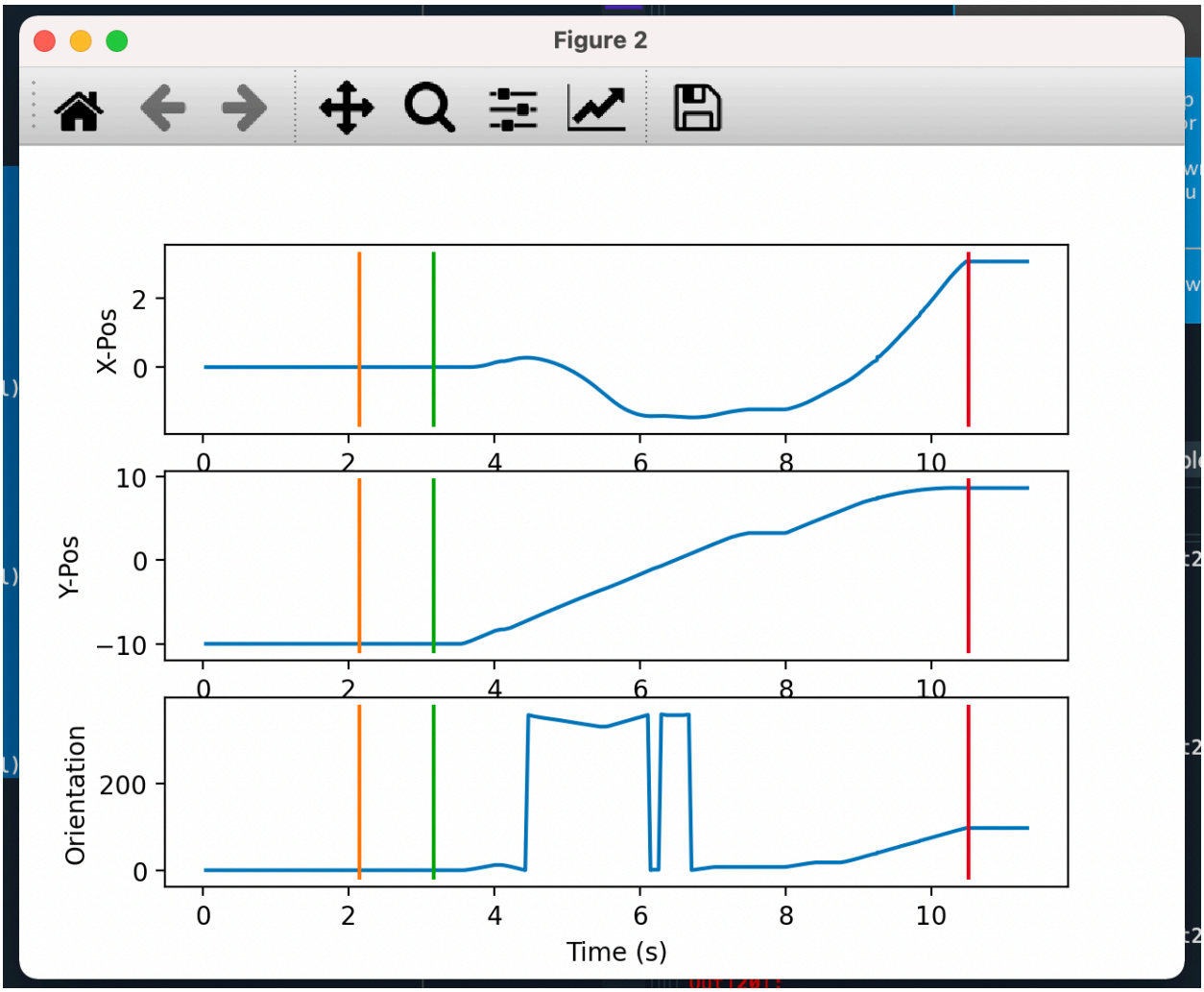
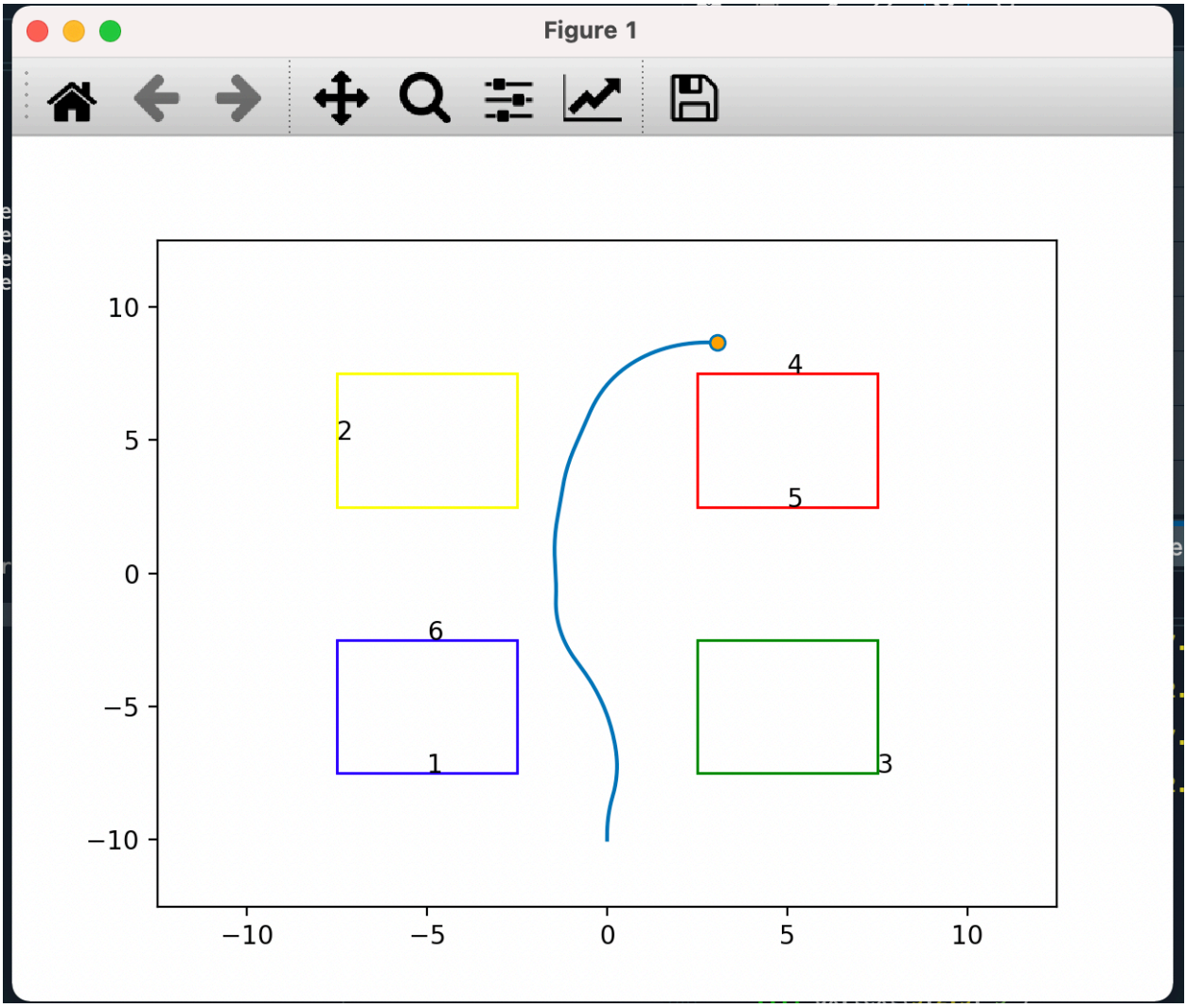
Part 2h



Part 2i

```
fig,ax = plt.subplots()
#boudaries of the maze
ax.set_xlim([-12.5,12.5])
ax.set_ylim([-12.5,12.5])
#pillars  (center point) (width, height)
rect1 = mpatches.Rectangle((-7.5,  2.5), 5, 5, linewidth=1, edgecolor='Yellow', facecolor='none')
rect2 = mpatches.Rectangle(( 2.5,  2.5), 5, 5, linewidth=1, edgecolor='Red', facecolor='none')
rect3 = mpatches.Rectangle((-7.5, -7.5), 5, 5, linewidth=1, edgecolor='Blue', facecolor='none')
rect4 = mpatches.Rectangle(( 2.5, -7.5), 5, 5, linewidth=1, edgecolor='Green', facecolor='none')
ax.add_patch(rect1)
ax.add_patch(rect2)
```

```
ax.add_patch(rect3)
ax.add_patch(rect4)
#text in the plot text(x,y,s)
ax.text(-7.5,5,'2')
ax.text(5,7.5,'4')
ax.text(5,2.5,'5')
ax.text(-5,-2.5,'6')
ax.text(-5,-7.5,'1')
ax.text(7.5,-7.5,'3')
#plot marker & markevery[0, -1] mfc markerfacecolor, mec edgecolor
plt.plot(data[uind,2], data[uind,3], marker = 'o', markevery=[-1], mfc='orange')
plt.show()
```



Part 3n

```
fig, axs = plt.subplots(2)

#data
eind = np.arange(15000)
ex = samples['gx_left'][eind]
ey = samples['gy_left'][eind]
et = samples['time'][eind]

#data wrangling
ex[ex>1920] = np.nan
ex[ex<0] = np.nan
ey[ey>1920] = np.nan
ey[ey<0] = np.nan

#plot

xval1 = messages['trialid_time'][1]
xval2 = messages['Cue_time'][1]
xval3 = messages['End_time'][1]
yl = plt.ylim(0,2000)

#plot ax1 ax2
axs[0].clear()
axs[0].plot(et,ex)
axs[0].set(xlabel='Time (s)', ylabel='Eye Position X')

axs[0].plot([xval1,xval1],yl)
axs[0].plot([xval2,xval2],yl)
axs[0].plot([xval3,xval3],yl)
axs[0].set_xlim(2203, 2219)

axs[1].clear()
axs[1].plot(et,ey)
axs[1].set(xlabel='Time (s)', ylabel='Eye Position Y')
axs[1].set_xlim(2203, 2219)

axs[1].plot([xval1,xval1],yl)
axs[1].plot([xval2,xval2],yl)
axs[1].plot([xval3,xval3],yl) #yl
```

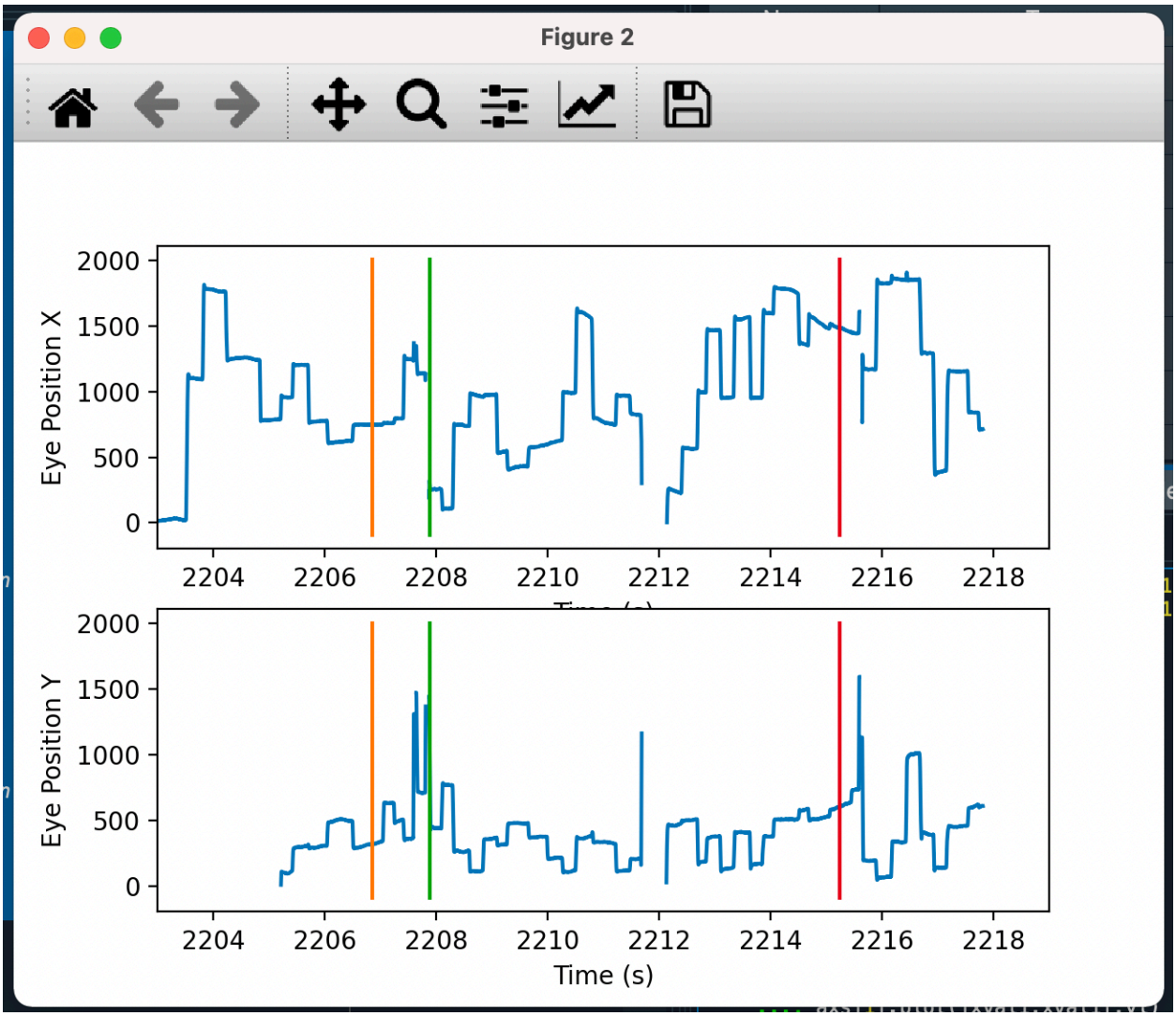


Figure 2
Eye Position X
2000
1500
1000
500
0
2204
2206
2208
2210
2212
2214
2216
2218
Eye Position Y
2000
1500
1000
500
0
2204
2206
2208
2210
2212
2214
2216
2218
Time (s)

Part 4g

```
#data
broadband_data = data[0:500000]
broadband_data = np.squeeze(broadband_data)  #y axis
time = [i/30000 for i in range(0,500000)]  #x axis
rawtimes = ev_rawtimes/30000

plt.figure()
plt.plot(time, broadband_data)
plt.xlabel('Time (s)')
plt.ylabel('Voltage (uV)')

#markers
byl = plt.ylim()
mi2 = np.array(list(map(int, ev_markers[0:9]))).nonzero()
bt2 = rawtimes[mi2[0][1:4]]
bpt2 = np.kron(np.ones((2,1)),bt2)
bpy2 = np.kron(np.ones((np.size(bpt2,1),1)),byl).transpose()
plt.plot(bpt2,bpy2)
```

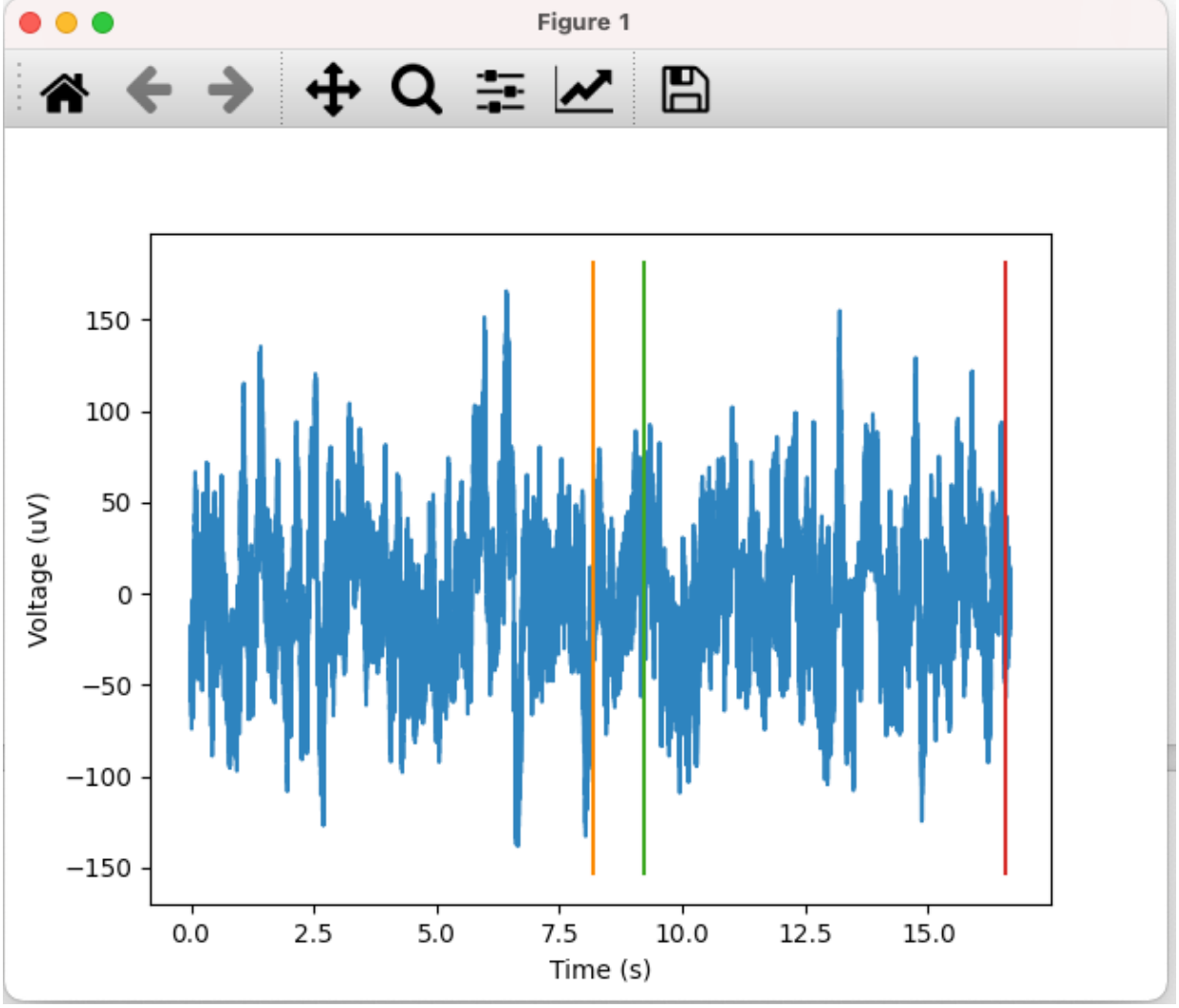


Figure 1
150
100
50
0
−50
−100
−150
Voltage (uV)
0.0
2.5
5.0
7.5
10.0
12.5
15.0
Time (s)

Part 5a

```
f, ax = plt.subplots(3,1)

ax[0].set_xlabel('Time (s)')
ax[0].set_ylabel('X-Pos')
ax[0].plot(ut,unity_data[uind,2])
ax[0].plot(pt2,py2)

ax[1].set_xlabel('Time (s)')
ax[1].set_ylabel('Eye Position X')
ax[1].plot(et,ex)
ax[1].plot([xval1,xval1],yl)
ax[1].plot([xval2,xval2],yl)
ax[1].plot([xval3,xval3],yl)


ax[2].set_xlabel('Time (s)')
ax[2].set_ylabel('Voltage (uV)')
ax[2].plot(time, broadband_data)
ax[2].plot(bpt2,bpy2)
```

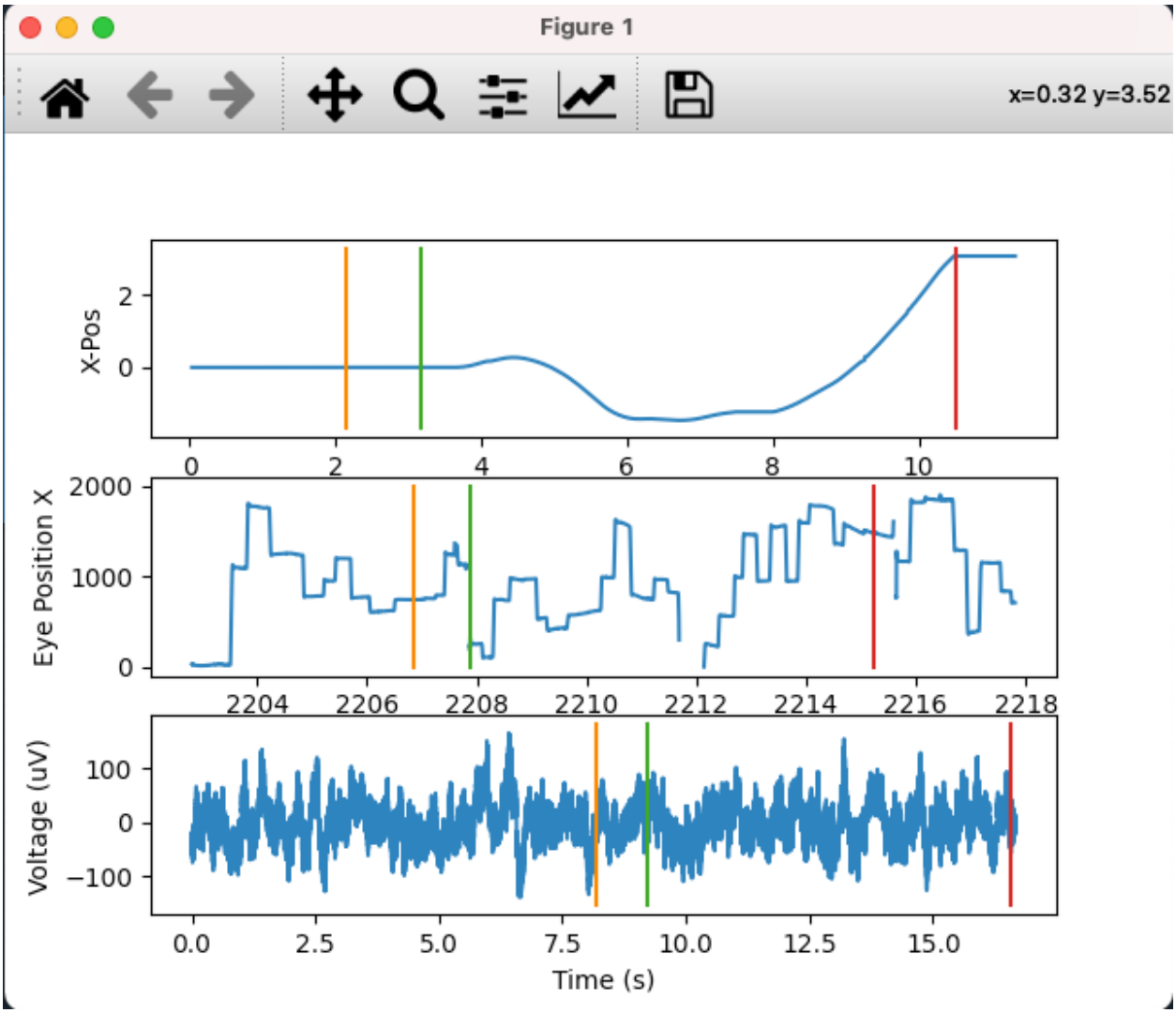
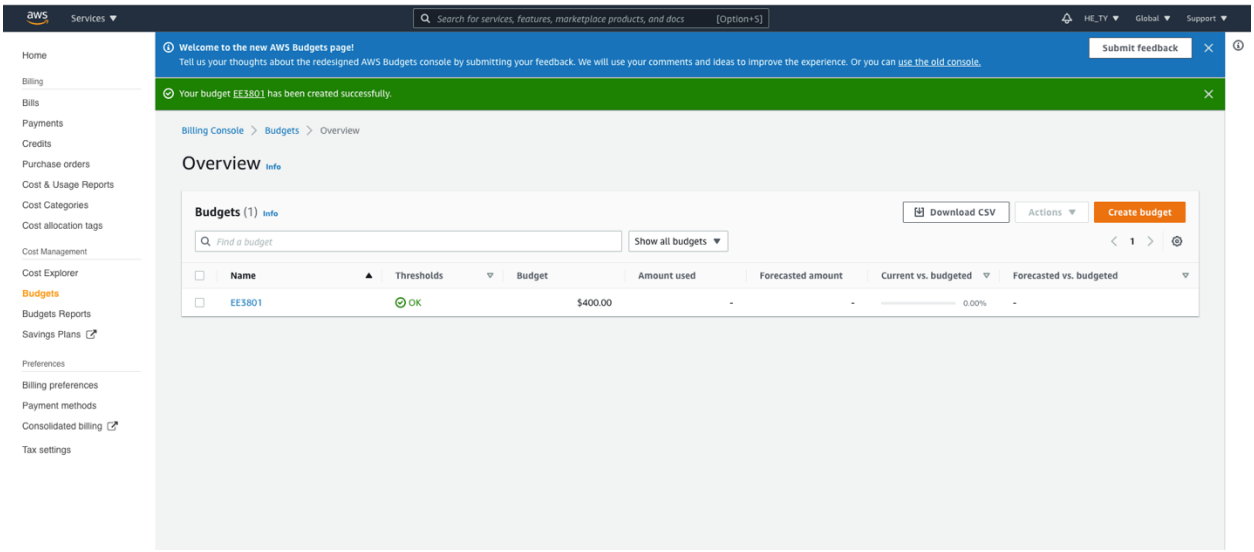
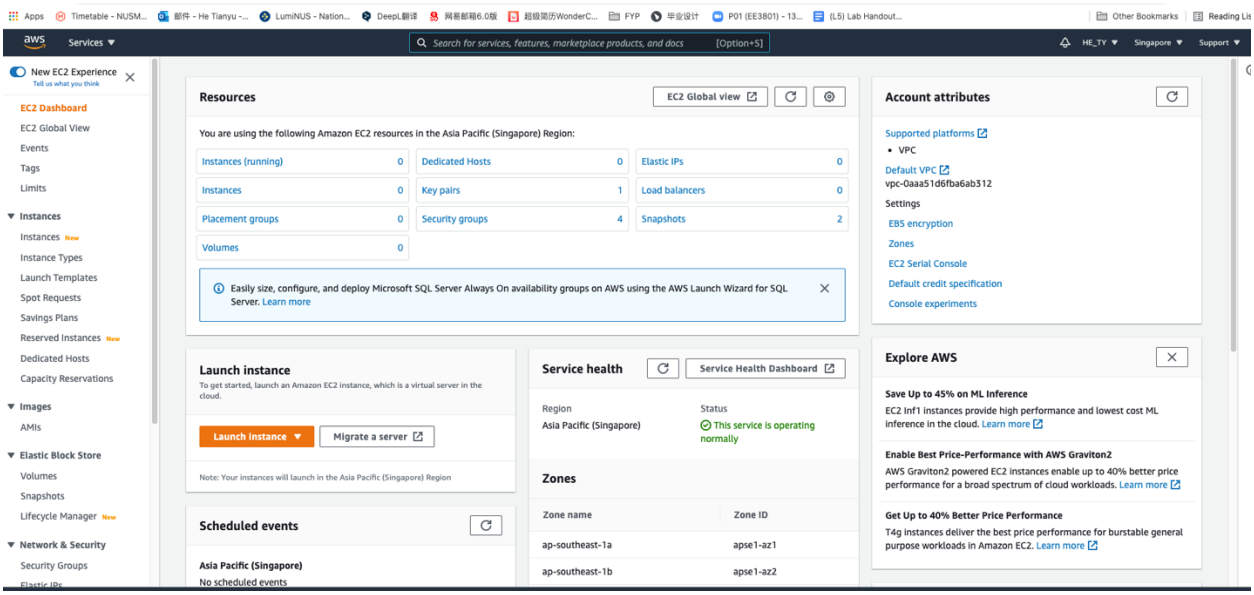


Figure 1
x=0.32 y=3.52
X-Pos
2
0
0
2
4
6
8
10
Eye Position X
2000
1000
0
2204
2206
2208
2210
2212
2214
2216
2218
Voltage (uV)
100
0
−100
0.0
2.5
5.0
7.5
10.0
12.5
15.0
Time (s)

Part 6



EC2 Dashboard



Elastic Compute Clouds-charges

AWS Credits