VS 265, Fall 2012
Neural Computation

Natalia Bilenko

**Total
10 points**

Problem Set 1 Solution

1. Membrane model.
   When time-varying current is injected into a cell, the equilibrium voltage is governed by this equation:

1 point

$$C_m \frac{dV}{dt} + (V - V_{leak})G_{leak} - I_{in}(t) = 0$$

Solving this differential equation, voltage can be expressed as a function of the time-varying current via convolution:

$$V = V_{leak} + \left\{ I_{in} * \frac{1}{C_m} e^{-t/\tau} \right\}$$

(a) Uniform current.

1 point

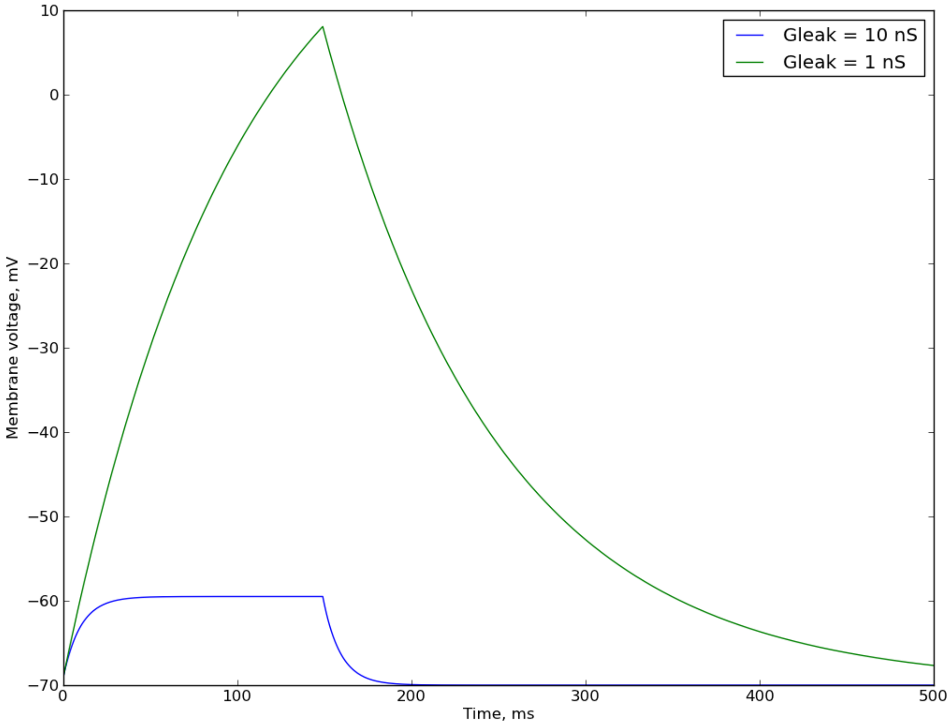


Figure 1: Problem 1(a).

0.5 point

Injecting a constant current for 150 ms causes the membrane voltage to increase from the leak potential of $-70mV$. For $G_{leak} = 10nS$, the voltage saturates at $-60mV$ and falls off after the current is turned off. For $G_{leak} = 1nS$, the voltage does not reach saturation before the current is turned off. It peaks at approximately $8mV$ and decreases after the current is turned off.

(b) Sinewave current.

1 point

1 point

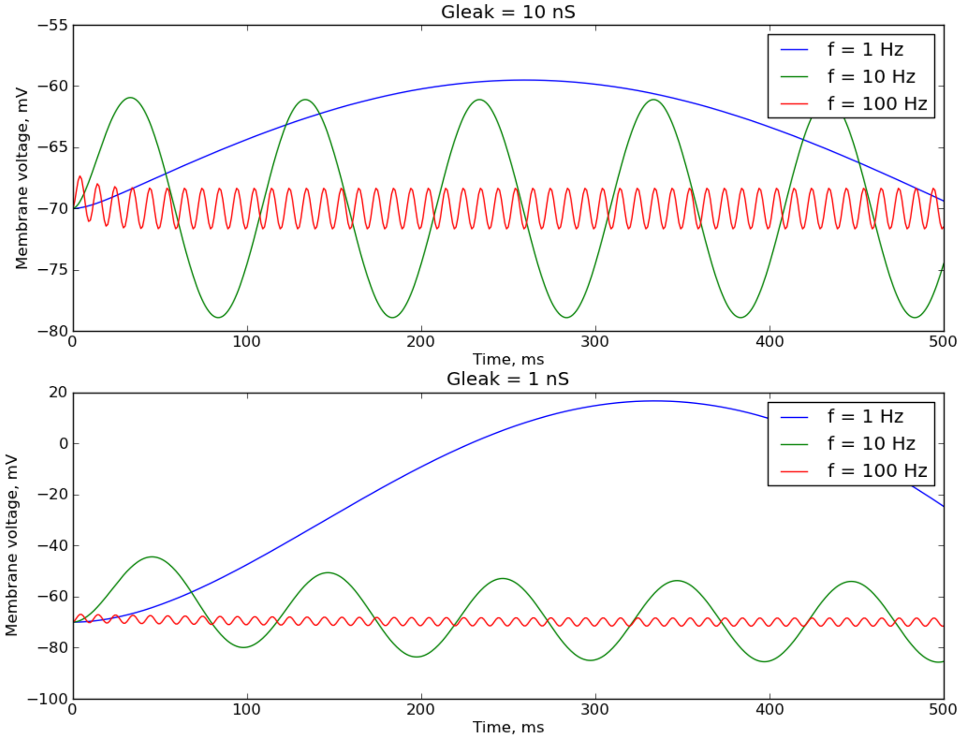


Figure 2: Problem 1(b).

0.5 point

Injecting a sinusoidal current produces a sinusoidal change in voltage. The frequency of the voltage fluctuation is equal to the frequency of the current. The amplitude and phase are dependent on the conductance: the amplitude is smaller for the higher conductance, but the phase shift is higher for the smaller conductance.

(c) White Gaussian noise.

1 point

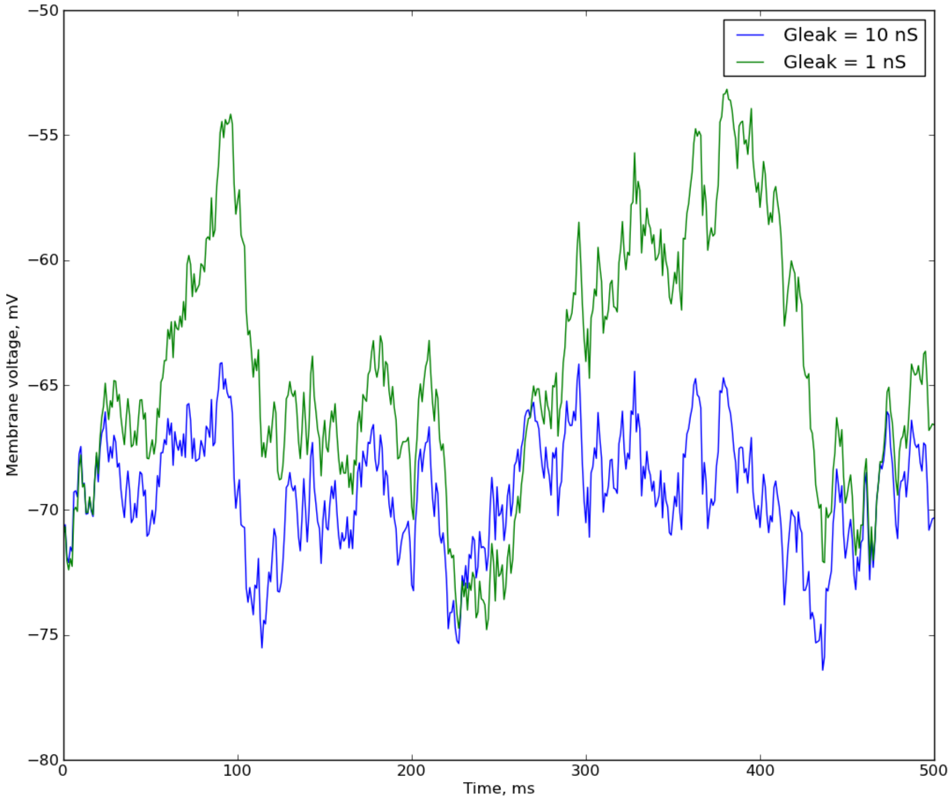


Figure 3: Problem 1(c).

Injecting a white noise current produces greater fluctuations in voltage for the lower conductance.

2. Shunting inhibition.
   At equilibrium, the membrane voltage can be calculated from the reversal potentials and conductances of the ion channels. Taking into account only the sodium and chloride channels and leak voltage and conductance, the equation for calculating the membrane potential is:

1 point

$$V = \frac{V_{leak}G_{leak} + V_{Na}G_{Na} + V_{Cl}G_{Cl}}{G_{leak} + G_{Na} + G_{Cl}}$$

2 points

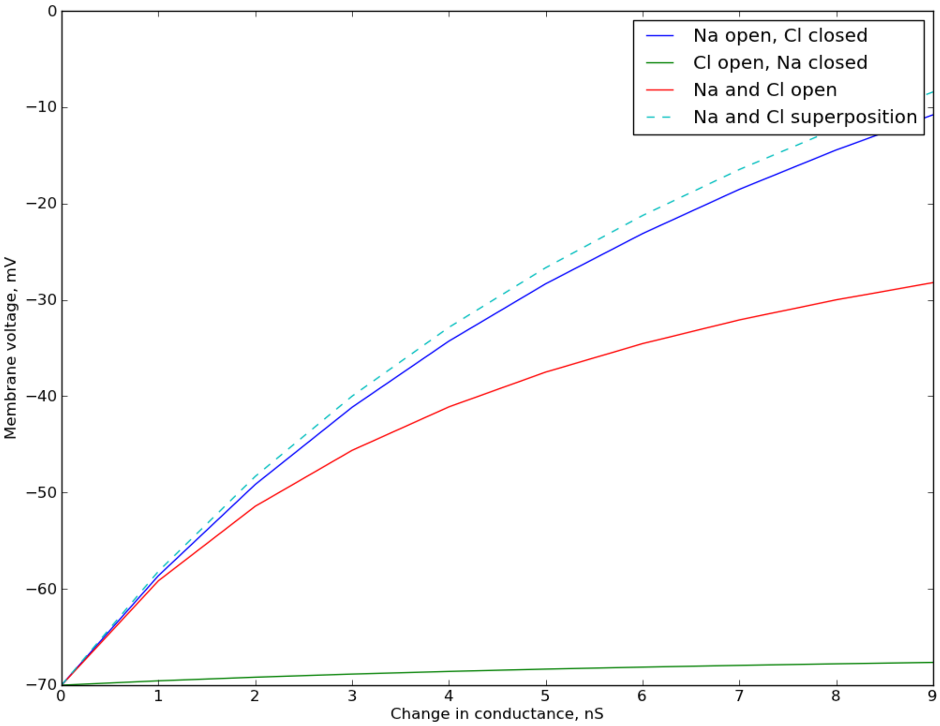
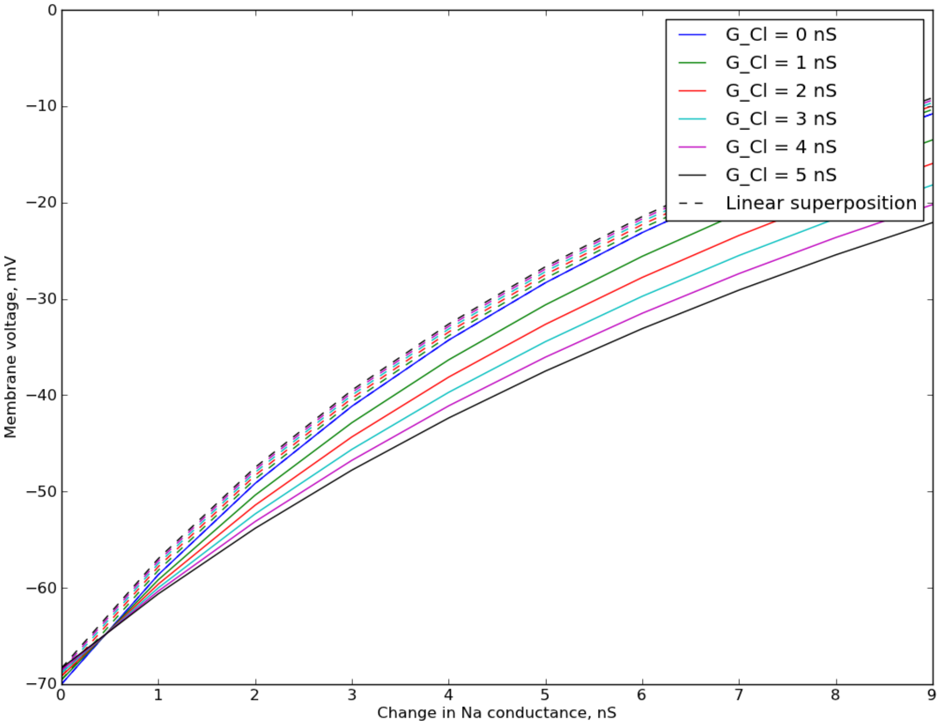


Figure 4: Problem 2.

1 point

Opening Na channels and keeping Cl channels closed produces a fairly large positive change in voltage, whereas opening only Cl channels and keeping Na channels closed produces a much smaller positive change. Opening both Na and Cl channels produces a smaller positive change than opening Na channels only. Linear superposition predicts a much higher change in voltage, suggesting that the interaction of Na and Cl channel opening is nonlinear.



Gold Star

Figure 5: Problem 2.

Greater Cl conductance inhibits the positive effect of increasing Na conductance on the membrane voltage, whereas linear superposition predicts the opposite effect. This result further highlights the nonlinear interaction between Na and Cl conductance.

Other students worked with: Alina Liberman, Anwar Nunez-Elizalde

Code:

```
import numpy as np
from matplotlib.pyplot import figure, savefig

## Problem 1.
## Given:
Vleak = -70 # leak voltage, mV
Cm = 100.e-12 # membrane capacitance, pF
Gleak1 = 10.e-9 # leak conductance 1, nS
Gleak2 = 1.e-9 # leak conductance 2, nS
tau1 = Cm/Gleak1 # time constant for conductance 1
tau2 = Cm/Gleak2 # time constant for conductance 2
nT = 501 # for 500 ms
tps = np.array(range(nT))*1.e-3 # the timepoints

## Voltage for two different conductance values.
V1 = np.zeros((nT,))
V2 = np.zeros((nT,))

## Initial condition:
V1[0] = Vleak
V2[0] = Vleak

## Problem 1a.
## Input current I0 for 0>=t>150 ms; 0 for t<0, t>150 ms
I0 = 100.e-12 # Input current, pA
Iin = np.zeros((nT,))
Iin[:150] = I0

## Voltage is equal to the convolution of injected current and 1/Cm*exp(-t/tau), plus initial voltage
Va1 = Vleak+np.convolve(Iin, 1/Cm*exp(-tps/tau1))[:nT]
Va2 = Vleak+np.convolve(Iin, 1/Cm*exp(-tps/tau2))[:nT]

fa = figure(num=1, figsize=(12, 9), dpi=80, facecolor='w', edgecolor='k')
ax = fa.add_subplot(111)
va1 = ax.plot(Va1)
va2 = ax.plot(Va2)
ax.set_ylabel('Membrane voltage, mV')
ax.set_xlabel('Time, ms')
ax.legend([va1[0], va2[0]], ['Gleak = 10 nS', 'Gleak = 1 nS'])
savefig('Problem1a.png')

## Problem 1b.

A = 100.e-12 #Input current, pA
f1 = 1 # Hz
f2 = 10 # Hz
f3 = 100 #Hz

## Input current for each frequency:
Iin1 = A*sin(2*pi*f1*tps)
```

```
Iin2 = A*sin(2*pi*f2*tps)
Iin3 = A*sin(2*pi*f3*tps)

## Calculate the current using the same formula as above, for each conductance and frequency:
Vb_g1_f1 = Vleak+np.convolve(Iin1, 1/Cm*exp(-tps/tau1))[:nT]
Vb_g1_f2 = Vleak+np.convolve(Iin2, 1/Cm*exp(-tps/tau1))[:nT]
Vb_g1_f3 = Vleak+np.convolve(Iin3, 1/Cm*exp(-tps/tau1))[:nT]

Vb_g2_f1 = Vleak+np.convolve(Iin1, 1/Cm*exp(-tps/tau2))[:nT]
Vb_g2_f2 = Vleak+np.convolve(Iin2, 1/Cm*exp(-tps/tau2))[:nT]
Vb_g2_f3 = Vleak+np.convolve(Iin3, 1/Cm*exp(-tps/tau2))[:nT]

fb = figure(num=2, figsize=(12, 9), dpi=80, facecolor='w', edgecolor='k')
axb1 = fb.add_subplot(211)
axb1.set_title('Gleak = 10 nS')
vb_g1_f1 = axb1.plot(Vb_g1_f1)
vb_g1_f2 = axb1.plot(Vb_g1_f2)
vb_g1_f3 = axb1.plot(Vb_g1_f3)
axb1.set_ylabel('Membrane voltage, mV')
axb1.set_xlabel('Time, ms')
axb1.legend([vb_g1_f1[0], vb_g1_f2[0], vb_g1_f3[0]], ['f = 1 Hz', 'f = 10 Hz', 'f = 100 Hz'])

axb2 = fb.add_subplot(212)
axb2.set_title('Gleak = 1 nS')
vb_g2_f1 = axb2.plot(Vb_g2_f1)
vb_g2_f2 = axb2.plot(Vb_g2_f2)
vb_g2_f3 = axb2.plot(Vb_g2_f3)
axb2.set_ylabel('Membrane voltage, mV')
axb2.set_xlabel('Time, ms')
axb2.legend([vb_g2_f1[0], vb_g2_f2[0], vb_g2_f3[0]], ['f = 1 Hz', 'f = 10 Hz', 'f = 100 Hz'])
savefig('Problem1b.png')

## Problem 1c.
## Injected current is Gaussian noise, st. dev = 100 pA
Iinc = 100.e-12*np.random.randn(500,)

Vc1 = Vleak + np.convolve(Iinc, 1/Cm*exp(-tps/tau1))[:nT]
Vc2 = Vleak + np.convolve(Iinc, 1/Cm*exp(-tps/tau2))[:nT]

fc = figure(num=3, figsize=(12, 10), dpi=80, facecolor='w', edgecolor='k')
axc = fc.add_subplot(111)
vc1 = axc.plot(Vc1)
vc2 = axc.plot(Vc2)
axc.set_ylabel('Membrane voltage, mV')
axc.set_xlabel('Time, ms')
axc.legend([vc1[0], vc2[0]], ['Gleak = 10 nS', 'Gleak = 1 nS'])
savefig('Problem1c.png')

## Problem 2
Gleak = 10.e-9 # leak conductance, nS
Vleak = -70 # leak voltage, mV
```

```
V_Na = 55 # Na reversal potential, mV
V_Cl = -65 # Cl reversal potential, mV

## Equilibrium potential is:
## V = (V_leak*G_leak + V_Na*G_Na + V_Cl*G_Cl)/(G_leak + G_Na + G_Cl)

## Changing the sodium conductance alone:
G_Na = np.array(range(10))*1.e-9
G_Cl = 0
Vn = (Vleak*Gleak + V_Na*G_Na + V_Cl*G_Cl)/(Gleak+G_Na+G_Cl)

## Changing the chloride conductance alone:
G_Na = 0
G_Cl = np.array(range(10))*1.e-9
Vc = (Vleak*Gleak + V_Na*G_Na + V_Cl*G_Cl)/(Gleak+G_Na+G_Cl)

## Changing both conductances together:
G_Na = np.array(range(10))*1.e-9
G_Cl = np.array(range(10))*1.e-9
Vnc = (Vleak*Gleak + V_Na*G_Na + V_Cl*G_Cl)/(Gleak+G_Na+G_Cl)

f2 = figure(num=4, figsize = (12, 9), dpi = 80, facecolor = 'w', edgecolor = 'k')
ax2 = f2.add_subplot(111)
v2n = ax2.plot(Vn)
v2c = ax2.plot(Vc)
v2nc = ax2.plot(Vnc)
v2sp = ax2.plot(Vn+Vc-Vleak, '--') # linear superposition, subtracting off Vleak since it was added twice
ax2.legend([v2n[0], v2c[0], v2nc[0], v2sp[0]], ['Na open, Cl closed', 'Cl open, Na closed', 'Na and Cl open', 'Na and Cl superposition'])
ax2.set_ylabel('Membrane voltage, mV')
ax2.set_xlabel('Change in conductance, nS')
savefig('Problem2.png')

## Changing GNa for each value of G_Cl between 0 and 5 nS:
f2all = figure(num=5, figsize = (12, 9), dpi = 80, facecolor='w', edgecolor='k')
ax3 = f2all.add_subplot(111)
G_Na = np.array(range(10))*1.e-9
leg = []
colors = ['b', 'g', 'r', 'c', 'm', 'k']
for ii, G_Cl in enumerate(range(6)):
  G_Cl = G_Cl*1.e-9
  Vnc_i = (Vleak*Gleak + V_Na*G_Na + V_Cl*G_Cl)/(Gleak+G_Na+G_Cl)
  Vsp_i = (Vleak*Gleak + V_Na*G_Na)/(Gleak+G_Na) + (Vleak*Gleak +V_Cl*G_Cl)/(Gleak+G_Cl) - Vleak # linear superposition
  vnc_i = ax3.plot(Vnc_i, colors[ii])
  vsp_i = ax3.plot(Vsp_i, colors[ii]+'--')
  leg.append(vnc_i[0])
leg.append(vsp_i[0])
ax3.set_ylabel('Membrane voltage, mV')
ax3.set_xlabel('Change in Na conductance, nS')
ax3.legend(leg, ['G_Cl = 0 nS', 'G_Cl = 1 nS', 'G_Cl = 2 nS', 'G_Cl = 3 nS', '
```

```
    G_Cl = 4 nS', 'G_Cl = 5 nS', 'Linear superposition'])
savefig('Problem2all.png')
```

hw1.py