Homework 3, Due Thursday May 2, 2024

Instructions: Please use File→ Make a Copy to save your own copy of this assignment document. Please insert your answers inline, below each question. When you are ready to submit, please make a pdf of your document and submit that in Canvas.

Your name: Michael Ibrahim

Install Jupyter Notebook. [Link to instructions.]
Download and save `rtl_sdr_fm_audio_2024_hw3.ipynb`

Additional Setup: If you run into `librtlsdr` import errors, you may need to download the [librtlsdr dependencies] (use the most recent version for your platform). Once downloaded and extracted, ensure that the file location is in your PATH for the runtime you're using.

With your RTL SDR plugged in, you will be running each cell in the `rtl_sdr_fm_audio_2024_hw3.ipynb` notebook (link above). You will need to download this notebook from Canvas. Read through the questions below first. You may need to modify values in certain cells before running them. You also may find yourself iterating through these questions a few times and trying different values. These questions are intended (1) to direct your attention to interesting knobs (variables) in the code that you may want to play with, (2) to verify that you have gotten the code running, and (3) to help debug if your system isn't working...some of the answers to the questions may indicate problems.

[1 point]
Q1) What is the initial value assigned to `fsps`? (It appears as a formula in the code...I am asking for the numerical value). By initial value, I mean the value in the code as I've given it to you. You will be changing the value later.

A1) 1048576

[1 point]
Q2) What is the initial value assigned to `faudiosps`?

A2) 48000

[1 point]
Q3) What value of `fc` are you using? Set `fc` to a strong station in your area, for example `94.9e6` for 94.9MHz, which is KUOW, UW's public radio station. (`fc` is the carrier frequency, ie the frequency of the station you want to listen to.) Note that `fc` is initially set to a different value.

A3) 94.9e6

[1 point]
Q4) What is the initial value of `Tmax`? This is the time duration that you will be recording.

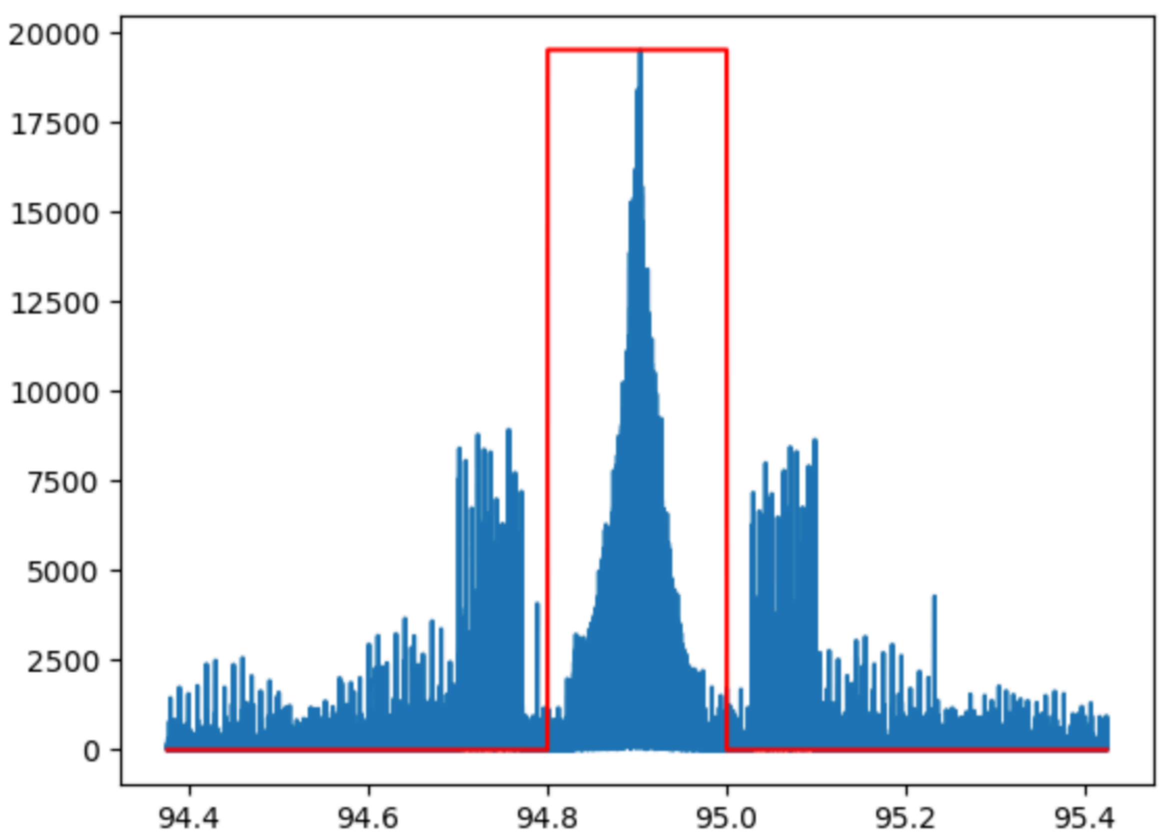
A4) 2.5 s

[1 point]
Q5) Run the cell that starts with comment "`# Create and plot the bandpass mask`."
What value of `fcutoff` is assigned in this cell? The value `fcutoff` represents the frequency above which the signal will be filtered by the next cell.

A5) 100 kHz

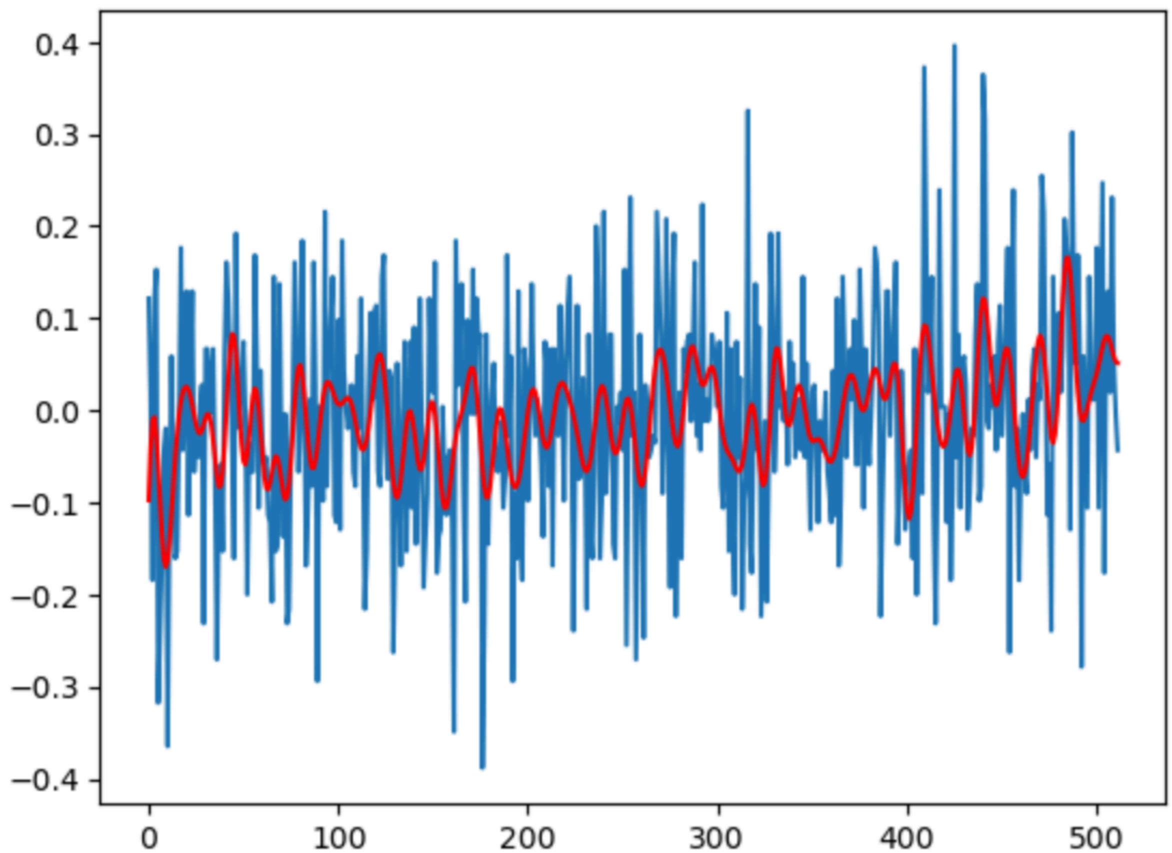
[1 point]
Q6) Take a screenshot of the plot made by this cell (the one that starts with comment "`# Create and plot the bandpass mask`". You can scale it to be not too big… maybe 2.5 to 3 inches across.

A6)



[1 point]
Q7) After you've run the cell that starts with the comment "`# Compare time domain view of filtered and un-filtered signal`," take a screenshot of the plot it creates and paste it in here:

A7)



[1 point]
Q8) What value of downsampling factor will be used? You can find the cell that includes `print("Down sampling factor is", dsf)`

A8) 22

[4 points]
Q10) Listen to the audio. How does it sound? Now go back up to the cell that starts with "`# Create and plot the bandpass mask`" and change `fcutoff` to 200000. Does the audio sound worse or better now? (Note that you will still be using the samples recorded earlier.) You may want to switch back and forth between the original value (the answer to question 6), and the new value. What is the difference in the audio quality, and why does changing `fcutoff` have this effect? Leave `fcutoff` at its original value (your answer to question 6) when you finish this question.

A10) It sounds normal, after changing the fcutoff, it sounds a little raspier I guess? Probably a little more noise is included.

[2 points]
Q11) In the cell labeled "`# Play audio`", insert a new line at the beginning of the cell "`faudiosps = 60000.`" Execute the cell to play the audio with this modified parameter setting. What happened to the audio? Why?

A11) When the new line faudiosps = 60000 was inserted at the beginning of the cell and executed, the audio became higher pitched. Increasing the sampling frequency without altering the data means that the samples are played back more quickly. This accelerated playback results in a higher pitch, making the audio sound faster and more chipmunk-like. The pitch shift happens because the audio waveform cycles are squeezed into a shorter timeframe, effectively increasing the frequency of the sound waves.

[8 points]
Q12) Restore `faudiosps` to its original value of `48000`. The downsampling portion of the notebook currently has two methods: a first one where 1 out of every `dsf` samples is kept, and a second one where blocks of `dsf` samples are averaged together. Make a new downsampling routine that is kind of a hybrid of these two. Instead of averaging block by block as in the existing code, use `np.convolve` to do a *moving average* with a window size of `dsf` samples. Then keep 1 just out of every `dsf` samples. Does this sound better, worse, or the same as the blockwise averaging scheme? [I haven't tried this, so I don't actually know…there isn't a right answer about how it sounds.]

A12)

To me personally, it sounds like there is more noise picked up using the new method compared to the blockwise averaging method. I don;t know if other outside variables came into play, as I used different recording examples for comparing the two.