Homework 3, Problem 3 on inhomogeneous Poisson processes

ECE C143A/C243A, Spring Quarter 2023, Prof. J.C. Kao, TAs T. Monsoor, R. Gore, D. Singla

In this problem, we will use the same simulated neuron as in Problem 2, but now the reaching angle s will be time-dependent with the following form:

$$s(t) = t^2 \cdot \pi,$$

where t ranges between 0 and 1 second. This will be refered as *s(t)* equation in the questions.

In [ ]:
```python
"""
ECE C143/C243 Homework-3 Problem-3

"""
import numpy as np
import matplotlib.pyplot as plt
import nsp as nsp # these are helper functions that we provide.
import scipy.special

# Load matplotlib images inline
%matplotlib inline

# Reloading any code written in external .py files.
%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

(a) (6 points) Spike trains

Generate 100 spike trains, each 1 second in duration, according to an inhomogeneous Poisson process with a firing rate profile defined by tuning equation,

$$\lambda(s) = r_0 + (r_{\max} - r_0)\cos(s - s_{\max})$$

and the $s(t)$ equation,

$$s(t) = t^2 \cdot \pi$$

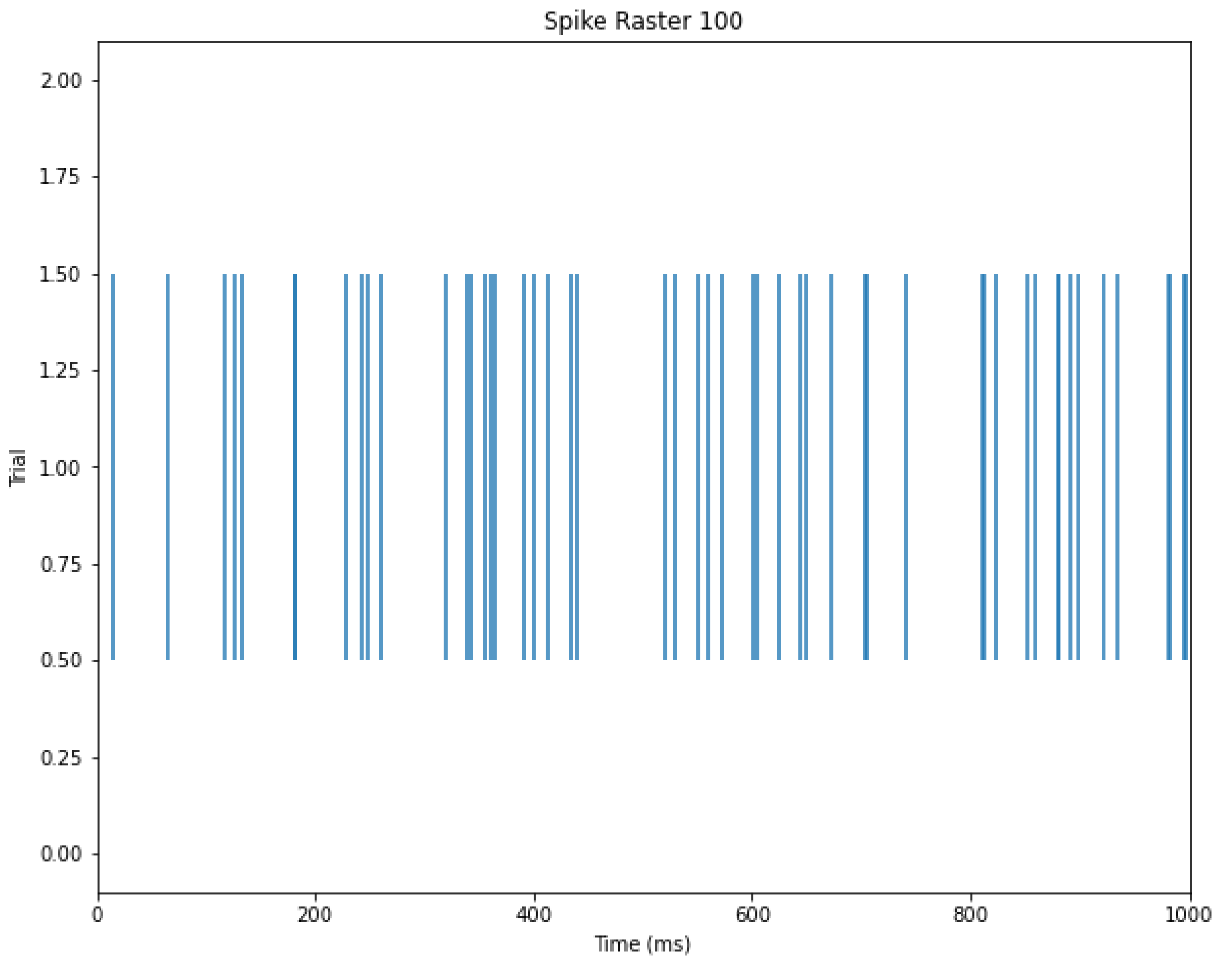
In [ ]:
```python
r_0 = 35 # (spikes/s)
r_max = 60 # (spikes/s)
s_max = np.pi/2 # (radians)
T = 1000 # trial length (ms)
```

In [ ]:
```python
np.random.exponential(1.0/r_max * 1000)
```

Out[ ]:
```
1.3161906679489954
```

In [ ]:
```python
## 3a
num_trials = 100 # number of total spike trains
```

```
num_rasters_to_plot = 5 # number of spike trains to plot

spike_trains = []

plt.figure(figsize=(10,8))


#========================================================#
# YOUR CODE HERE:
#   Generate the spike times for 100 trials of an inhomogeneous
#   Poisson process.  Plot 5 example spike rasters.

def tuning_curve(s):
    return r_0 + (r_max - r_0) * np.cos(s - s_max)

def reach_angle(t):
    return t**2 * np.pi

for i in range(num_trials):
    spikes = []
    t = 0

    while t < T:
        fr = tuning_curve(reach_angle(t))
        t += np.random.exponential(1.0/r_max * 1000)

        if t < T:
            spikes.append(t)

    spike_trains.append(spikes)

plt.eventplot(spike_trains[i])
plt.xlim([0, T])
plt.xlabel('Time (ms)')
plt.ylabel('Trial')
plt.title(f'Spike Raster {i+1}')


plt.show()


#========================================================#
pass

#========================================================#
# END YOUR CODE
#========================================================#
```



(b) (5 points) Spike histogram

Plot the spike histogram by taking spike counts in non-overlapping 20 ms bins, then averaging across the 100 trials. The spike histogram should have firing rate (in spikes / second) as the vertical axis and time (in msec, not time bin index) as the horizontal axis. Plot the expected firing rate profile defined by equations tuning equation and *s(t)* equation on the same plot.

In [ ]:

```
# 3b
bin_width = 20 # (ms)
#======================================================#
# YOUR CODE HERE:
#   Plot the spike histogram
#======================================================#
pass

#======================================================#
# END YOUR CODE
#======================================================#

plt.ylabel('spikes/s')
plt.xlabel('Time(ms)')
```

Question:

Does the spike histogram agree with the expected firing rate profile?

Your Answer:

(c) (6 points) Count distribution

For each trial, count the number of spikes across the entire trial. Plot the normalized distribution of spike counts. Fit a Poisson distribution to this empirical distribution and plot it on top of the empirical distribution.

In [ ]:

```
#========================================================#
# YOUR CODE HERE:
#   Plot the normalized distribution of spike counts
#========================================================#
pass

#========================================================#
# END YOUR CODE
#========================================================#
plt.xlabel('spike count')
plt.ylabel('p(spikecount)')
plt.show()
```

Question:

Should we expect the spike counts to be Poisson-distributed?

Your Answer:

(d) (5 points) ISI distribution

Plot the normalized distribution of ISIs. Fit an exponential distribution to the empirical distribution and plot it on top of the empirical distribution.

In [ ]:

```
#========================================================#
# YOUR CODE HERE:
#   Plot the normalized distribution of ISIs
#========================================================#
pass

#========================================================#
# END YOUR CODE
#========================================================#
plt.xlabel('ISI (ms)')
plt.ylabel('P(ISI)')
plt.show()
```

Question:

Should we expect the ISIs to be exponentially-distributed? (Note, it is possible for the empirical distribution to strongly resemble an exponential distribution even if the data aren't exponentially distributed.)

Your Answer: