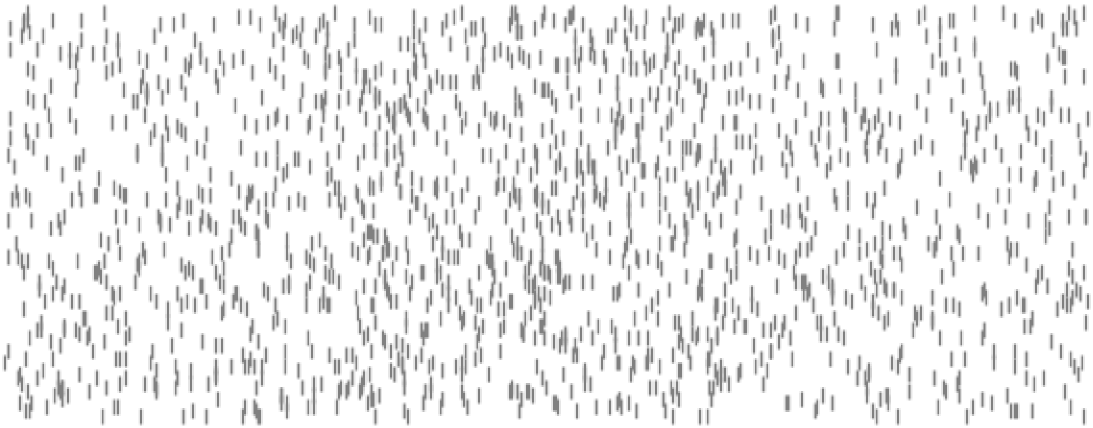


Simulating neural computation and information processing (with *Brian*)

Marcel Stimberg

Institut de la Vision/Sorbonne Université

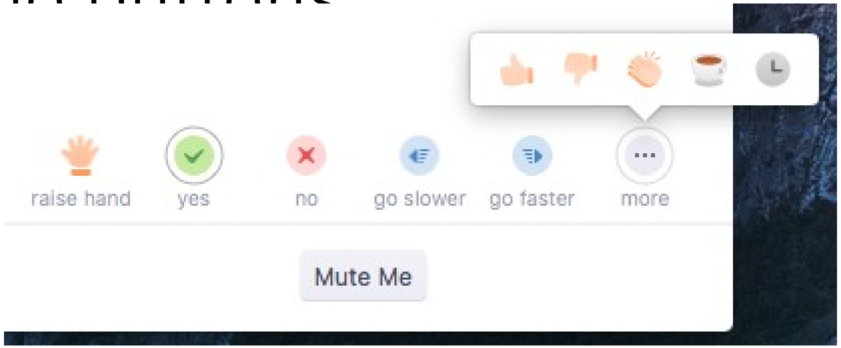
marcel.stimberg@inserm.fr

The jupyter notebooks for the practical introductions to neural simulation can be found at:

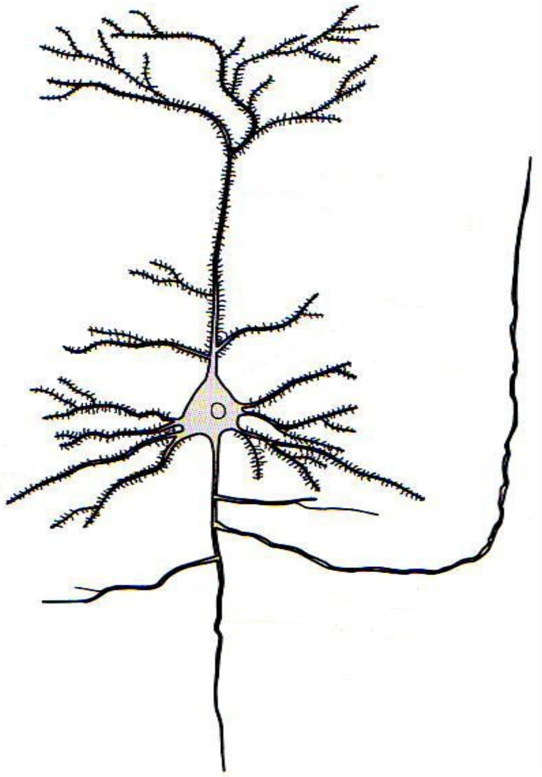
https://github.com/brian-team/brian-material/tree/master/2020-TD-Brian-Sorbonne

Zoom features

- Please keep yourself muted but with camera on (but of course not mandatory)
- Use **raise hand** for questions (but feel free to interrupt me if I don’t see it)
- Please give feedback with yes/no buttons
  I might also ask you to answer question in chat



Plan for today



(More or less) practical introduction to neural modeling

Part 1: neurons

Part 2: networks of neurons

Part 3: case study (binaural sound localisation)

(short break after each part)

Each part:

some slides + practical simulation in Brian

The

BRIAN

simulator

Who is Brian?

- Simulator for spiking neuronal networks, written in Python
- Started by Dan Goodman and Romain Brette at ENS Paris in 2007
- “A simulator should not only save the time of processors, but also the time of scientists”
- Does not provide a library of fixed models but allows for a flexible definition of (almost) arbitrary models
- Focusses on “medium-sized” neuronal networks
  (“a few” to ~100000 neurons), simulations on standard PCs, not supercomputers
- Tool for research and teaching
- Free-and-open-source

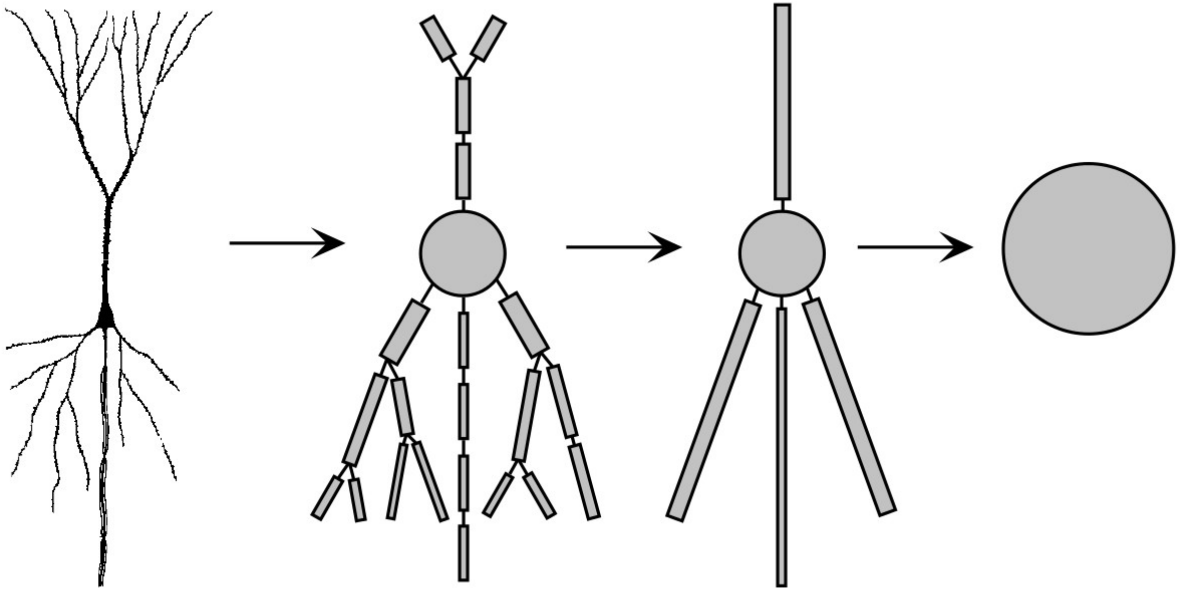
Brian's approach

- *Philosophy*: Mathematical model descriptions
  - Models are defined in mathematical notation
  - Everything is expressed using physical units
- *Technology*: Code generation
  - High-level descriptions transformed into low-level code
  - Modular architecture allows for extensions (e.g. to run code on GPU)

Part 1: **Neurons**

Modelling neurons

Individual elements

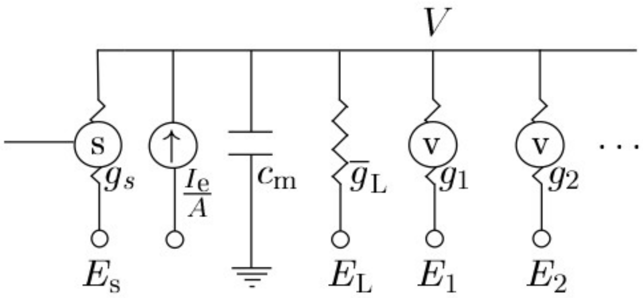


Detailed neuronal morphologies → point-neuron models

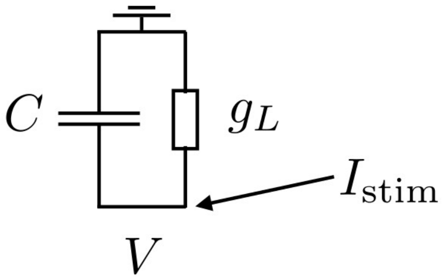
Modelling neurons

Individual elements

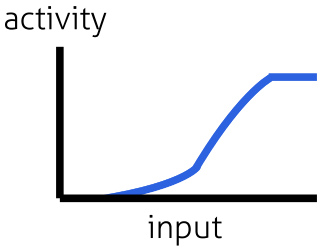
Point-neuron models



Hodgkin-Huxley formalism



integrate-and-fire model

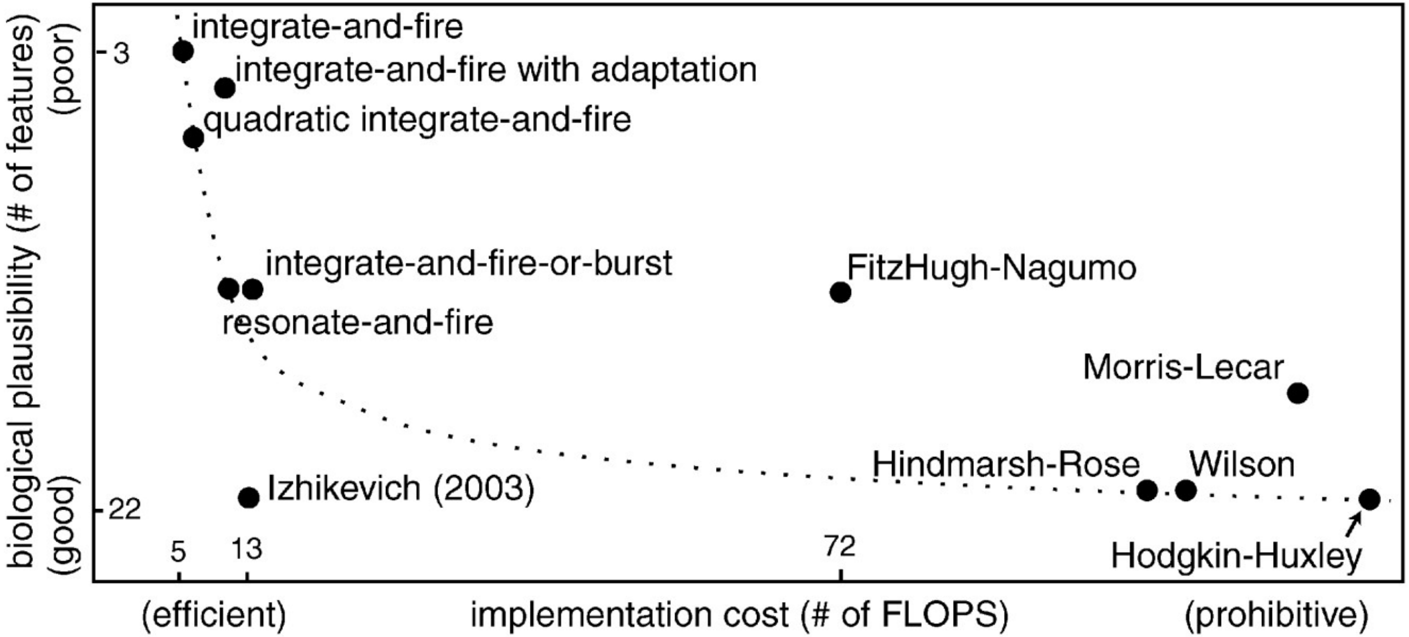


firing rate models

Modelling neurons

Individual elements

Point-neuron models

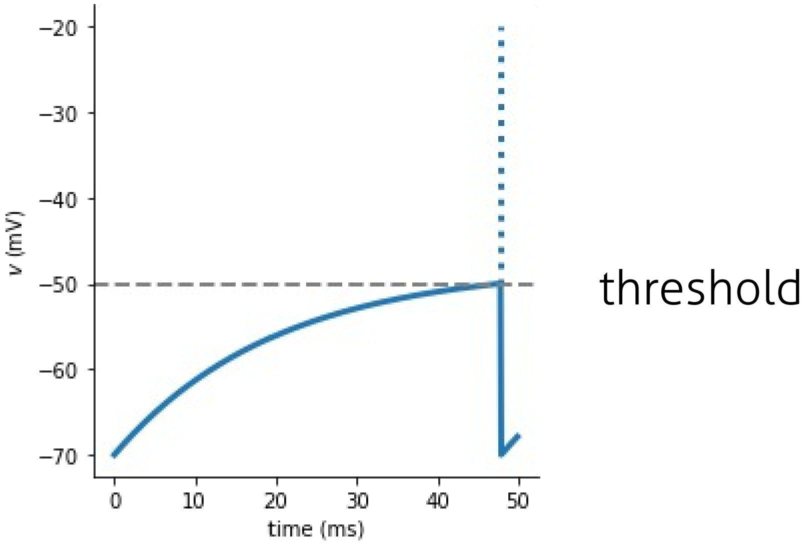


Izhikevich (2004) IEEE Neural Networks

Integrate-and-fire neuron

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = g_L(V_{\mathrm{rest}} - V) + I_{\mathrm{stim}}$$

$$V(t) > V_{\mathrm{threshold}} \quad \rightarrow \quad \mathrm{spike} + V(t) = V_{\mathrm{reset}}$$



Computing with spikes

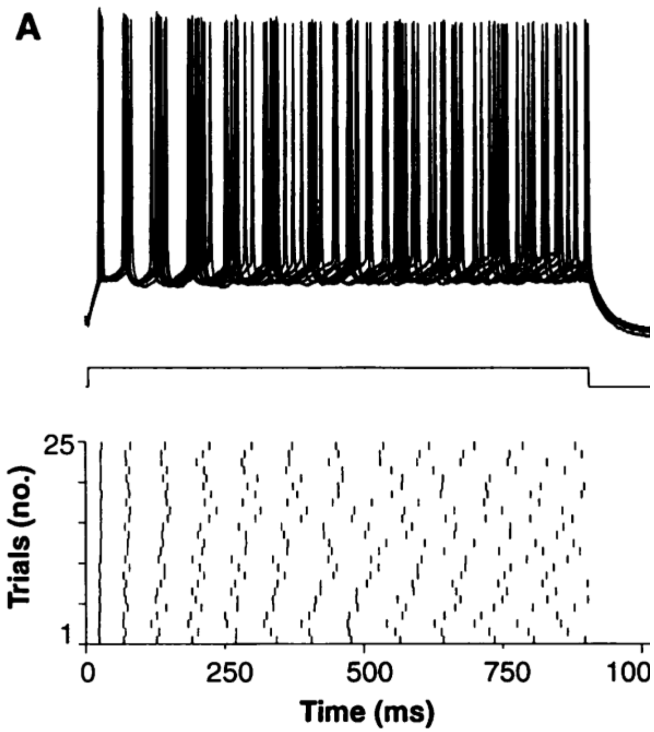
- An argument that is sometimes made:

  "Spike timing in individual neurons is unreliable. Therefore, only the firing rate (averaged over neurons or over time) matters."

- Empirical evidence for "unreliable timing" is unclear.

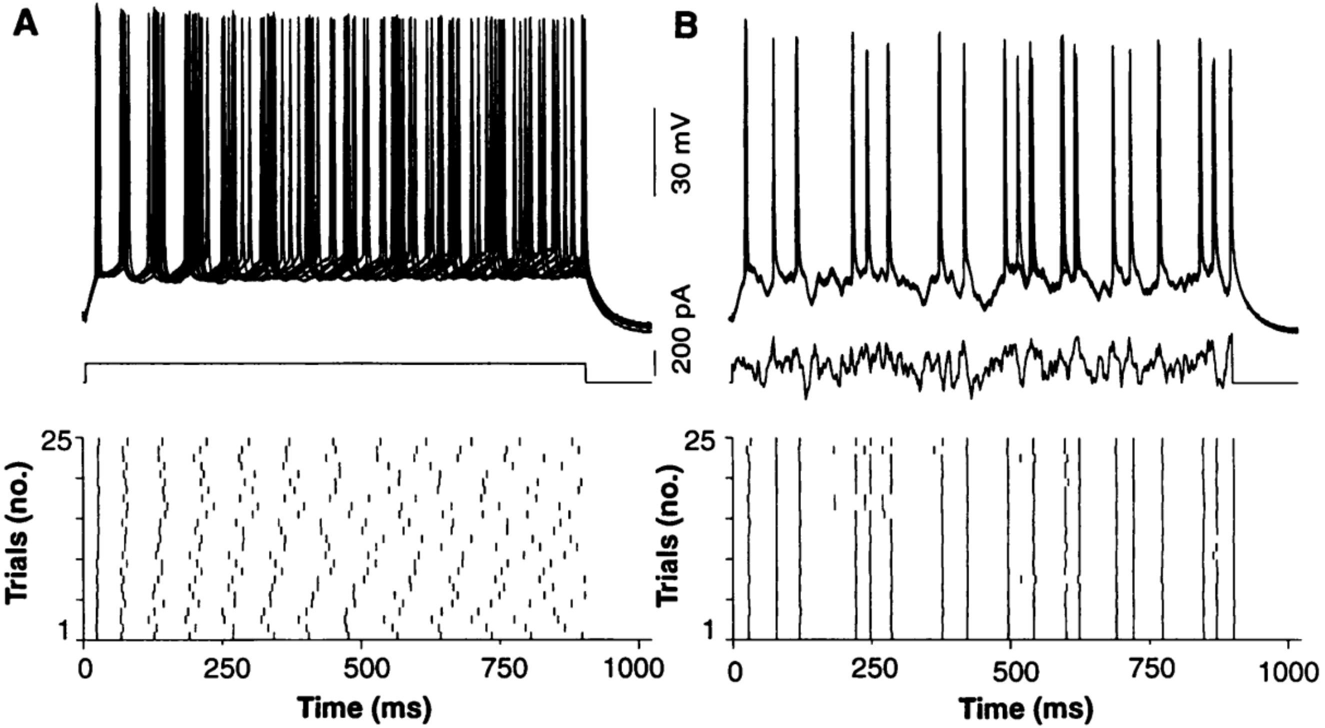
Computing with spikes

Constant current injection = **unreliable** spike times



Computing with spikes

Constant current injection = **unreliable** spike times
Fluctuating current injection = **reliable** spike times



Let's try with

Part 2: **Networks**

Modelling networks of neurons

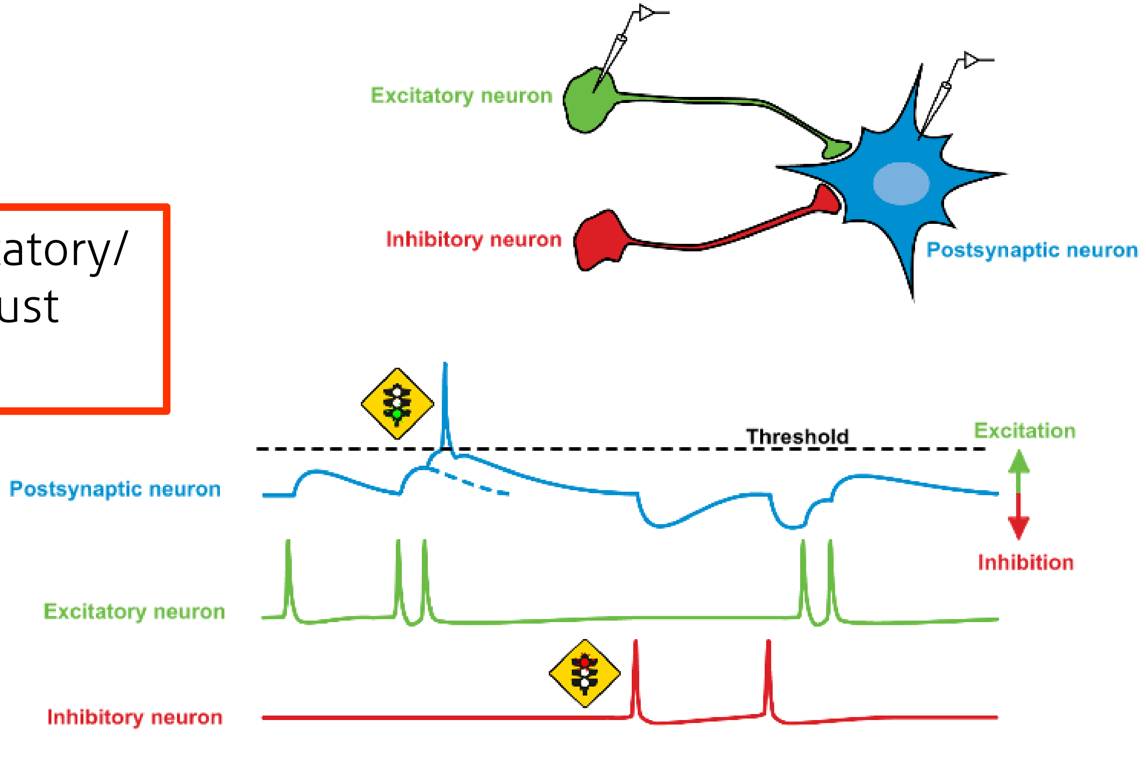
Synapses

?

Why can we talk about excitatory/ inhibitory *neurons* and not just synapses?

!

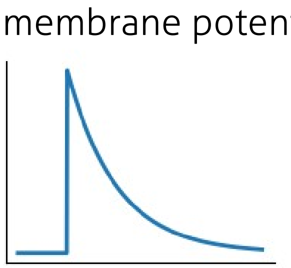
→ "Dale's law"
Neurons release the same neurotransmitter(s) on every synapse



Neuroscience online

Modelling networks of neurons

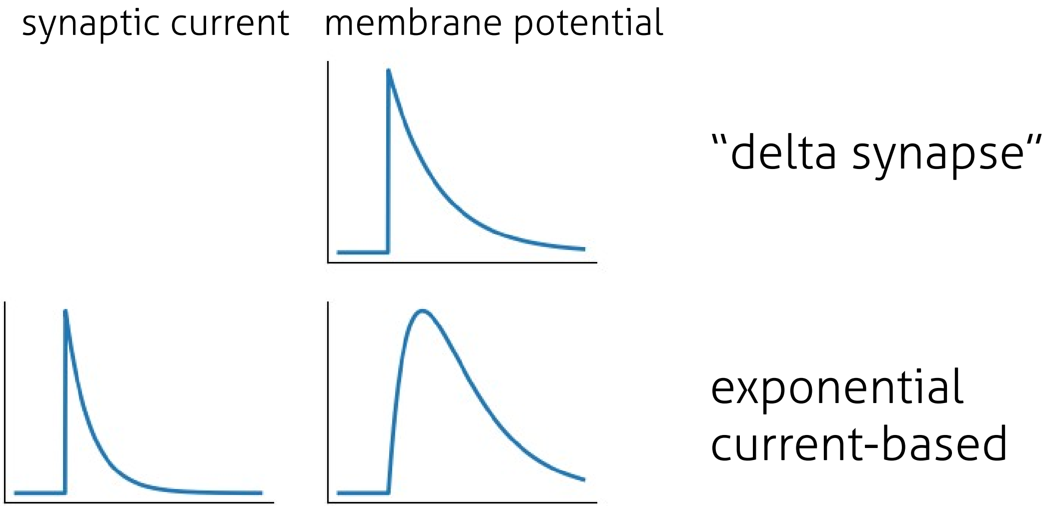
Synapses



"delta synapse"

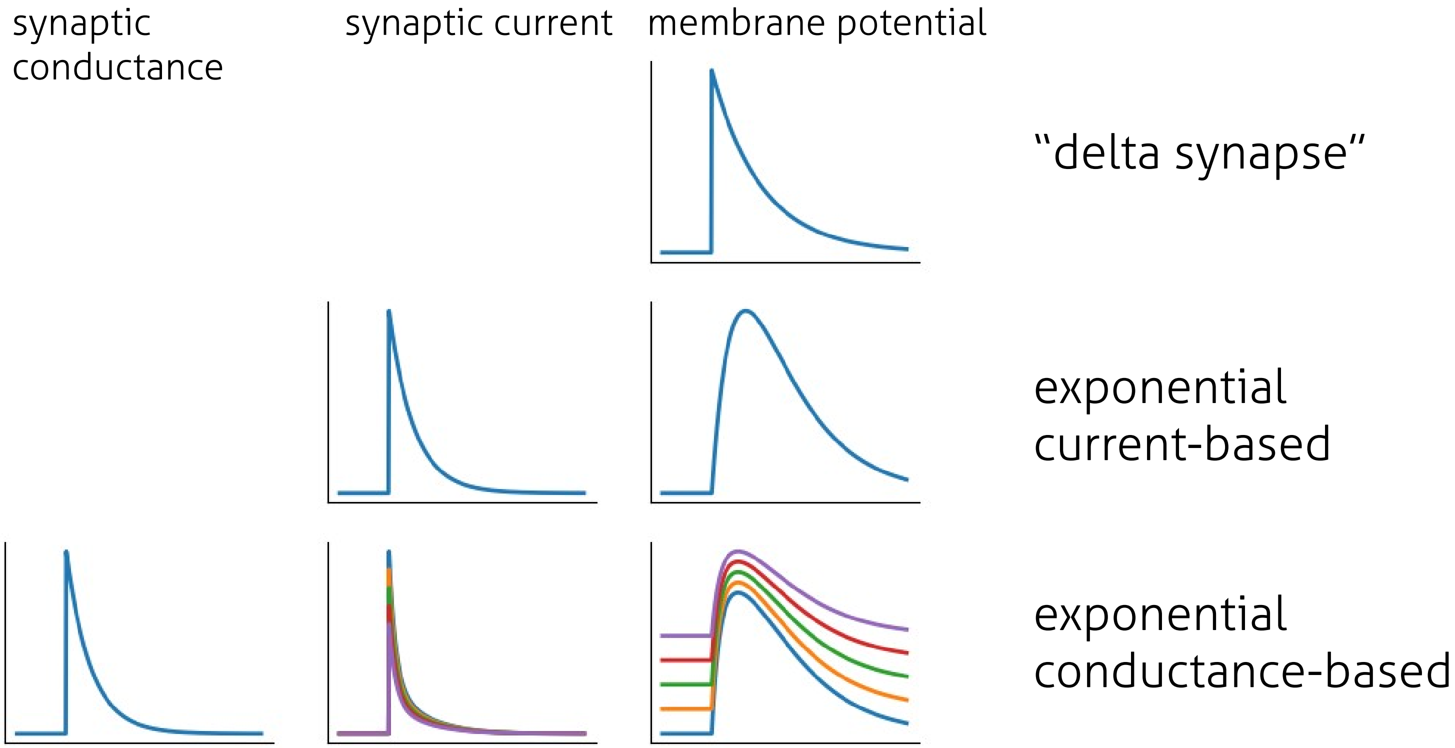
Modelling networks of neurons

Synapses



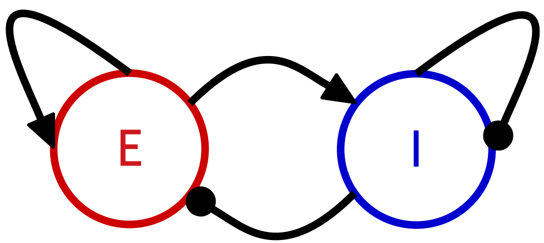
Modelling networks of neurons

Synapses

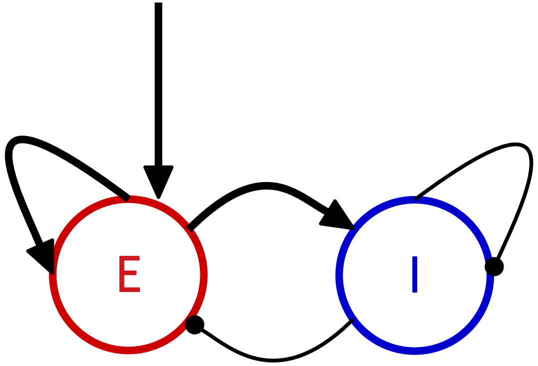


Dynamics in spiking models

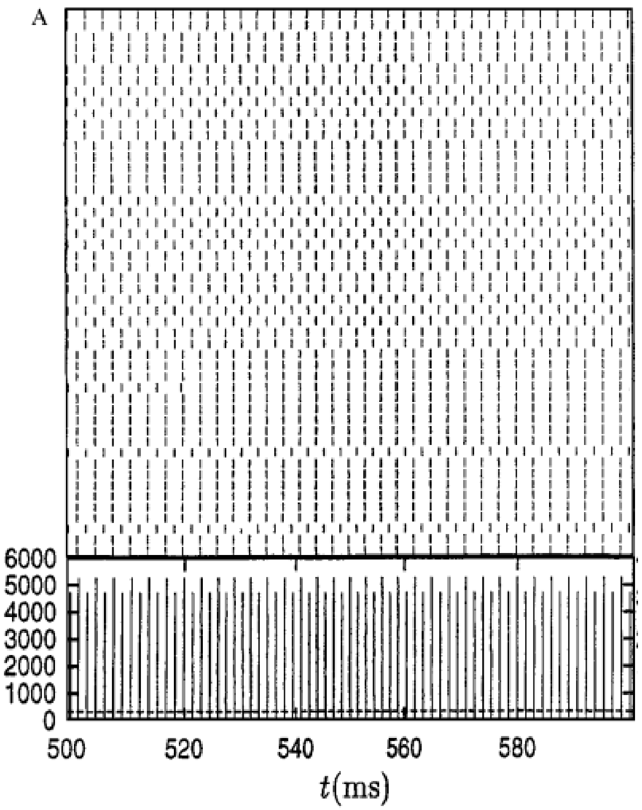
- Randomly connected (often: sparsely) neurons
- excitatory and inhibitory



Activity regimes



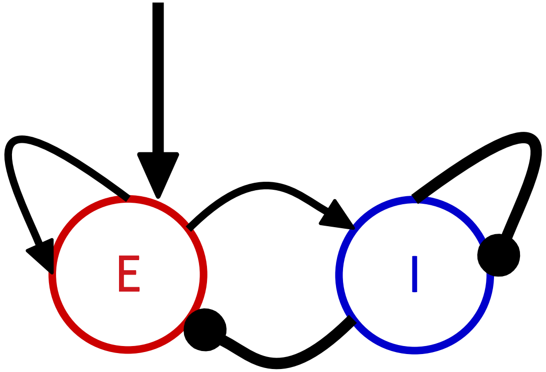
regular firing
global synchronization

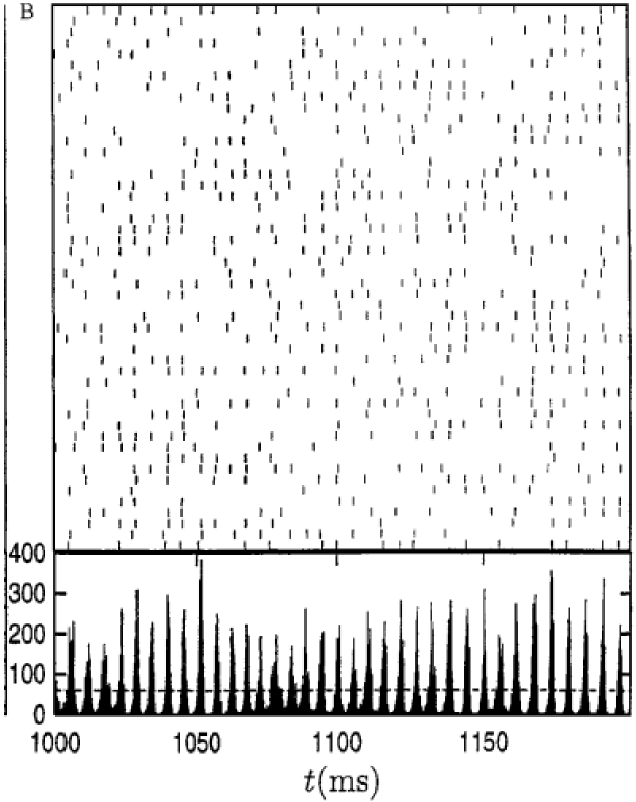


Brunel (2000)

Activity regimes

irregular firing
global synchronization

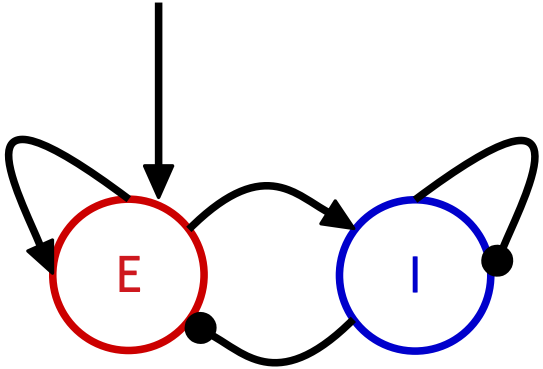


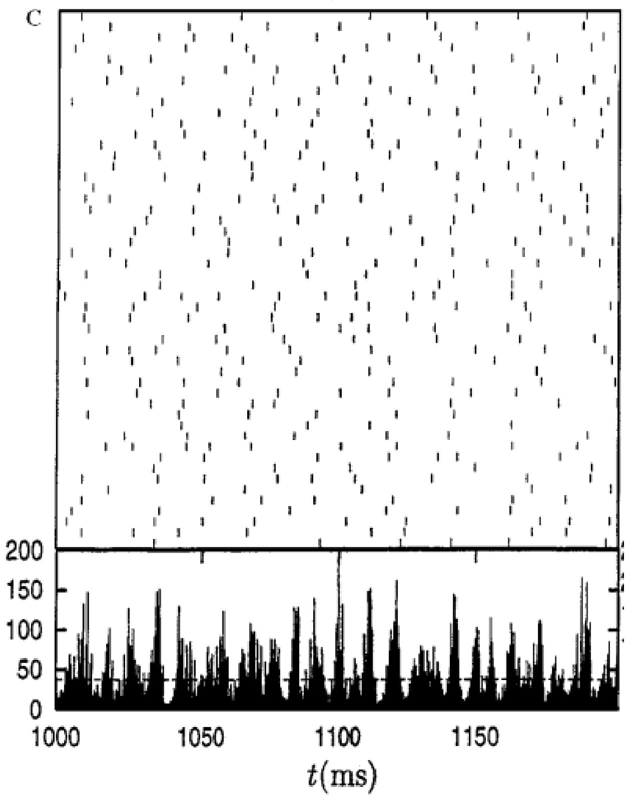


Brunel (2000)

Activity regimes

irregular firing
asynchronous activity

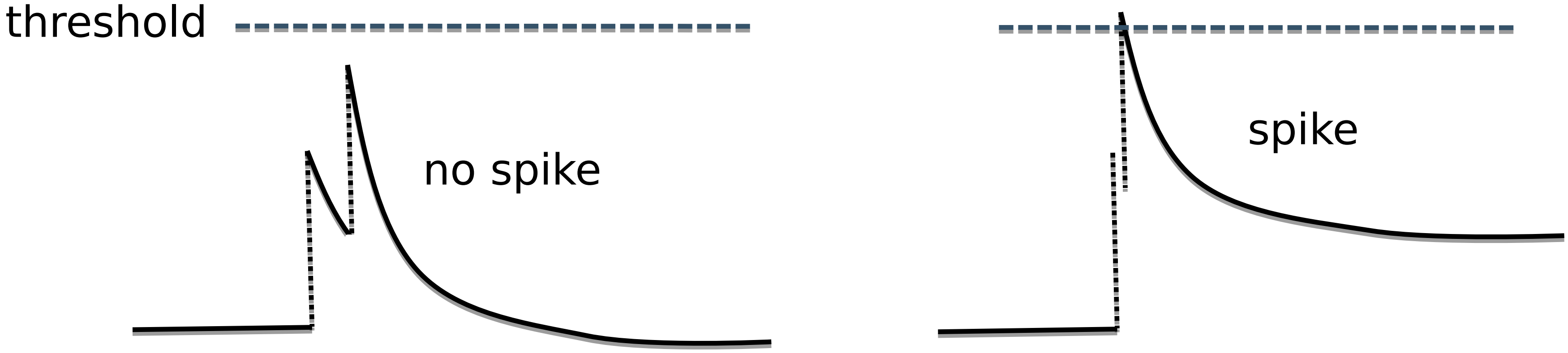




Brunel (2000)

Input integration in neurons

- Neurons sum inputs over space (synapses) and over time
- Synchronous activation is more efficient than asynchronous activation

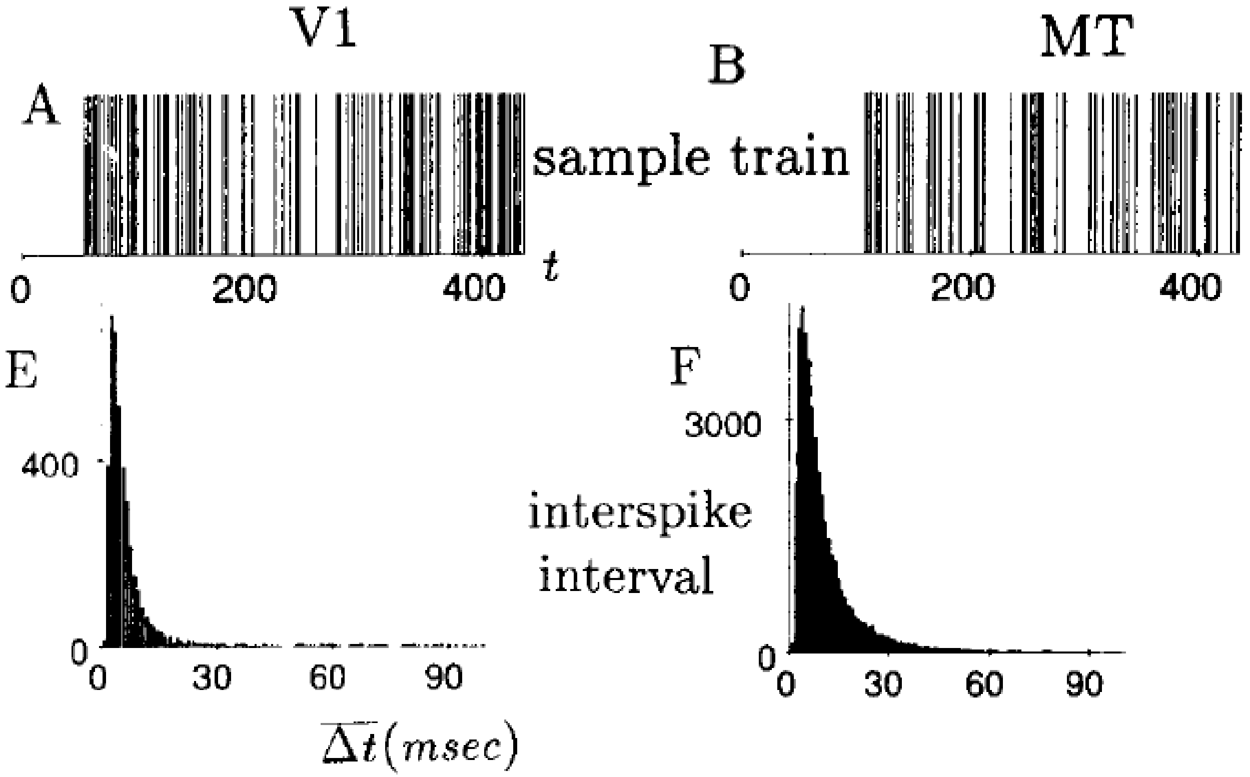


Input integration in neurons

- Cortical neurons: ~10000 synapses
- If spikes at synapses are independent
  → total input relatively constant
     (law of big numbers)
  → neuron should fire regularly

Firing regularity

Neurons in the cortex fire *irregularly*

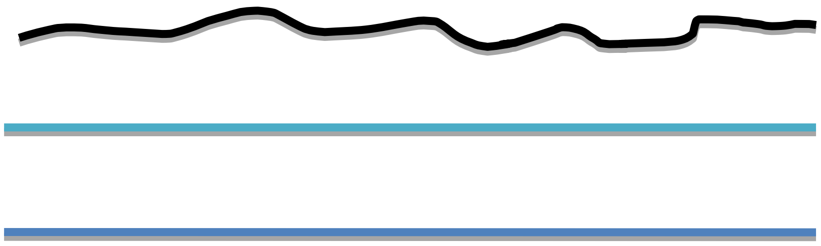


Coefficient of variation

$$CV = \frac{\sigma_{ISI}}{\langle ISI \rangle}$$

Mean-driven vs. fluctuation-driven

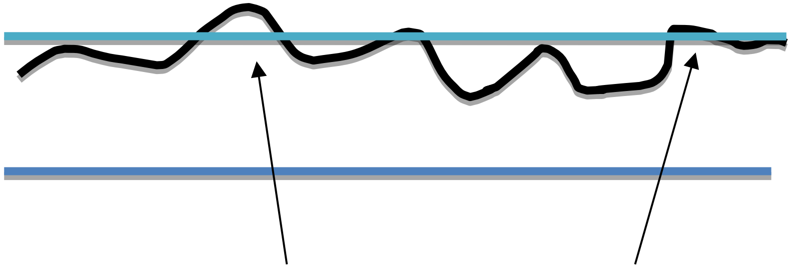
Mean-driven



$$\langle I \rangle > I_{\text{threshold}}$$

Small variability (average of many inputs)

Fluctuation-driven



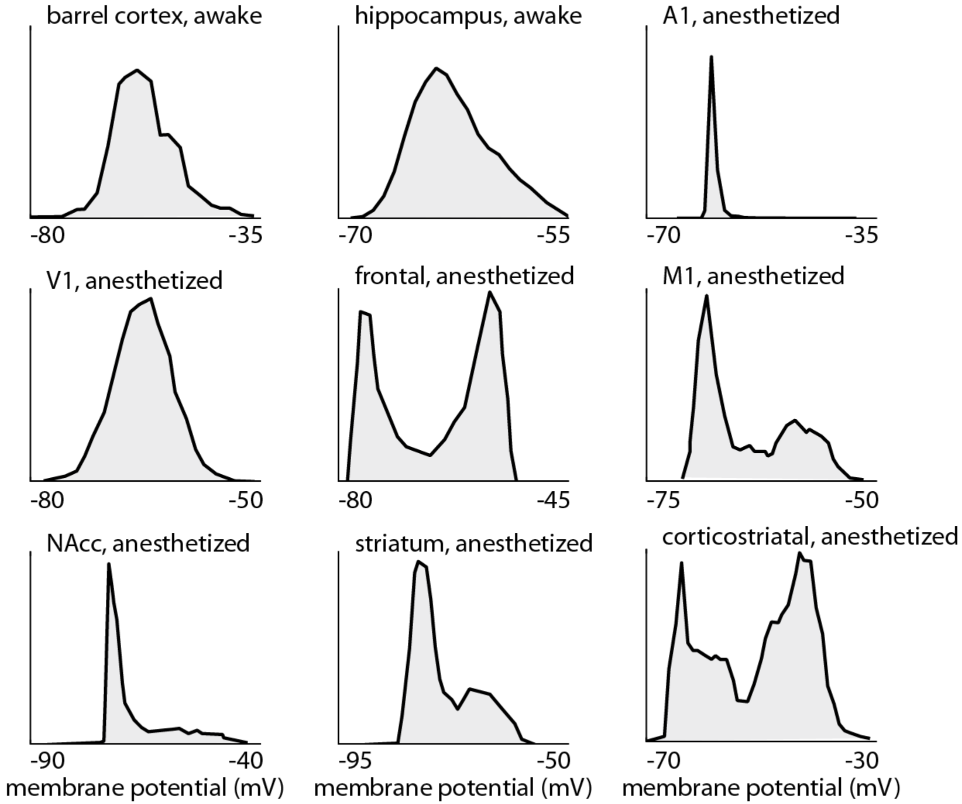
$$\langle I \rangle < I_{\text{threshold}}$$

0

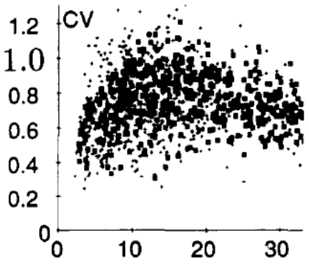
Let's try with BRIAN

spikes can only occur at times when the input fluctuates above the mean

Firing regularity



Rossant et al. (2011) *J Neurosci*

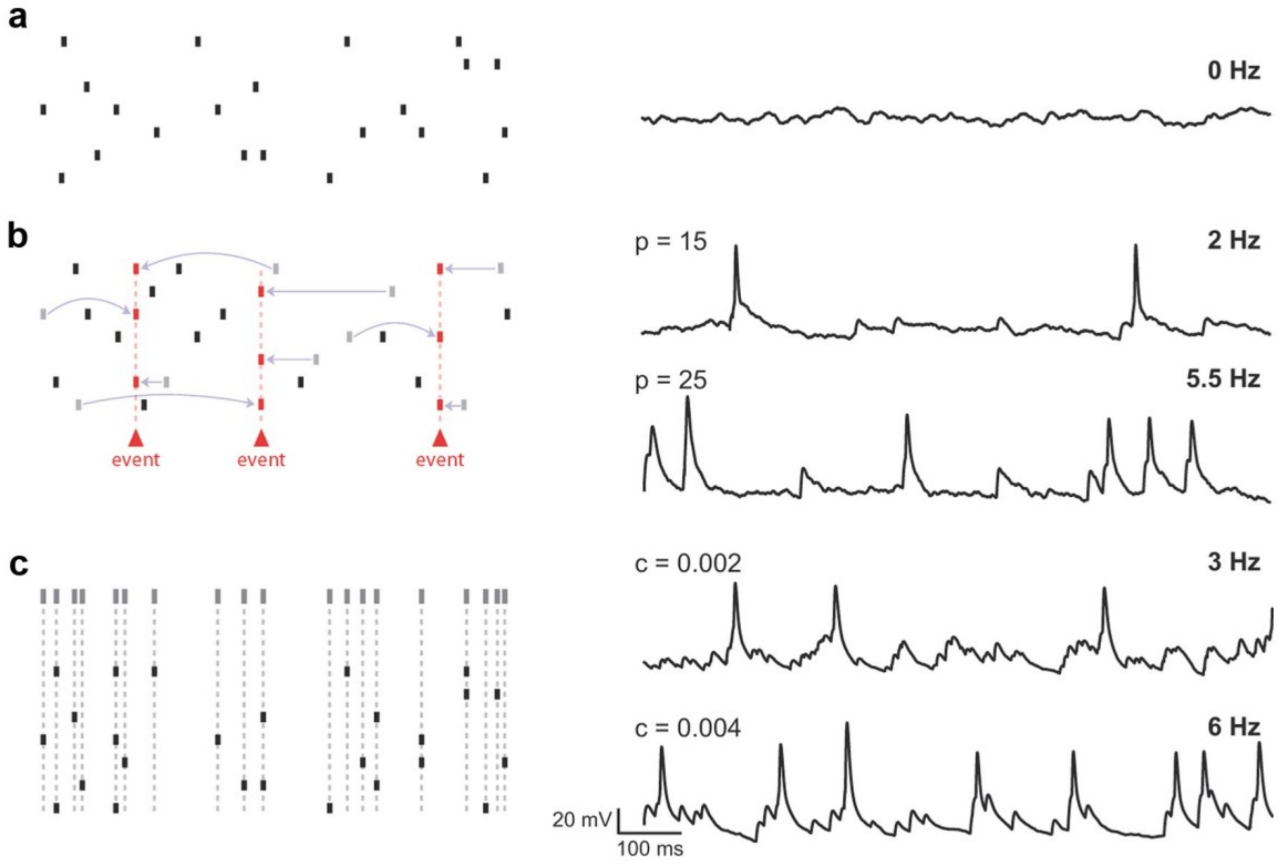


Membrane potential distribution peaks below threshold

irregular firing with CV ≈1

→ fluctuation-driven

Input integration in fluctuation-driven regime



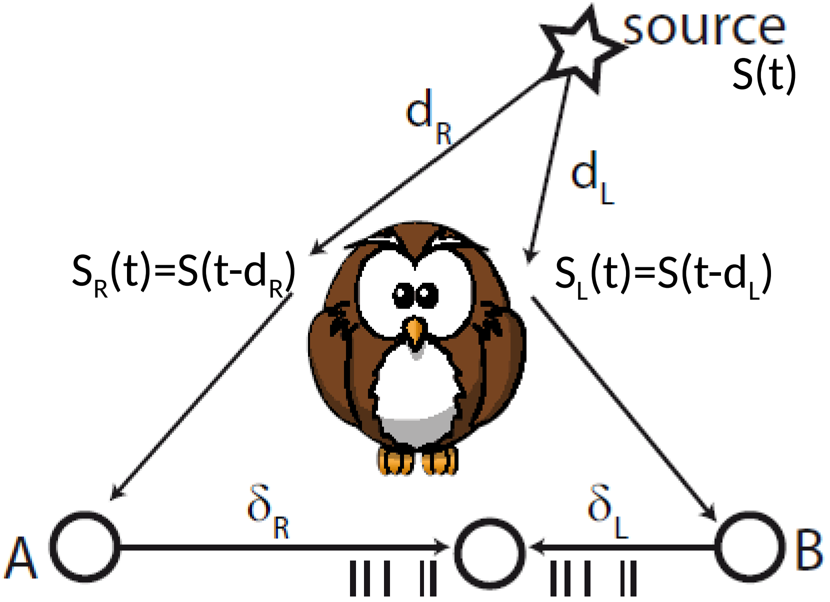
Highly sensitive to synchronous activation / correlated inputs

Rossant et al. (2011)

Part 3: Case study
Binaural sound localisation

Binaural sound localisation

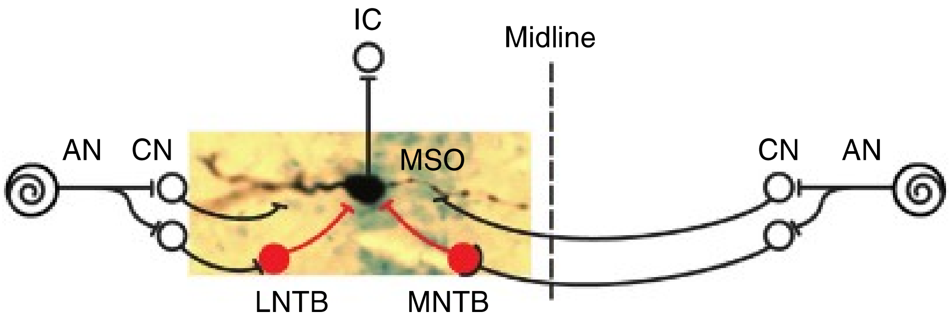
The Jeffress model

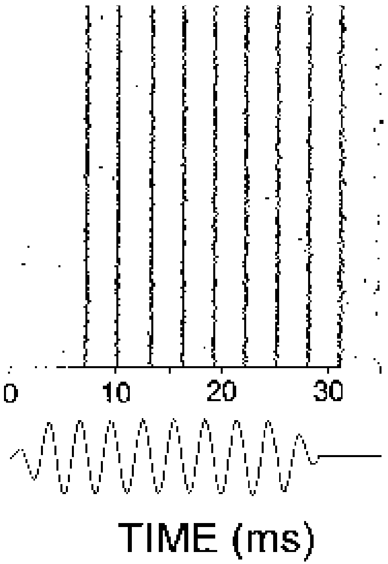


Interaural time delay
(ITD)

Binaural sound localisation

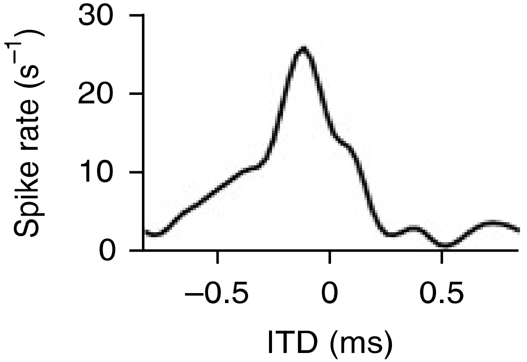
Anatomical structures from the ear to the brainstem





Louage et al. (2005)

Franken et al. (2015)



Let's try with

BRIAN

More info about Brian

Documentation: https://brian2.readthedocs.io

Web site: https://briansimulator.org

Articles:

Stimberg, Marcel, Romain Brette, and Dan FM Goodman. "Brian 2, an Intuitive and Efficient Neural Simulator." ELife 8 (2019): e47314. https://doi.org/10.7554/eLife.47314.

Stimberg, Marcel, Dan F. M. Goodman, Victor Benichoux, and Romain Brette. "Equation-Oriented Specification of Neural Models for Simulations." Frontiers in Neuroinformatics 8 (2014). https://doi.org/10.3389/fninf.2014.00006