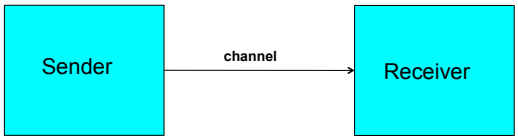
Every Spike is Sacred

John Harris
harris@cnel.ufl.edu
Computational NeuroEngineering Lab (CNEL)
University of Florida

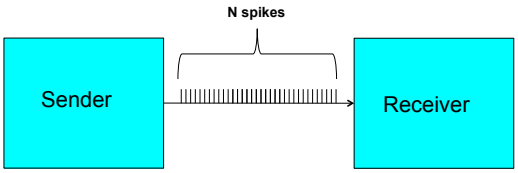
Every spike is sacred.
Every spike is great.
If a spike is wasted,
God gets quite irate.

Start with a thought experiment



What is the fewest spikes needed to transmit a DC value?
Assume value is an integer from 1 to a million

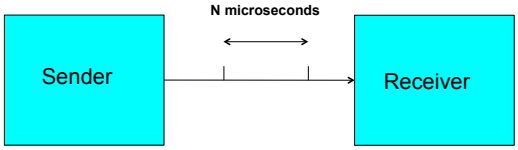
Use Rate Code



Just transmit the number of spikes
Could require up to a million spikes
Very robust but wasteful

If a spike is wasted,
God gets quite irate.

Use Temporal Code



Transmit two spikes and measure the time difference

What happens if we lose a spike?

Every spike is sacred

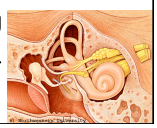
Outline

Part 1: Time-to-first-spike imager

Part 2: Neural recording implant

All animal pictures removed

Part 3: Bio-inspired speech processing

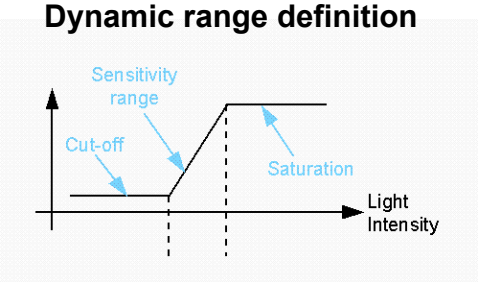


Use engineering performance metrics, always willing to sacrifice biological realism.

Part 1: Time-to-first-spike imager

Dynamic range quantifies the ability to image bright and dark areas simultaneously.

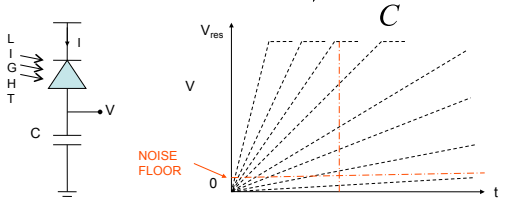
Dynamic range definition



Defined as the ratio of brightest nonsaturated signal to darkest detectable signal.

Dynamic Range Limitation

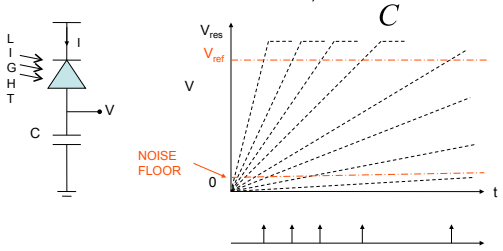
$$V = \frac{I \times t}{C}$$



Dynamic range 60-70dB

Rate Coding Scheme

$$V = \frac{I \times t}{C}$$



Dynamic range extended to ~140dB.

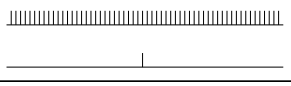
Rate Coding Schemes

Advantages:

- **Each pixel chooses its own integration time to optimize noise and dynamic range**
- **No analog readout noise**
- **No ADC required**
- **Firing rate imagers are not new (Mahowald 1990, Yang 1994, Boahen 1995, Ralph's octopus etc)**

Disadvantages:

- **Takes too long to form an image (~seconds)**
- **Wasteful in terms of bandwidth (and therefore power)**



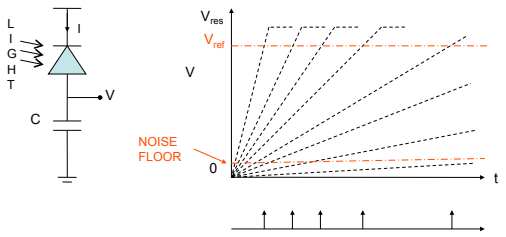
Bright pixel

Dark pixel

Let the heathen spill theirs
On the dusty ground
God shall make them pay for
Each spike that can't be found

Using Time-To-First-Spike Coding

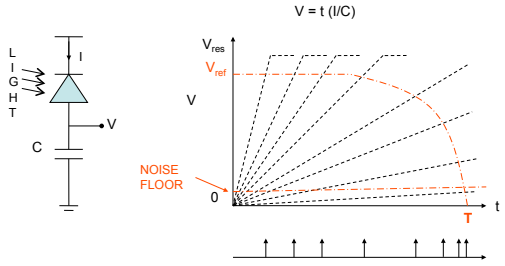
V = t (I/C)



Dynamic range extended to ~140dB.

Only one spike per pixel per frame

Adjusting the Threshold



Dynamic range extended to ~140dB within time T.

Time-to-First-Spike Coding

Advantages:

- Each pixel chooses its own integration time to optimize noise and dynamic range
- No analog readout noise
- No A/D required
- Low power and small bandwidth
- Parallels in biology (Simon Thorpe)

Prototype Chip

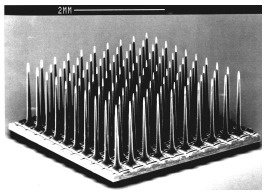
Technology: 0.18*µm TSMC* CMOS
Supply Voltage: 1.8/3.3 Volt
Transistors per pixel: 24
Array size: 128 *x* 128
Pixel size: 12*µm x* 12*µm*
Photosensitive area: 3*µm x* 3*µm*
Power dissipation: 8.7 mW at 30fps
Dark current: 1.04 nA/cm²
SNR: 46.4 dB, measured
FPN: 2.3%, estimated
Dynamic Range: 110 dB (one pixel, measured at 30fps)
Dynamic Range: 100 dB (array, measured), limited by the optics

Sample Image

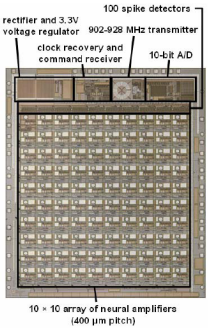
Part 2: Neural Recording Implant

Challenges of tethered Electrophysiological Preparations

State of the Art, in Utah!



Utah Electrode Array



Recording chip by Reid Harrison
100 electrodes, 100 amplifiers
10-bit ADC on one selectable channel, spike detectors on 99 other channels.

Constraints on Bandwidth

- **8 bits/sample x 20 Ksamples/sec = 160 Kbits/sec/channel**
- **Bandwidth constraints allow transmission of only a few channels but want to scale up to 100's of channels**
- **Current solutions can only send one channel full resolution and crudely spike detect on the other channels (no spike sorting possible).**
- **Need compression method but power constraints rule out a DSP solution.**

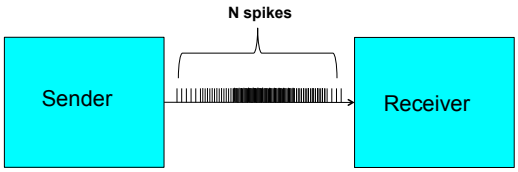
Time for another thought experiment

Another Thought Experiment

What is the fewest spikes needed to transmit x(t)?
Assume x(t) is:

1) **Bandlimited to 10KHz,**
2) **Zero mean**
3) **Reconstructed with 8-bit precision.**

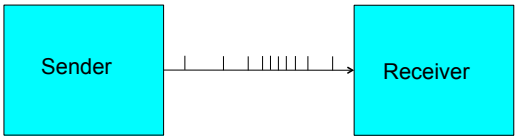
Rate Coding



Transmit a rate coded signal (such that a LPF can recover x(t))
Could require quite a few spikes

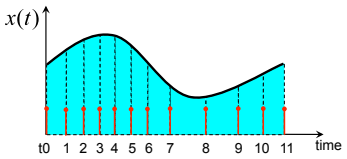
If a spike is wasted,
God gets quite irate.

Temporal Code



Send x(t) into an integrate and fire neuron
And achieve perfect reconstruction on receiver

Sampling With Integrate-and-fire (IF) Neuron Model



Encoding equation

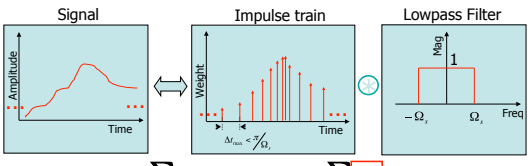
$$\int_{t_i}^{t_{i+1}} x(t)dt = \theta$$

Can we reconstruct x(t) only given the spike timing?

YES!

Signal Reconstruction

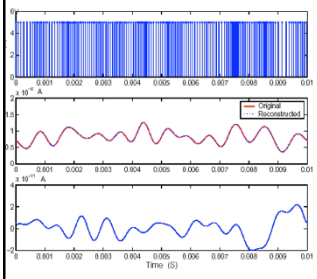
Any bandlimited signal can be expressed as a low-pass filtered version of an appropriately weighted sum of delayed impulse functions. (Derived from Duffin et al. 1952, Feichtinger et al. 1994, Lazar et al.)



$$x(t) = h(t) * \sum_j w_j \delta(t - s_j) = \sum_j w_j h(t - s_j)$$

Where w_j is computed by solving a linear system

Simulation Results (Matlab)



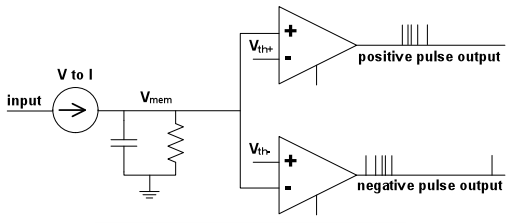
X(t) is a Gaussian random noise signal bandlimited to 1.5kHz

Maximum ISI = 0.14ms < T

SNR = 103dB

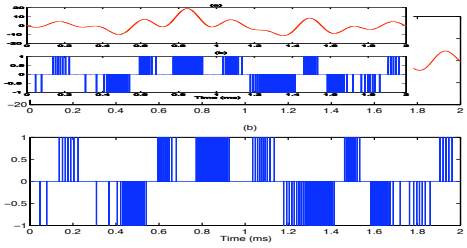
SNR is limited by the finite number of spikes and finite computational precision

Biphasic Schematic



$$\frac{1}{C}\int_{t_i}^{t_{i+1}} x(\Delta)e^{\frac{\Delta - t_{i+1}}{RC}}d\Delta = \theta_i$$

A Synthetic Signal and Its Biphasic Pulse Representation



We have a neuron board here with real-time reconstruction on PC—talk to Manu

IF Advantages

- **Pulses are noise robust and efficiently transmitted at low bandwidth**
- **Front-end is simple and low power**
- **No conventional ADC required**
- **Reduces power, bandwidth and size**

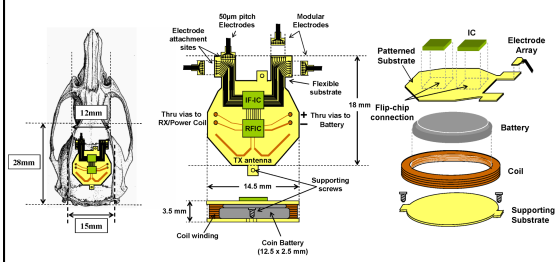
One drawback is that the back-end requires a reconstruction algorithm

Florida Wireless Implantable Recording Electrodes (FWIRE)

FWIRE Faculty

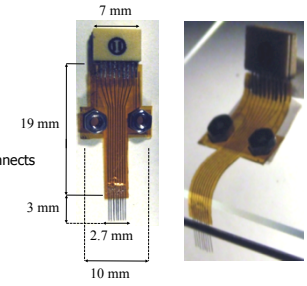
John Harris | **Jose Principe** | **Rizwan Bashirullah** | **Toshi Nishida** | **Justin Sanchez**

Overall FWIRE Schematic

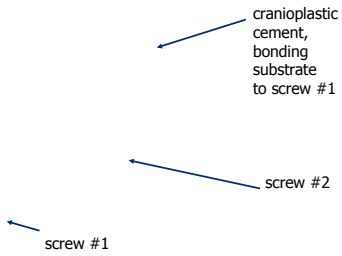


1st Generation Microelectrode

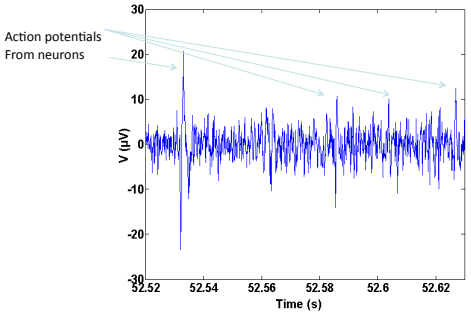
- Polyimide substrate
- Hybrid process
  - 50 um W microwires
  - Micropatterned Au interconnects
  - Omnetics connector



In vivo implantation

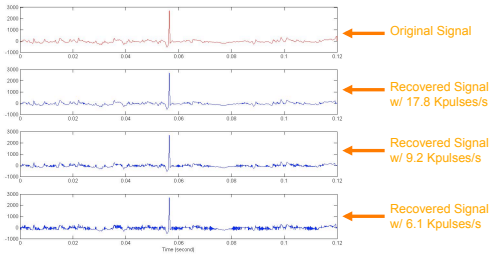


In vivo Recording Results



Can we use spike trains to represent this kind of signal?

Sub-Nyquist Compression

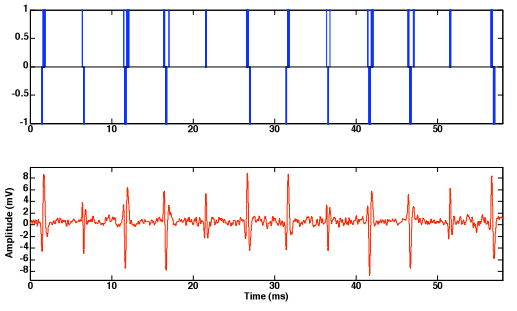


160Kbits/sec → 10Kspikes/sec =16x compression (not enough)

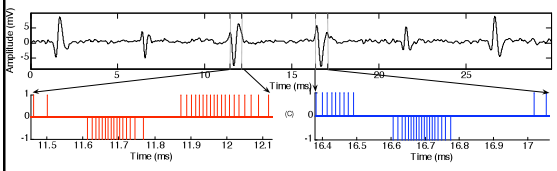
Pulse Based Feature Extraction

- **Realize that the output needs spike sorting**
- **Can achieve further compression with feature extraction**
- **Using exactly the same biphasic circuit, extract pulse-based features**

Pulse-based feature extraction



Feature Extractor Output



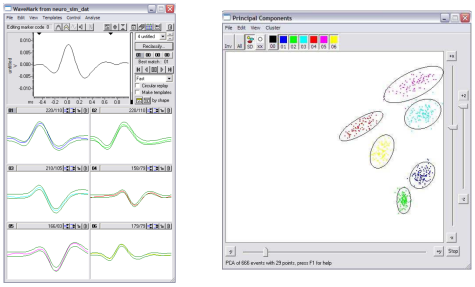
Pulse-Based Spike Sorting

Form template signatures for each neuron

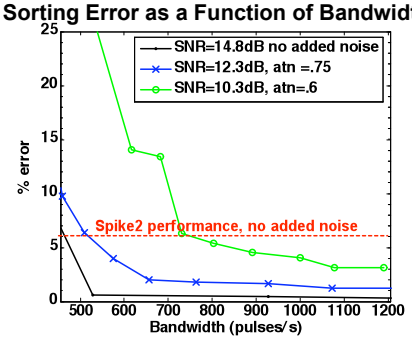
Convolve signatures with Gaussian.
- **σ small - coincidence detector**
- **σ larger - spike count**

Find min MSE and if < noise classify as that neuron

Spike2's Templates

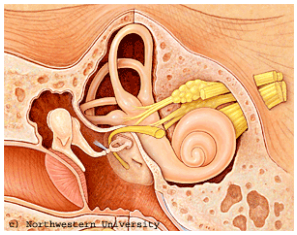


Sorting Error as a Function of Bandwidth



10Kbits/sec → 600 spikes/sec = 16x compression (256x total)

Part 3: Bio-Inspired Speech Processing

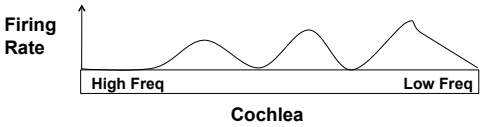


Will show a better view of the cochlea

How to process the spikes?

Should be easy

Doesn't the firing rate at each point on cochlea determines the energy at the corresponding frequency in the input signal (place coding)?



Place code has some troubles

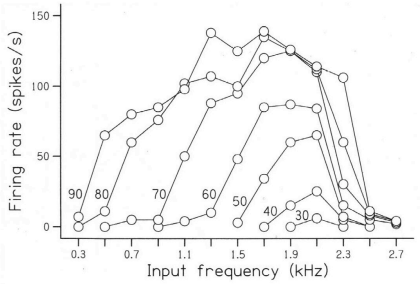
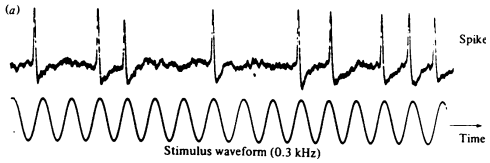


FIGURE 1.16 Isointensity contours for a single fiber in the auditory nerve of an anaesthetized squirrel monkey. The number next to each curve indicates the sound level (dB SPL) used to obtain that curve. Note that the frequency producing maximal firing varies as a function of level. Data from Rose *et al.* (1971).

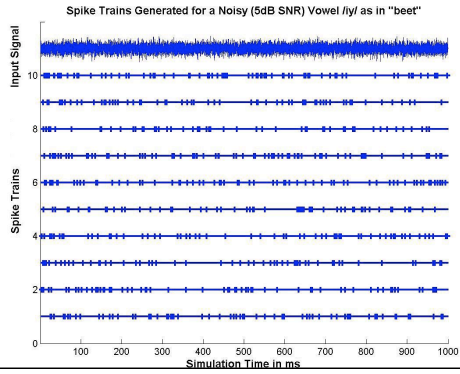
Can Use Phase Locking



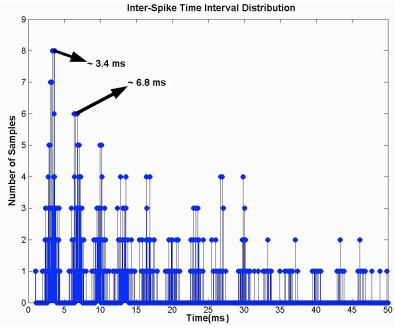
Can extract frequency information from spike timing

Not only should spikes not be wasted, but there is information available in their precise timing

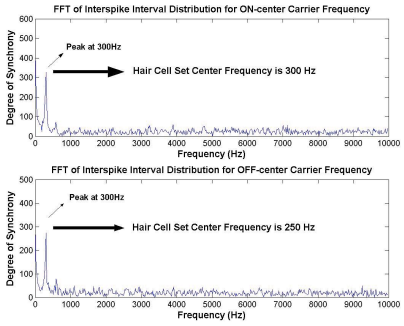
Phase Locking



Inter-Spike Interval Histogram

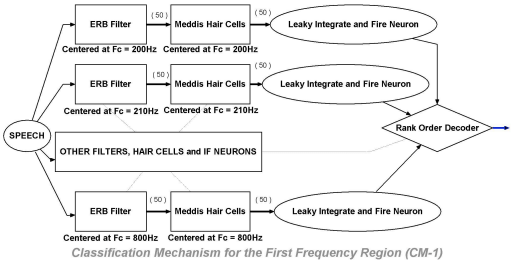


Degree of Phase Locking



16/30

The Overall Algorithm (1)

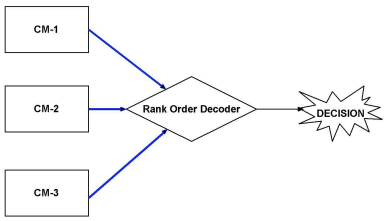


Classification Mechanism for the First Frequency Region (CM-1)

The integrate and fire neuron can compute the degree of phase lock

18/30

The Overall Algorithm (2)

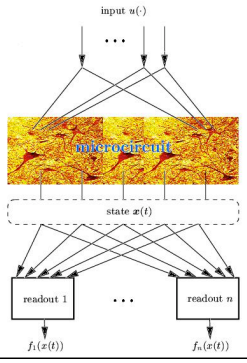


Spike-Based Vowel Classification Algorithm

Revised Algorithm

- Rank order coding is simple and can learn with very few training patterns
- Unfortunately it's performance was poorer than conventional methods (MFCC/HMM)
- Then used Liquid State Machine (LSM), a pattern classifier for spikes

Liquid State Machines



Provides a mechanism for processing spike trains.

Uses a recurrent collection of spiking neurons to form a large dimensional basis set.

A linear readout can be trained to classify the inputs.

Experiment with Vowels

5 vowel classes from the TIMIT database

Two types of noise
- White noise
- Pink noise

Training => Two vowels for each class, one male and one female

Testing => 25 vowels randomly chosen for each class, multi-speaker

Baseline classifier => 39 dimension MFCC with standard HMM

23/30

Final Results (% correct)

	SNR		
	25dB	10dB	5dB
MFCC-HMM	94.0	88.0	78.5
Phaselock	93.0	92.5	91.0

Results only for pink noise, white noise is similar
Promising results, particularly for low SNR cases

Needs to be extended to:
- more phoneme classes
- more noise types
- word recognition

Can we augment the feature vector?

Conclusions

1. Save bandwidth (and power) by using fewer spikes
2. Use the spikes more efficiently by taking advantage of their timing (if you can)
3. Try to understand what implications all this has to biology

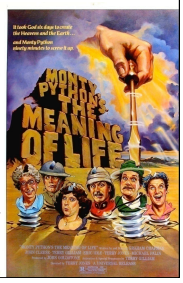
OR

Conclusions

Every spike is sacred

Acknowledgements

NIH/NINDS NS053561
Texas Instruments
My grad students especially
Xin Qi, Xiaochuan Guo, Du Chen, Christy Rogers, Dazhi Wei, and Ismail Uysal
And



Questions?