NumPy, Matplotlib, SciPy, NeuroTools, PyNN

Jens Kremkow

Neurobiology and Biophysics, Albert-Ludwigs-Universität Freiburg
Bernstein Center for Computational Neuroscience Freiburg

Overview

NumPy - fundamental package needed for scientific computing

Matplotlib - matplotlib is a python 2D plotting library

SciPy - package for mathematics, science, and engineering

PyNN - simulator-independent language for building neuronal network models

NeuroTools - collection of tools to support all tasks associated with a neural simulation projects

Jens Kremkow

How to present code and its results?

Emacs@quercus.local

```
"""
$Id: sfn_example_stgen.py $

Example to show off some capabilities of the stgen module
and the SpikeTrain class.

- Creates two poisson spike trains with approx. rate of 100 Hz.
- Calculates rate, coefficient of variance and fano factor using the
  functions form SpikeTrain.
- Calculates cross correlation between the two SpikeTrains.
- generates a third SpikeTrain with rate ~10Hz and merges it into the
  previous two, effectively injecting correlation.
- Plots the cross correlation for the correlated and uncorrelated SpikeTrains.
- Generates shot noise from one SpikeTrain using the StGEn class and plots it.
- Calculates spike triggered average from shot noise to reconstruct the initial
  shot noise kernel.

Performed at the NeuroTools demo session, INCF booth,
SfN annual meeting 2008, Washington. DC.
"""
import NeuroTools.stgen as stgen
import pylab
pylab.close('all')

sg = stgen.StGen()
duration = 10000.
rate_independent = 100. #Hz
rate_shared = 10. #Hz, 10 % correlation

st1 = sg.poisson_generator(rate=rate_independent, t_stop = duration)
print "Spiketrain 1:"
print "mean rate: %f" % st1.mean_rate()
print "coefficient of variation: %f" % st1.cv_isi()
print "fano factor: %f" % st1.fano_factor_isi()
st1.raster_plot(display=True)
pylab.figure()
--:--  sfn_example_stgen.py   Top L15   SVN:362   (Python)----------------------------
```

Complex, time-consuming way: Copy&paste code and figures into a presentation

---

```
import NeuroTools.stgen as stgen
import pylab

sg = stgen.StGen()
duration = 1000.
rate_independent = 100. #Hz
rate_shared = 10. #Hz, 10 % correlation
```

```
st1 = sg.poisson_generator(rate=rate_independent, t_stop = duration)
print "Spiketrain 1:"
print "mean rate: %f" % st1.mean_rate()
Out: 94.0
print "coefficient of variation: %f" % st1.cv_isi()
Out: 1.03099116849
print "fano factor: %f" % st1.fano_factor_isi()
Out: 11.211944837
st1.raster_plot(display=True)
```

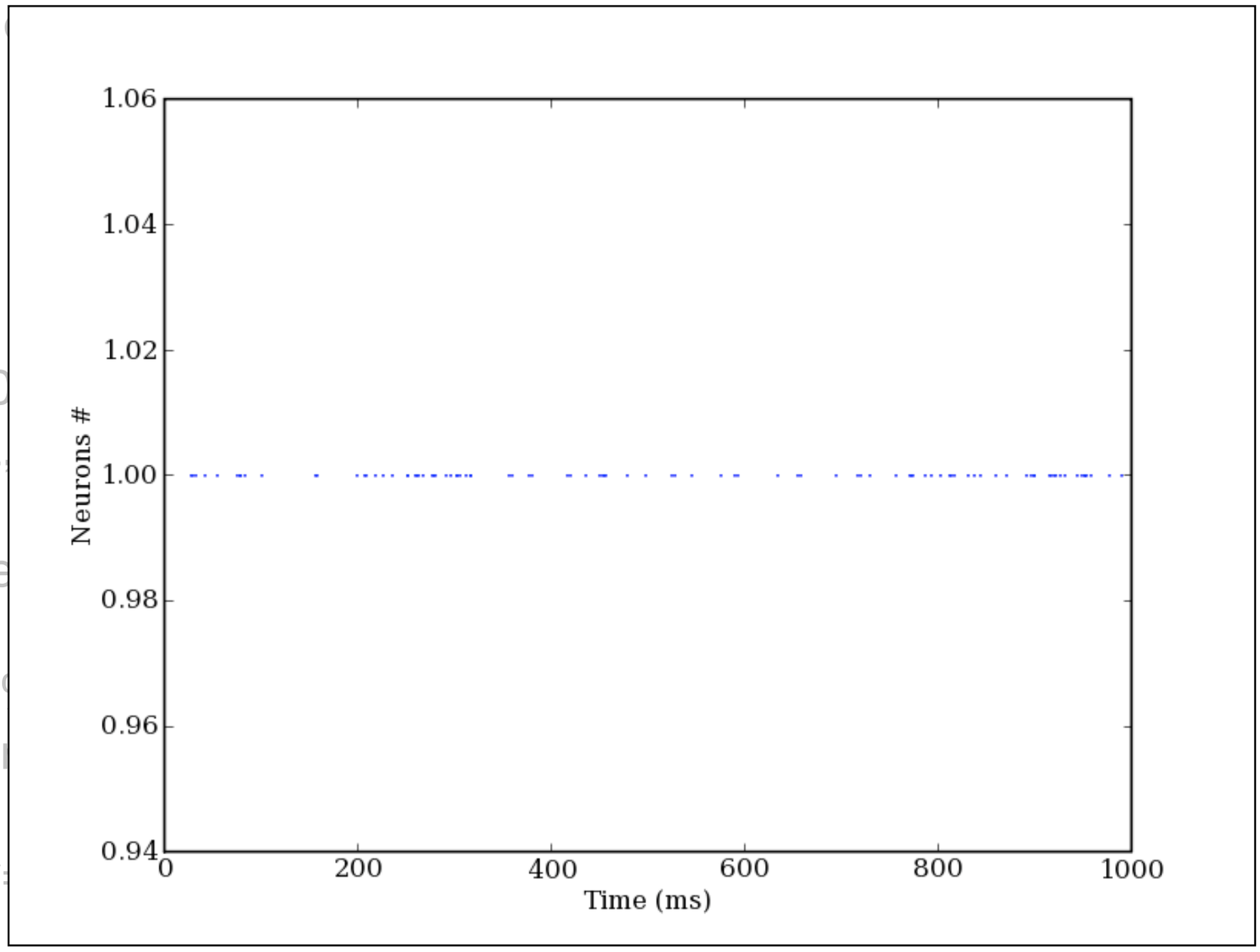

Jens Kremkow

Complex, time-consuming way: Copy&paste code and figures into a presentation

```
import NeuroTools.stg
import pylab
pylab.close('all')


sg = stgen.StGen()
duration = 1000.
rate_independent = 10
rate_shared = 10. #Hz

st1 = sg.poisson_gene
print "Spiketrain 1:"
print "mean rate: %f"
print "coefficient of va
print "fano factor: %f"
st1.raster_plot(display
```



Jens Kremkow

Easy, efficient way:
pyreport

- http://gael-varoquaux.info/computers/pyreport/

- pyreport is a program that runs a python script and captures its output, compiling it to a pretty report in a pdf or an html file.

- It can display the output embedded in the code that produced it and can process special comments (literate comments) according to markup languages (rst or LaTeX) to compile a very readable document.

- This allows for extensive literate progamming in python, for generating reports out of calculations written in python, and for making nice tutorials.


Easy, efficient way:
pyreport

pyreport sfn_example_stgen.py

open sfn_example_stgem.pdf


Jens Kremkow

NumPy

- http://numpy.scipy.org

- NumPy is the fundamental package needed for scientific computing with Python. It contains:
- a powerful N-dimensional array object
- basic linear algebra functions
- basic Fourier transforms
- sophisticated random number capabilities
- ...


Information about NumPy

General information: http://www.scipy.org/NumPy

Reference: http://docs.scipy.org/doc/numpy/reference/

Examples for each function: http://www.scipy.org/Numpy_Example_List_With_Doc

Cookbook: http://www.scipy.org/Cookbook

NumPy examples

show examples pdf

Jens Kremkow

Matplotlib

- matplotlib is a python 2D plotting library which produces publication quality figures

- matplotlib can be used in python scripts, the python and ipython shell (ala matlab or mathematica), web application servers, and six graphical user interface toolkits

- matplotlib tries to make easy things easy and hard things possible

- You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code

Information about Matplotlib

General information: http://matplotlib.sourceforge.net/

Examples: http://matplotlib.sourceforge.net/examples/index.html

Cookbook: http://www.scipy.org/Cookbook/Matplotlib

User Guide: http://matplotlib.sourceforge.net/users/index.html

Gallery: http://matplotlib.sourceforge.net/gallery.html

Jens Kremkow

Matplotlib examples

show examples pdf

Jens Kremkow

SciPy

- SciPy (pronounced "Sigh Pie") is open-source software for mathematics, science, and engineering.

- The SciPy library depends on NumPy

- The SciPy library is built to work with NumPy arrays, and provides many user-friendly and efficient numerical routines such as routines for numerical integration and optimization.


Information about SciPy

General information: http://www.scipy.org

Reference Guide: http://docs.scipy.org/doc/scipy/reference/

Cookbook: http://www.scipy.org/Cookbook

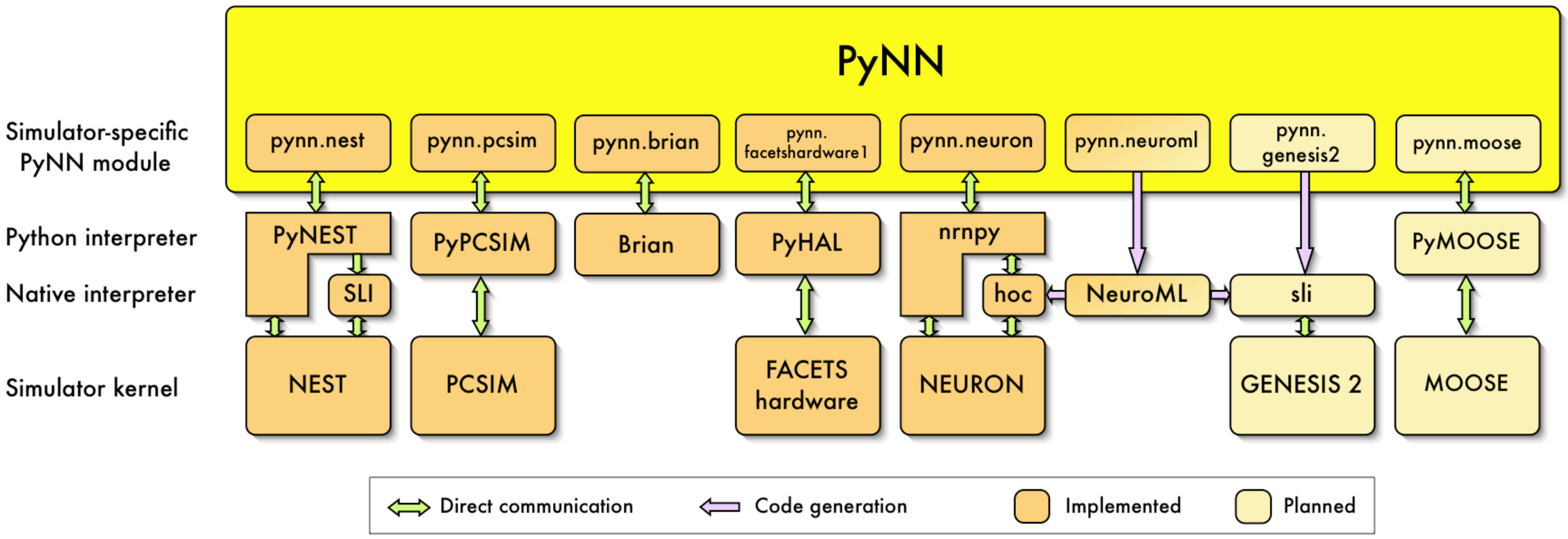

SciPy examples

show examples pdf


Jens Kremkow

PyNN

- PyNN (pronounced 'pine') is a is a simulator-independent language for building neuronal network models



PyNN

General information: http://neuralensemble.org/trac/PyNN

User Guide: http://neuralensemble.org/trac/PyNN/wiki/UsersGuide


Jens Kremkow

PyNN examples

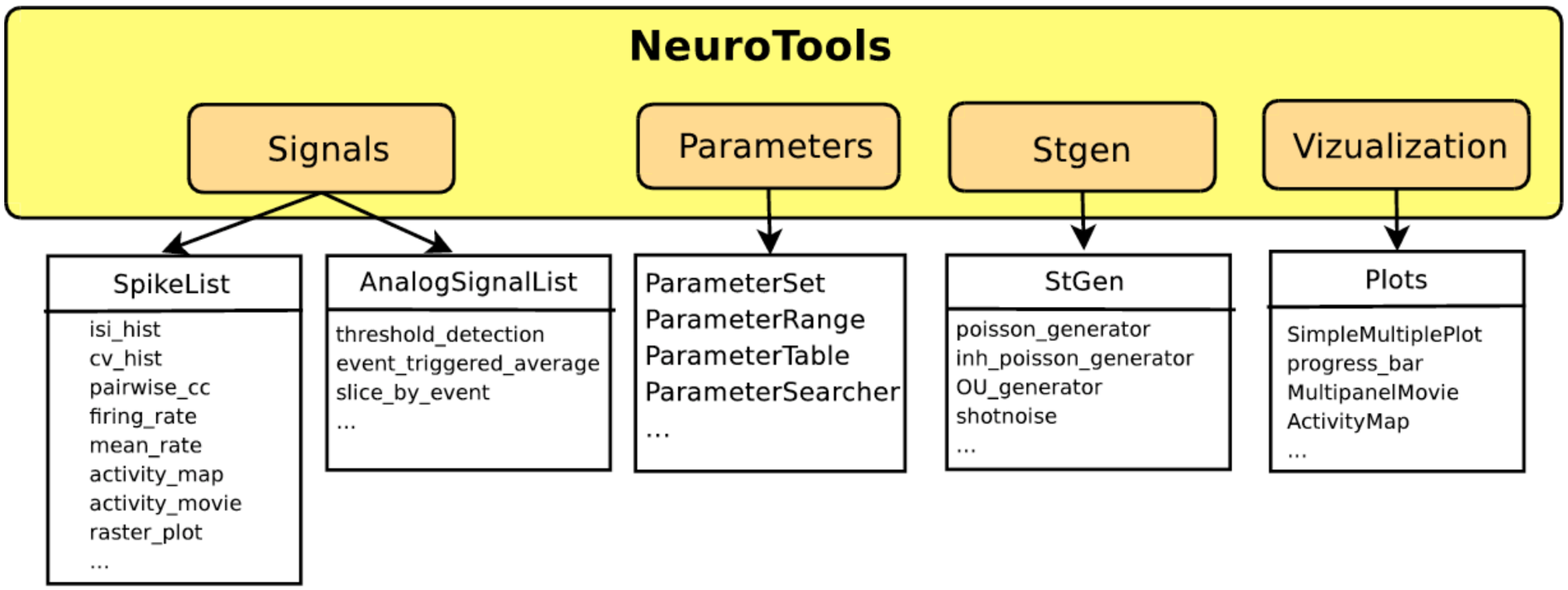
show examples pdf

Jens Kremkow

NeuroTools

- NeuroTools is a collection of tools to support all tasks associated with a neural simulation project which are not handled by the simulation engine

- NeuroTools provides modules to facilitate simulation setup, parameterization, data management, analysis and visualization



Jens Kremkow

NeuroTools

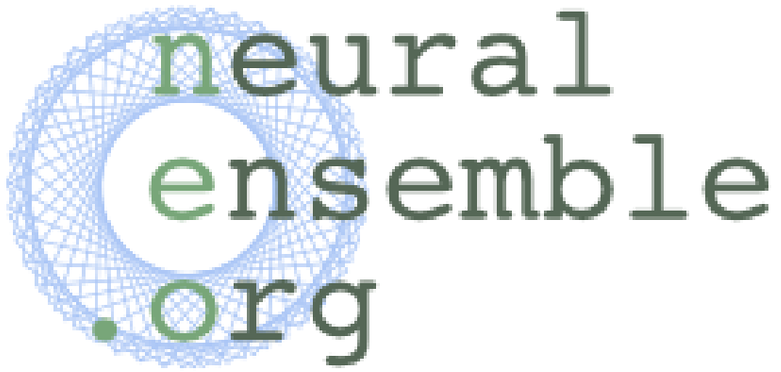
General information & User Guide: http://neuralensemble.org/trac/NeuroTools

Jens Kremkow

NeuroTools examples

show examples pdf

Jens Kremkow



Contributers:

Daniel Brüderle, Andrew Davison, Jens Kremkow, Eilif Muller, Laurent Perrinet, Michael Schmucker

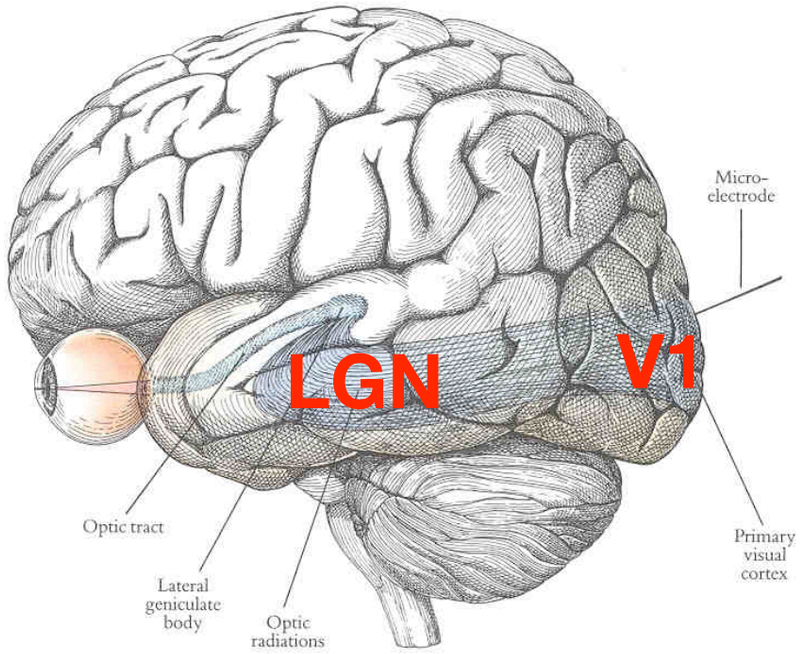
Exercises

- Simple exercises on the Wiki

- More complex exercise: build a visual system using numpy, matplotlib and NeuroTools

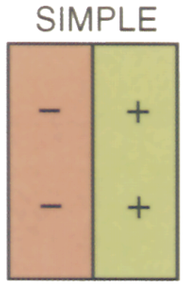
- Go through some examples in the cookbooks of numpy, scipy, matplotlib

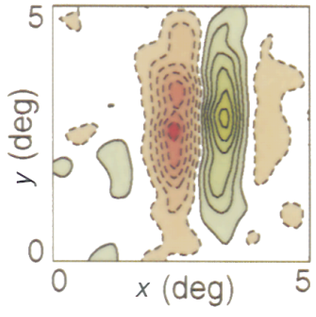

More complex exercise:
Build a visual system model



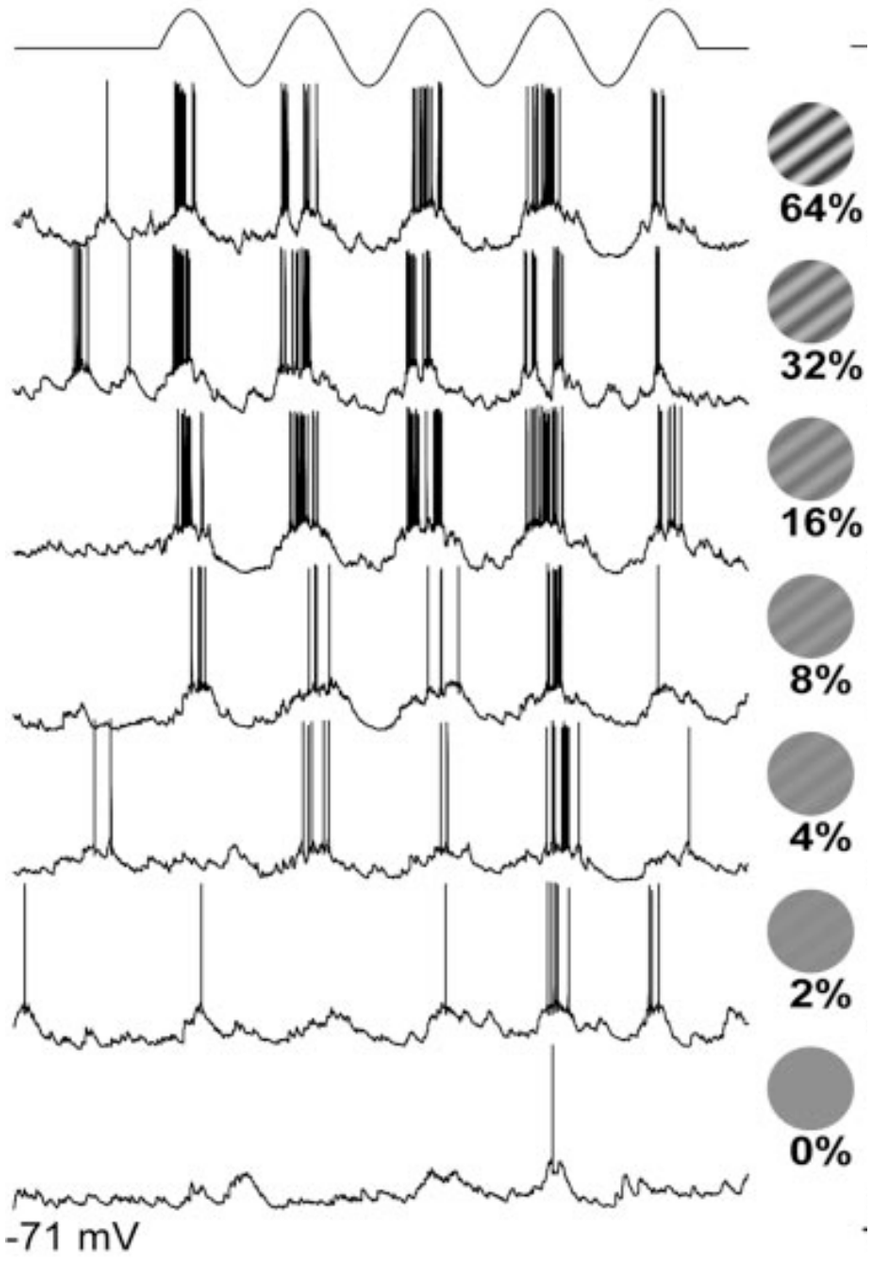
Adapted from Hubel & Wiesel
1987

V1 simple cell receptive field





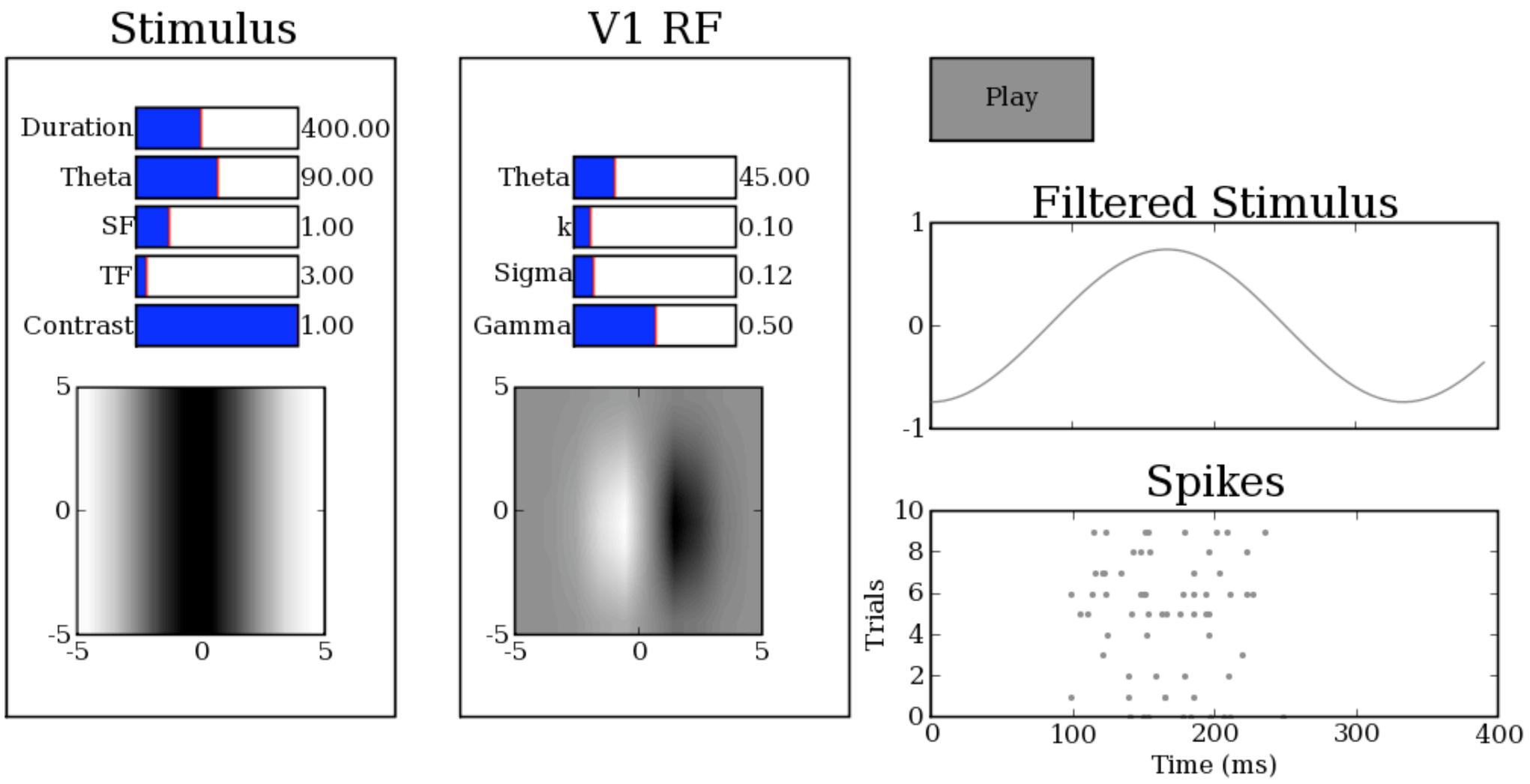
DeAngelis et al.
Trends Neurosci 1995



Adapted from Contreras & Palmer
J Neurosci 2003

Jens Kremkow

More complex exercise:
Build a visual system model



More complex exercise:
Build a visual system model