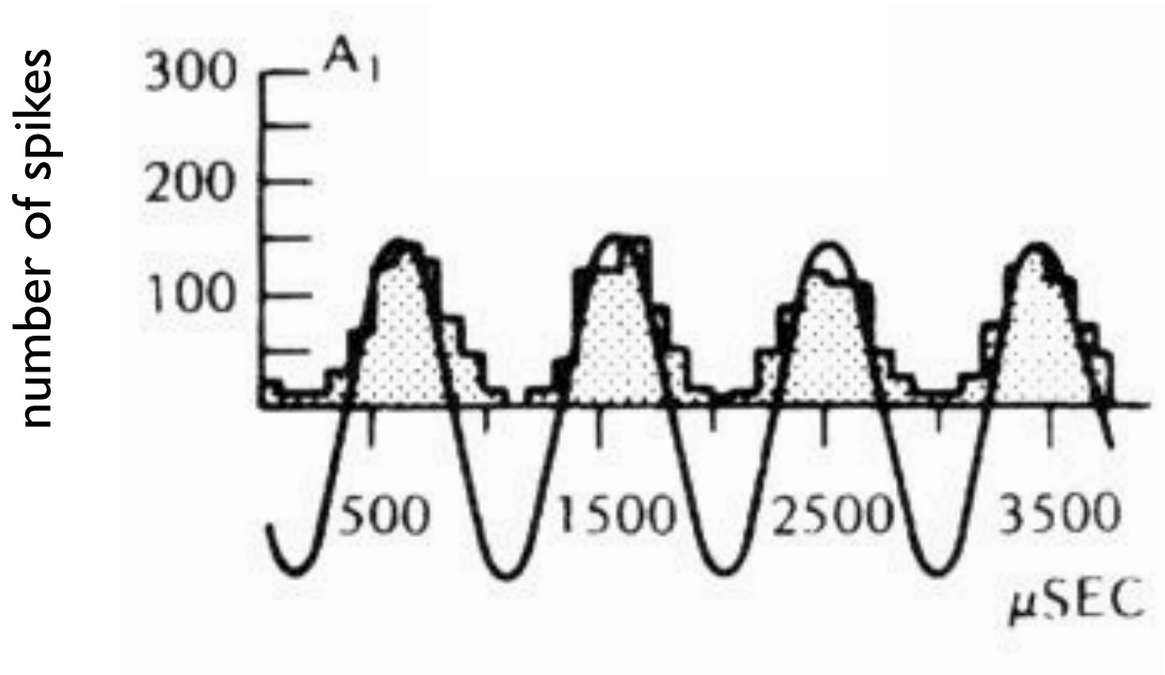
- basilar membrane motion results in stereocilia vibration
- this results in current flow as an influx of K+ ions
- If influx cross threshold, this results in an action potential at the auditory nerve

from Gold and Morgan, 2000

Phase locking of auditory nerve spikes

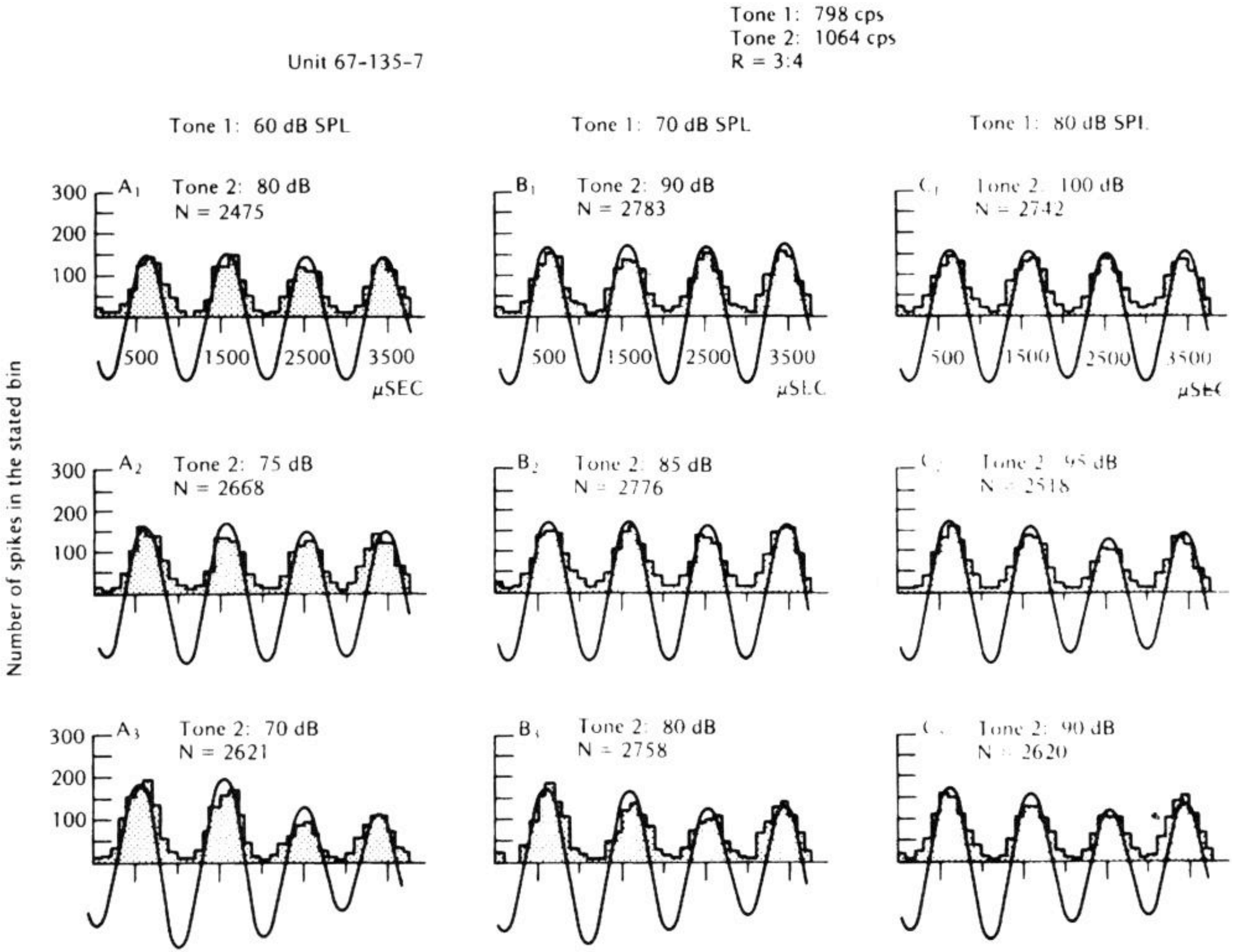


Histograms of auditory nerve spikes to repeated presentations of stimulus

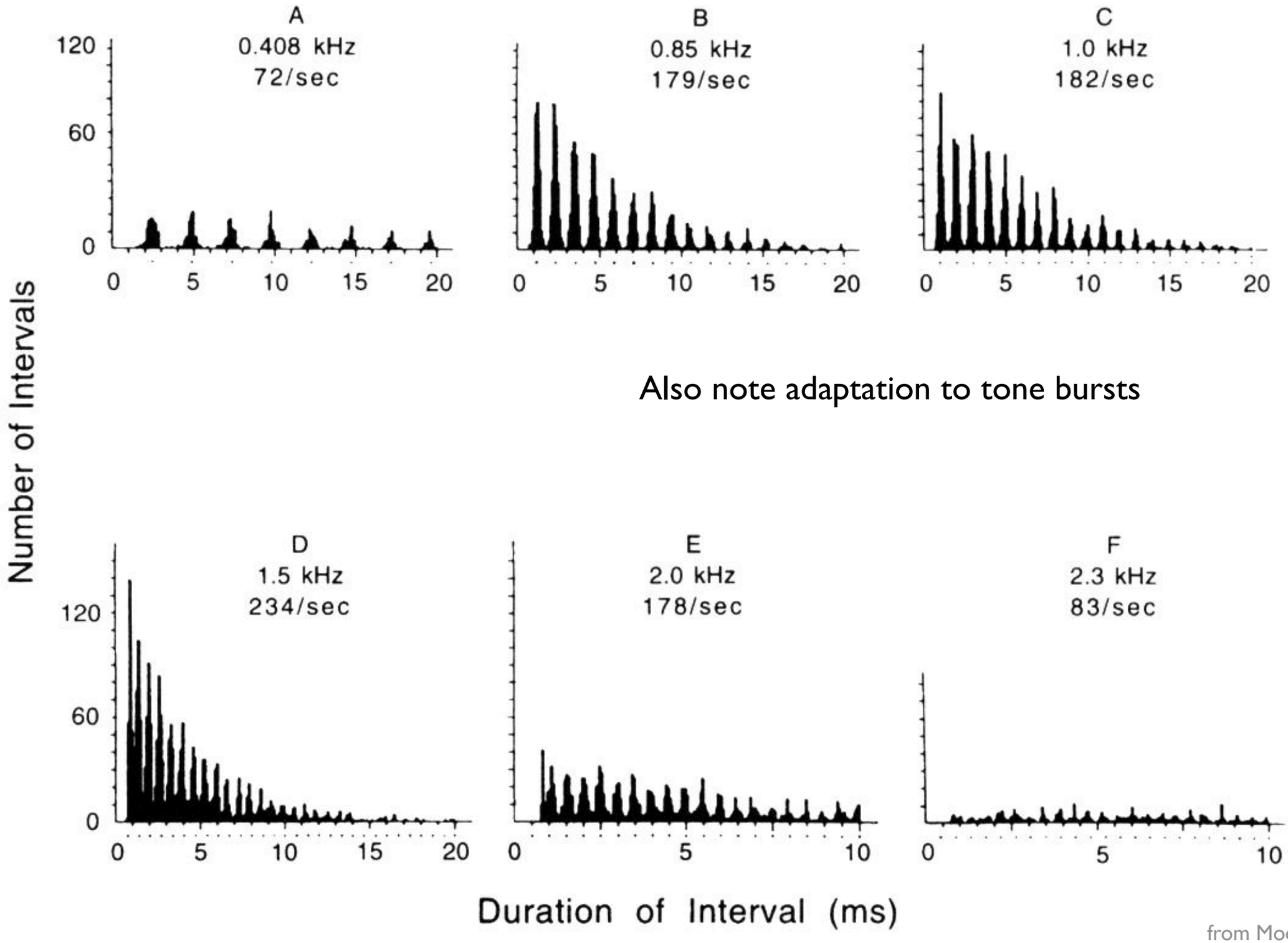


from Yost, 2000

Spiking probability follows stimulus amplitude



Phase locking breaks down for higher freqencies

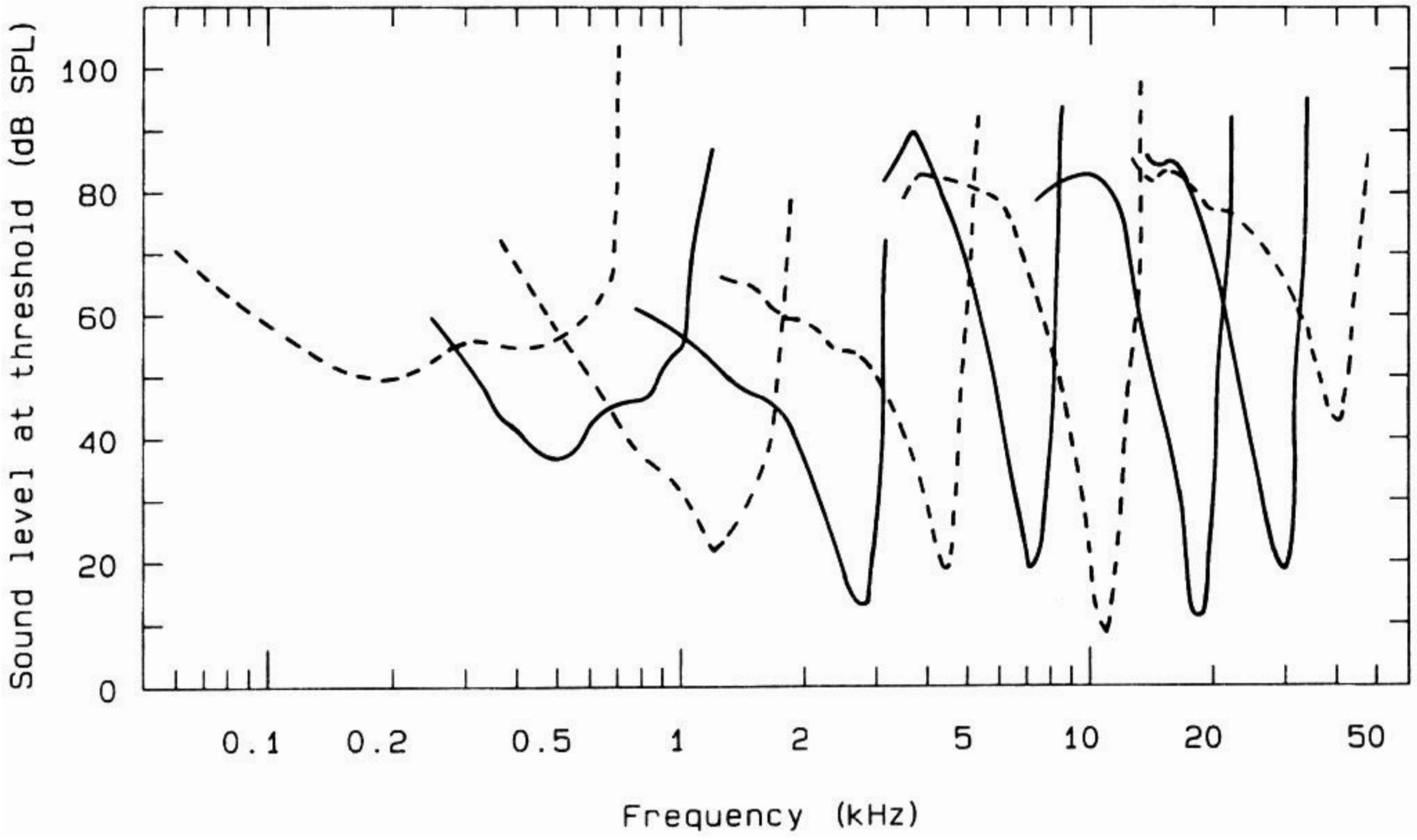


Also note adaptation to tone bursts

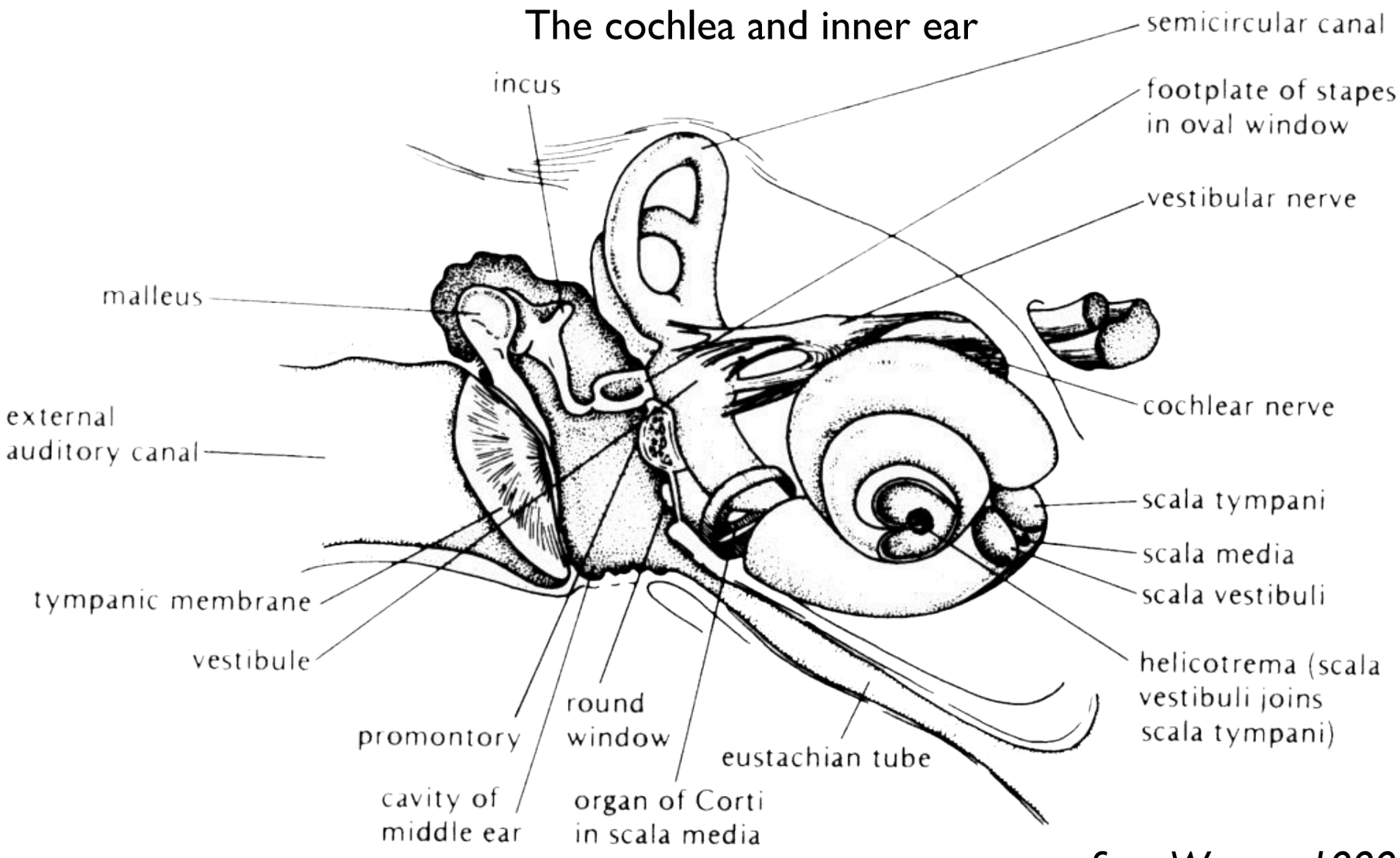
from Moore, 1997

Auditory nerve fiber responses: frequency tuning

threshold sensitivity curves for different auditory nerves



The real system is much more complex than linear model

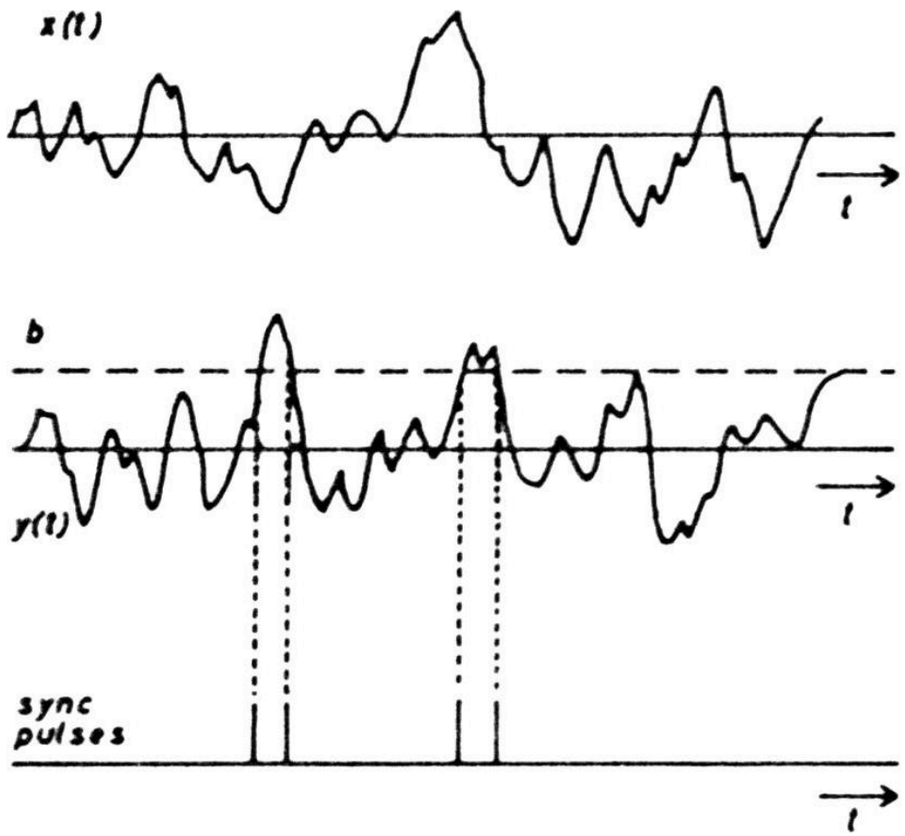


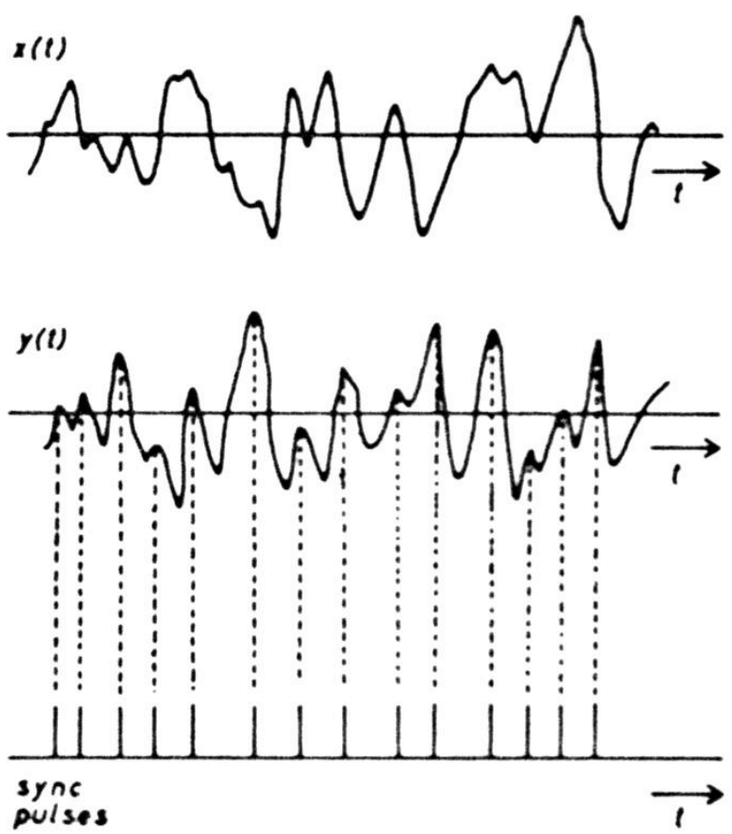
from Warren, 1999

How do we describe what it does?

A linear characterization of auditory nerves

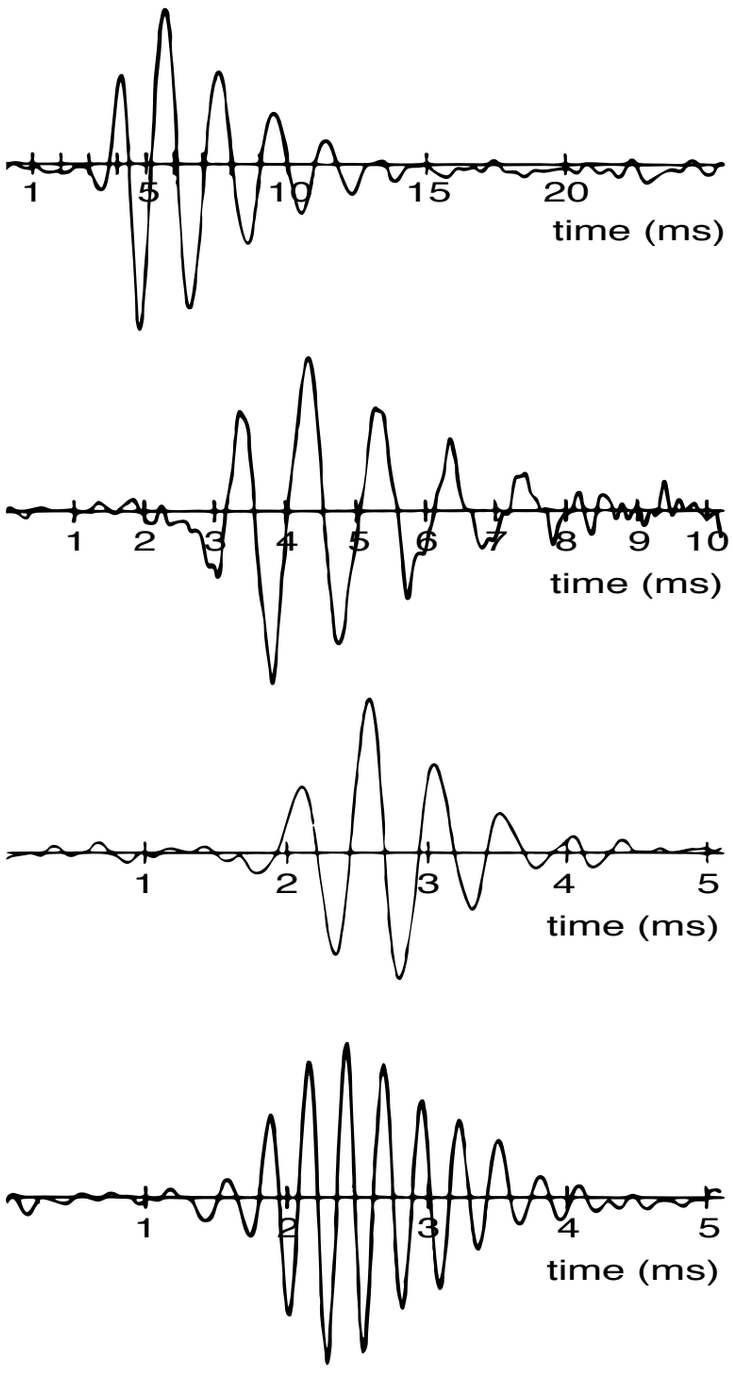
Spike-triggered averaging estimates auditory nerve impulse response functions: "revcor" filters

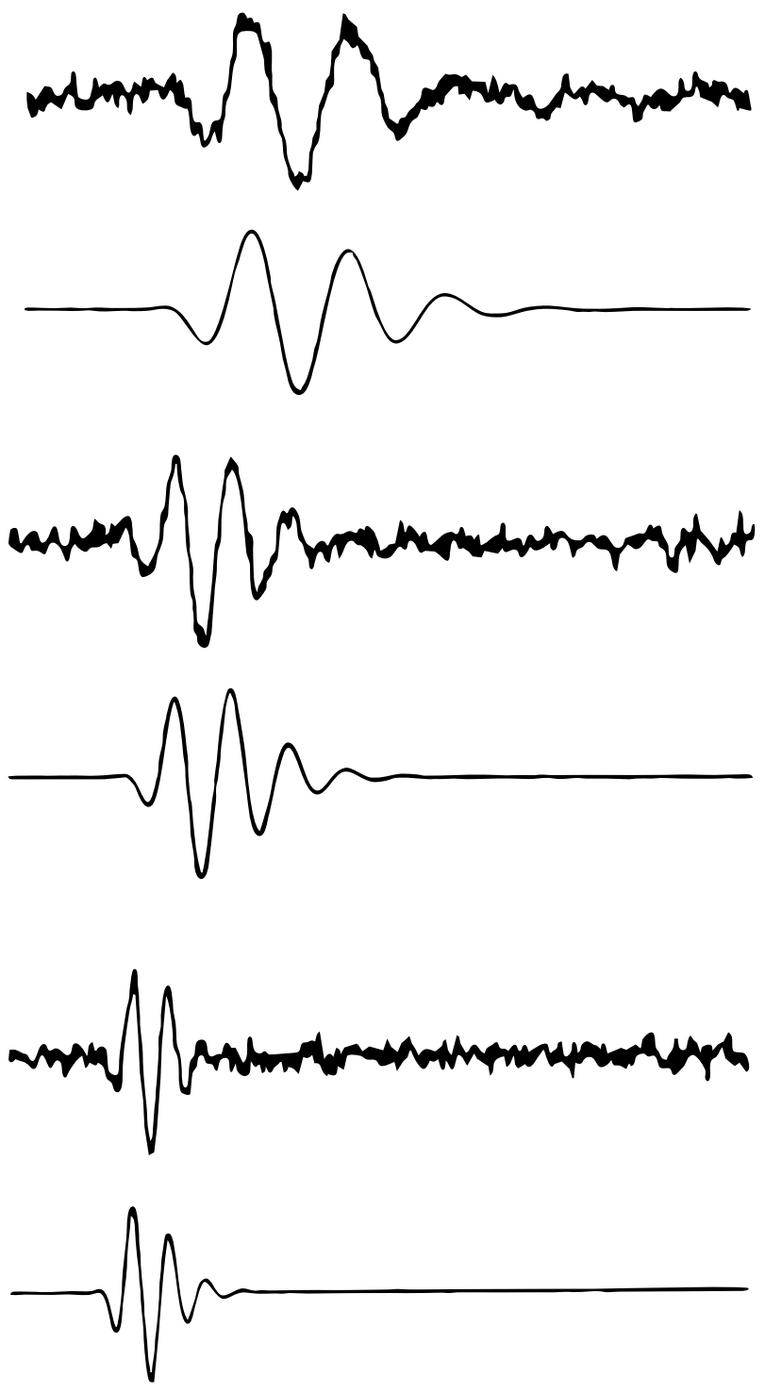




from deBoer and Kuyper, 1968

Auditory nerve revcor filters



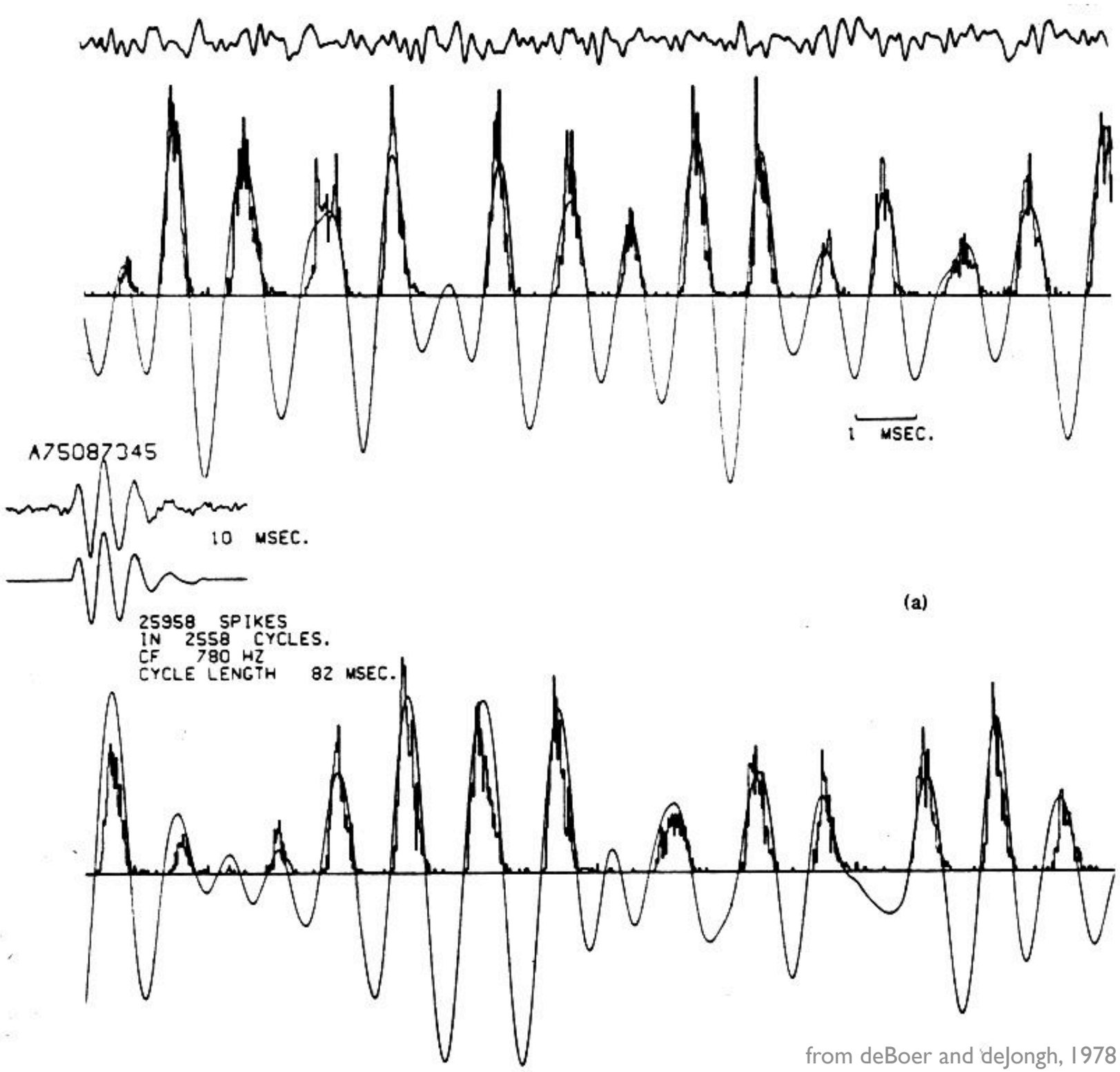


deBoer and deJongh, 1978

Carney and Yin, 1988

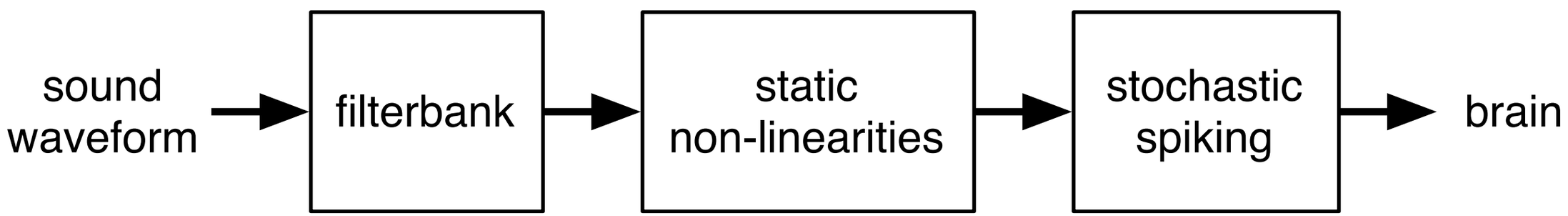
compare to BM impulse response

Predictions of the linear model



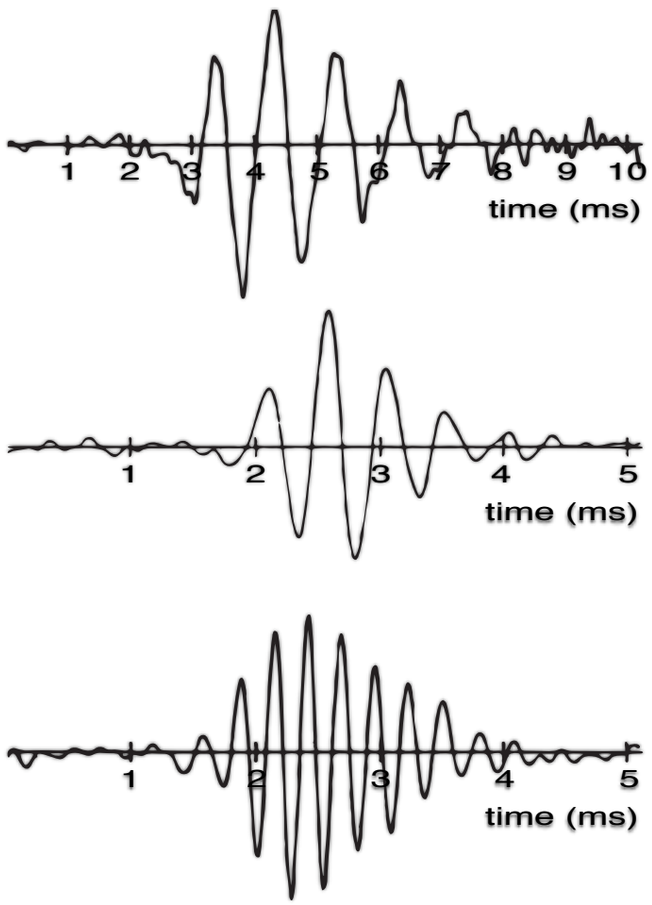
from deBoer and deJongh, 1978

Information theory arose from the problem of speech coding

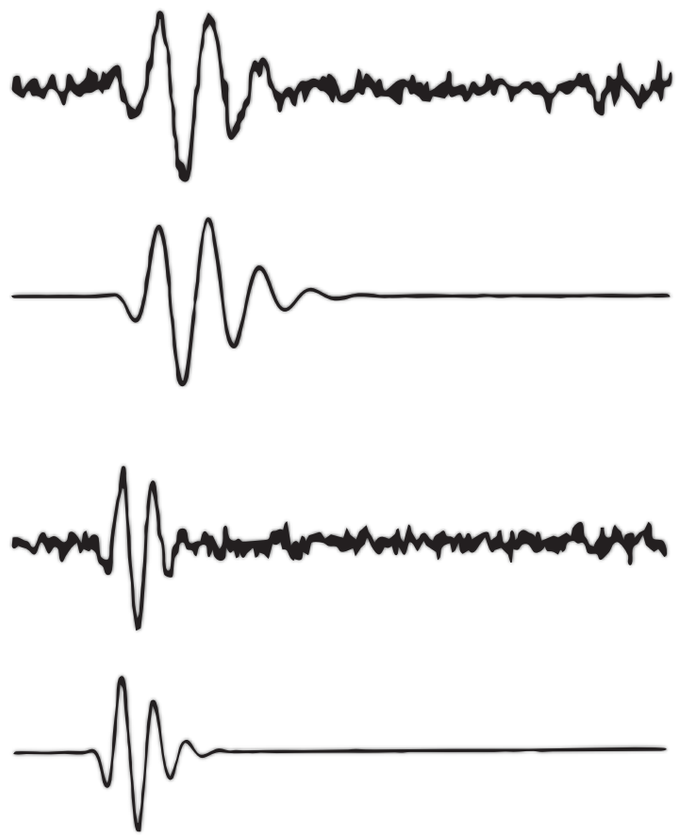


A simple model of auditory coding

Auditory nerve filters can be estimated using
reverse correlation (spike-triggered averaging)



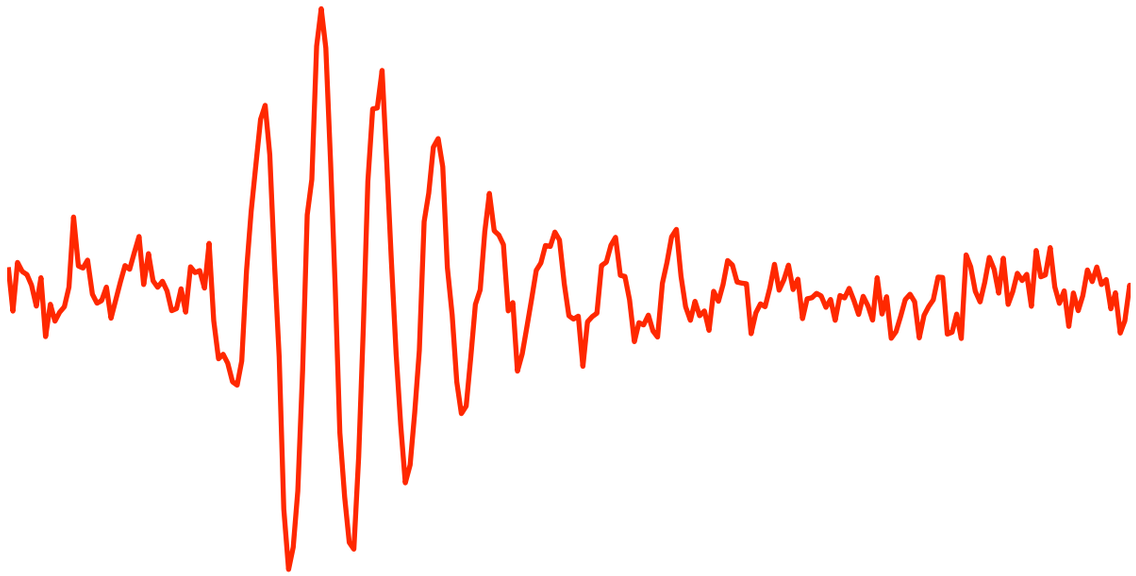
deBoer and deJongh, 1978



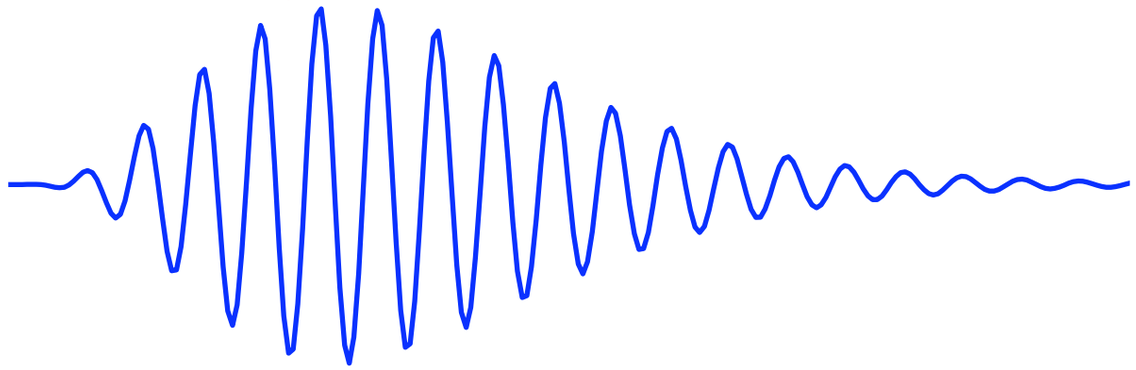
Carney and Yin, 1988

Models are data driven

data: revcor filter



model: gammatone

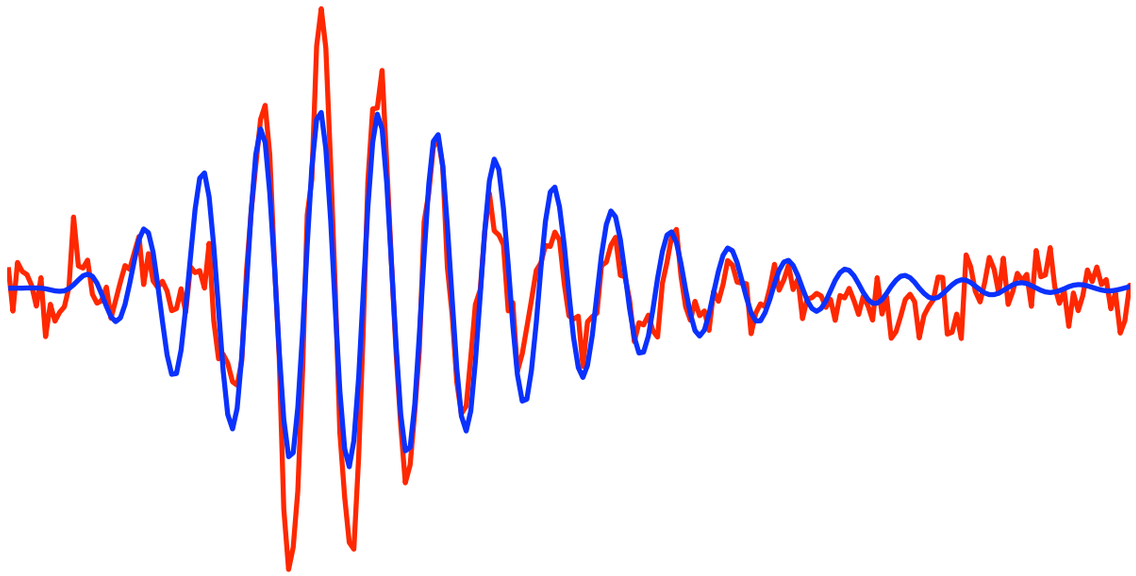


$$g(t) = at^{n-1}e^{-bt}\cos(2\pi ft + \phi)$$

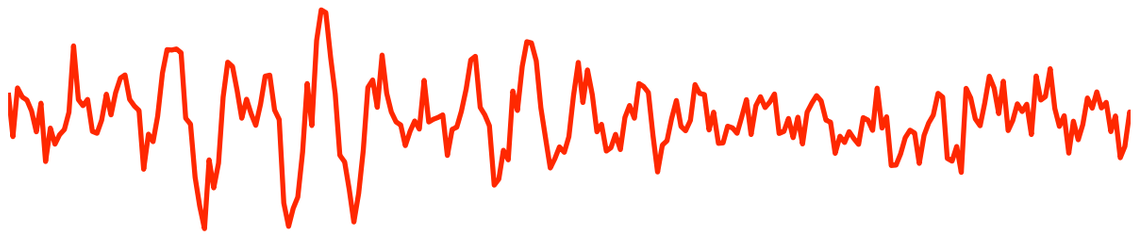
"gammatone" function

Models are data driven

data: revcor filter with fitted model



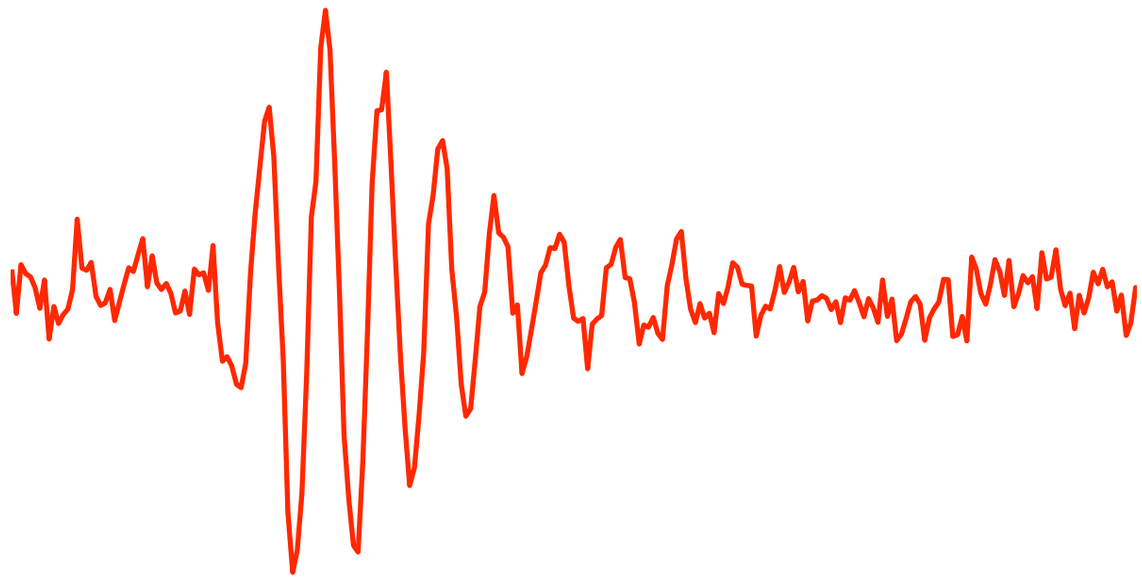
residual error



$$g(t) = at^{n-1}e^{-bt}\cos(2\pi ft + \phi)$$

"gammatone" function

A theoretical approach



Theoretical questions:

- Why gammatones?
- Why spikes?
- How is sound coded by the spike population?

How do we develop a theory?

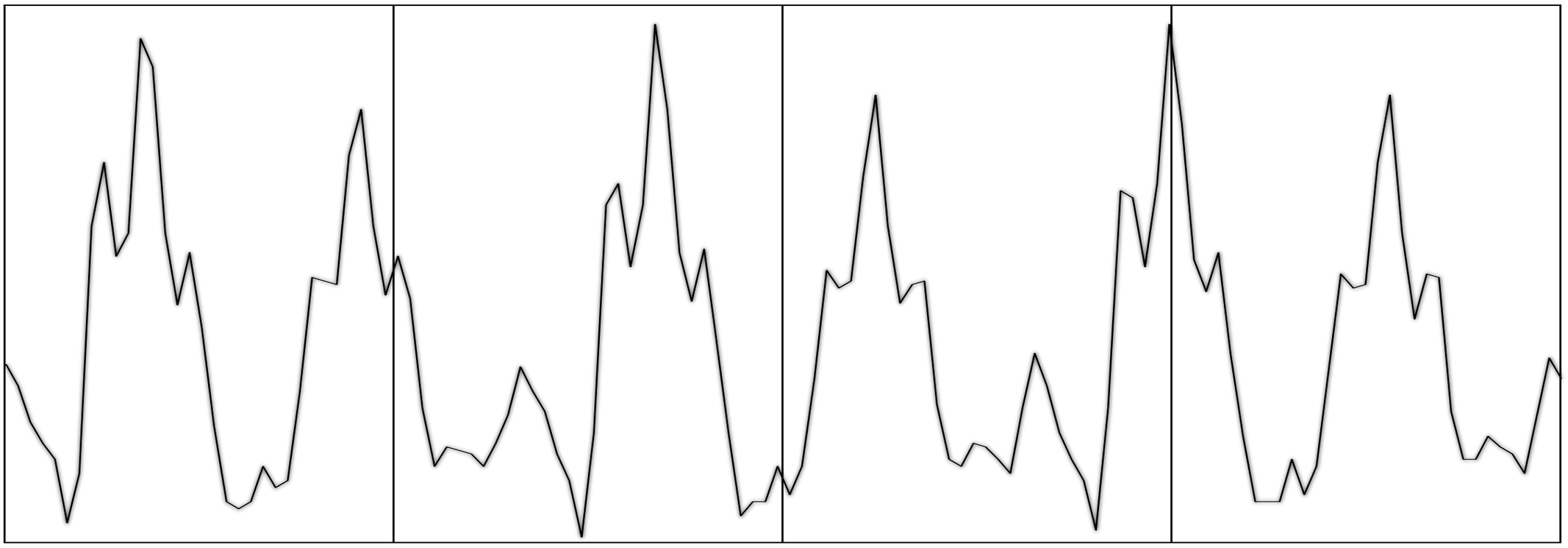
Efficient coding theory

- Barlow, 1961; Attneave, 1954
  - main goal of sensory coding is to code signals *efficiently*
  - sensory codes are adapted to the sensory environment
  - each code feature should have minimal *redundancy*
  - each "feature" should describe *independent* information

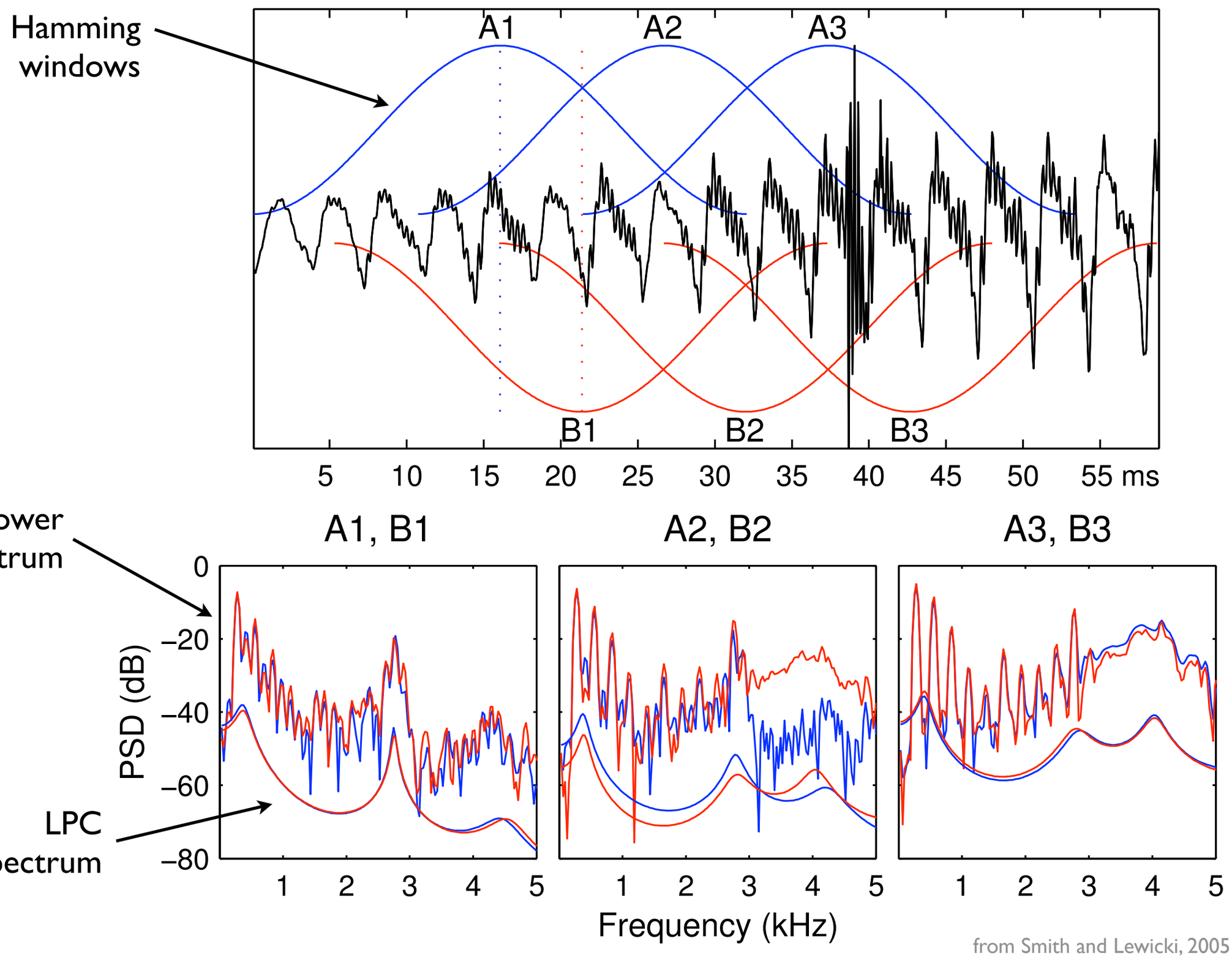
- caveats:
  - applies to *behaviorally relevant* information
  - not all redundancy is bad, e.g. when compensating for noise

Limitations of this theoretical model

- filter bank model is linear
- code is optimal only within a block, not for whole signal
- offers no explanation of phase locking and spikes
- representation depends on the relative alignment of the signal and block

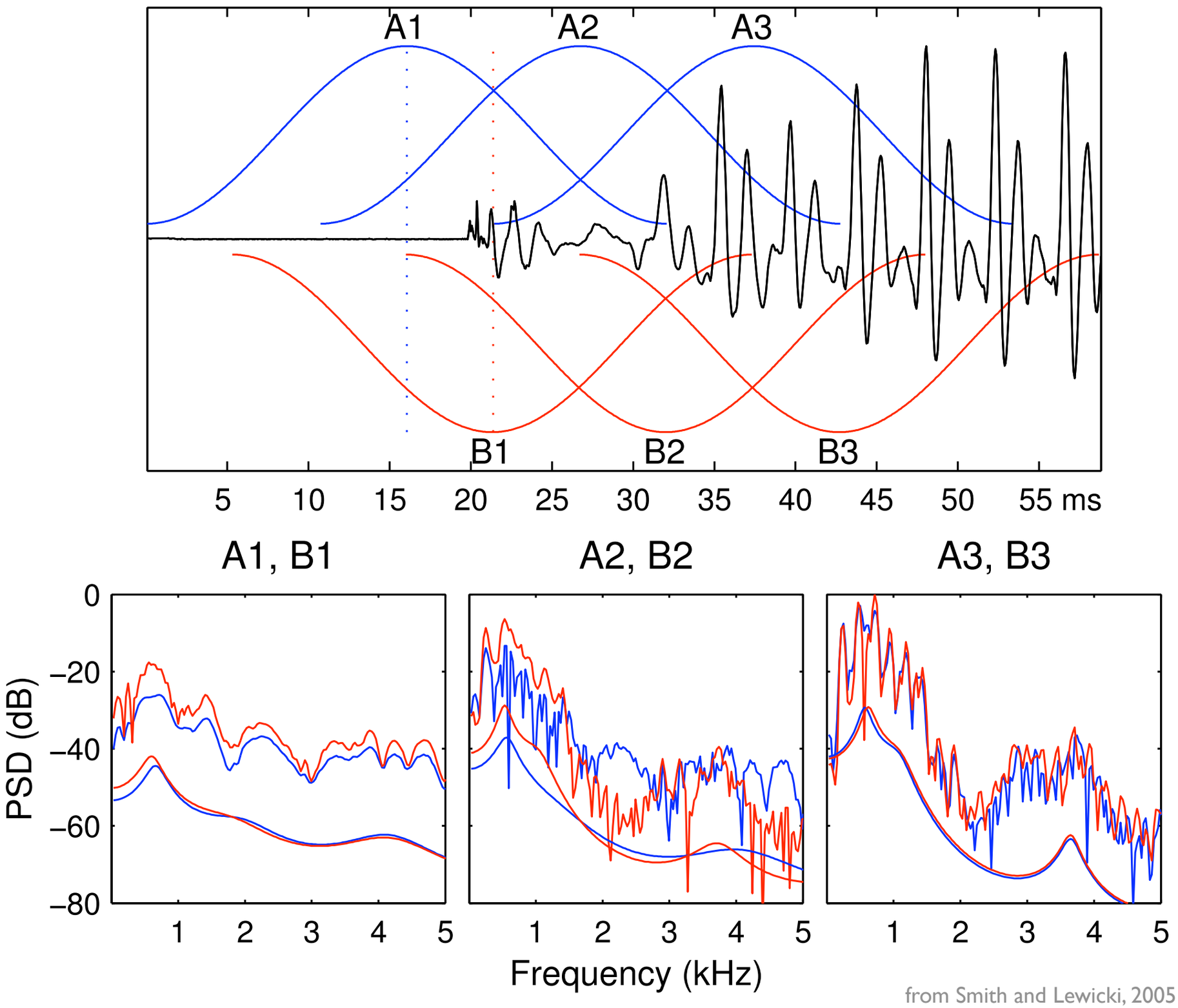


Block coding does not yield time-relative codes



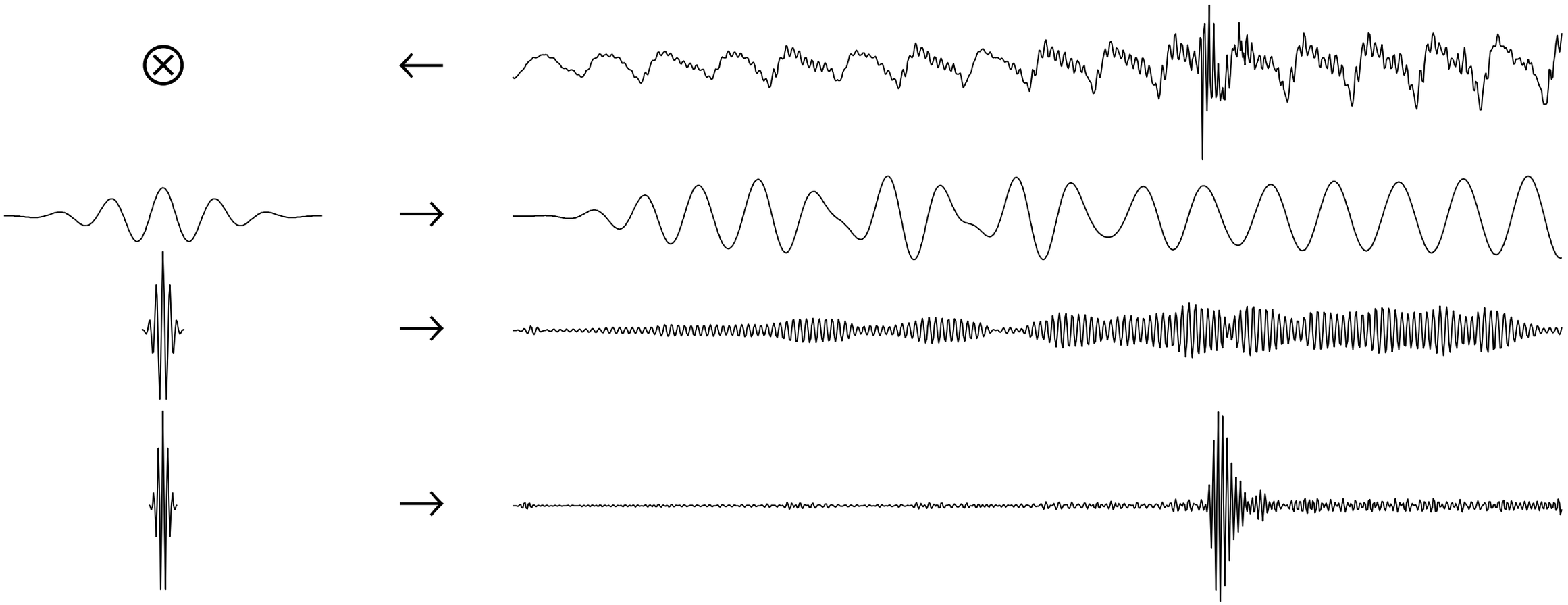
from Smith and Lewicki, 2005

Block coding does not yield time-relative codes



from Smith and Lewicki, 2005

A continuous filterbank does not form an efficient code

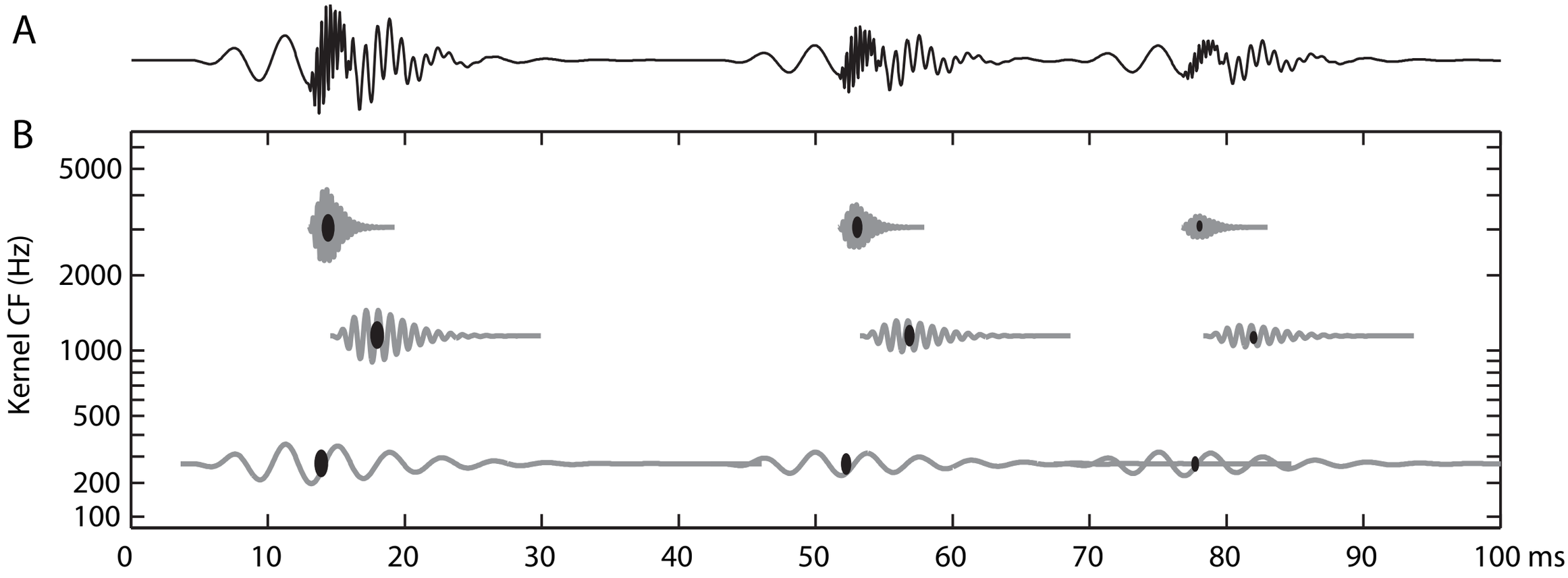


from Smith and Lewicki, 2005

Goal:

find a representation that is both *time-relative* and *efficient*

Efficient signal representation using time-shiftable kernels (spikes)

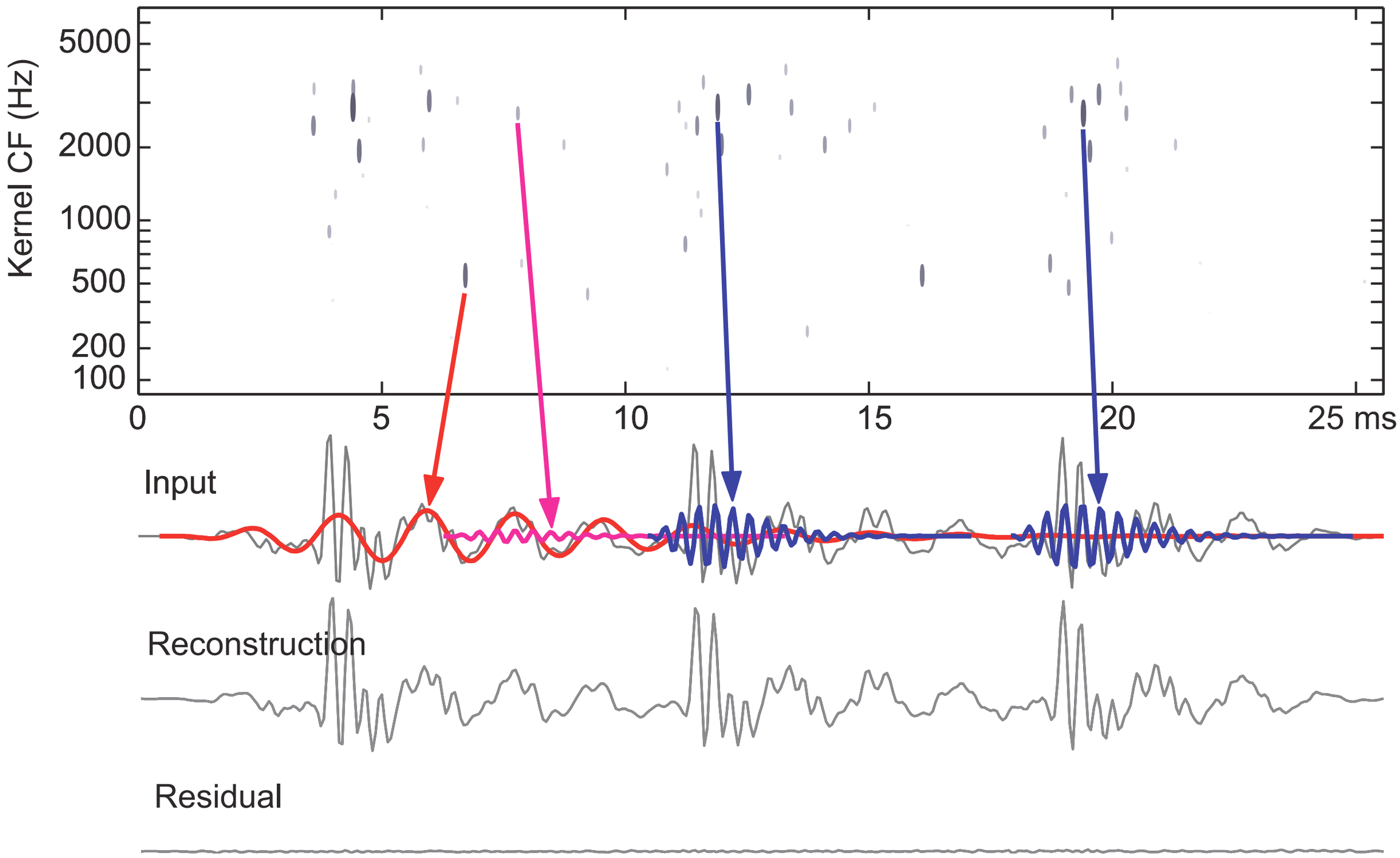


from Smith and Lewicki, 2005

$$x(t) = \sum_{m=1}^{M} \sum_{i=1}^{n_m} s_{m,i}\, \phi_m(t - \tau_{m,i}) + \epsilon(t)$$

- Two important theoretical abstractions for "spikes":
  - not binary, each has an analog value
  - not probabilistic
- Each spike encodes the precise *time* and *magnitude* of a particular kernel
- Population forms a non-redundant signal representation
- Can convert to a population of stochastic, binary spikes

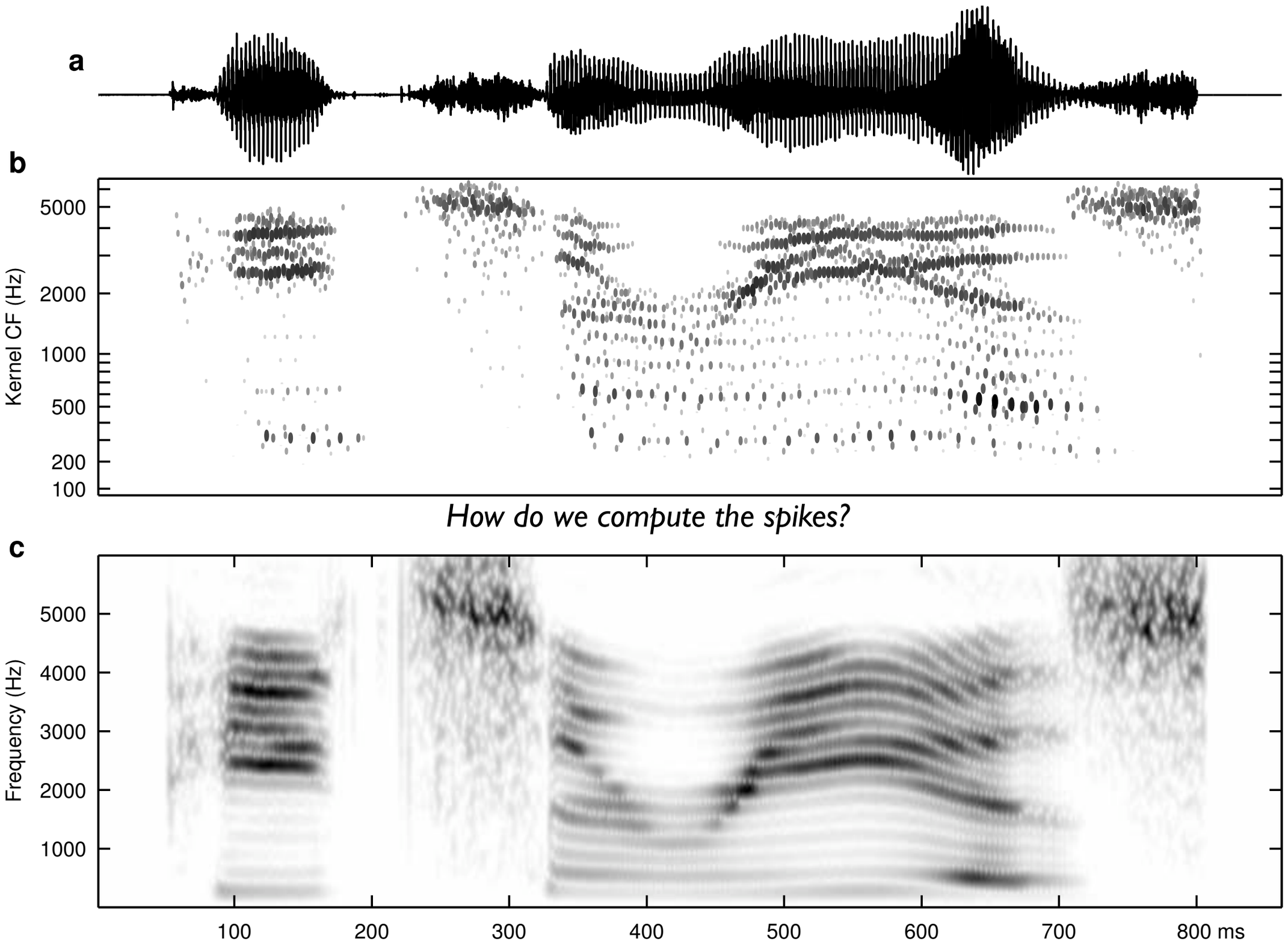
The spikegram



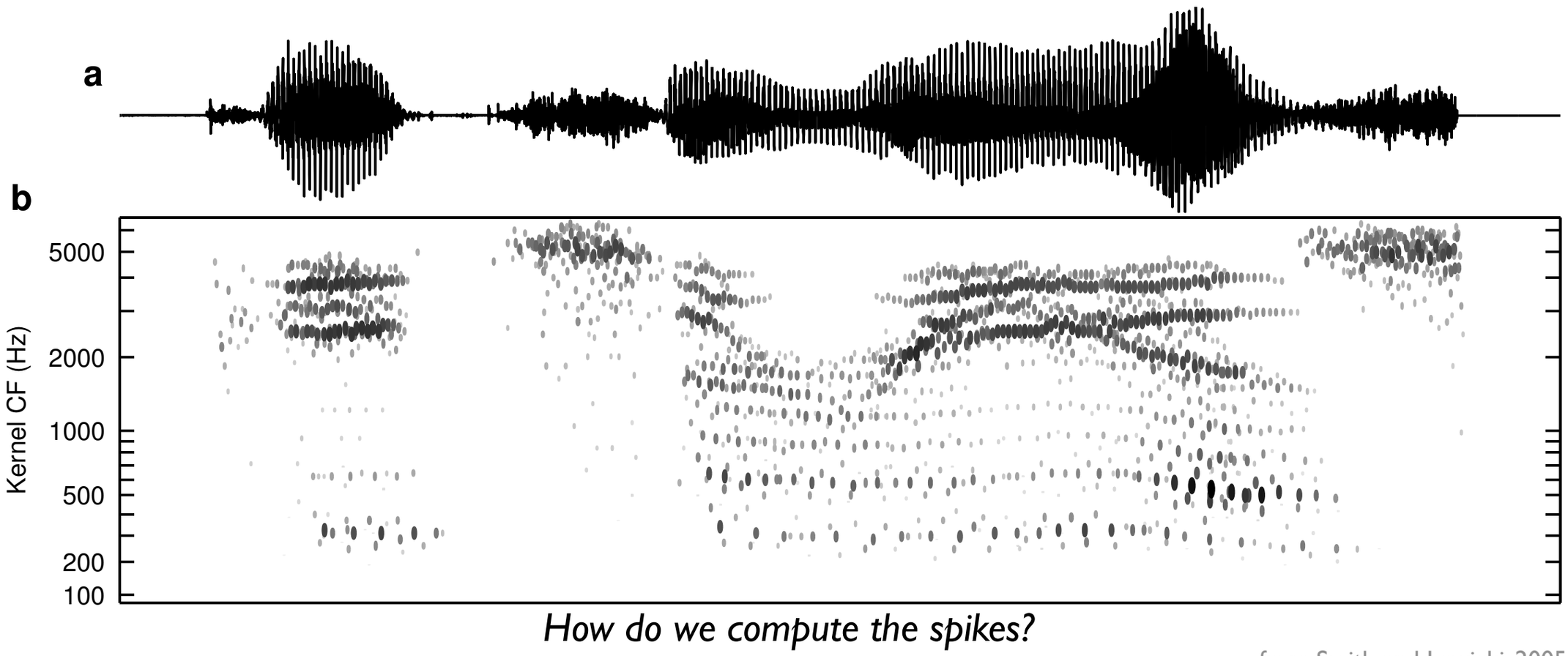
from Smith and Lewicki, 2005

Comparing a spike code to a spectrogram

from Smith and Lewicki, 2005



Comparing a spike code to a spectrogram



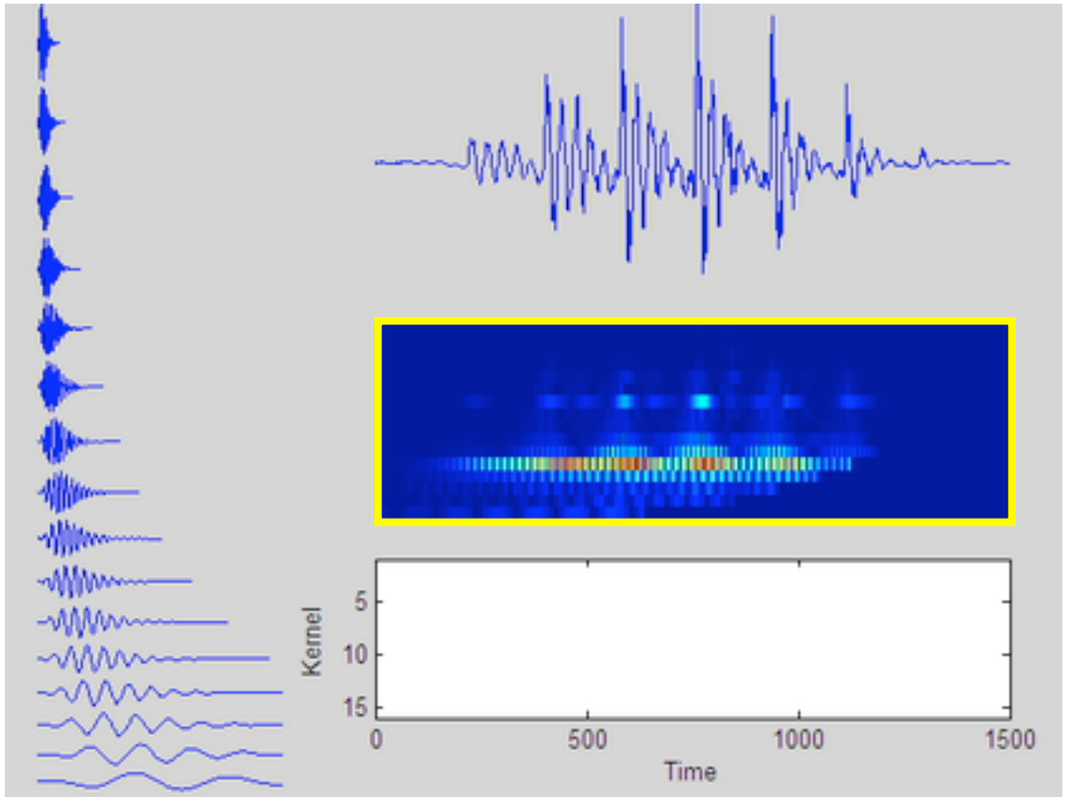
How do we compute the spikes?

from Smith and Lewicki, 2005

- There are many possible algorithms, varying degrees of biological plausibility
- Here, we use a variation of *Matching Pursuit* (Mallat and Zhang, 1993)
  - yields near optimal spike representation, but not biologically plausible
  - assume there exists a biol. plausible algorithm that achieves the same end

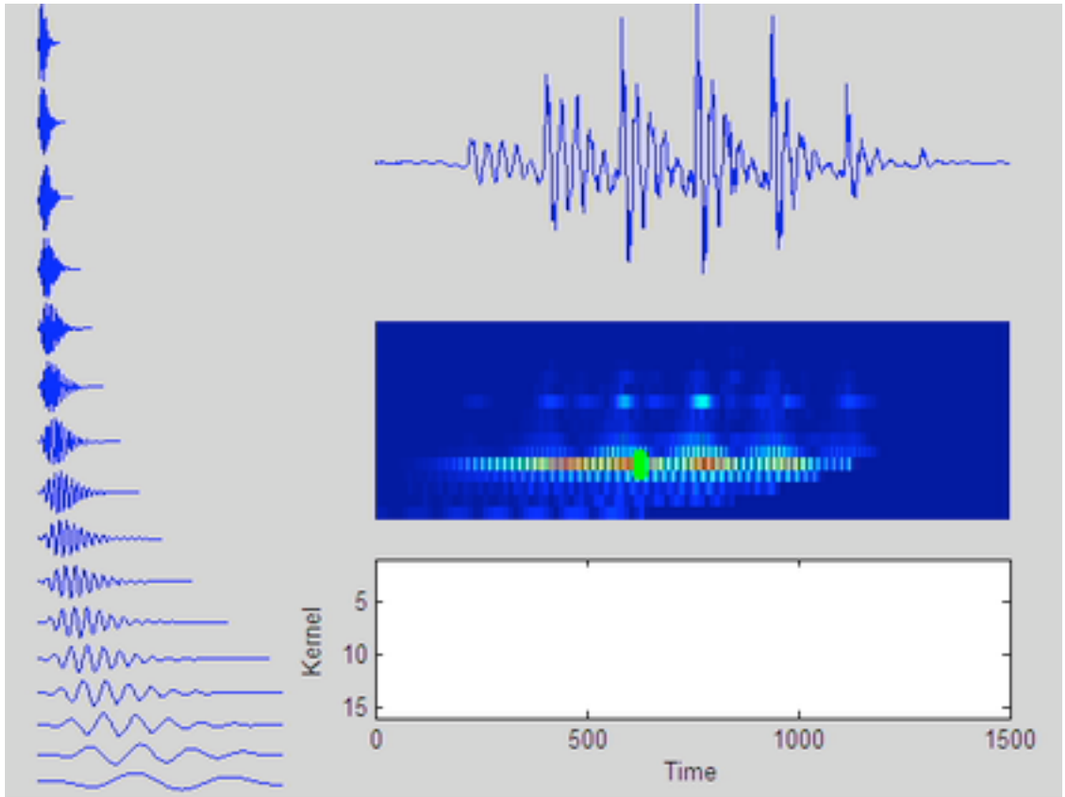
Spike Coding with Matching Pursuit

1. convolve signal with kernels



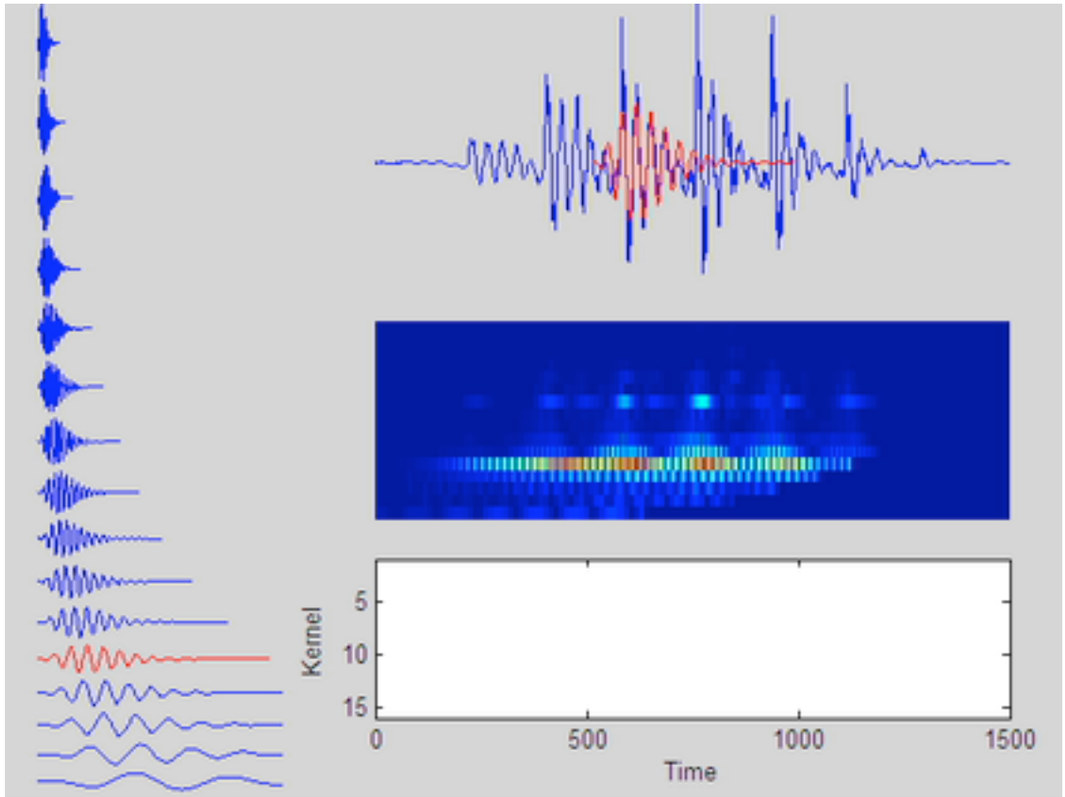
Spike Coding with Matching Pursuit

1. convolve signal with kernels
2. find largest peak over convolution set



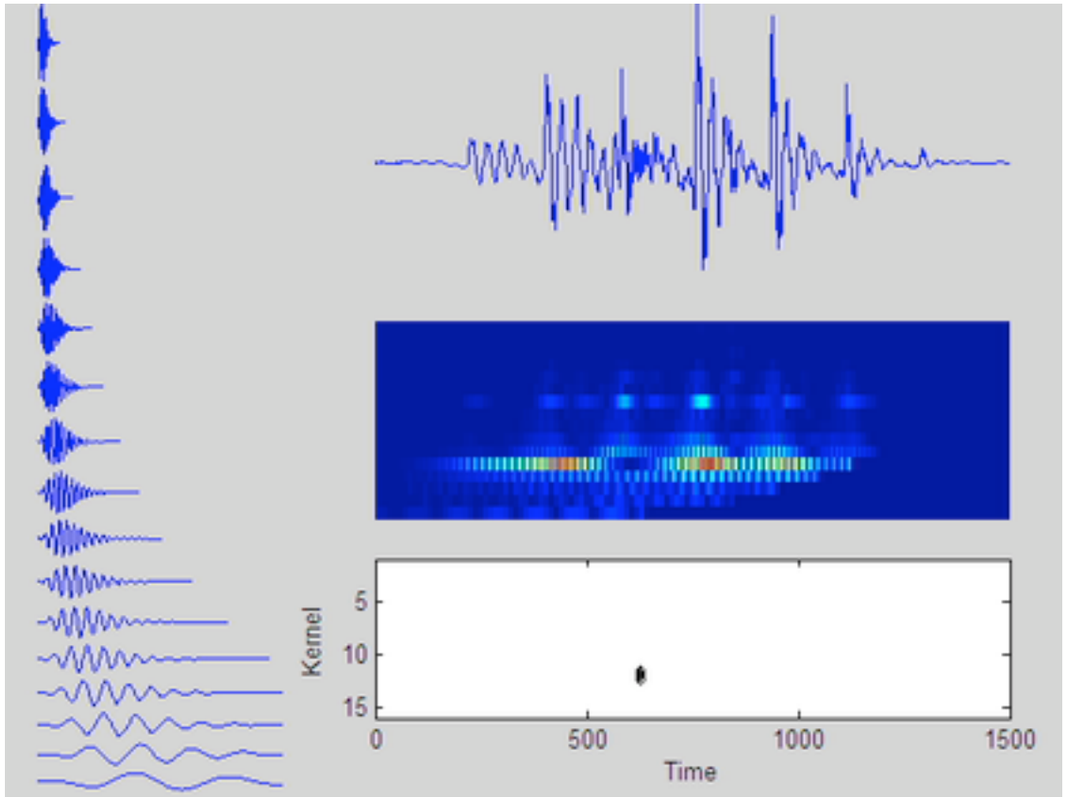
Spike Coding with Matching Pursuit

1. convolve signal with kernels
2. find largest peak over convolution set
3. fit signal with kernel



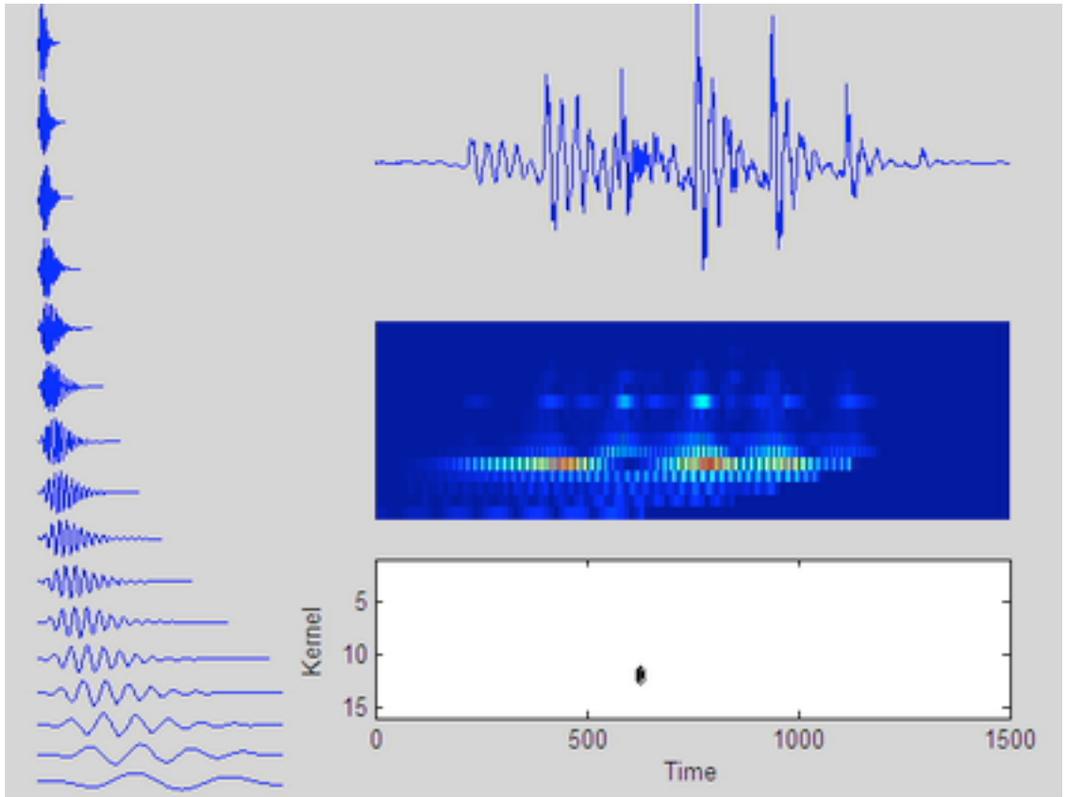
Spike Coding with Matching Pursuit

1. convolve signal with kernels
2. find largest peak over convolution set
3. fit signal with kernel
4. subtract kernel from signal, record spike, and adjust convolutions



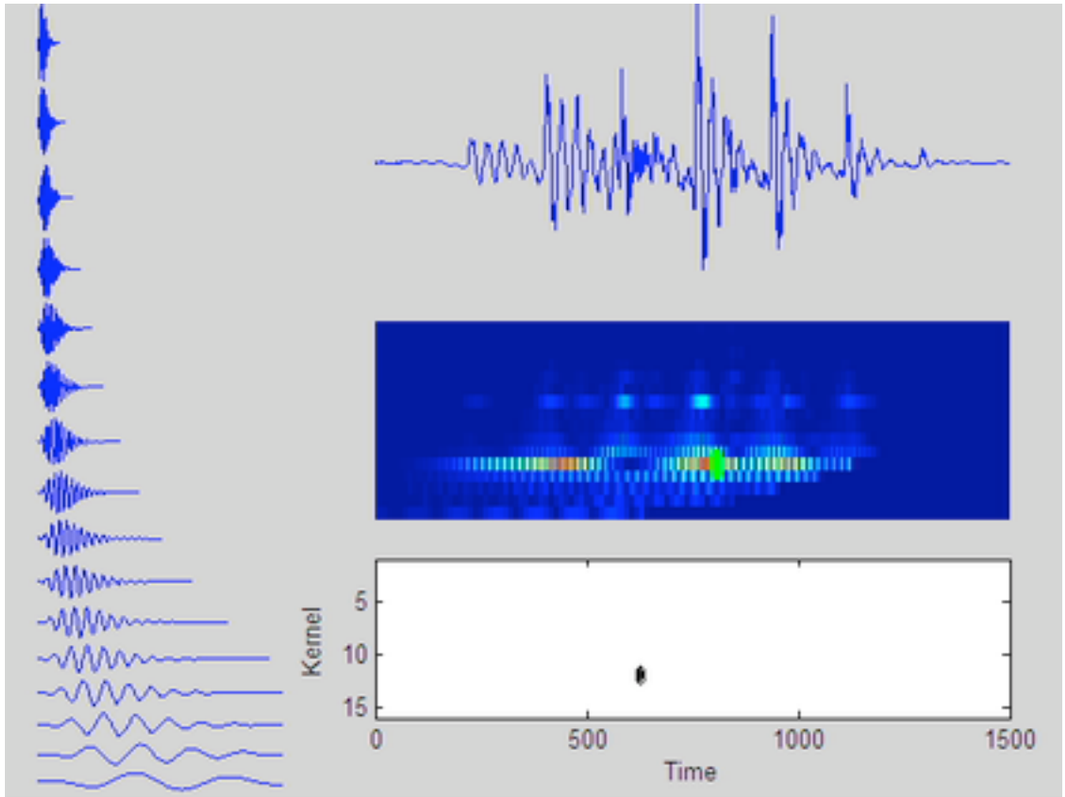
Spike Coding with Matching Pursuit

1. convolve signal with kernels
2. find largest peak over convolution set
3. fit signal with kernel
4. subtract kernel from signal, record spike, and adjust convolutions
5. repeat



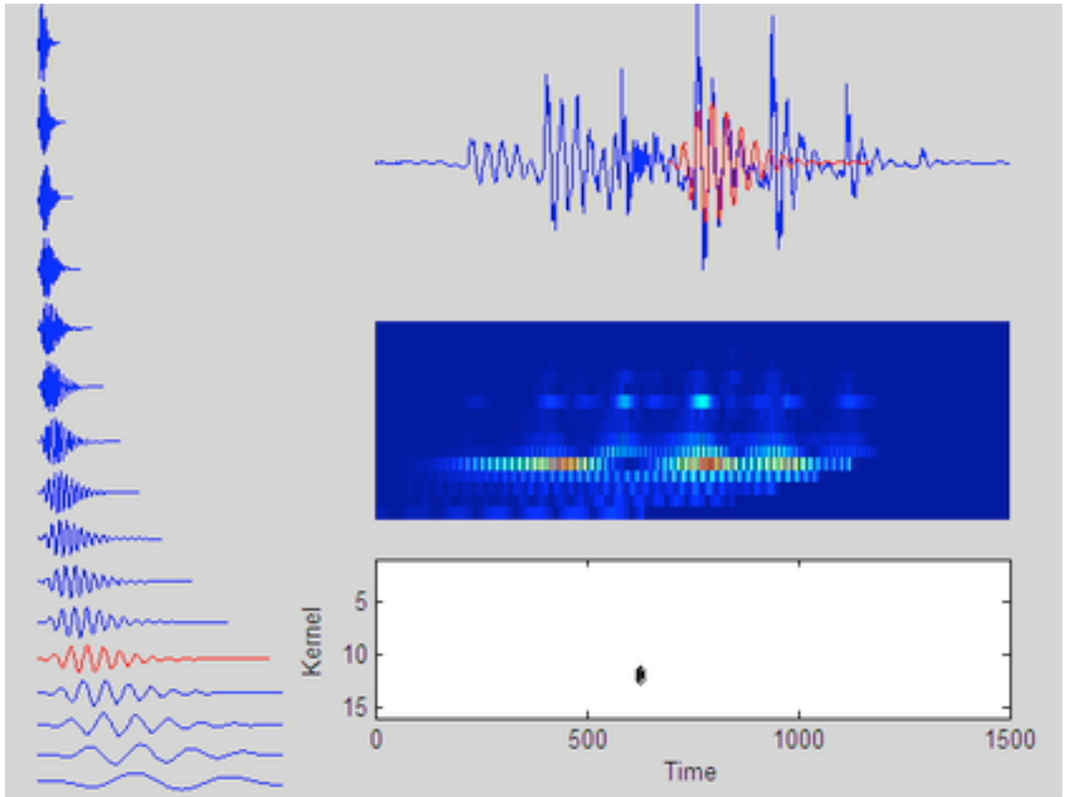
Spike Coding with Matching Pursuit

1. convolve signal with kernels
2. **find largest peak over convolution set**
3. fit signal with kernel
4. subtract kernel from signal, record spike, and adjust convolutions
5. repeat



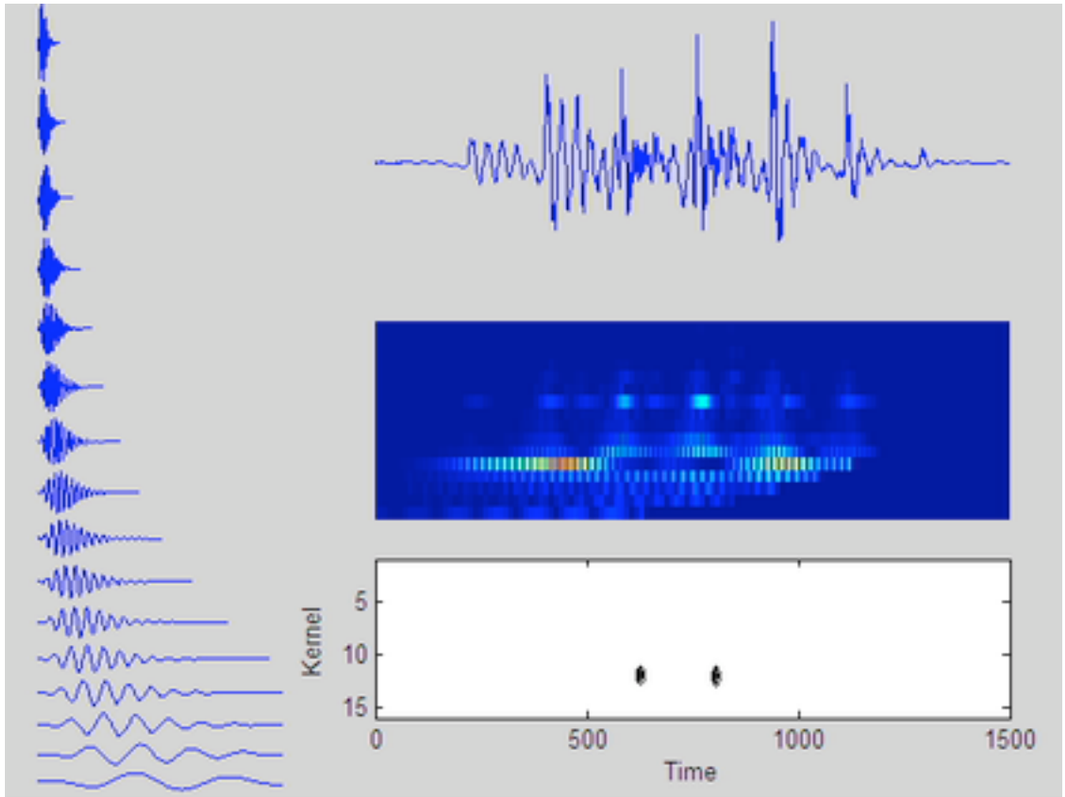
Spike Coding with Matching Pursuit

1. convolve signal with kernels
2. find largest peak over convolution set
3. **fit signal with kernel**
4. subtract kernel from signal, record spike, and adjust convolutions
5. repeat



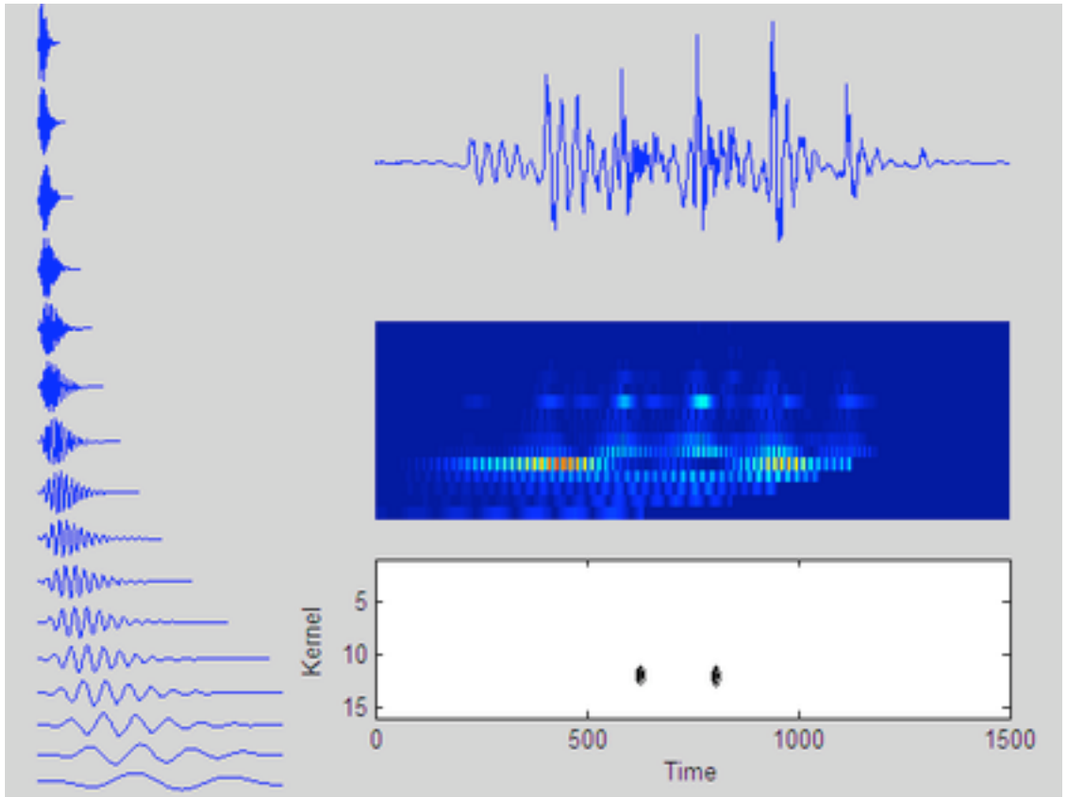
Spike Coding with Matching Pursuit

1. convolve signal with kernels
2. find largest peak over convolution set
3. fit signal with kernel
4. **subtract kernel from signal, record spike, and adjust convolutions**
5. repeat



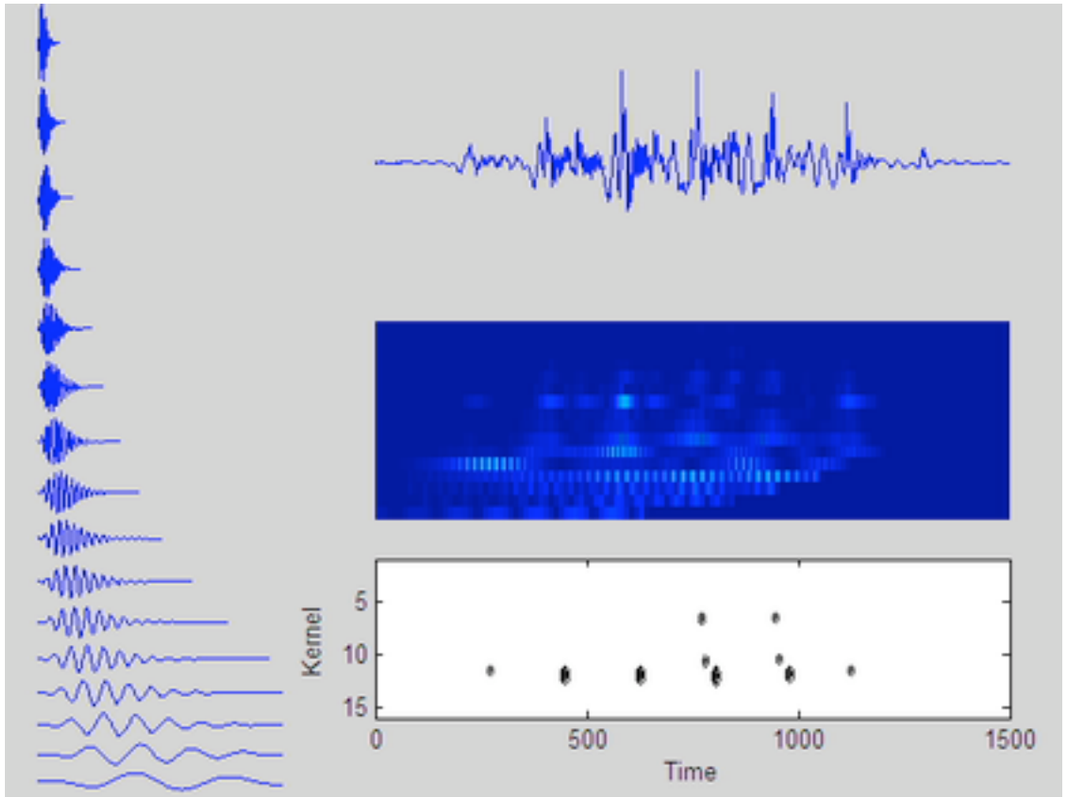
Spike Coding with Matching Pursuit

1. convolve signal with kernels
2. find largest peak over convolution set
3. fit signal with kernel
4. subtract kernel from signal, record spike, and adjust convolutions
5. **repeat . . .**

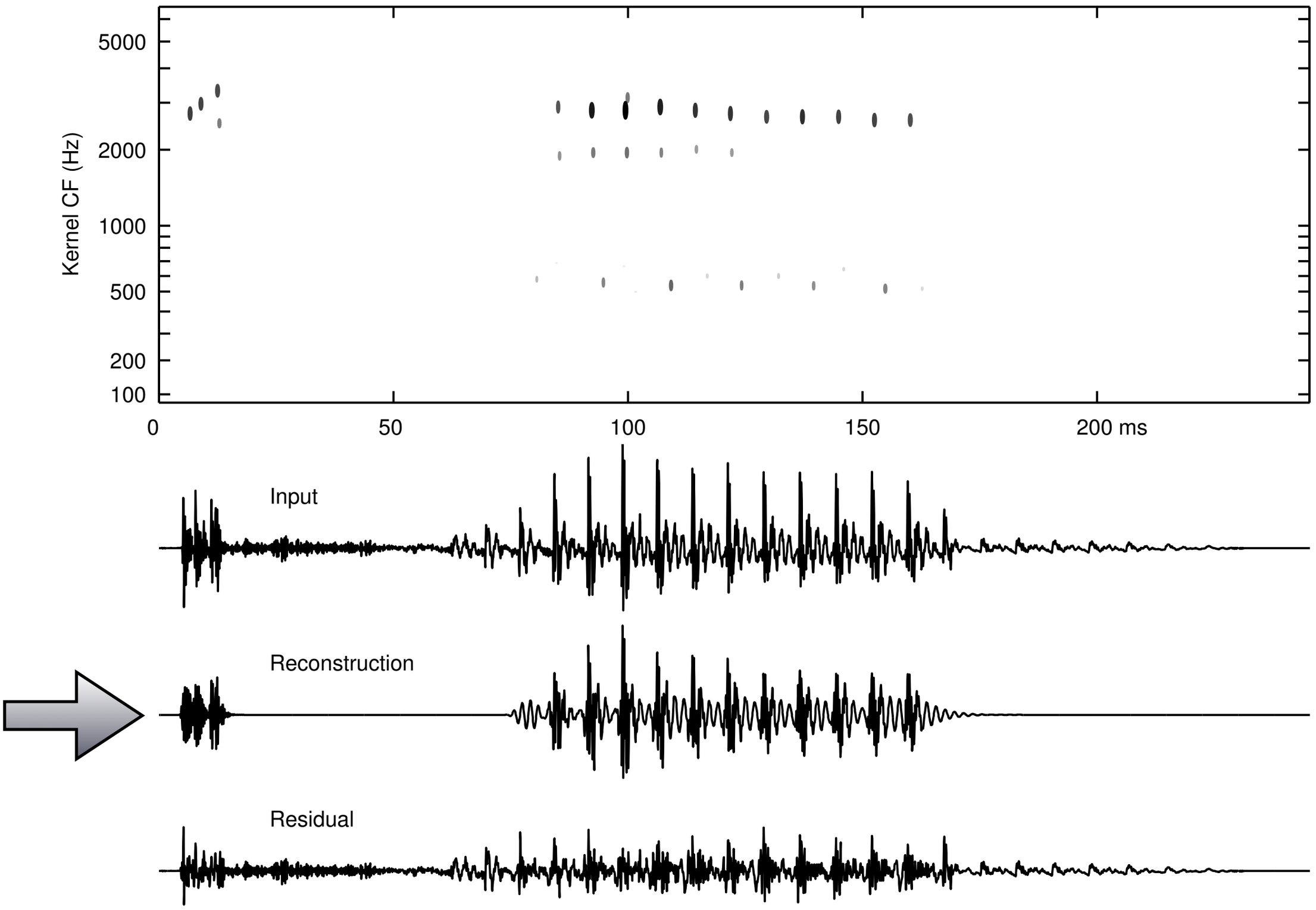


Spike Coding with Matching Pursuit

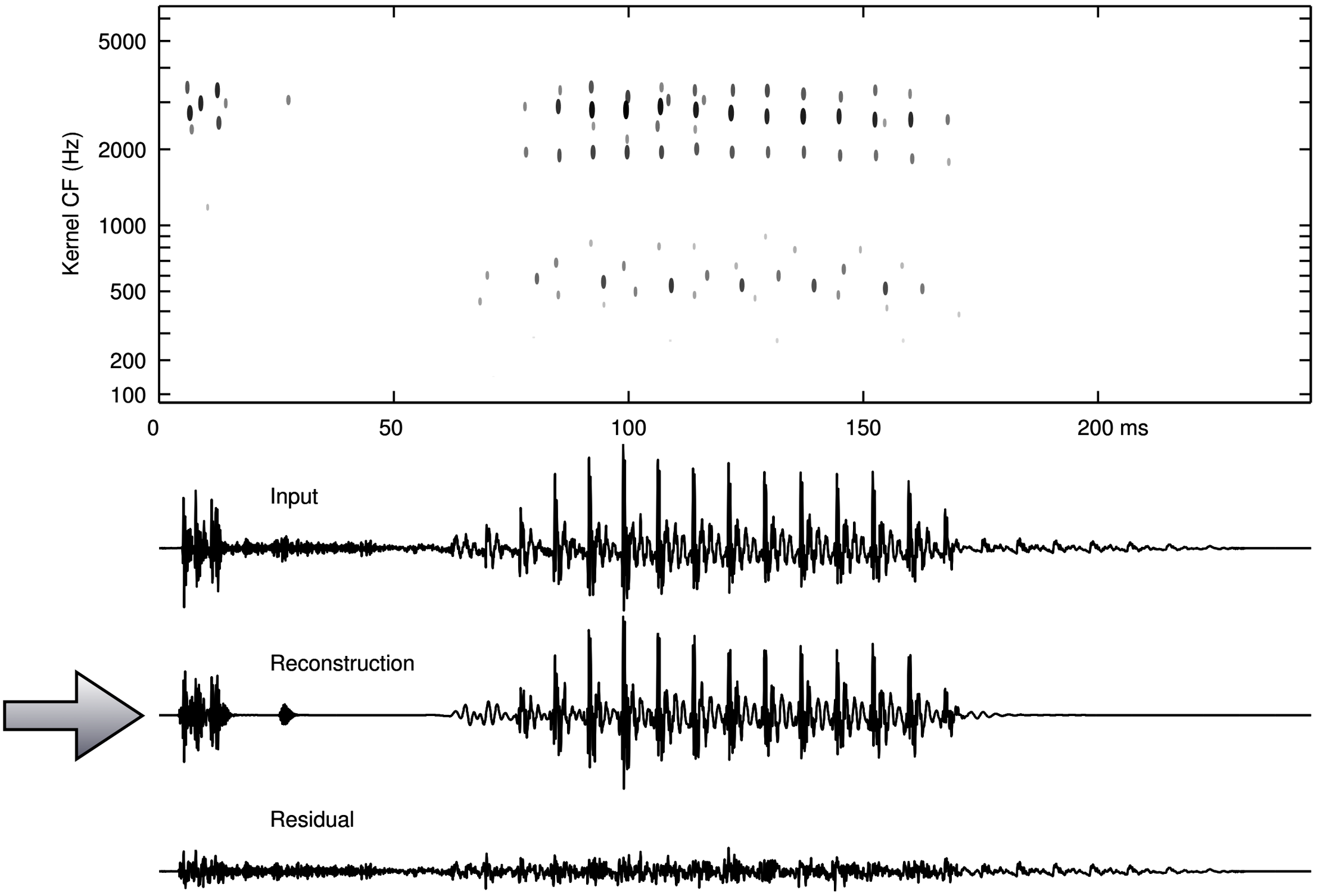
1. convolve signal with kernels
2. find largest peak over convolution set
3. fit signal with kernel
4. subtract kernel from signal, record spike, and adjust convolutions
5. repeat . . .
6. halt when desired fidelity is reached



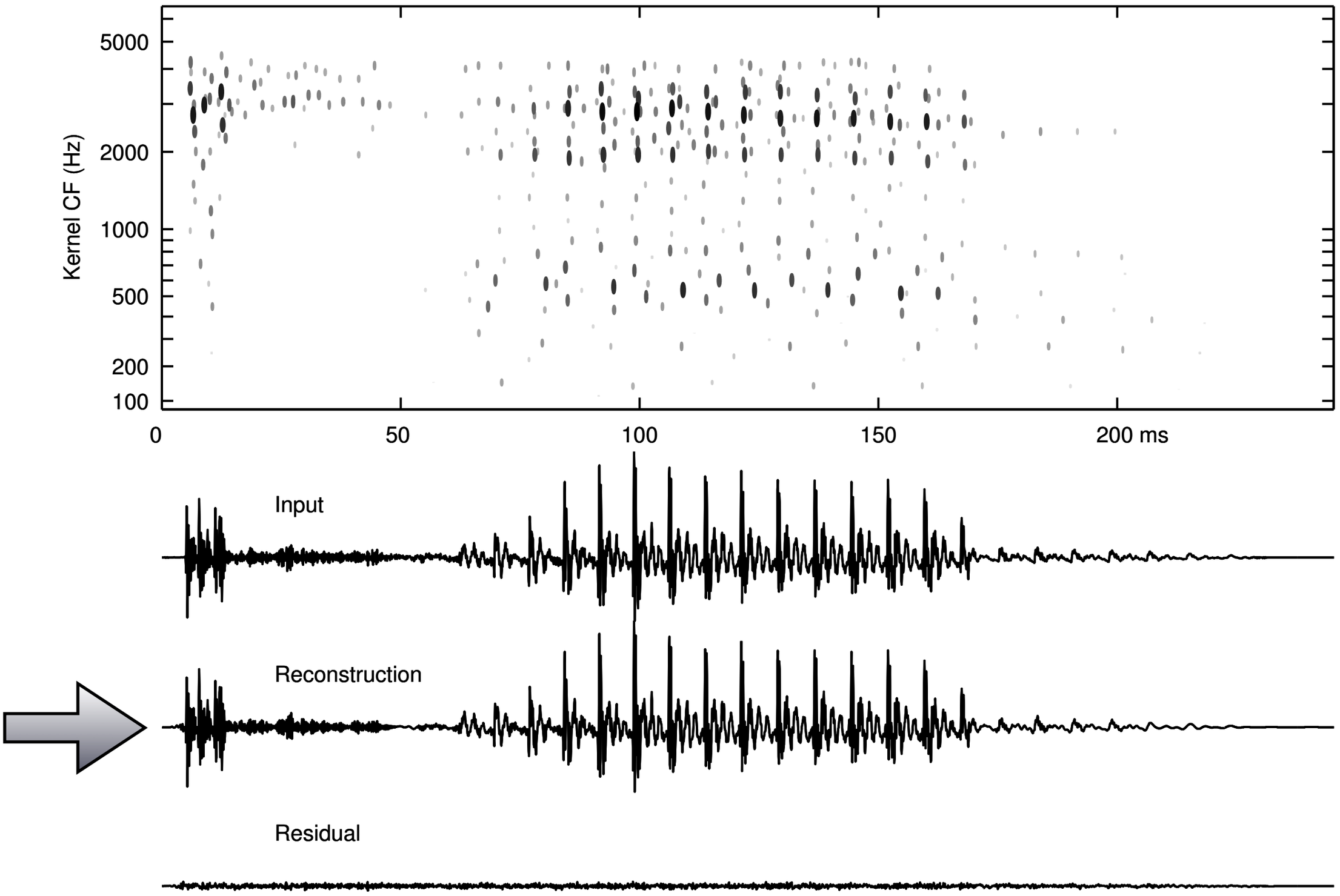
“can” 5 dB SNR, 36 spikes, 145 sp/sec



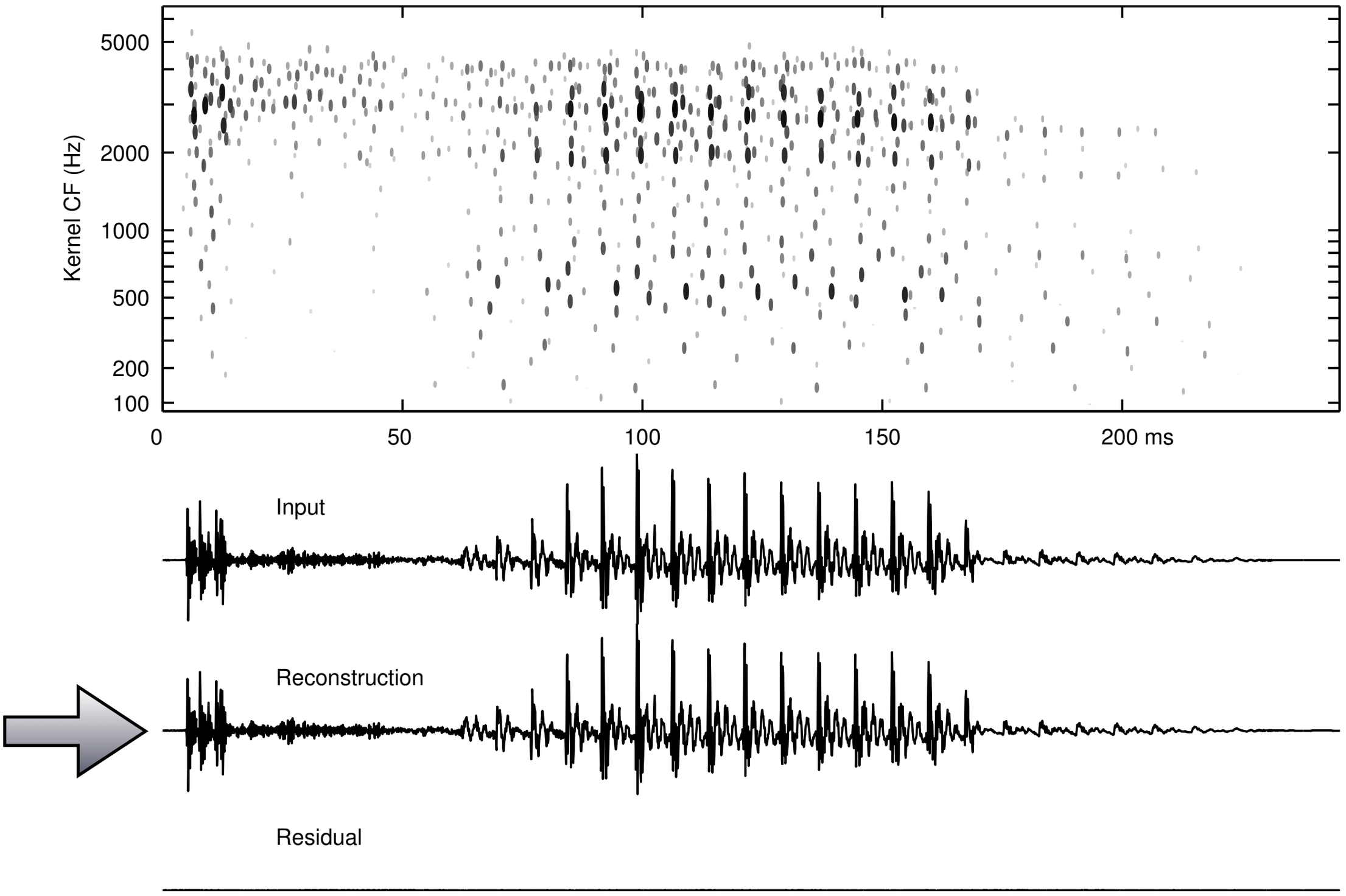
“can” 10 dB SNR, 93 spikes, 379 sp/sec



“can” 20 dB SNR, 391 spikes, 1700 sp/sec

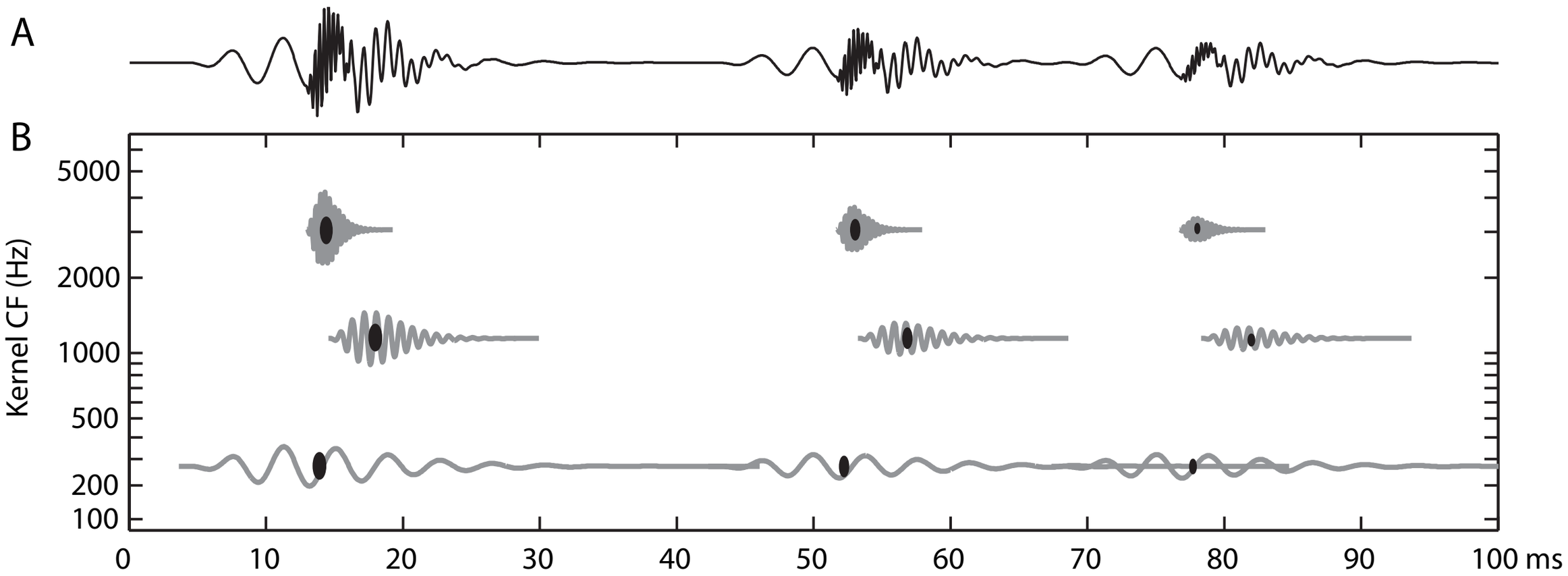


“can” 40 dB SNR, 1285 spikes, 5238 sp/sec



What are the optimal shapes for the kernels?

$$x(t) = \sum_{m=1}^{M} \sum_{i=1}^{n_m} s_{m,i}\, \phi_m(t - \tau_{m,i}) + \epsilon(t)$$



from Smith and Lewicki, 2005

Optimizing the probabilistic model

$$x(t) = \sum_{m=1}^{M} \sum_{i=1}^{n_m} s_i^m \, \phi_m(t - \tau_i^m) + \epsilon(t),$$

$$\begin{aligned} p(x|\Phi) &= \int p(x|\Phi, s, \tau) p(s) p(\tau) ds d\tau \\ &\approx p(x|\Phi, \hat{s}, \hat{\tau}) p(\hat{s}) p(\hat{\tau}) \end{aligned}$$

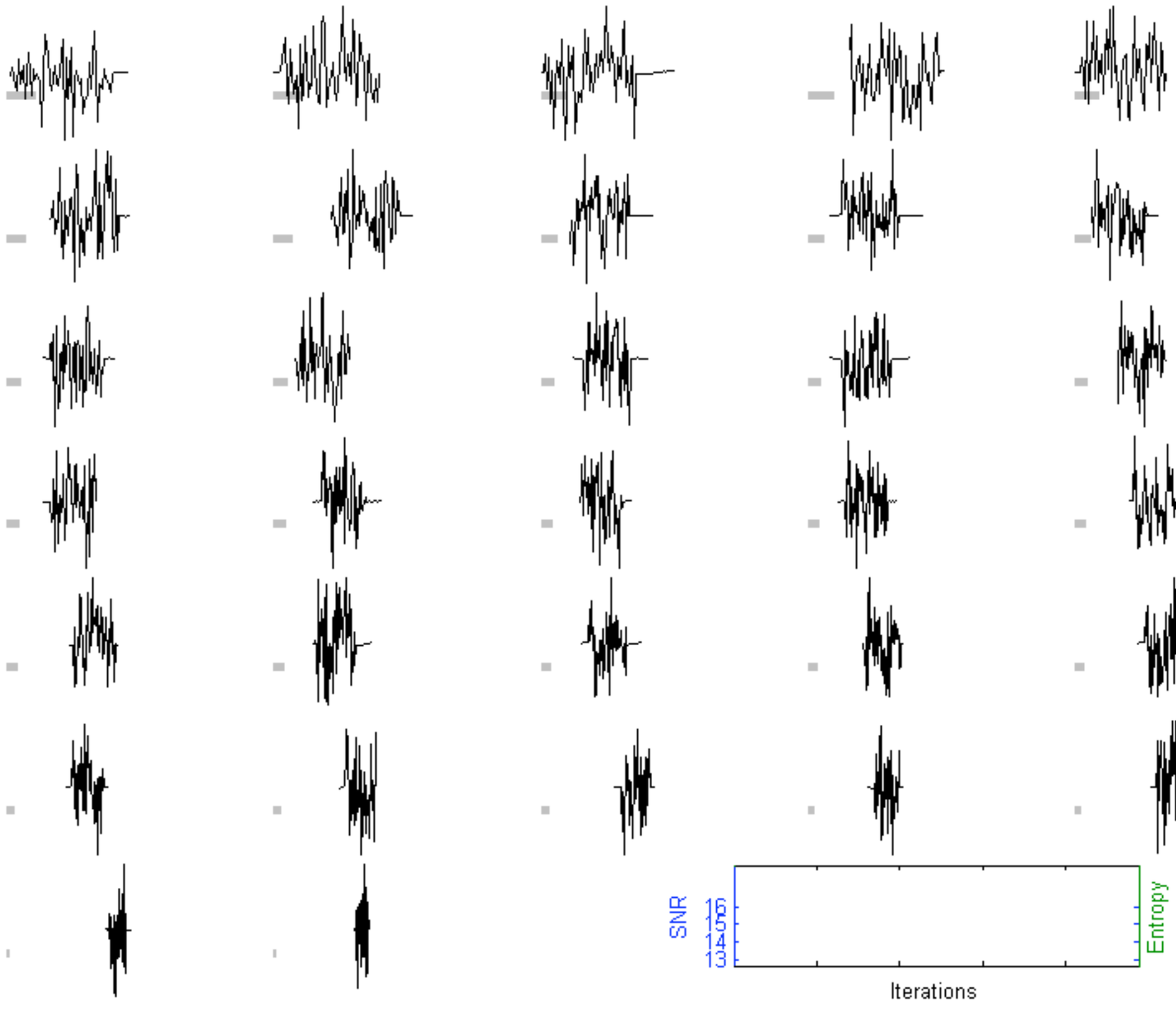
$$\epsilon(t) \sim \mathcal{N}(0, \sigma_\epsilon)$$

Learning (Olshausen, 2002):

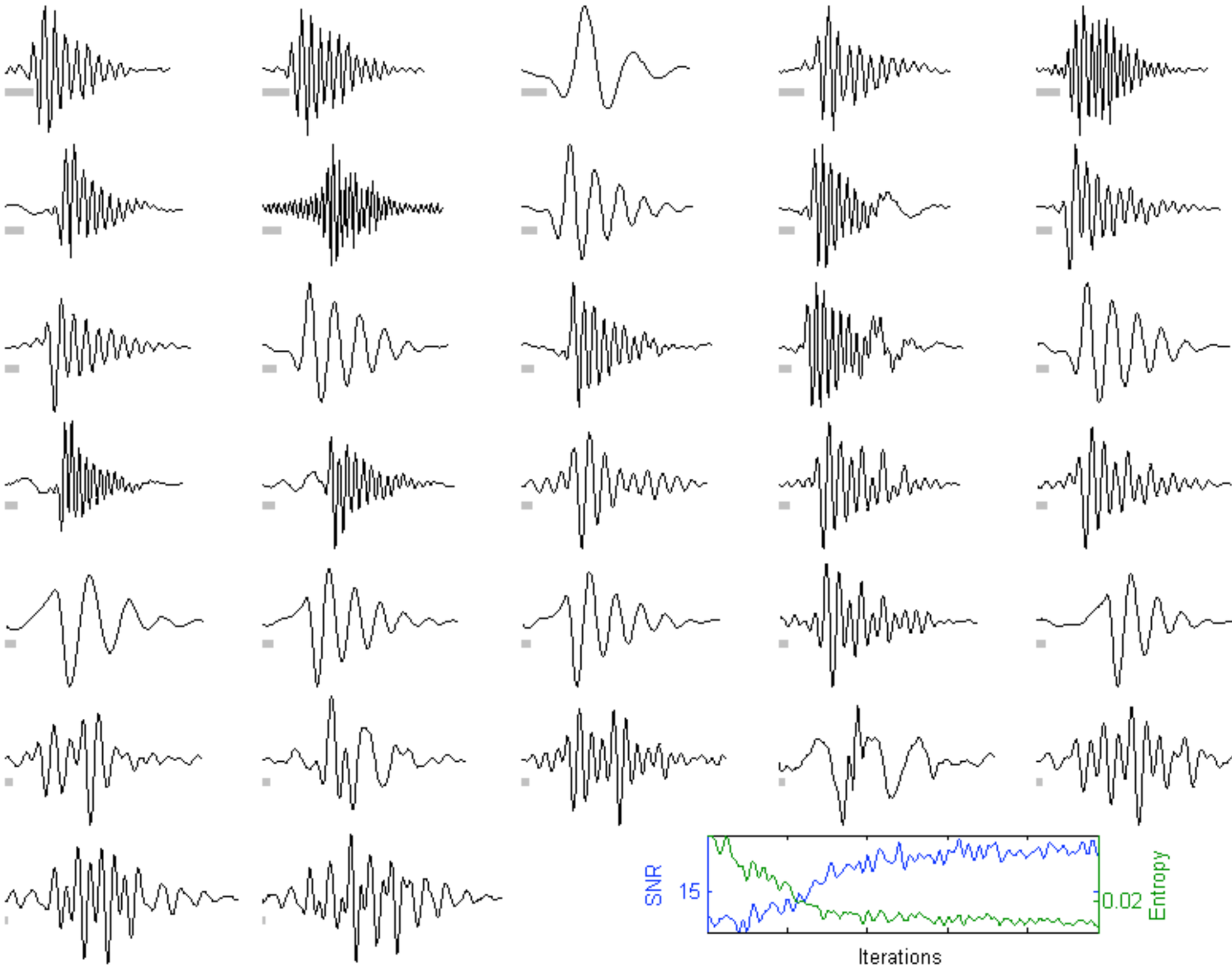
$$\begin{aligned} \frac{\partial}{\partial \phi_m} \log p(x|\Phi) &= \frac{\partial}{\partial \phi_m} \log p(x|\Phi, \hat{s}, \hat{\tau}) + \log p(\hat{s}) p(\hat{\tau}) \\ &= \frac{1}{2\sigma_\varepsilon} \frac{\partial}{\partial \phi_m} [x - \sum_{m=1}^{M} \sum_{i=1}^{n_m} \hat{s}_i^m \, \phi_m(t - \tau_i^m)]^2 \\ &= \frac{1}{\sigma_\varepsilon} [x - \hat{x}] \sum_i \hat{s}_i^m \end{aligned}$$

Also extend algorithm to adapt kernel lengths

Adapting the optimal kernel shapes



Kernel functions optimized for coding speech



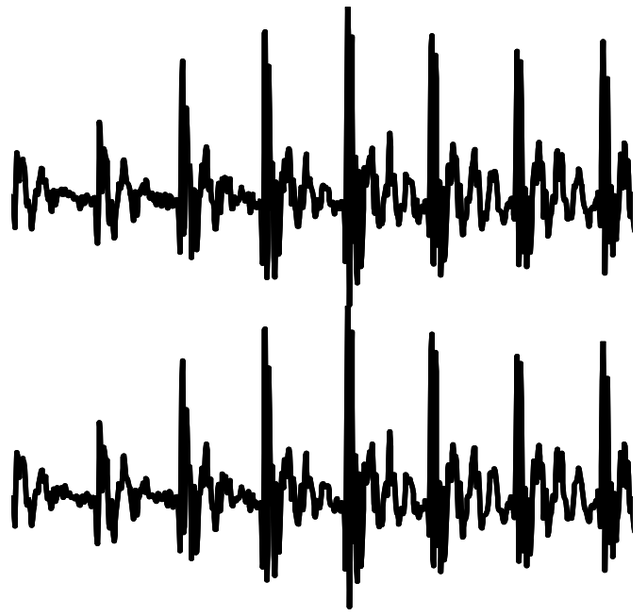
Quantifying coding efficiency

1. *fit* signal
2. *quantize* time and amplitude values
3. *prune* zero values
4. *measure coding efficiency* using the entropy of quantized values
5. *reconstruct signal* using quantized values
6. *measure fidelity* using signal-to-noise ratio (SNR) of residual error

- identical procedure for other codes (e.g. Fourier and wavelet)

$$x(t) = \sum_{m=1}^{M} \sum_{i=1}^{n_m} s_{m,i}\, \phi_m(t - \tau_{m,i}) + \epsilon(t)$$

spike code

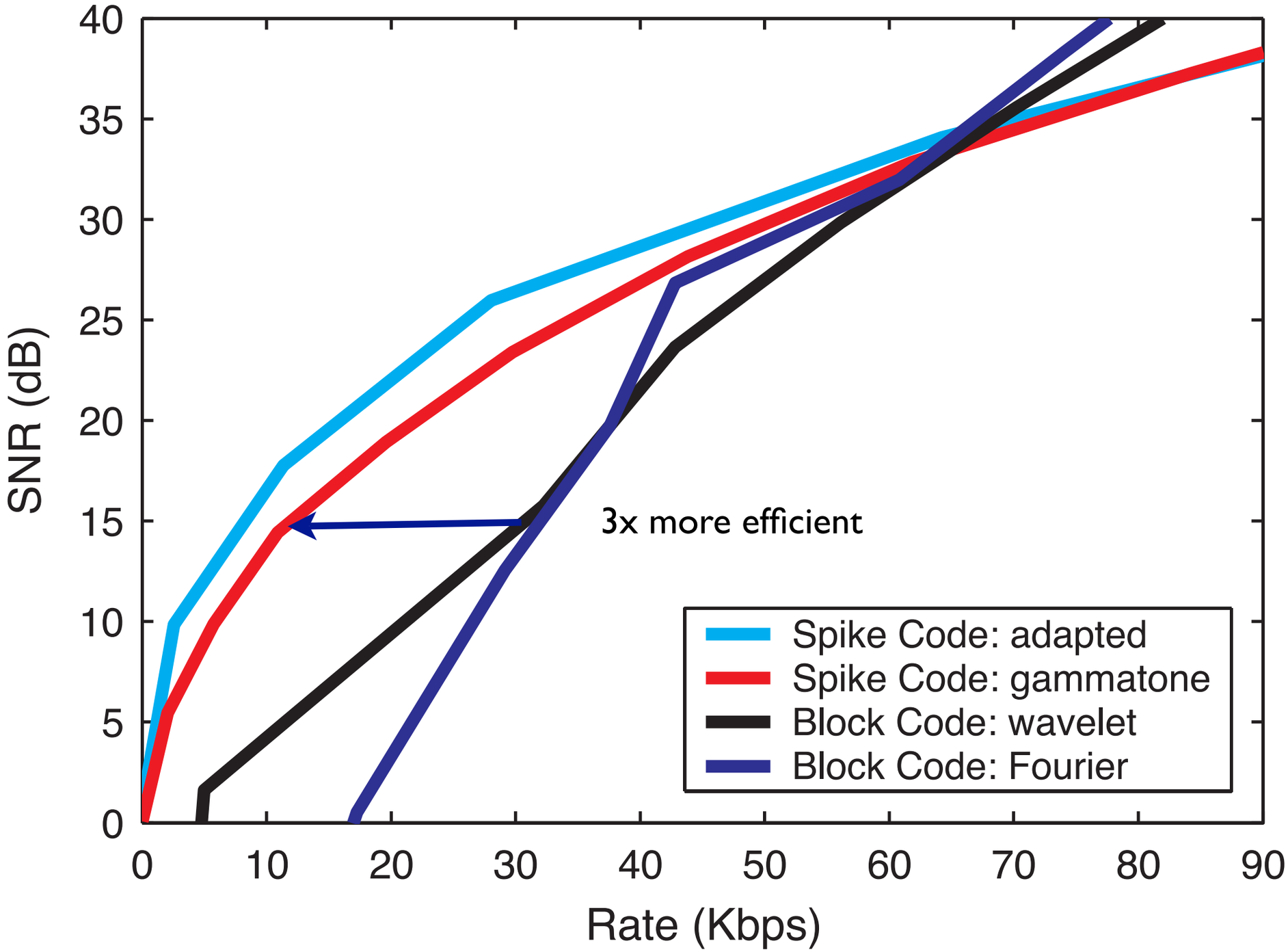


original

reconstruction

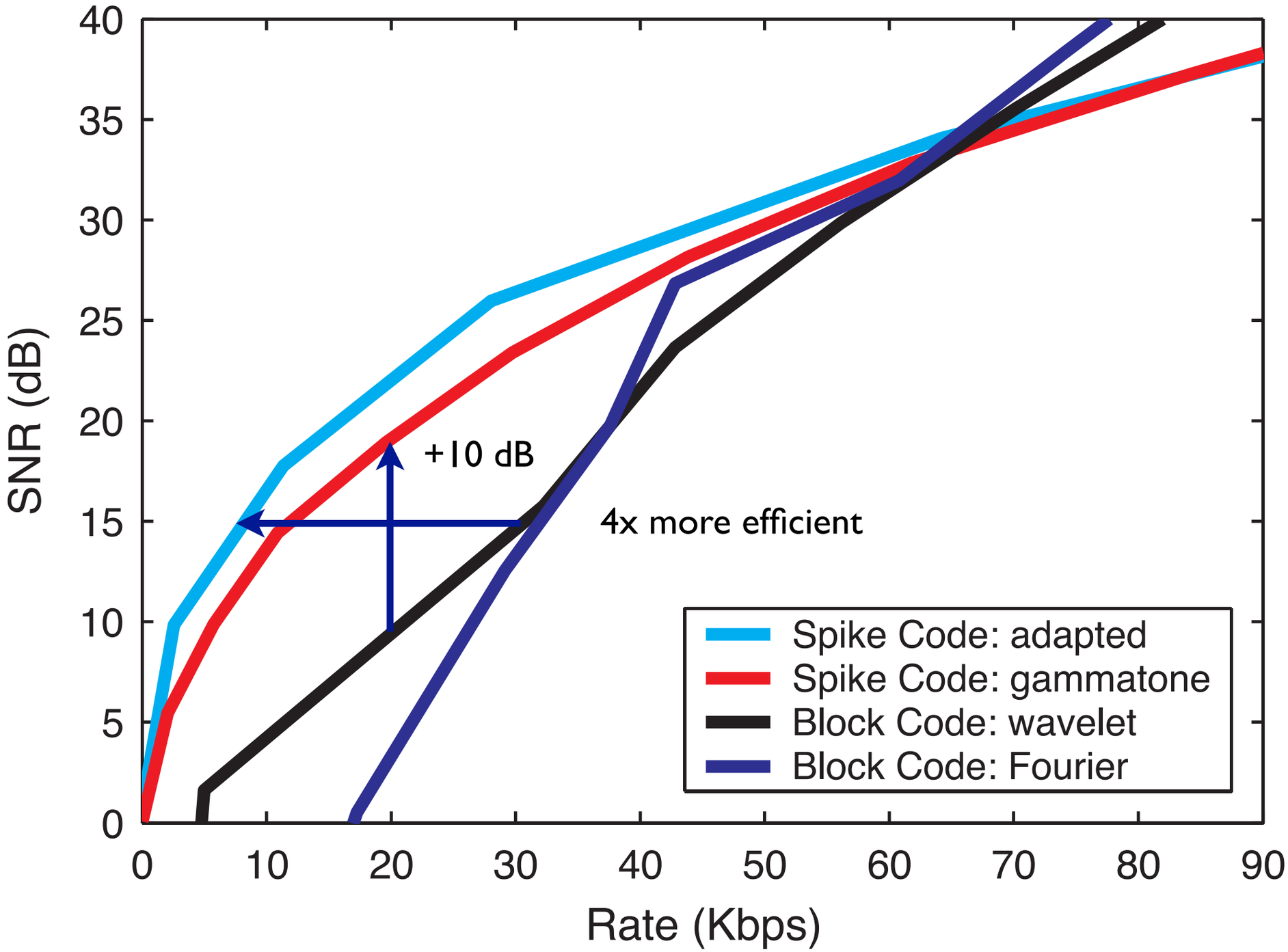
residual error

Coding efficiency curves



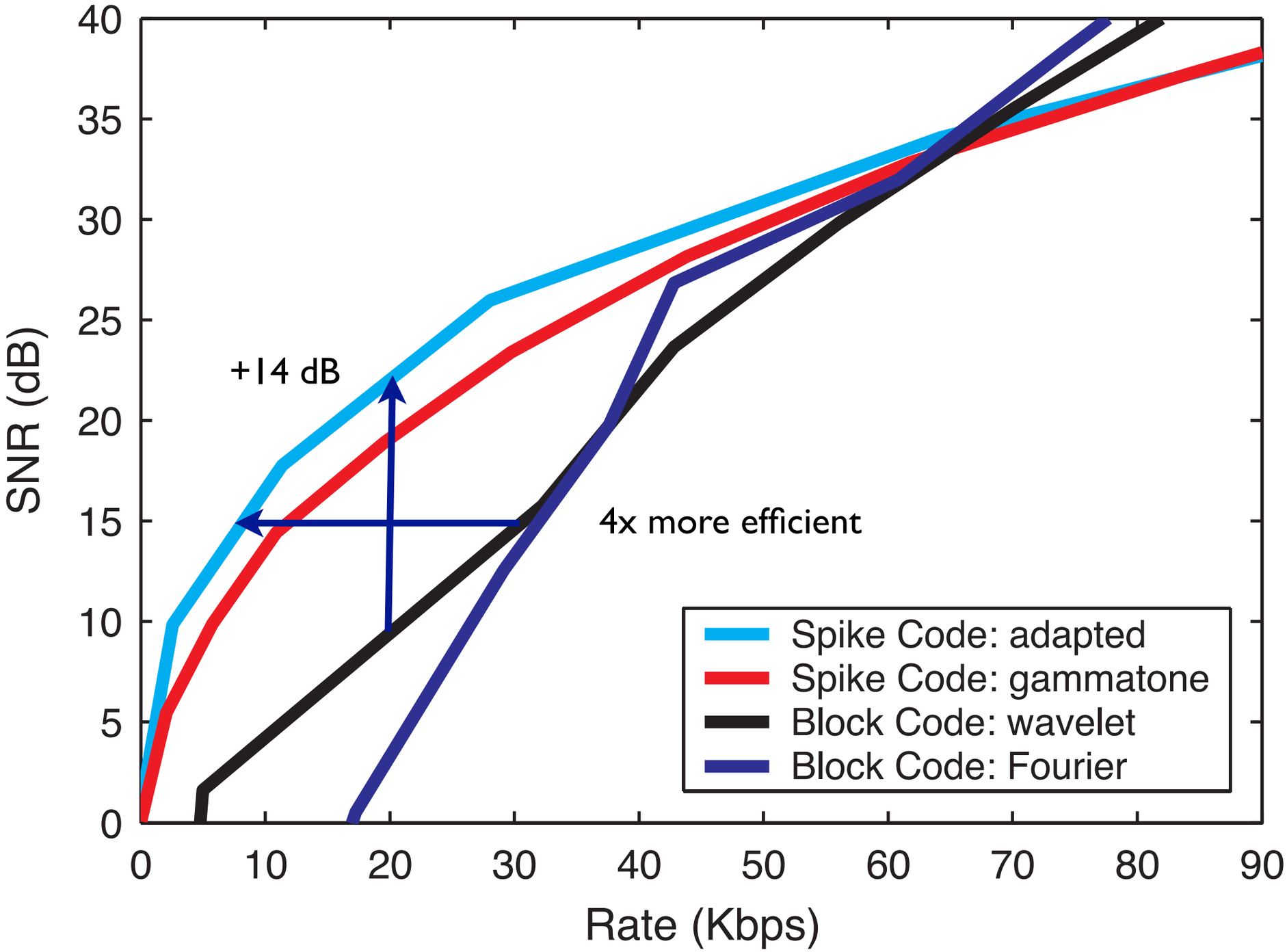
from Smith and Lewicki, 2005

Coding efficiency curves



from Smith and Lewicki, 2005

Coding efficiency curves



from Smith and Lewicki, 2005

Using efficient coding theory to make theoretical predictions

Environment of Natural Sounds
(speech, env. sounds, vocalizations)

optimize kernel coding efficiency

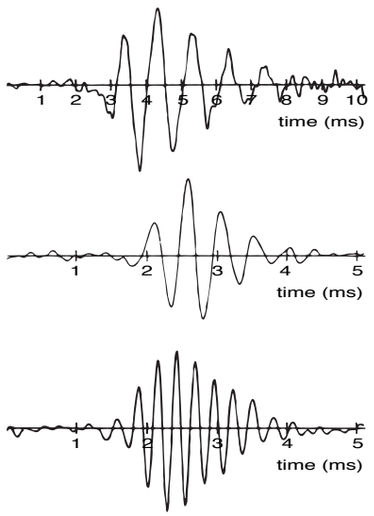
optimal kernels:
- properties
- coding efficiency

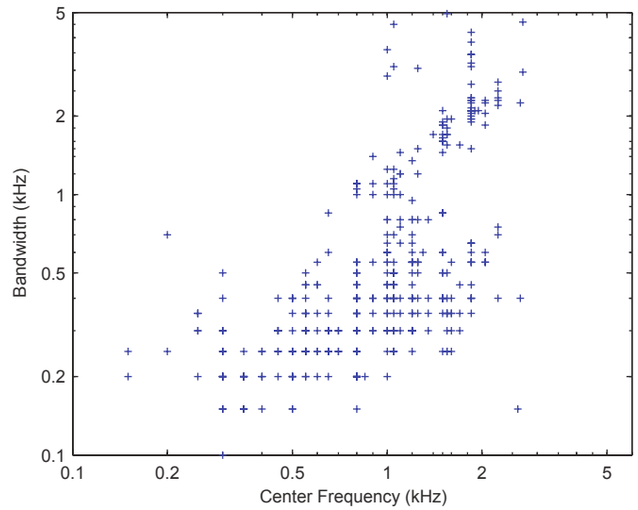
evolution

physiological data:
- auditory nerve filter shapes
- population trends

?

only compare to the data *after* optimizing
we do not fit the data



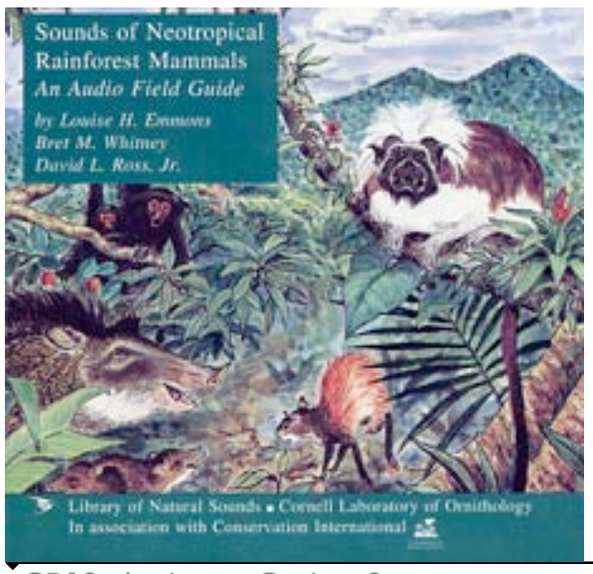


Natural sounds

	environmental sounds	
vocalizations	transient	ambient
fox	walking on leaves	rustling leaves
squirrel	cracking branches	stream by waterfall

Natural sounds

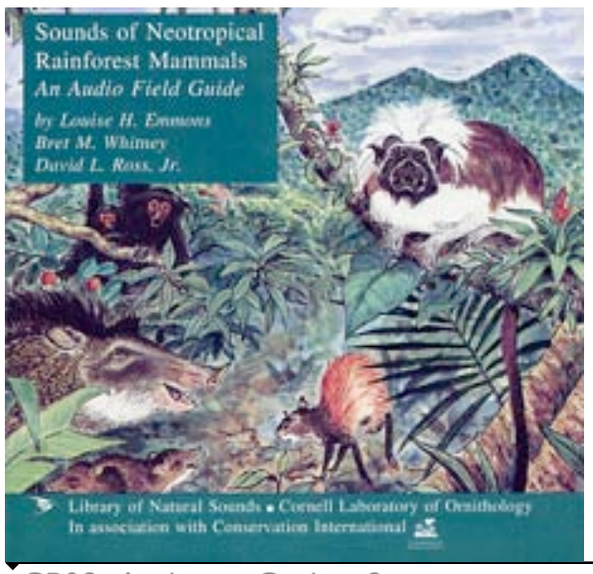
	environmental sounds	
vocalizations	transient	ambient
fox	walking on leaves	rustling leaves
squirrel	cracking branches	stream by waterfall



CORNELL LAB of ORNITHOLOGY
MACAULAY LIBRARY

Natural sounds

	environmental sounds	
vocalizations	transient	ambient
fox	walking on leaves	rustling leaves
squirrel	cracking branches	stream by waterfall



CORNELL LAB of ORNITHOLOGY
MACAULAY LIBRARY

Natural sounds

	environmental sounds	
vocalizations	transient	ambient
fox	**walking on leaves**	rustling leaves
squirrel	cracking branches	stream by waterfall

Natural sounds

	environmental sounds	
vocalizations	transient	ambient
fox	walking on leaves	rustling leaves
squirrel	**cracking branches**	stream by waterfall

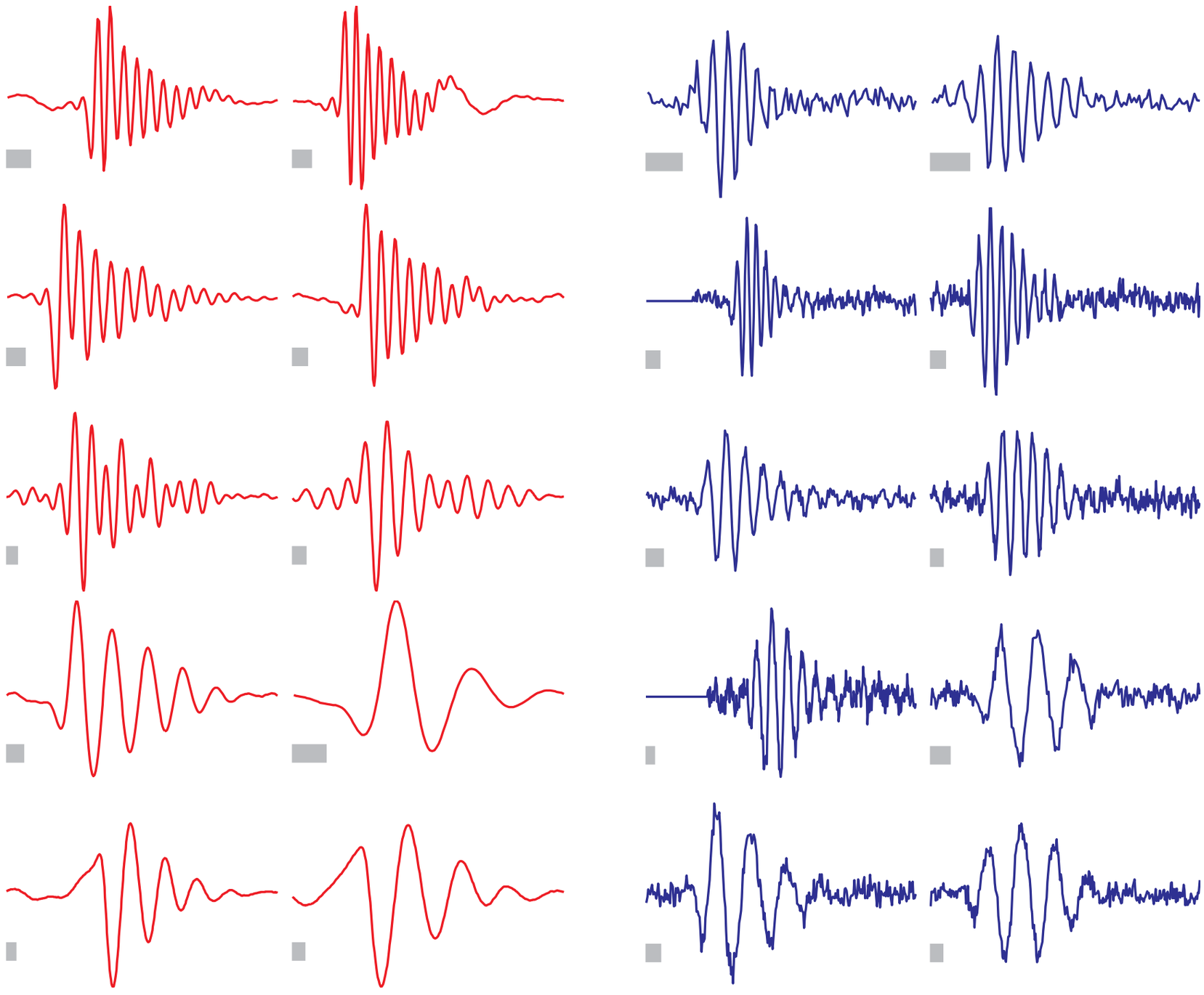
Natural sounds

	environmental sounds	
vocalizations	transient	ambient
fox	walking on leaves	**rustling leaves**
squirrel	cracking branches	stream by waterfall

Natural sounds

	environmental sounds	
vocalizations	transient	ambient
fox	walking on leaves	rustling leaves
squirrel	cracking branches	**stream by waterfall**

Learned kernels share features of auditory nerve filters



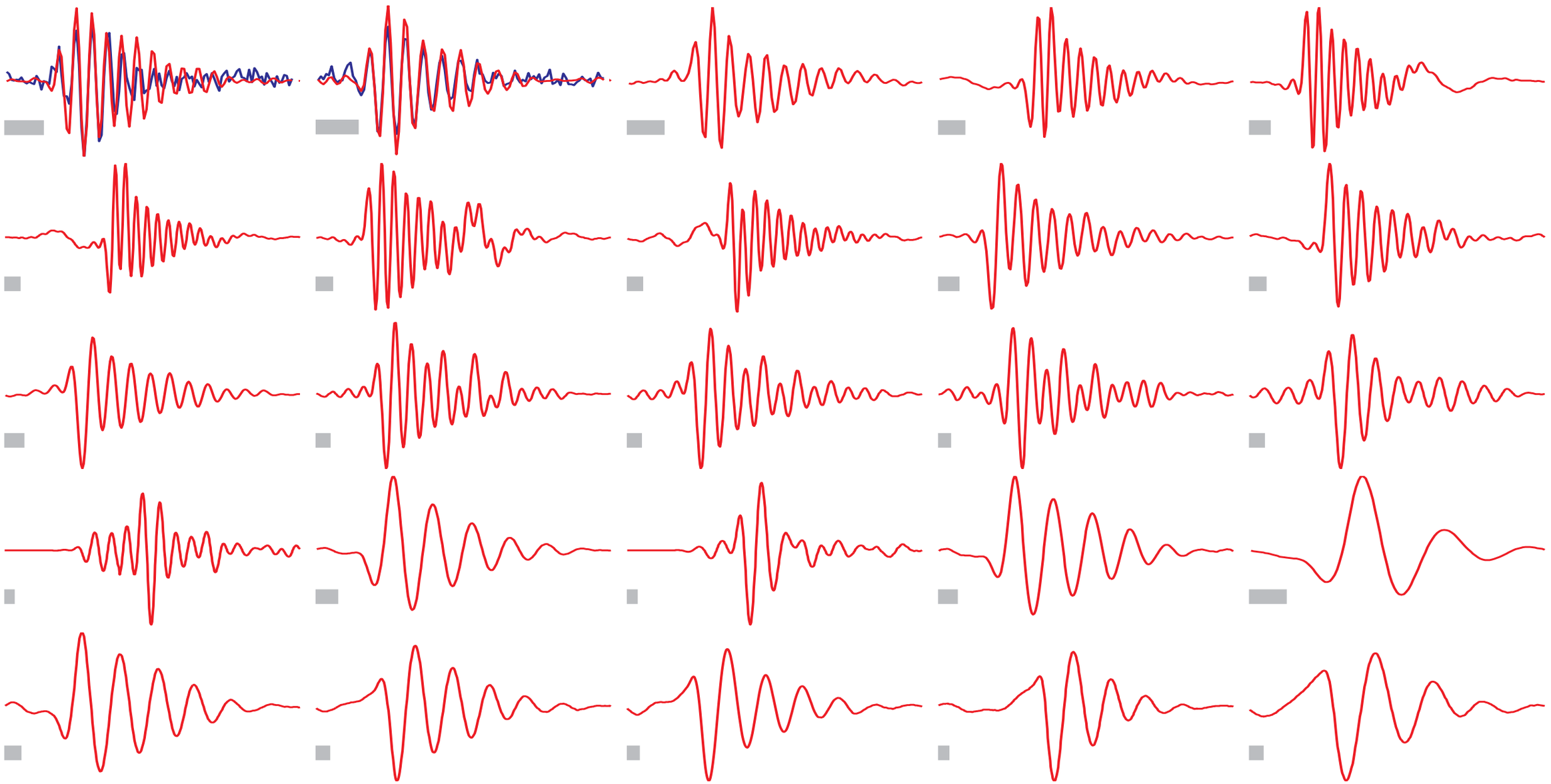
Optimized kernels

scale bar = 1 msec

Auditory nerve filters

from Carney, McDuffy, and Shekhter, 1999

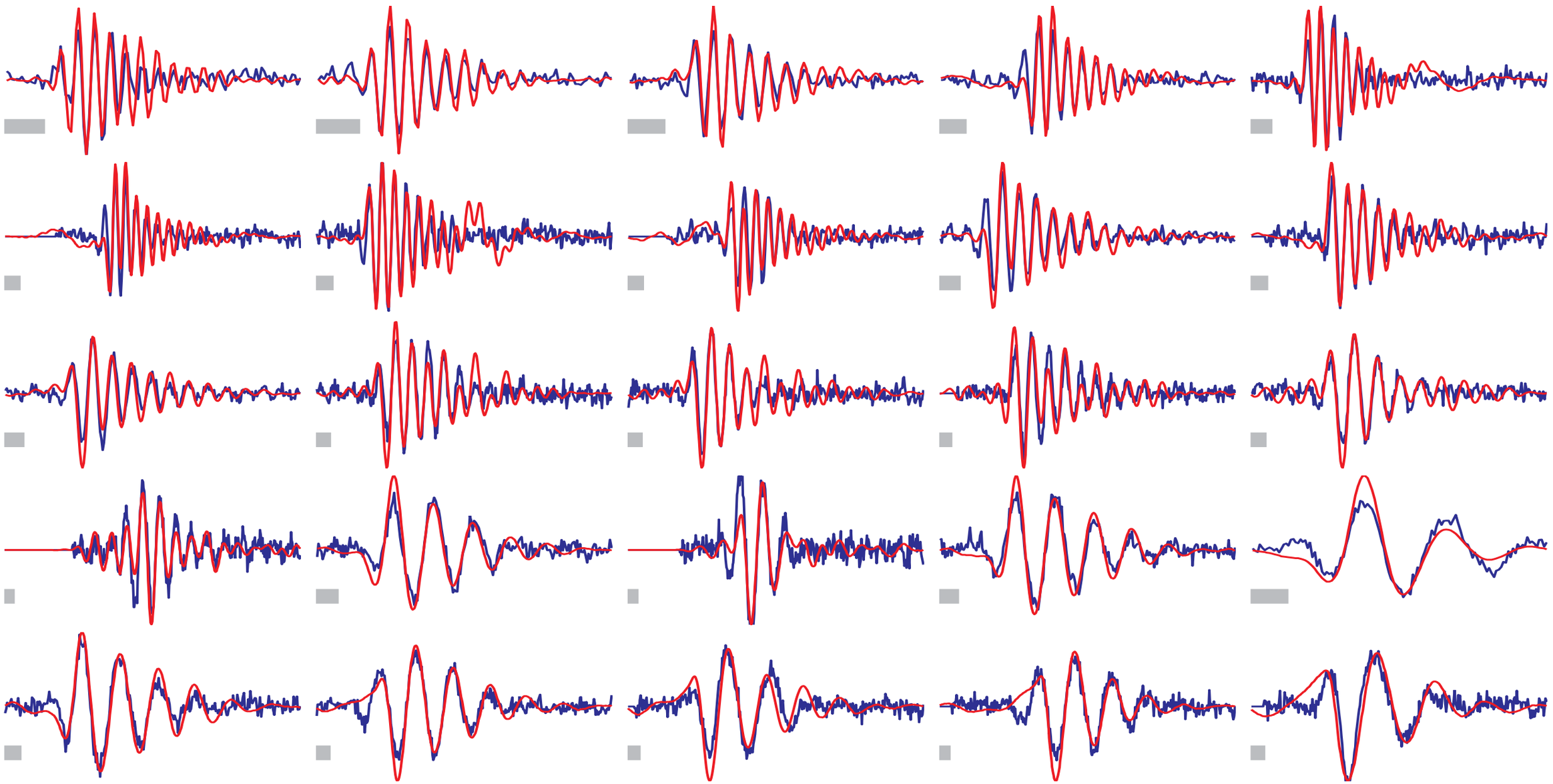
Learned kernels closely match individual auditory nerve filters



for each kernel find closet matching auditory nerve filter
in Laurel Carney's database of ~100 filters.

from Smith and Lewicki, 2005

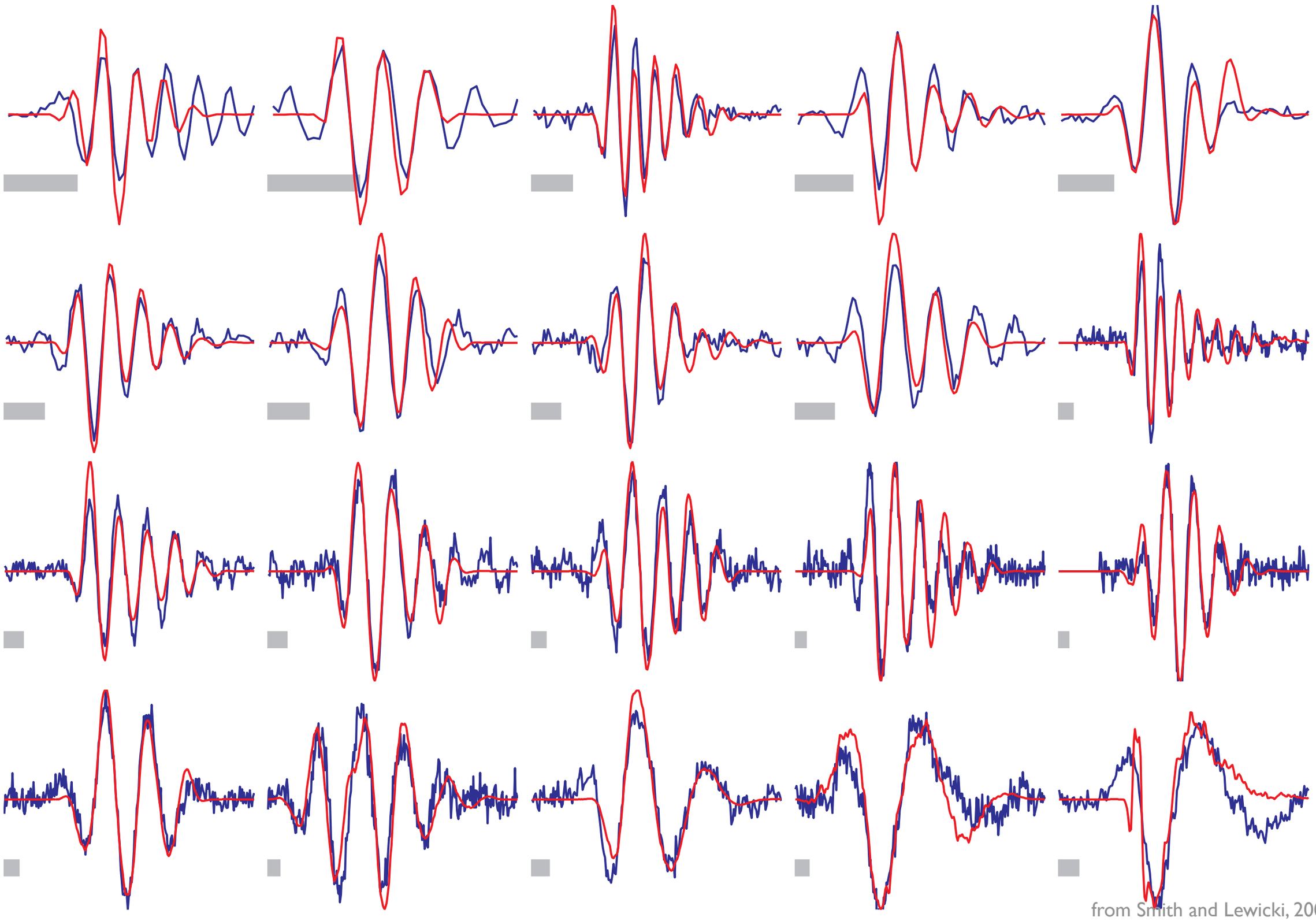
Learned kernels overlaid on selected auditory nerve filters



For almost all learned kernels there is a closely matching auditory nerve filter.

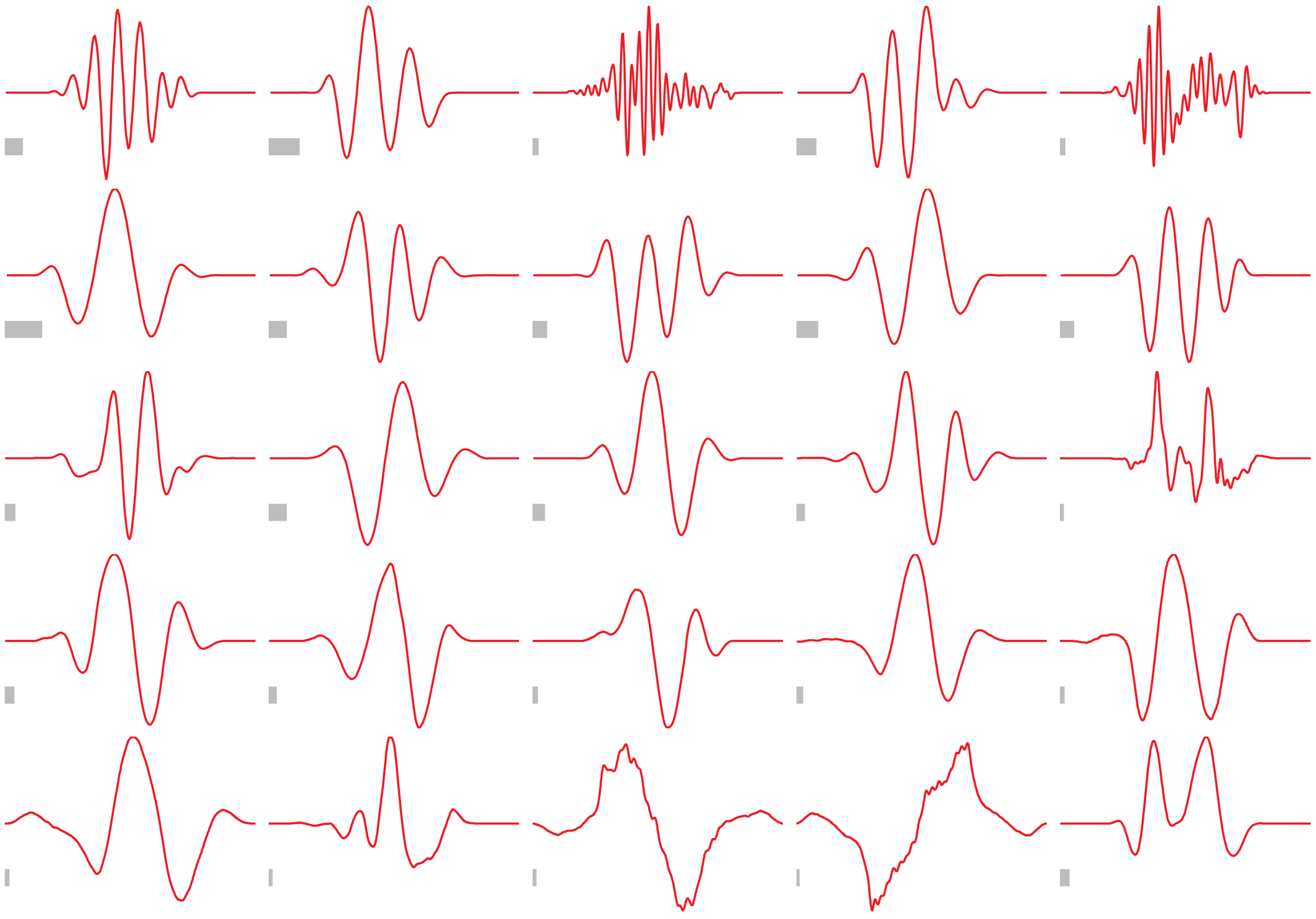
from Smith and Lewicki, 2005

Spike kernels for natural sound mix matches revcor filters

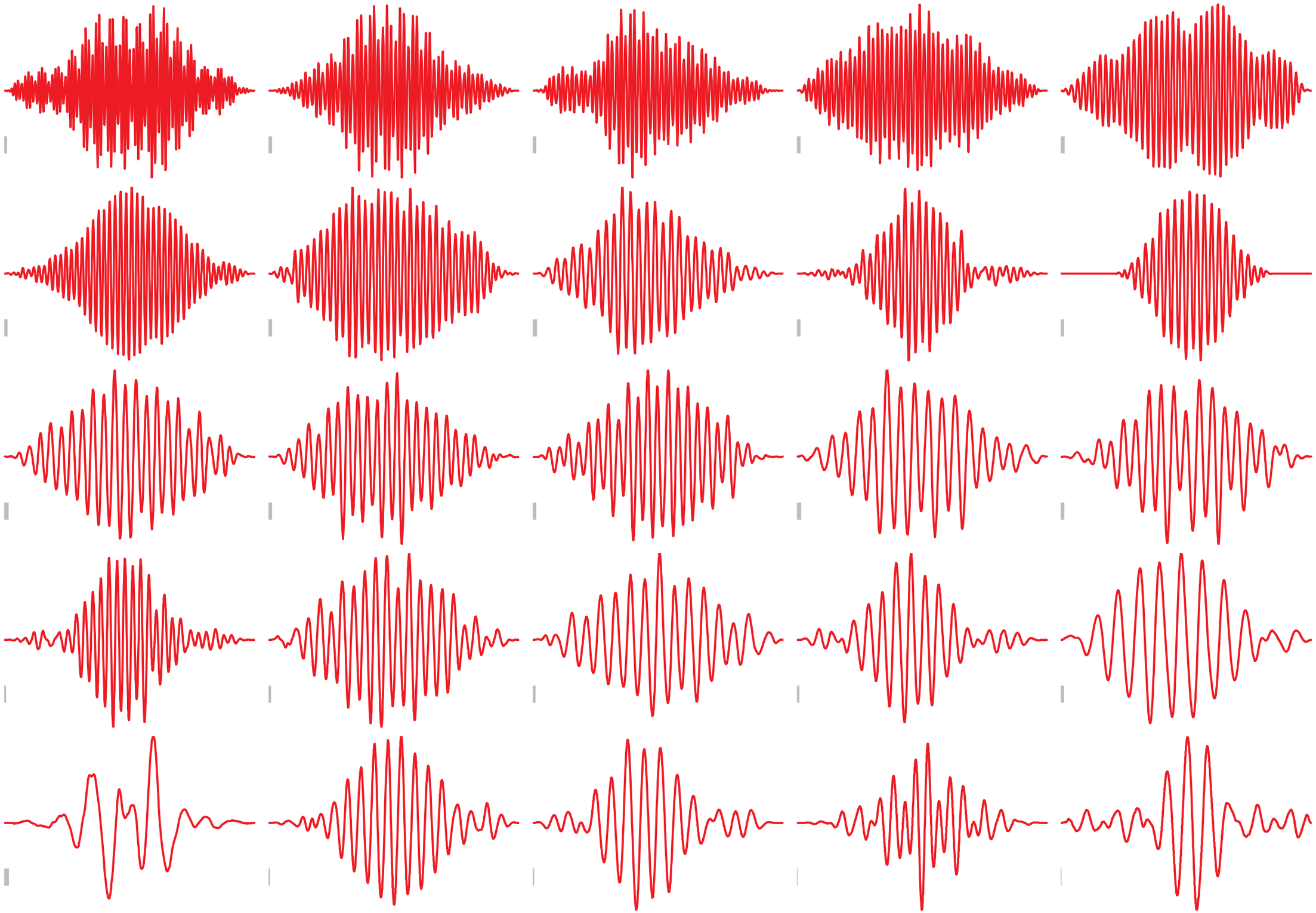


from Smith and Lewicki, 2005

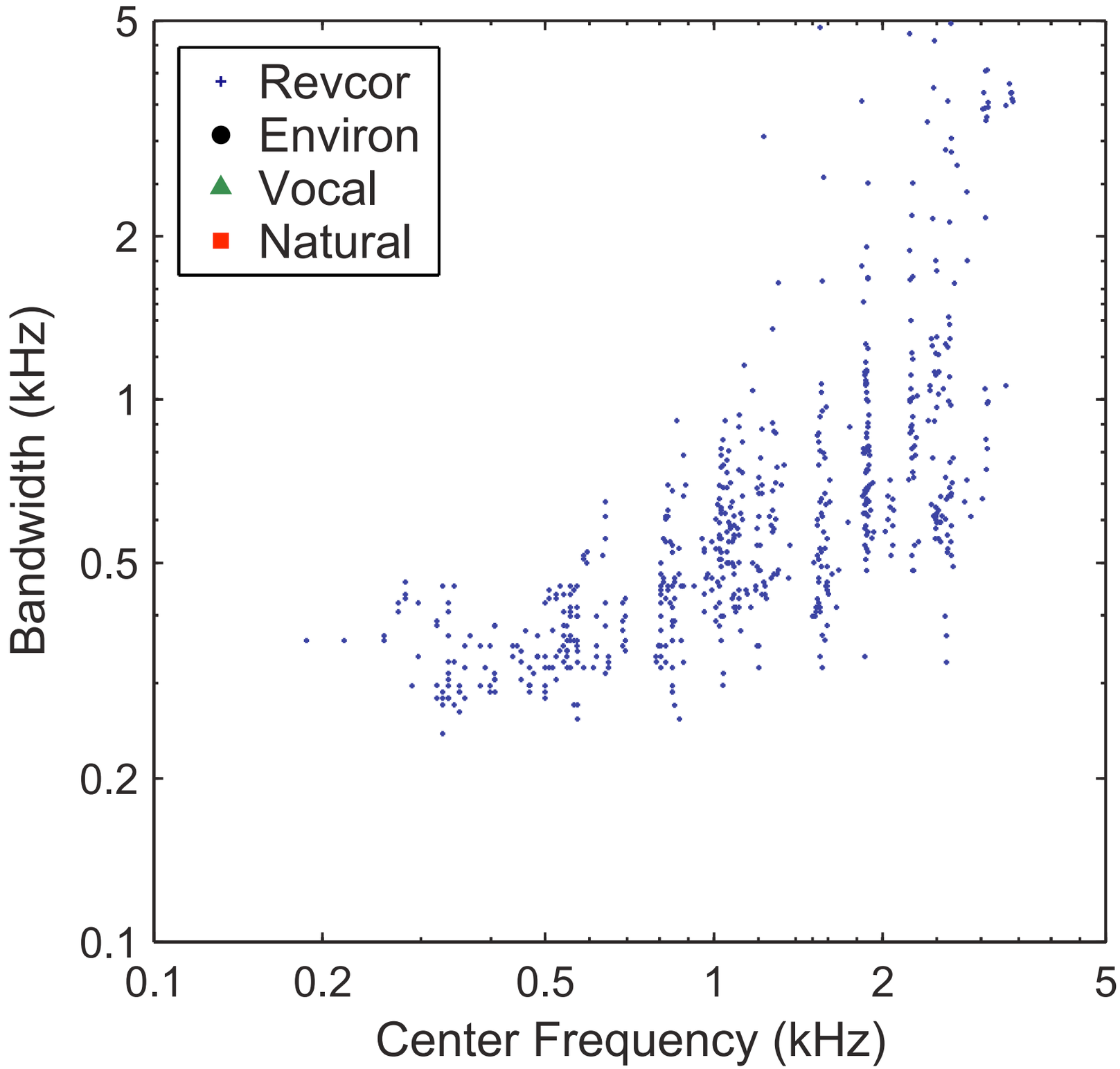
Optimal kernels for environmental sounds are very short



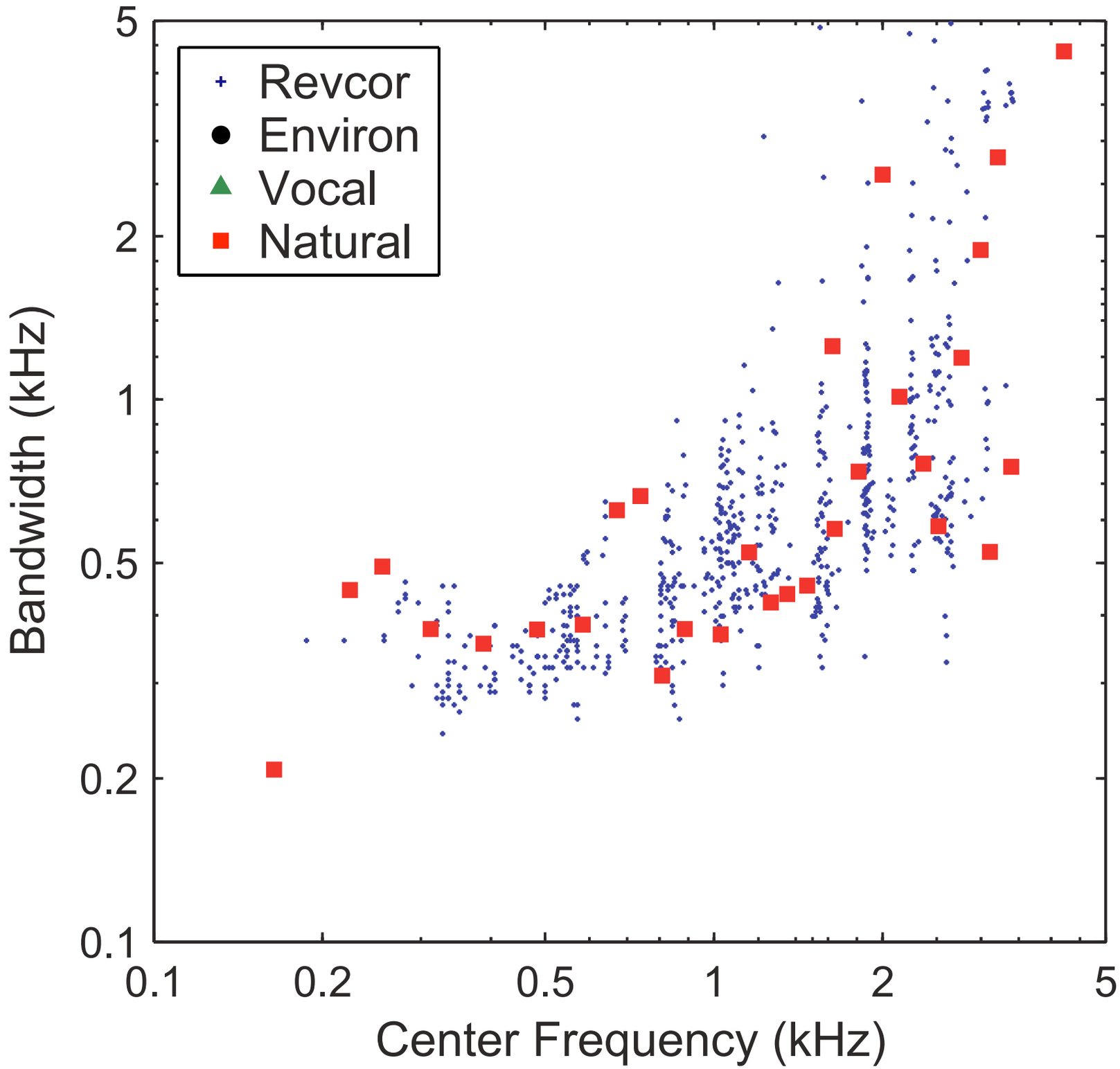
Spike kernels for vocalizations are much longer and symmetric



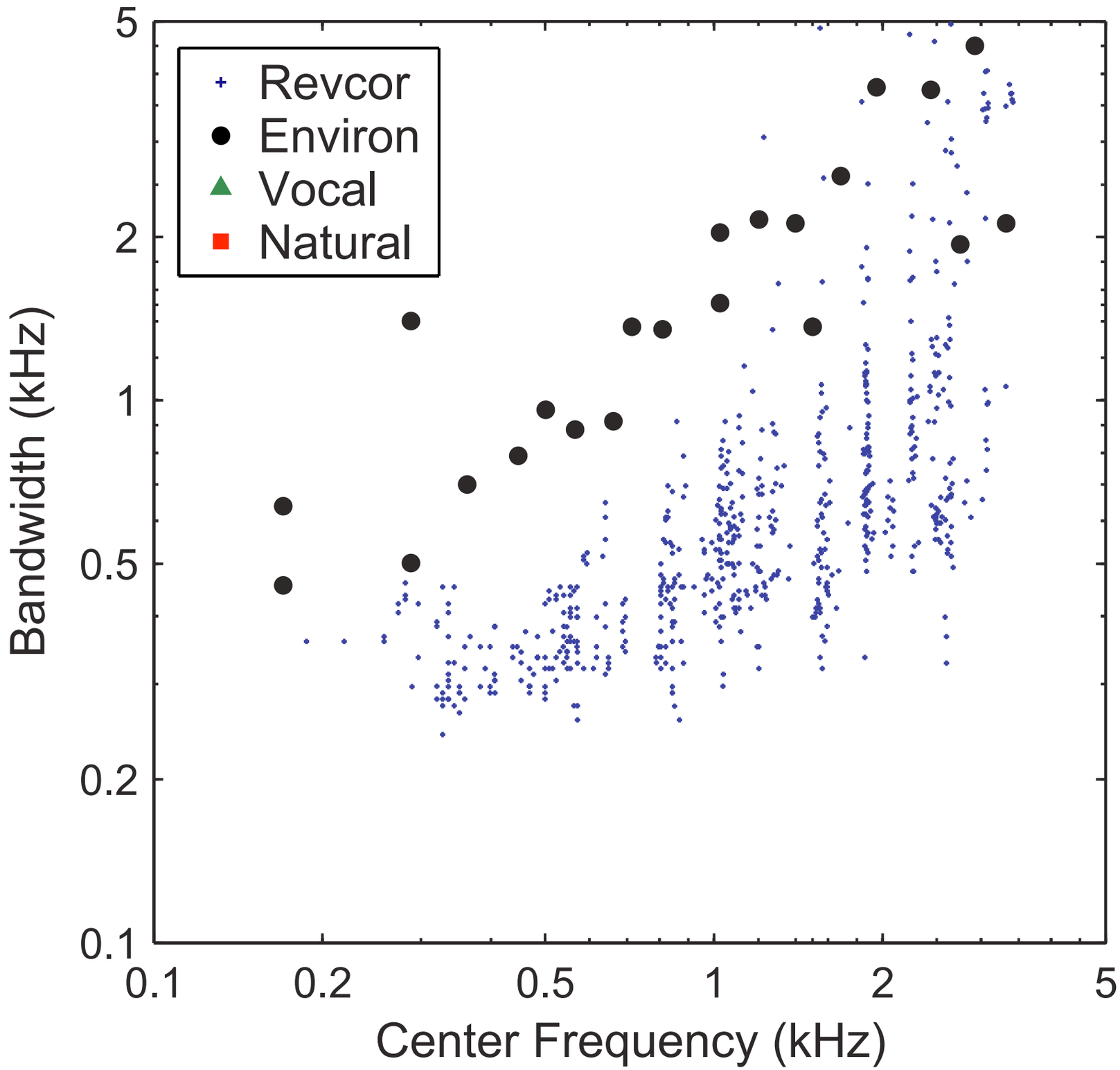
Comparing learned kernels to auditory nerve population



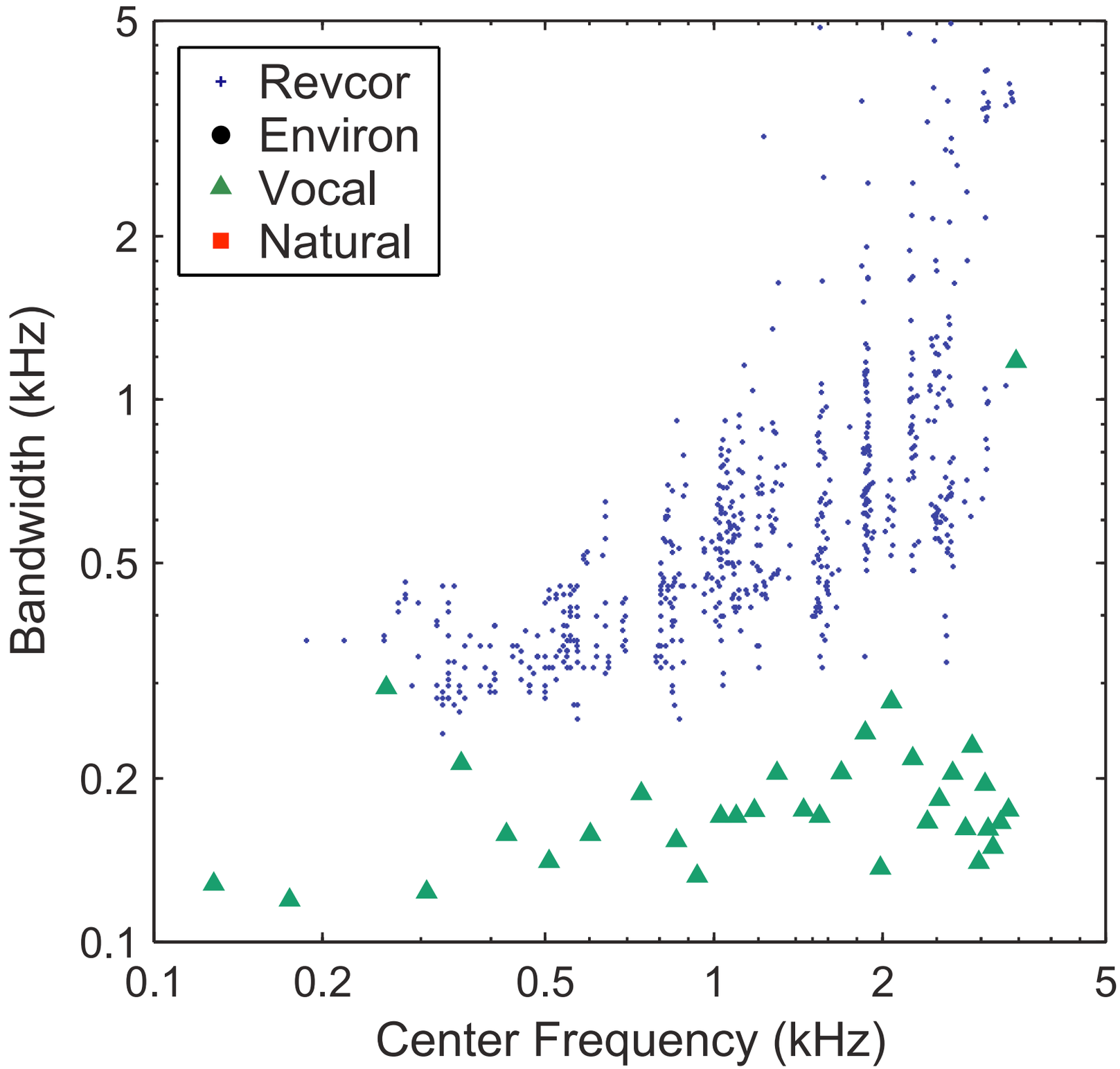
Population distribution of kernels for natural sounds



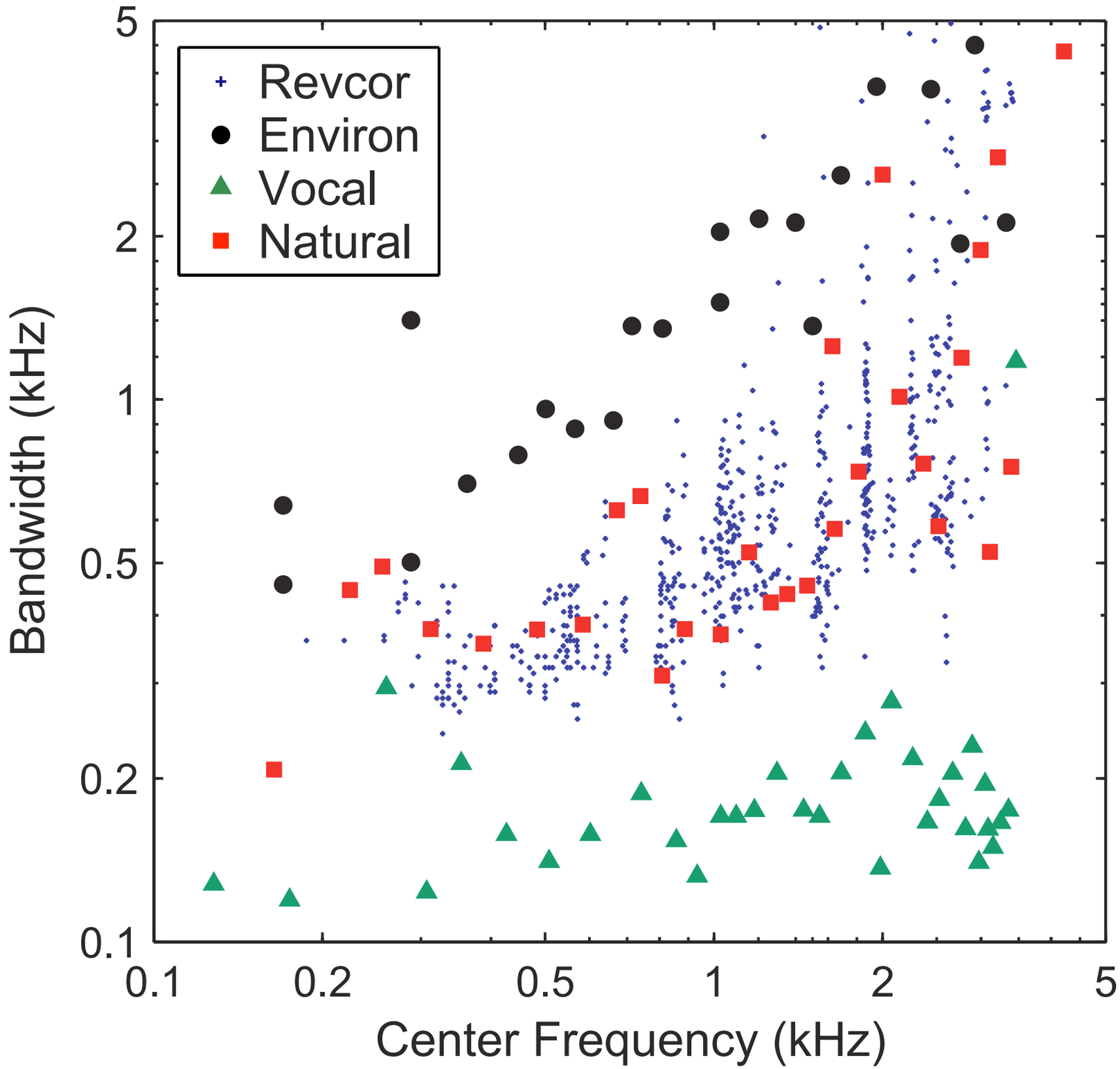
Population distribution of kernels for environmental sounds



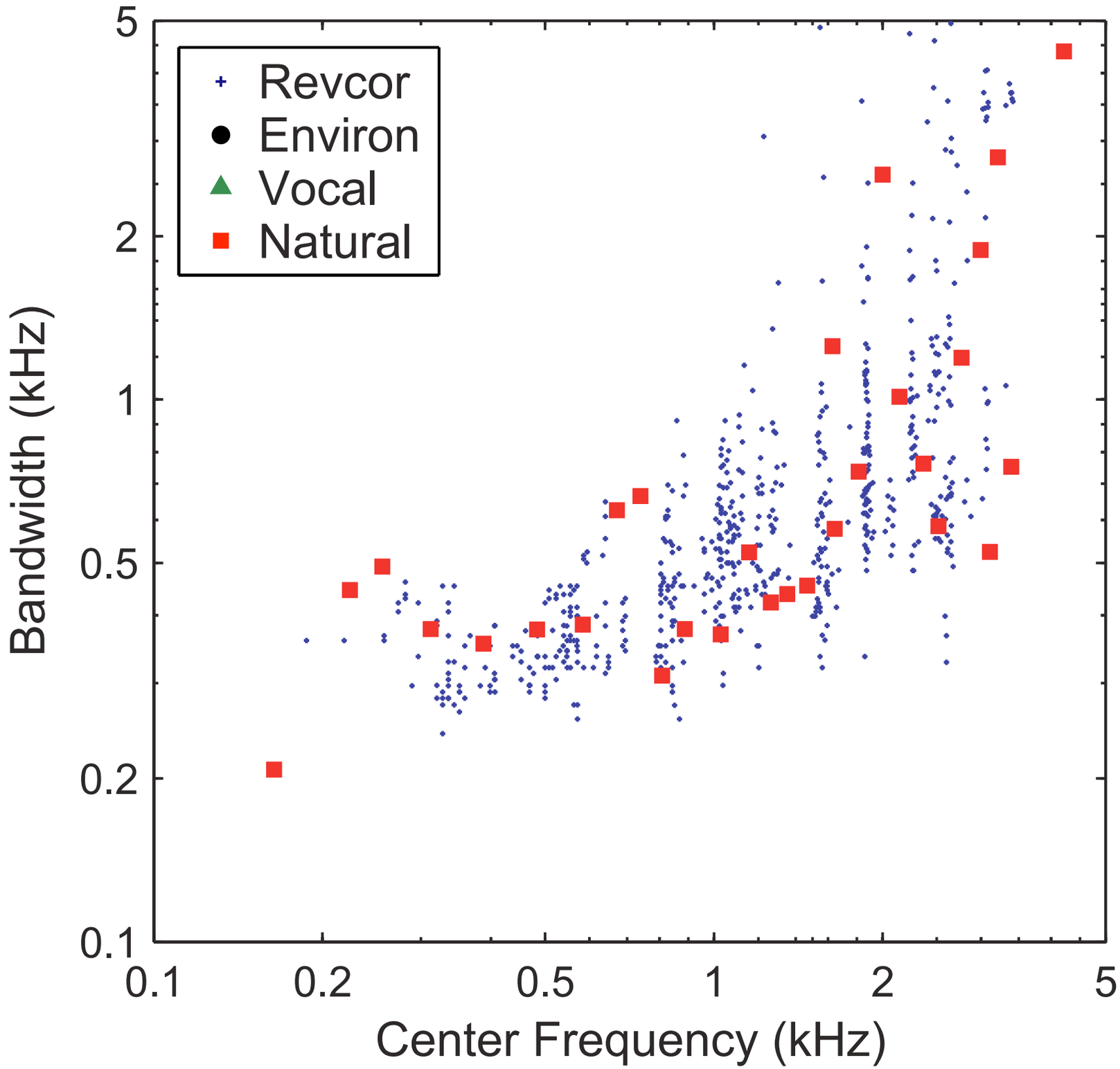
Population distribution of kernels for animal vocalizations



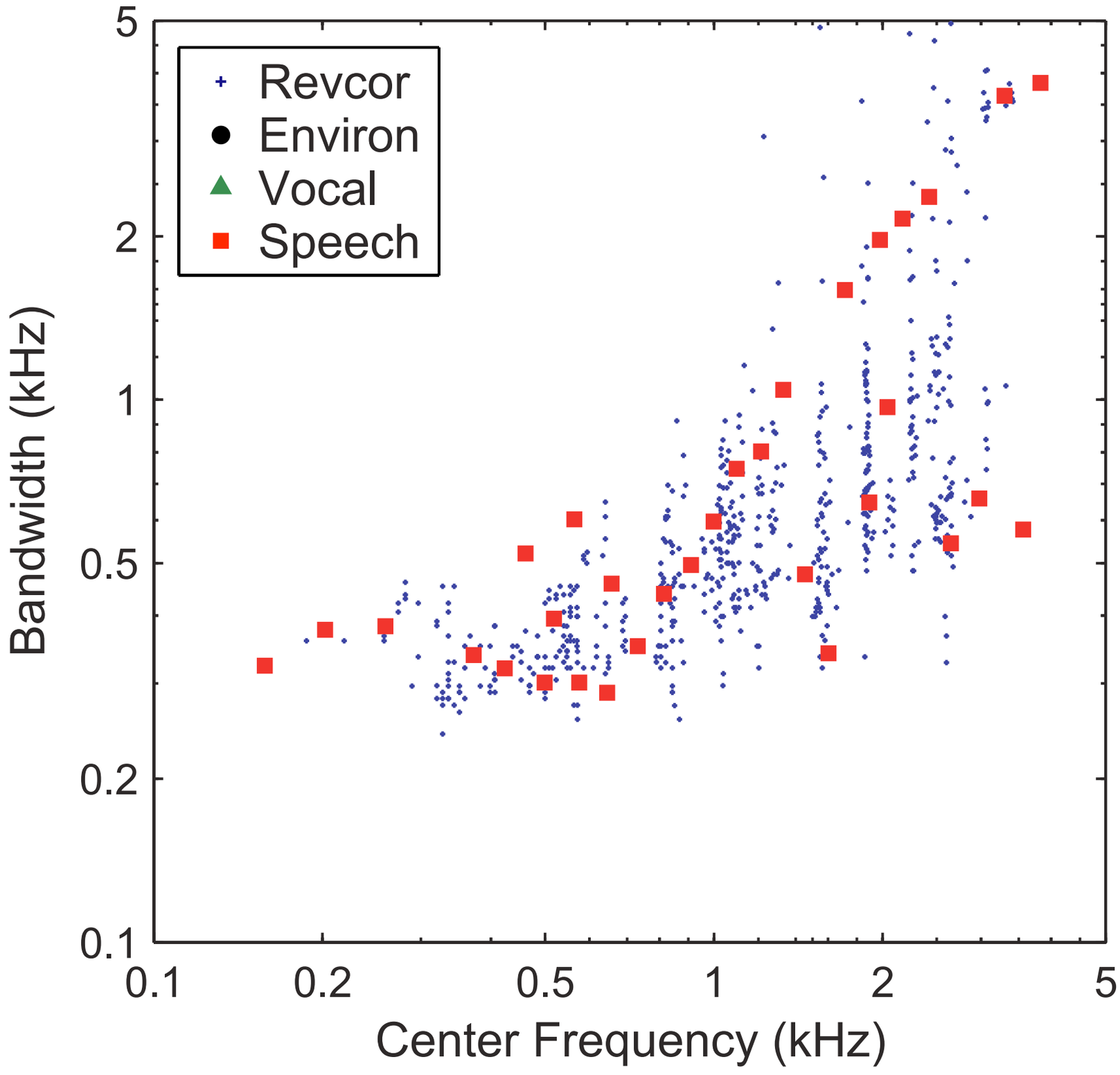
Kernel distributions for different sound ensembles



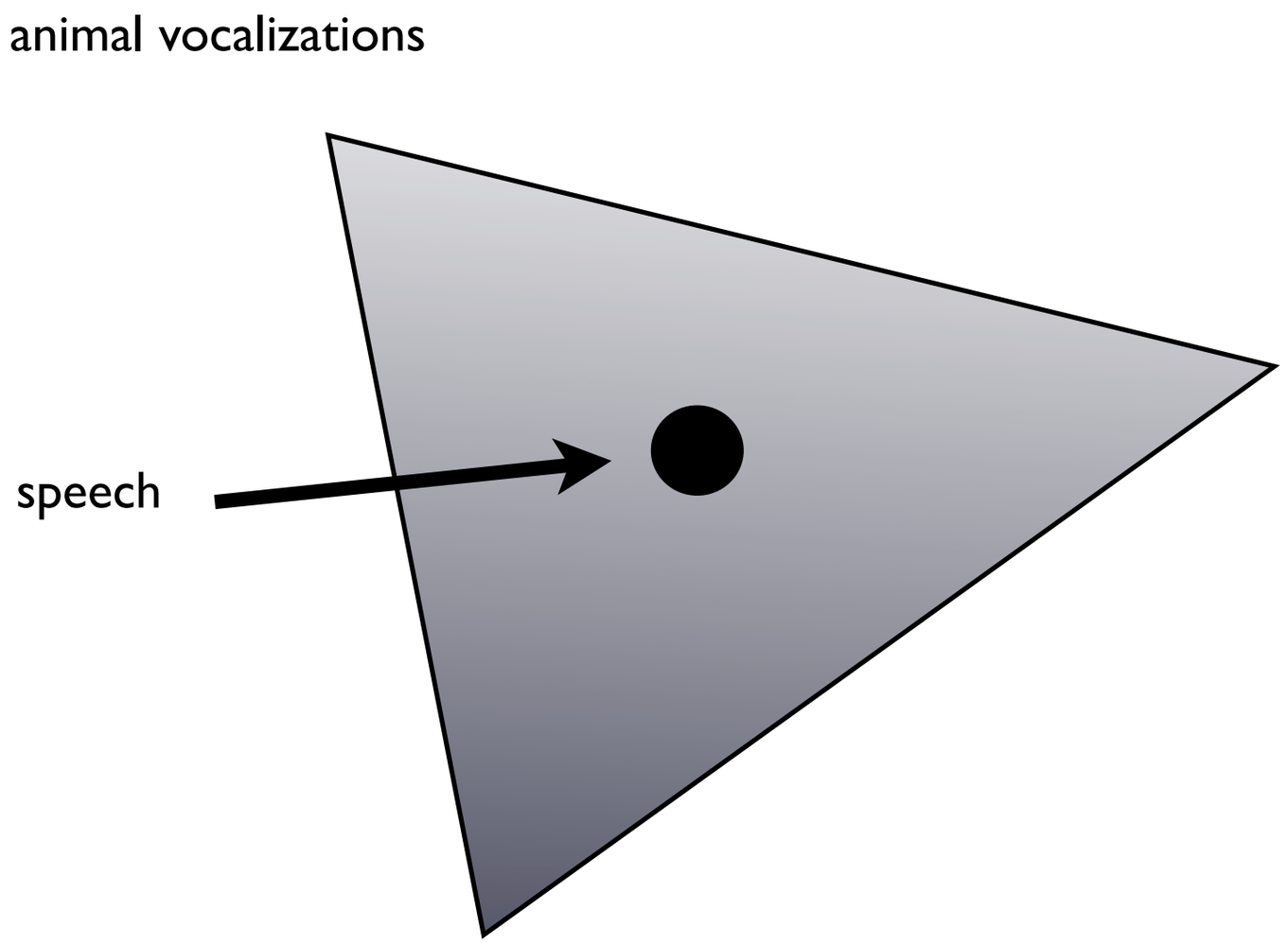
Population distribution of kernels for natural sounds



Population distribution of kernels for speech (TIMIT)

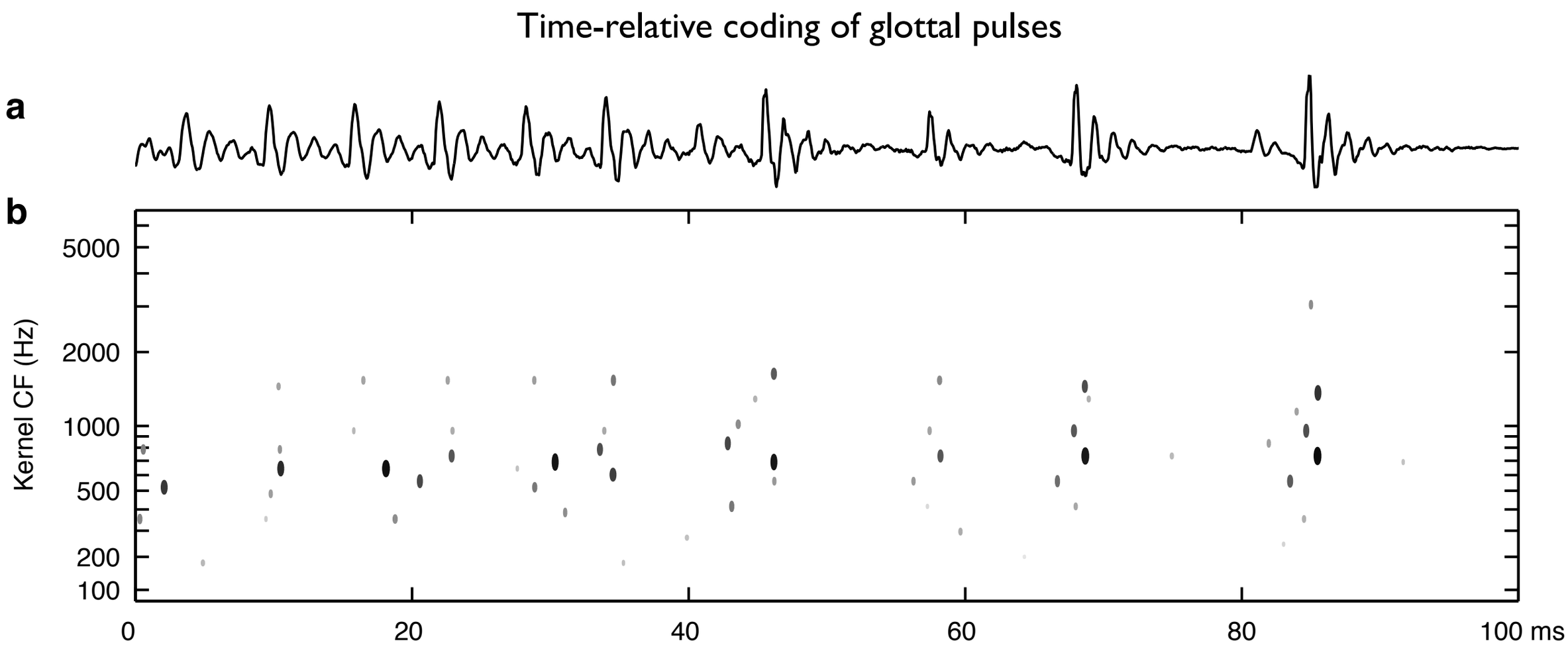


Speech matches composition of natural sounds



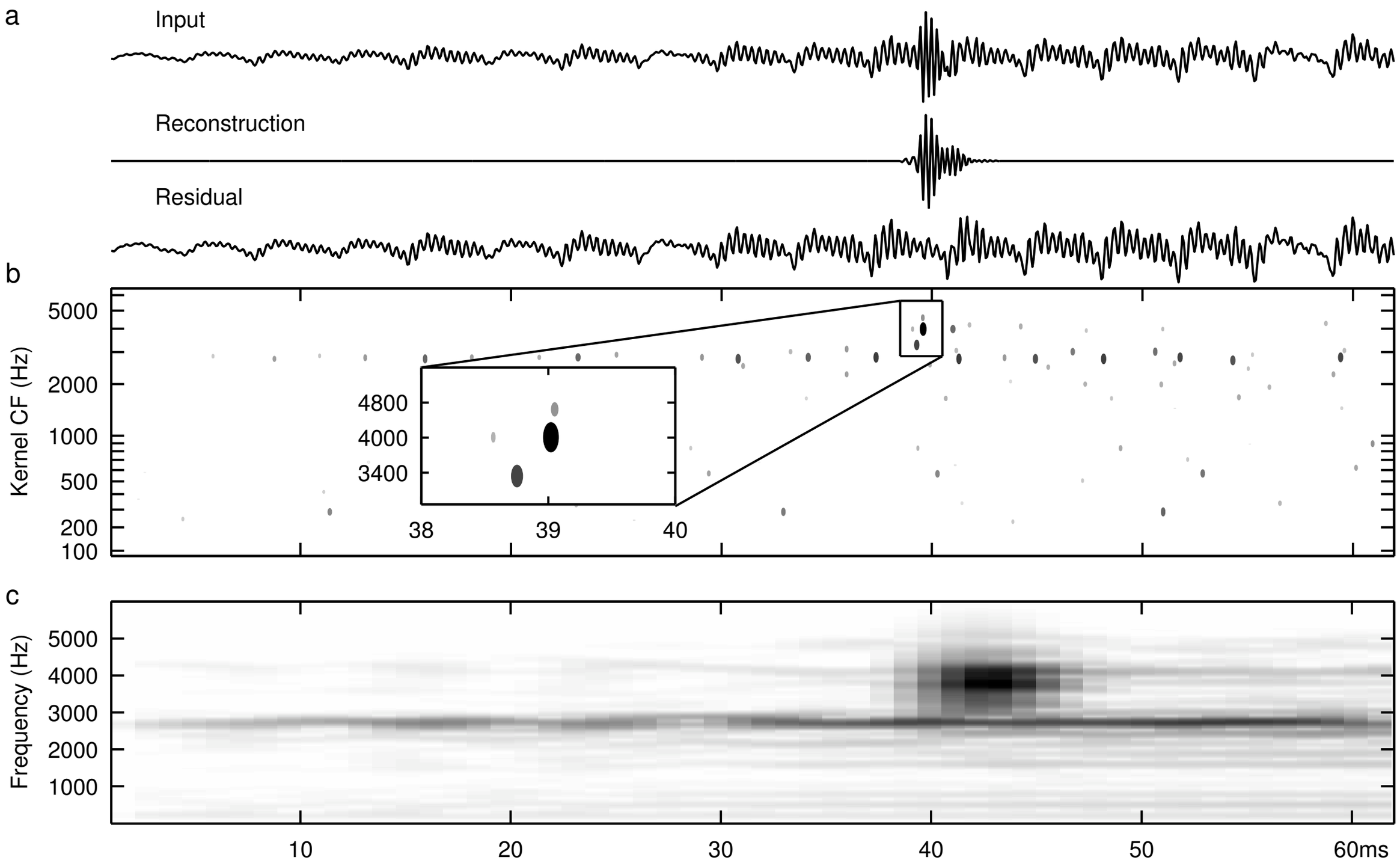
Best mix for predicting auditory coding: 1.0 : 0.8 : 1.2

How is this achieving an efficient, time-relative code?



from Smith and Lewicki, 2005

Coding of a speech consonant



from Smith and Lewicki, 2005