NeuralTractSpikes

Julius C. F. Schulz

January 28, 2017

Simulation of 10000 ms of 400 independent descending commands following a gamma distribution with mean of 142 ms and order 10

Input #1

```
import sys
sys.path.insert(0, '..')
import time
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import scipy.stats

from Configuration import Configuration
from NeuralTract import NeuralTract
```

Input #2

```
conf = Configuration('confNeuralTractSpikes.rmto')
```

Input #3

```
t = np.arange(0.0, conf.simDuration_ms, conf.timeStep_ms)
```

Input #4

```
pools = dict()
pools[0] = NeuralTract(conf, 'CMExt')

tic = time.clock()
for i in xrange(0,len(t)-1):
    pools[0].atualizePool(t[i])
toc = time.clock()
print str(toc - tic) + ' seconds'
```

Output #4

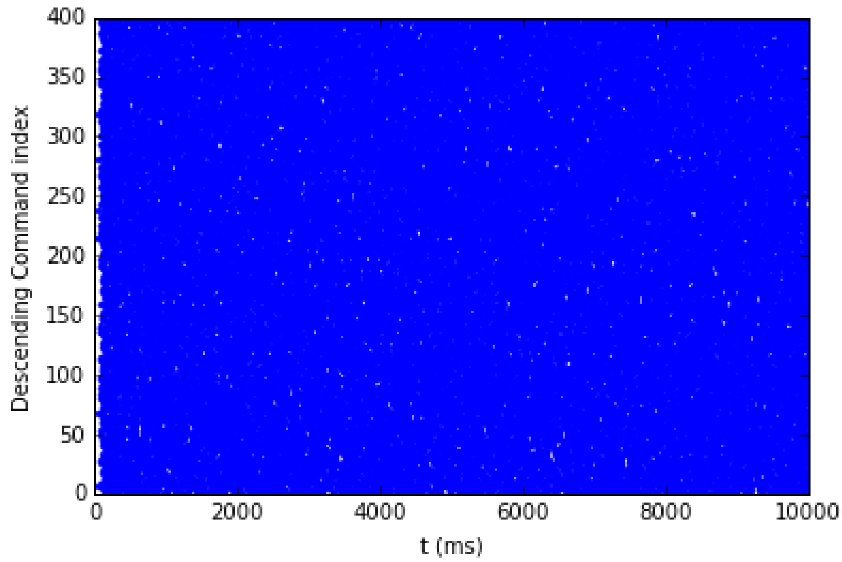
```
Descending Command CMExt built
114.326818 seconds
```

Input #5

```
pools[0].listSpikes()
```

The spike times of all descending commands along the 10000 ms of simulation is shown in Fig. 1.

Input #17



Spike times of all descending commands along the 10000 ms of simulation.

```
plt.figure()
plt.plot(pools[0].poolTerminalSpikes[:, 0],
         pools[0].poolTerminalSpikes[:, 1]+1, '.')
plt.xlabel('t (ms)')
plt.ylabel('Descending Command index')
```

Output #17

```
Out[17]: <matplotlib.text.Text at 0x7f2a4944db10>
```

The spike times of all descending commands during the last 1000 ms of the simulation is shown in Fig. 2.

Input #7

```
plt.figure()
plt.plot(pools[0].poolTerminalSpikes[pools[0].poolTerminalSpikes[:, 0]>9000, 0],
         pools[0].poolTerminalSpikes[pools[0].poolTerminalSpikes[:, 0]>9000, 1]+1, '.')
plt.xlabel('t (ms)')
plt.ylabel('Descending Command index')
```

Output #7

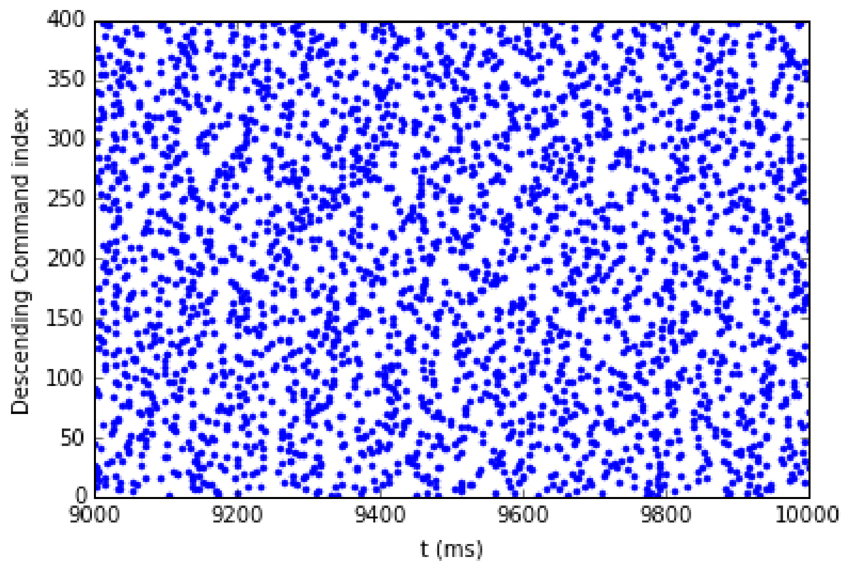
```
Out[7]: <matplotlib.text.Text at 0x7f2a49606150>
```

The histogram of the interspike intevals of all the descending commands is shown in Fig. 3. Note that the peak is in the specified ISI at the beginning of the simulation.

Input #8

```
ISI = np.array([])
for i in xrange(0,len(pools[0].unit)):
    ISI = np.append(ISI, np.diff(np.reshape(np.array(pools[0].unit[i].terminalSpikeTrain),
```

Input #9



Spike times of all descending commands along the last 1000 ms of simulation.

```
plt.figure()
plt.hist(ISI)
plt.xlabel('ISI (ms)')
plt.ylabel('Counts')
```

Output #9

```
Out[9]: <matplotlib.text.Text at 0x7f2a495fa190>
```

Below different statistics of the interspike intervals and firing rate are obtained.

Input #13

```
SD = np.std(ISI)
M = np.mean(ISI)
SK = scipy.stats.skew(ISI)
CV = SD / M

print 'ISI Mean = ' + str(M) + ' ms'
print 'ISI Standard deviation = ' + str(SD) + ' ms'
print 'ISI CV = ' + str(CV)
```

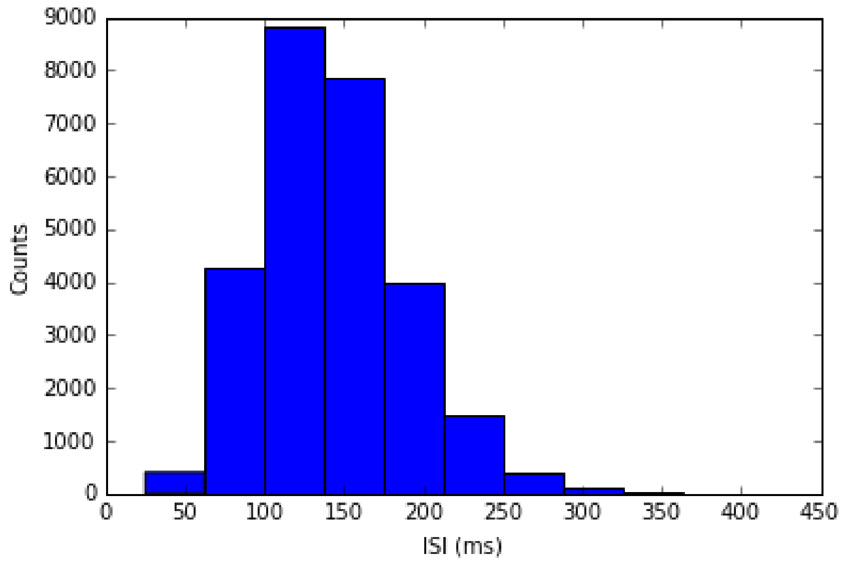
Output #13

```
ISI Mean = 143.121468017 ms
ISI Standard deviation = 45.4032545914 ms
ISI CV = 0.317235808299
```

Input #14

```
M_FR = 1000.0 / M
SD_FR = np.sqrt((SD**2) * 1000 / (M**3) + 1/6.0 + (SD**4) / (2*M**4) - SK/(3*M**3))


print 'Firing rate mean = ' + str(M_FR) + ' Hz'
print 'Firing rate standard deviation = ' + str(SD_FR) + ' Hz'
```



Histogram of the interspike intervals durin g the simulation.

Output #14

```
Firing rate mean = 6.98707198754 Hz
Firing rate standard deviation = 0.935360627226 Hz
```

Input #15

```
CV_FR = SD_FR / M_FR
print 'CV of Firing rate = ' + str(CV_FR)
```

Output #15

```
CV of Firing rate = 0.133870186094
```

Input #None