InjectedCurrentRenshawCellPool

January 24, 2017

1 This notebook presents a simulation of a pool of Renshaw cells (RC) with a current injected (iInjected) in their soma.

```
In [1]: import sys
        sys.path.insert(0, '..')
        import time
        import matplotlib.pyplot as plt
        %matplotlib inline
        import numpy as np

        from Configuration import Configuration
        from InterneuronPool import InterneuronPool
        from SynapsesFactory import SynapsesFactory
```

```
In [2]: conf = Configuration('confInjectedCurrentRenshawCellPool.rmto')
        conf.simDuration_ms = 500 # Here I change simulation duration without chang
```

```
In [3]: # Time vector for the simulation
        t = np.arange(0.0, conf.simDuration_ms, conf.timeStep_ms)

        membPotential = np.zeros_like(t, dtype = 'd')
```

```
In [5]: pools = dict()
        pools[0] = InterneuronPool(conf, 'RC')

        Syn = SynapsesFactory(conf, pools)
```

```
Interneuron Pool of RC built
Synaptic Noise on RC built
All the 350 synapses were built
```

```
In [9]: # The simulation  itself

        tic = time.clock()
        for i in xrange(0,len(t)-1):
            # Here you can change the injected current in the Resnhaw cell during t
```

```
            for j in xrange(len(pools[0].unit)):
                pools[0].unit[j].iInjected = 5
            pools[1].atualizePool(t[i])
            pools[0].atualizeInterneuronPool(t[i])
            membPotential[i] = pools[0].unit[175].v_mV # This command records the r
        toc = time.clock()
        print str(toc - tic) + ' seconds'
```

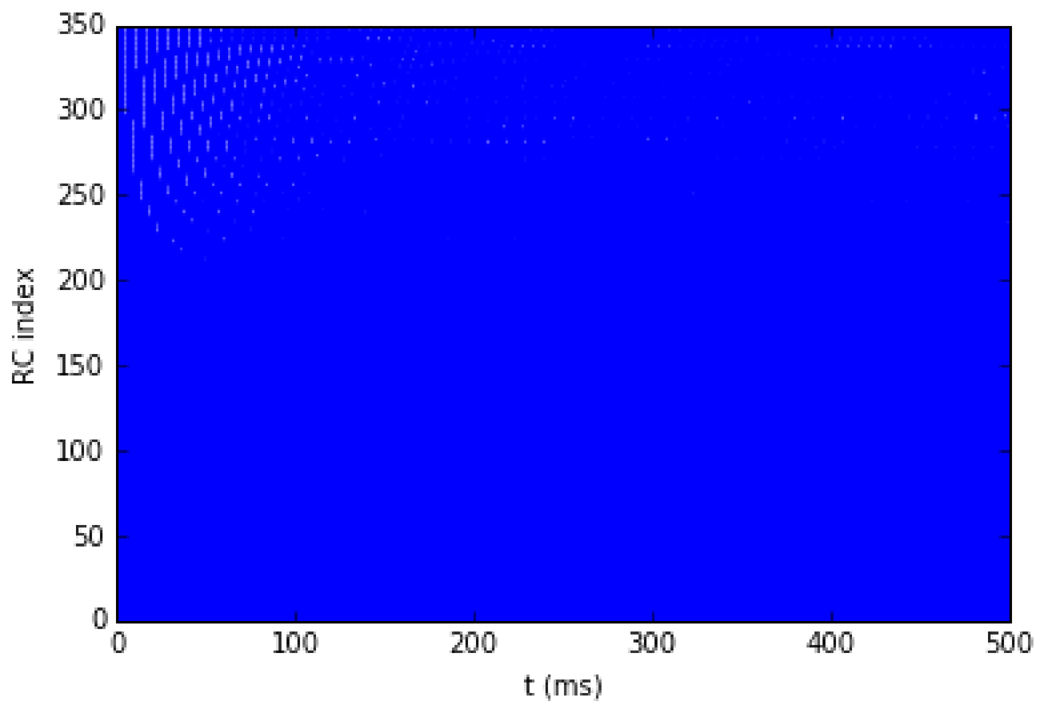
```
363.657428 seconds
```

```
In [10]: pools[0].listSpikes()

         plt.plot(pools[0].poolSomaSpikes[:, 0],
             pools[0].poolSomaSpikes[:, 1]+1, '.')

         plt.xlabel('t (ms)')
         plt.ylabel('RC index')

         plt.show()
```



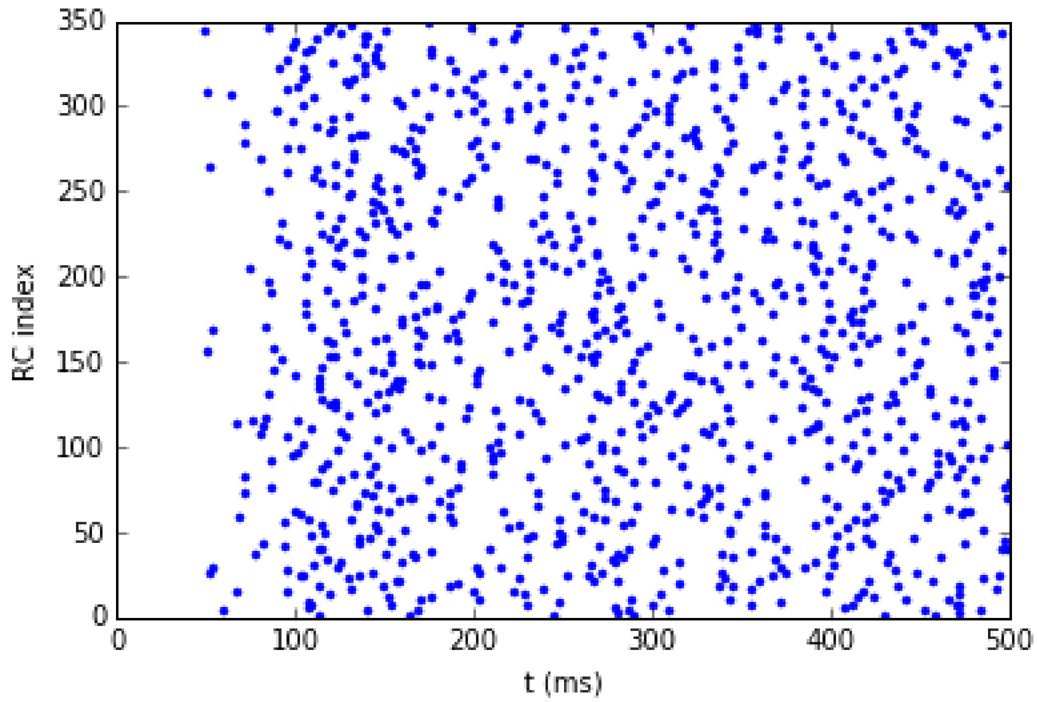

```
In [11]: pools[1].listSpikes()

         plt.plot(pools[1].poolTerminalSpikes[:, 0],
```

```
    pools[1].poolTerminalSpikes[:, 1]+1, '.')

plt.xlabel('t (ms)')
plt.ylabel('RC index')

plt.show()
```

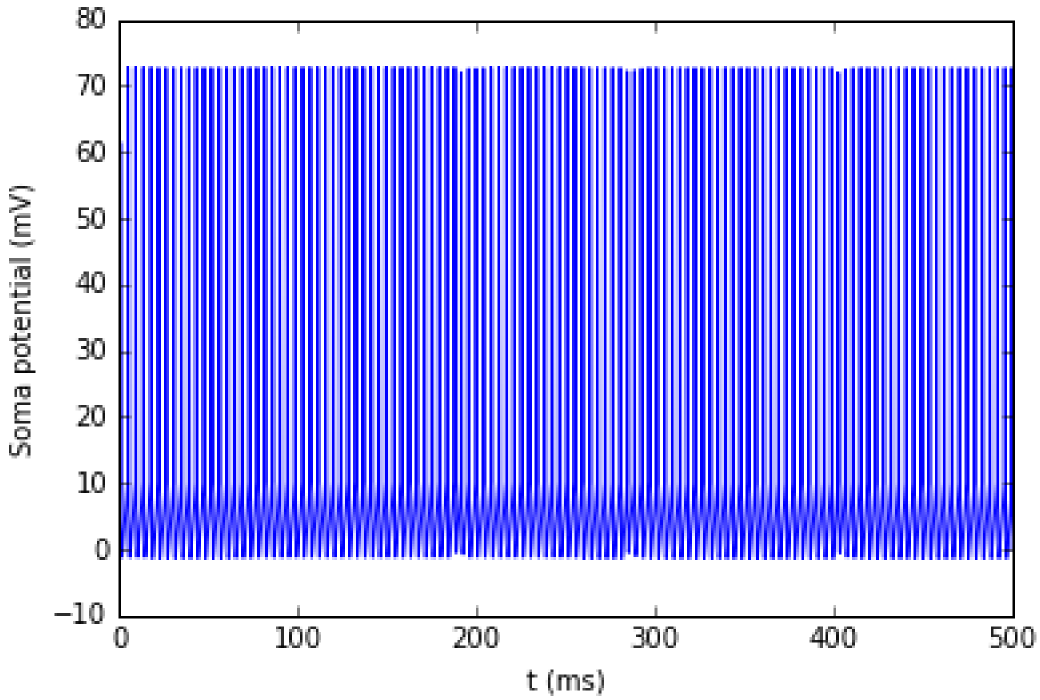



```
In [12]: plt.figure()

         plt.plot(t, membPotential, '-')

         plt.xlabel('t (ms)')
         plt.ylabel('Soma potential (mV)')

         plt.show()
```




```
In [13]: ISI = np.array([])
         for i in xrange(0,len(pools[1].unit)):
             ISI = np.append(ISI, np.diff(np.reshape(np.array(pools[1].unit[i].term
         print ISI
```

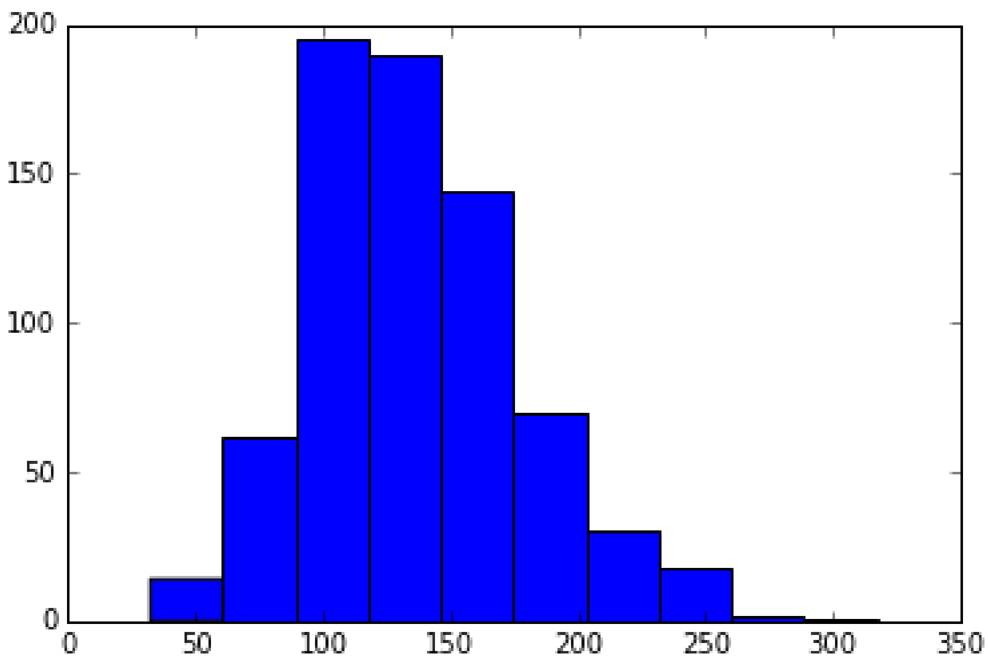
```
[ 131.1    185.4    125.5    168.3    191.15   118.2    197.95    80.7    168.5
  142.8    171.95   129.     120.5     70.3    112.15    59.25   201.7    114.95
  140.85   123.7    140.4     98.4    117.25   178.4    247.15   261.65   108.85
   76.5    167.2    155.65   195.1    104.2    197.     107.5    115.8    138.05
   73.35   150.95    98.85   195.3    128.25   124.1    110.65   153.3    122.3
   79.9     97.2     95.6    114.4    145.55    32.3    127.95   102.75   127.65
  149.3    145.35   135.2    108.35    91.5    129.05   102.4    209.35   131.1
  147.3    170.85    92.4    169.35   154.95   136.1    119.15   111.25   190.5
  115.85   159.8     90.05   177.7     62.8    114.9     88.25   172.7     93.4
  129.95   120.3    179.5    173.85   114.4     97.05    93.75   166.55   120.35
  175.8    140.55   162.85   228.7    164.4    122.3     95.45   246.7    159.55
  117.8     93.9    138.6     77.25   140.35   153.95   129.35   131.05    96.5
  112.3    181.6    141.9    190.35   110.35   136.6    139.3    107.05    94.9
  100.25   130.25   141.     126.45   139.95   165.35   117.7    142.05   125.5
   91.55   156.5    244.1    125.7    116.35   105.     154.2    109.15   216.85
  164.3    155.45   104.9    128.3    101.55   100.65    65.8    191.8     99.2
   72.25    77.7    110.2    225.4    164.45    66.2    112.5    105.75    97.15
   96.     101.45   115.45   176.55   112.35   151.2    156.25   105.1    138.25
  124.9    183.2    102.1     57.5    173.15   192.     197.9    177.65   166.65
```

210.45	96.8	144.25	168.1	159.9	238.9	316.95	76.6	221.45
160.85	165.05	75.7	117.	221.05	171.9	81.3	145.55	195.5
222.4	123.85	83.5	172.9	160.4	133.25	167.3	78.25	223.95
111.4	99.75	156.1	113.65	245.1	155.7	142.45	102.45	169.
56.15	108.6	170.1	96.2	139.35	146.7	102.65	145.3	125.55
133.95	113.05	138.4	96.3	118.1	166.15	152.3	167.25	169.15
144.7	105.45	166.9	90.55	156.05	105.	191.45	89.55	154.45
133.7	177.7	82.2	225.1	100.05	76.75	267.6	129.5	134.95
106.	96.1	181.85	102.55	115.1	214.15	167.45	161.05	179.4
169.85	91.65	115.85	74.7	145.3	165.15	66.55	179.75	132.1
90.75	88.75	115.65	166.05	108.9	78.7	111.	49.5	141.1
126.4	158.95	95.35	150.9	165.3	144.5	183.75	165.25	130.
110.65	178.55	118.55	136.7	138.3	229.25	147.7	90.95	89.55
91.25	60.8	167.9	112.5	177.65	145.3	145.2	129.3	101.2
121.65	124.1	190.1	171.75	218.5	114.75	84.65	96.2	98.35
248.9	124.85	158.55	166.55	180.2	96.5	120.35	125.7	113.95
182.85	98.25	188.85	140.05	141.7	143.35	103.05	115.1	128.9
169.35	106.05	152.55	159.1	145.65	164.05	142.3	82.75	66.05
192.2	165.4	78.65	142.05	158.55	73.15	151.15	111.2	216.35
82.1	89.7	103.	93.25	107.75	95.7	149.15	119.6	147.5
79.55	96.45	121.1	85.25	115.2	104.8	165.55	147.55	118.85
121.2	135.1	71.25	124.95	121.05	164.75	122.35	167.35	202.7
107.25	131.05	95.5	116.9	137.5	106.7	77.7	142.75	233.5
74.	100.8	114.55	93.75	145.15	200.95	101.95	137.45	103.75
169.	180.3	95.8	122.15	195.6	129.7	107.	84.3	166.15
106.4	132.4	204.1	140.4	103.1	103.55	106.8	111.85	157.65
115.5	158.2	142.95	140.9	94.	102.8	118.9	107.3	49.45
146.05	110.1	185.35	170.7	155.05	112.55	138.05	212.65	103.55
138.1	72.35	155.6	200.25	97.9	126.3	71.9	138.6	245.35
91.9	93.45	171.2	97.65	119.5	177.75	94.05	43.3	163.9
120.6	240.15	166.6	162.3	153.25	181.65	104.2	124.9	130.65
69.4	139.2	165.45	160.8	105.75	176.95	106.3	131.8	134.2
118.1	148.15	137.	178.9	208.85	117.65	131.1	110.95	166.6
104.6	83.5	150.35	144.35	128.	93.9	143.65	168.3	118.2
228.45	65.45	256.4	134.1	95.	140.95	184.35	84.65	56.3
156.3	177.2	91.05	117.95	183.8	168.5	129.4	141.45	216.55
76.9	86.35	156.05	139.	177.15	138.85	115.4	101.7	152.6
157.55	158.25	112.7	222.85	88.6	229.7	217.4	109.55	242.85
200.4	81.6	118.1	62.75	94.95	96.65	168.65	128.9	170.25
180.7	89.9	86.85	159.7	124.85	69.6	131.6	151.5	139.
151.85	145.85	88.05	117.65	159.55	119.95	121.25	202.45	75.65
76.4	89.95	136.2	44.	162.8	171.7	74.2	137.35	129.55
117.5	185.9	106.7	103.2	117.9	120.3	102.25	240.7	152.85
137.7	99.8	155.	69.	135.35	135.1	155.65	176.8	119.45
182.4	82.9	71.45	116.1	90.6	52.55	95.4	197.4	113.85
103.05	86.95	195.2	78.7	159.85	98.45	240.45	187.25	234.25
116.3	184.8	92.8	169.15	158.2	153.85	121.25	115.2	175.35
125.65	152.55	156.2	75.75	150.	165.05	147.45	188.75	210.65

```
 166.15   98.35  120.95  129.9   119.55  162.8   121.45  141.8   117.85
 137.55  111.35  206.35  129.65   82.2   135.65  105.05  125.75  152.7
 148.55  139.7    92.35  122.35  166.1    83.5   108.    103.5   143.4
  59.15  206.    140.7   134.9   234.5   109.25  120.65  134.95  199.7
 100.95  176.45  157.1    89.9   135.15  174.9   127.55   93.35  142.45
 151.5   158.9    94.55  114.45  102.2   135.8   115.25   46.1   244.7
 111.5   154.     91.1    71.7   135.4   133.9   111.35  118.4   156.55
 162.45  186.85  113.    197.8   222.     40.7   213.85  138.85   99.45
 150.85   91.7   160.15  111.45  180.05  127.5   140.55  235.85   56.3
 150.3   213.9   142.45  114.95  199.55  156.9   163.5   131.45  202.15
 128.45  155.6   156.55  216.85  128.5   222.05  110.    148.55  152.75
 107.65  151.45   98.     98.45   99.35  133.95  136.85  101.7   115.65
 177.65  236.25  103.6   130.95  119.6   114.55  158.5   126.2   190.3
 131.5   103.65  132.45  124.8   123.35  159.1 ]
```

```
In [14]: plt.figure()
         plt.hist(ISI)
```

```
Out[14]: (array([  14.,   62.,  195.,  190.,  144.,   70.,   30.,   18.,    2.,
          array([  32.3  ,   60.765,   89.23 ,  117.695,  146.16 ,  174.625,
                  203.09 ,  231.555,  260.02 ,  288.485,  316.95 ]),
          <a list of 10 Patch objects>)
```




```
In [ ]:
```