Homework 1 submission

ECET 512 — Wireless Systems

Chris Uzokwe
ID: cnu25

January 21, 2021

1 Submitted files

For this assignment, besides this report, the following archives were created:

1.1 SRC Folder

+ *"main.py"*: This **script** will run a simulation where we can see a mobile user moving through a set of cells and at any given time, we can see which cell is serving the mobile user. At the top of the script the user can define the number of cells **N**, the **cell radius** and the **center of the main cluster**. What's more, the user can define whether he wants to record a video and see a plot of the friis equation for the received power or not.

1.2 DOC Folder

+ *"anim.gif"*: This gif shows how a mobile user moves through a set of 7 seven clusters of 7 cells each. There is a blue link between the mobile user and the serving cell at any given time, as long as the user is inside of the range of a cell.

+ *"3.png, 4.png and 7.png"*: These images represent the different clusters for different N.

2 Code execution

The code for this homework has been developed with Python. To run it, the user has to open the folder "src" and run the script *"main.py"*. This script has all the necessary calls to the functions developed for the homework and will generate a simulation of the user moving through the cluster of cell clusters for different values of N. The value of N, along with the center of the main cluster and the cell radius can be defined at the top of the main script.

If one wants to draw the clusters without running the simulation, this can be done by calling the function *drawCluster()*.

3 Homework 1 Solution

3.1 Co-channel plots for interfering cells when N = 3, 4 and 7

Running drawCluster() plots the clusters around each cell. After getting the co-channel locations using channelCenters(), we draw a cluster at each point.

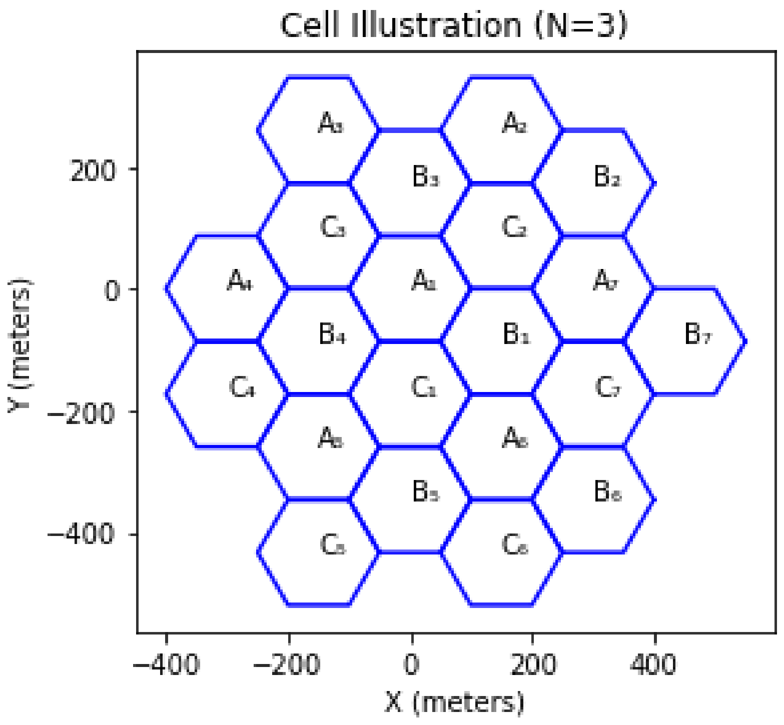


Figure 1: Co-channel plot for N=3.

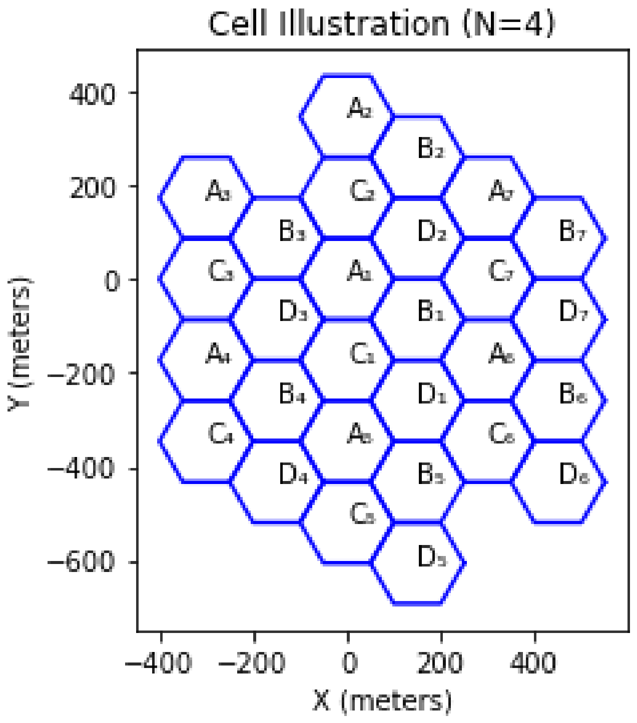


Figure 2: Co-channel plot for N=4.

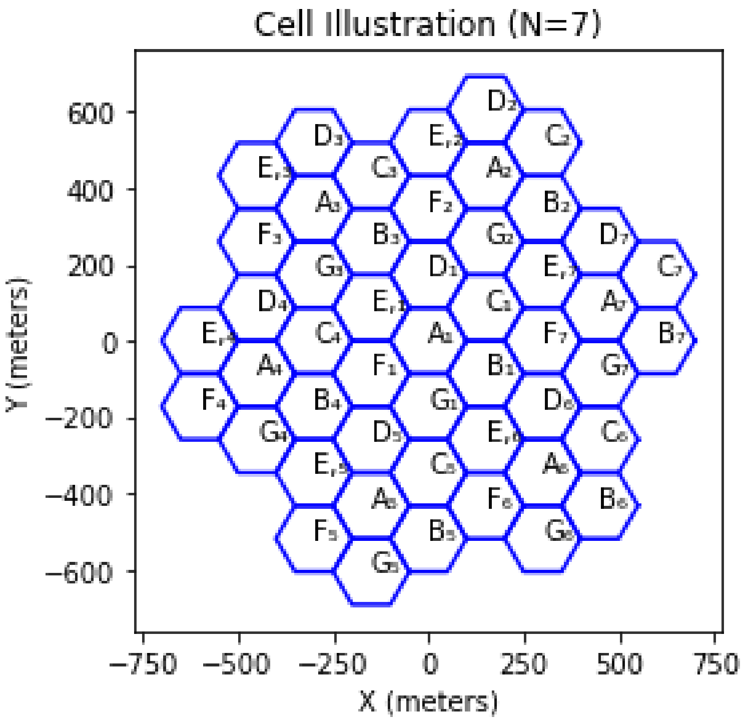


Figure 3: Co-channel plot for N=7.

3.2 Capturing a mobile user through channels

We can see how a user moves through a cell channel. Below is a screenshot but the full video is available in the src. Folder.

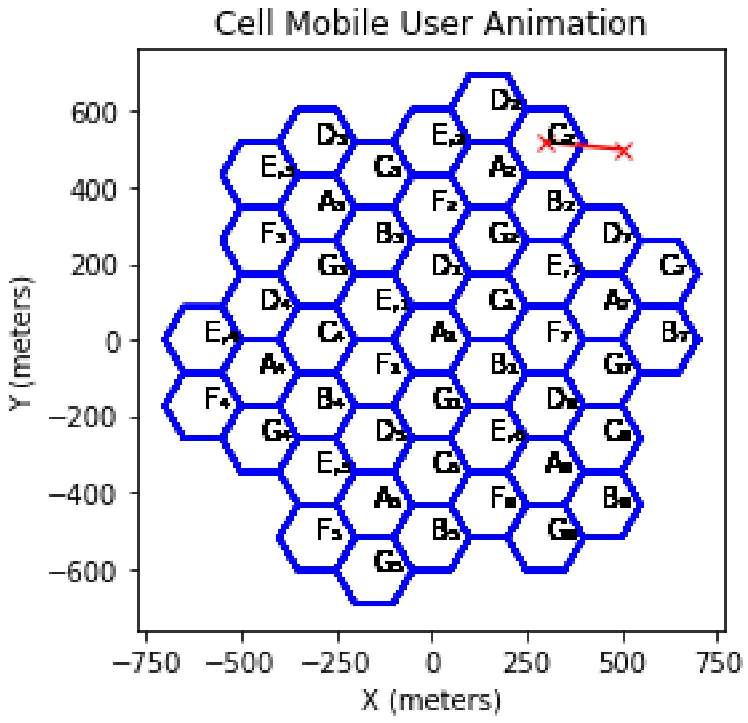
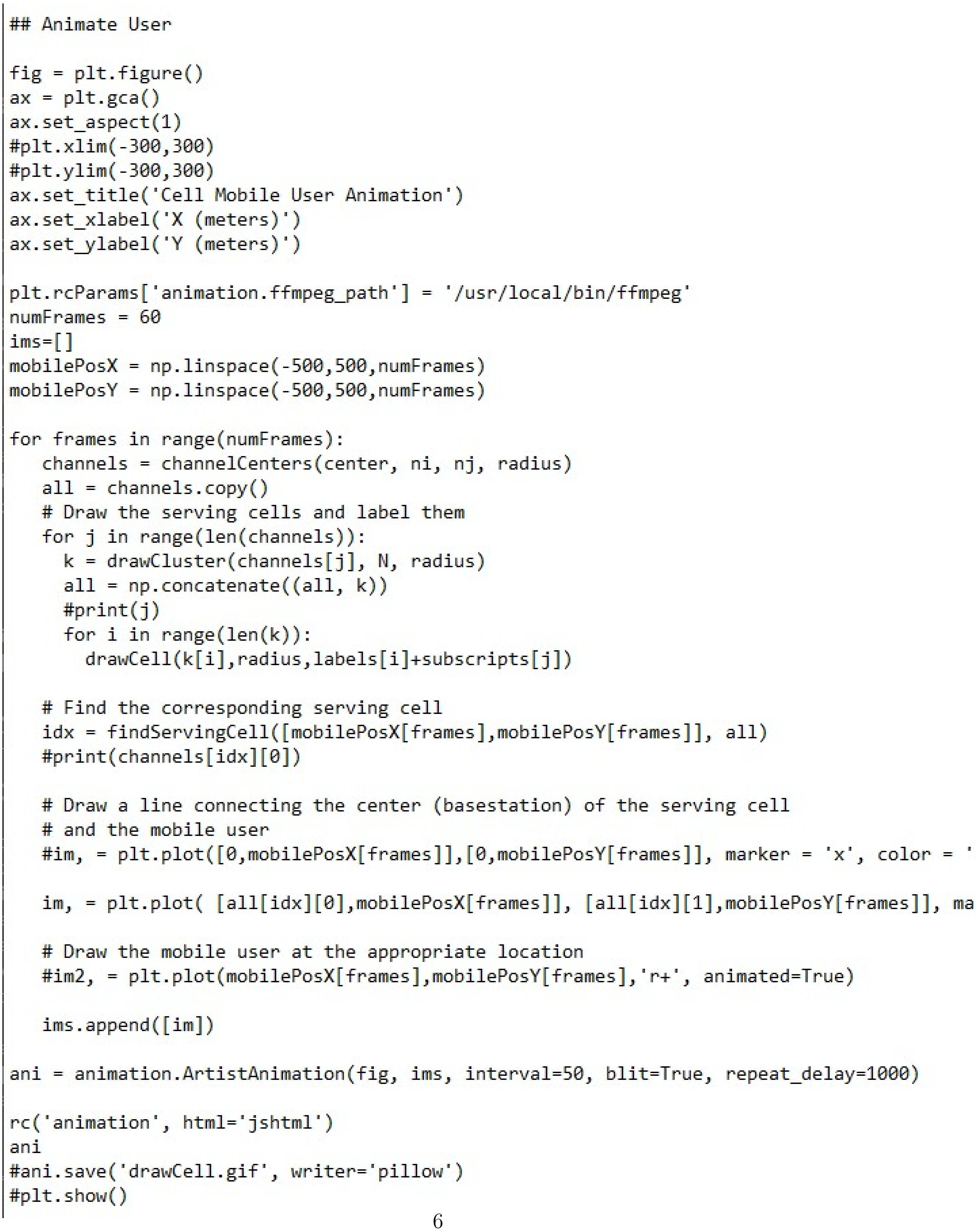


Figure 4: Co-channel mobile user animation for N=7.

4 Python code



```
## Animate User

fig = plt.figure()
ax = plt.gca()
ax.set_aspect(1)
#plt.xlim(-300,300)
#plt.ylim(-300,300)
ax.set_title('Cell Mobile User Animation')
ax.set_xlabel('X (meters)')
ax.set_ylabel('Y (meters)')

plt.rcParams['animation.ffmpeg_path'] = '/usr/local/bin/ffmpeg'
numFrames = 60
ims=[]
mobilePosX = np.linspace(-500,500,numFrames)
mobilePosY = np.linspace(-500,500,numFrames)

for frames in range(numFrames):
   channels = channelCenters(center, ni, nj, radius)
   all = channels.copy()
   # Draw the serving cells and label them
   for j in range(len(channels)):
     k = drawCluster(channels[j], N, radius)
     all = np.concatenate((all, k))
     #print(j)
     for i in range(len(k)):
       drawCell(k[i],radius,labels[i]+subscripts[j])

   # Find the corresponding serving cell
   idx = findServingCell([mobilePosX[frames],mobilePosY[frames]], all)
   #print(channels[idx][0])

   # Draw a line connecting the center (basestation) of the serving cell
   # and the mobile user
   #im, = plt.plot([0,mobilePosX[frames]],[0,mobilePosY[frames]], marker = 'x', color = '

   im, = plt.plot( [all[idx][0],mobilePosX[frames]], [all[idx][1],mobilePosY[frames]], ma

   # Draw the mobile user at the appropriate location
   #im2, = plt.plot(mobilePosX[frames],mobilePosY[frames],'r+', animated=True)

   ims.append([im])

ani = animation.ArtistAnimation(fig, ims, interval=50, blit=True, repeat_delay=1000)

rc('animation', html='jshtml')
ani
#ani.save('drawCell.gif', writer='pillow')
#plt.show()
```