Homework 2 submission

ECET 512 — Wireless Systems

Chris Uzokwe
ID: cnu25

January 26, 2021

1 Submitted files

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

1.1 SRC Folder

+ *"main.py"*: This **script** extends the simulation done in homework 1 to calculate the path loss from the exponent model. The script includes two functions *exppathloss* and *expshadowpathloss* which calculate the loss based on the **distances** and the **path loss exponent**.

1.2 DOC Folder

+ *"FriisRecievedPowerModel.png"* A graph of the mobile user recieved power based on the Friis Pathloss Model.

+ *"DownlinkSignalPowervsTime.png"* A graph of the mobile user recieved power based on the Exponent path loss Model.

+ *"drawCellFTD.gif and drawCellFTI.gif"* Animation of basestation and interfering cells as the mobile user moves through the first tier of cells.

+ *"DownlinkInterferencePowervsTime.png"* A graph of the mobile user recieved interference based on the Exponent path loss Model.

+ *"DownlinkSIRvsTime.png"* A graph of the mobile user signal to interference ratio based on the Exponent path loss Model.

2 Friis Path Loss Propagation

A graph of the mobile user received power based on the Friis Pathloss Model, which was to be included in Homework 1.

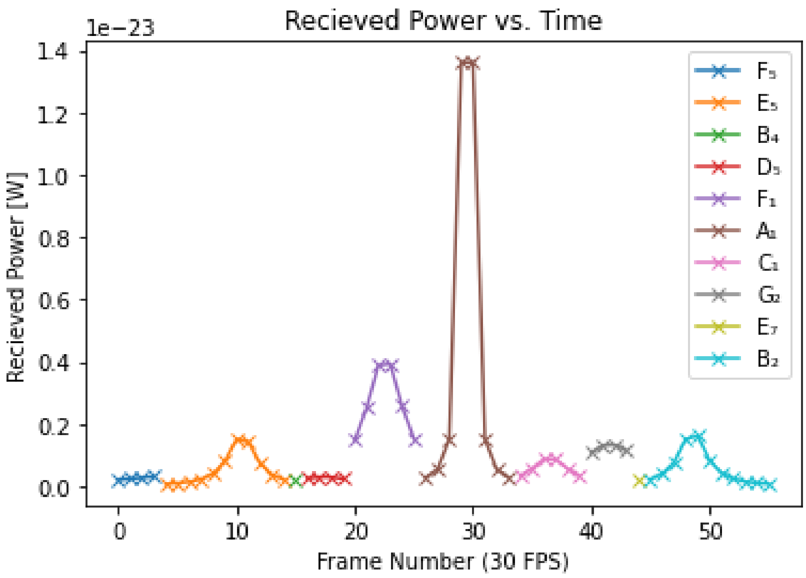

Figure 1: Downlink Signal Power vs. Time (Friis Model)

3 Homework 2 Solution

3.1 Exponent Path Loss Propagation

Downlink Signal Power vs. Mobile User Position: we can see that the shadowing effects cause the power to deviate by the same amount for both exponent models. Since in our case our random recieved power was positive, the ending signal power recieved greater power.

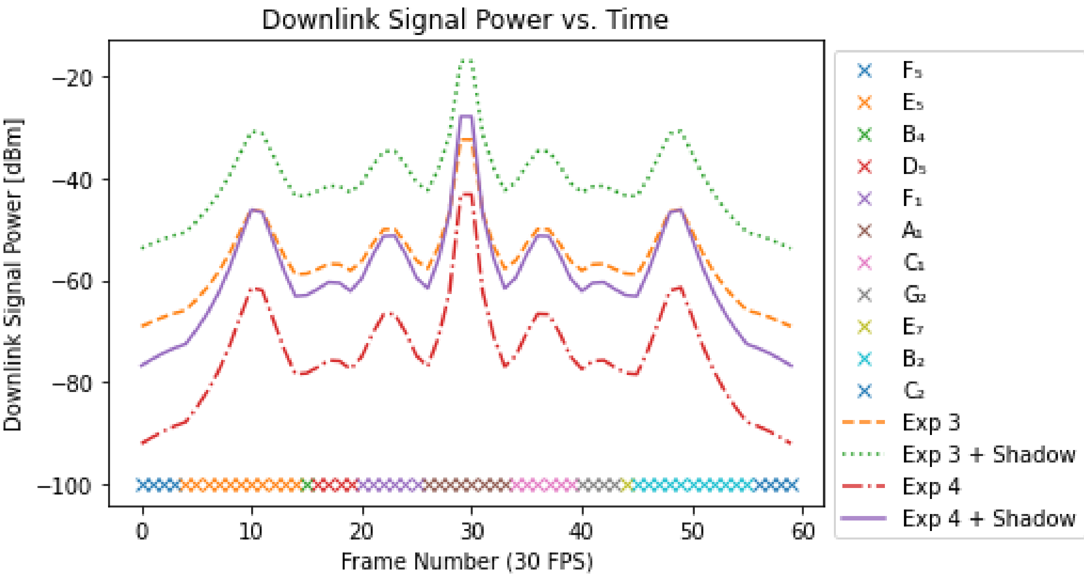

Figure 2: Downlink Signal Power vs. Time (Exp Model)

3.2 Downlink Interference Power and Cell Animation

Downlink Interferenve Power and Animation: We can see that from the intereference, there is essentially a barrier that must be over come, as there exists slight interference at all times from the upper tier channels.

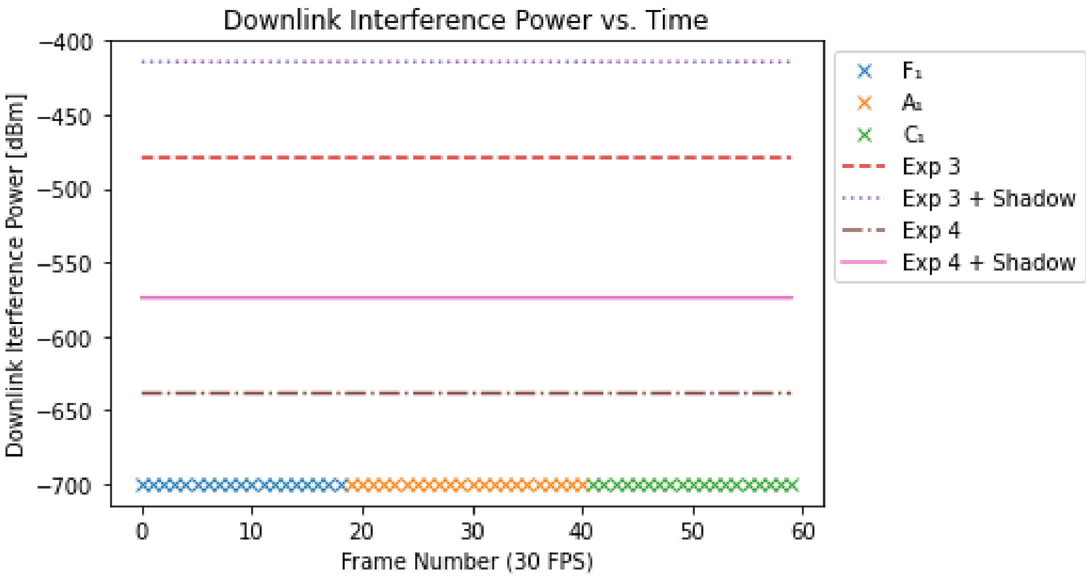

Figure 3: Downlink Interference Power vs. Time

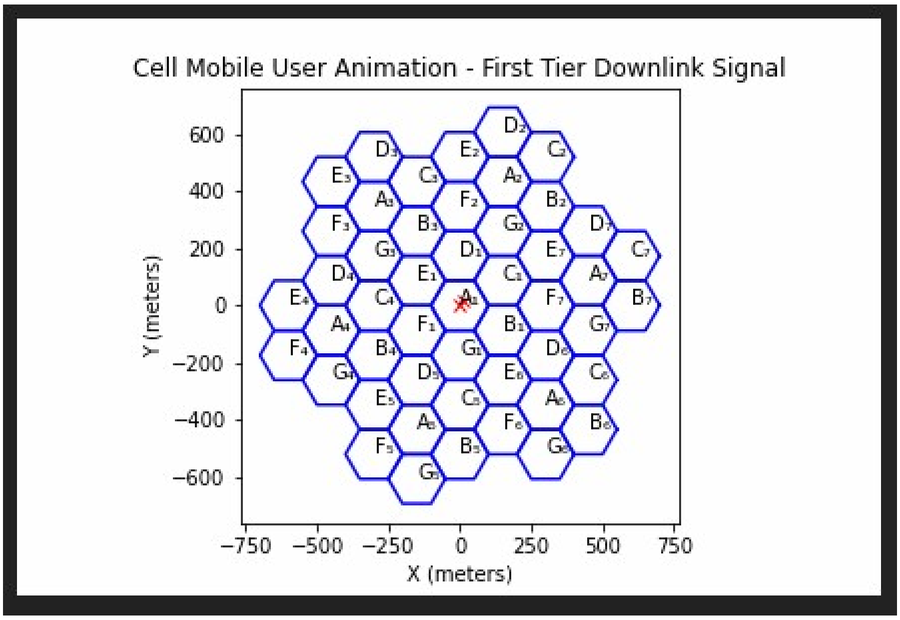

Figure 4: Base signal downlink.

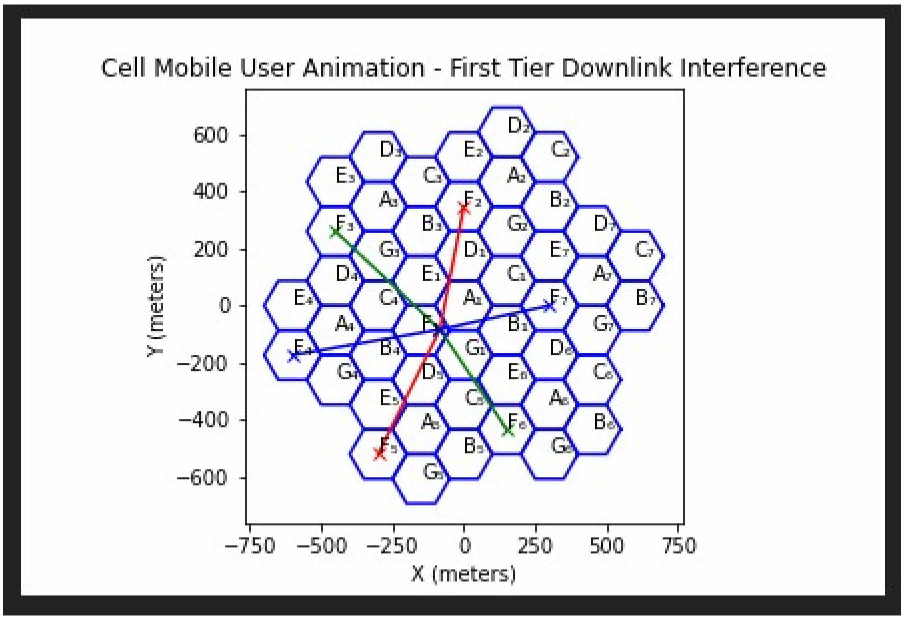

Figure 5: Co-channel interference.

3.3 Downlink Signal To Interference Ratio

We can see from the SIR graph that as a user gets closer to the center of a cell, there is less possibility for interference, as the power emitted from the serving basestation overcomes all interference thresholds. We can equate this to the idea of a link budget because depending on where the user is

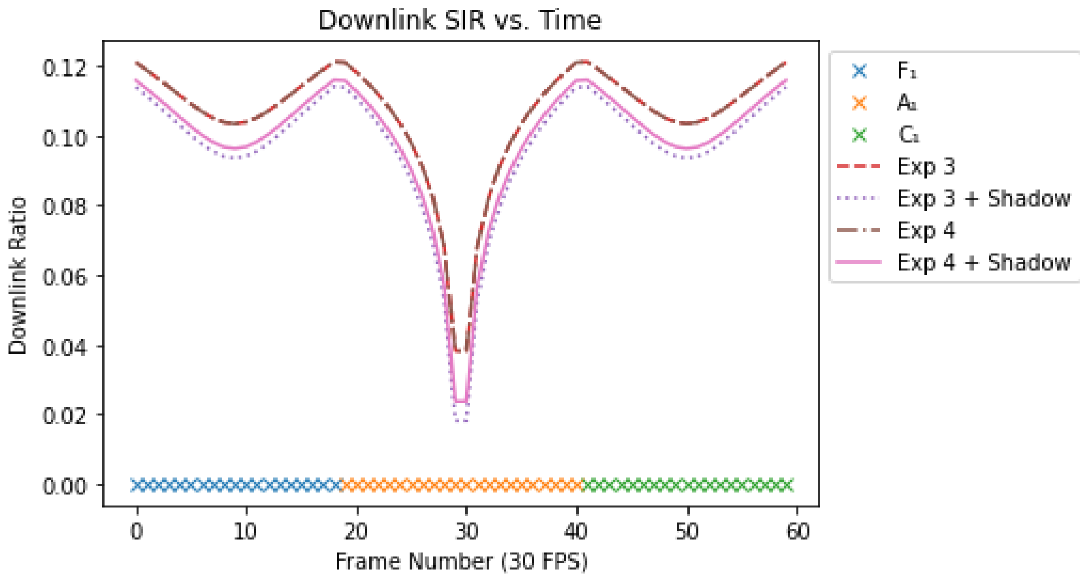

Figure 6: SIR as user moved through cell tier 1.

4 Python code

```
def friispathloss(wavelength, d):
  return (wavelength/(4*np.pi*d))**2

def exppathloss(d, B):
  return 1E-3 - 10*B*np.log10(d)

e = np.random.normal(0, 8, 1)

def expshadowpathloss(d, B):
  return 1E-3 - 10*B*np.log10(d) + e

targetcells = np.array(targetcells)
powers = np.array(list(map(friispathloss, targetcells[0:, 2], distances)))

powers.resize((60,1))
targetcells = np.concatenate((targetcells, powers), axis=1)

cellnum = targetcells[0][1]
splits = []
div = 0

for i in range(len(targetcells)):
  if targetcells[i][1] == cellnum:
    div += 1
  else:
    splits.append(div)
    div = 1
    cellnum = targetcells[i][1]

idx = 0
for i in splits:
  plt.plot(range(idx,idx+i), targetcells[idx:idx+i,3:4], 'x', linestyle='-' ,label=targetcells[idx][1])
  idx = idx+i

plt.title('Recieved Power vs. Time')
plt.xlabel('Frame Number (30 FPS)')
plt.ylabel('Recieved Power [W]')

plt.legend()
plt.show()

p3 = np.array(list(map(exppathloss, distances, np.full(
    shape=len(distances),
    fill_value=3,
    dtype=np.int))))

p3s = np.array(list(map(expshadowpathloss, distances, np.full(
    shape=len(distances),
    fill_value=3,
```

Figure 7: Snippet of main.py.