ECE 4670 Lab Report Grading

Lab 6: Software Defined Radio and the RTL-SDR USB Dongle; *Abbreviated for Spring 2022*

Points	Lab Exercise Number	Laboratory Exercise Description	Check off
		Overview of the RTL-SDR Dongle	
5		Explain why the output signal of the mixer will be complex. (Question arises on page 6)	
		Using the RTL-SDR Dongle with SDR#	
2	Part 1	Tune in NOAA, 162.475 MHz and play through the PC speakers, demo to instructor (set up for NFM)	
2	Part 1	Change the demodulator filter bandwidth, note the setting that gives the best audio fidelity.	
2	Part 1	Calibrate your dongle to this known frequency station (NOAA, 162.475 MHz). Note the value setting. (Note, be sure your dongle has been running for at least 5 minutes)	
2	Part 2	Note the frequencies of at least 3 WFM stations, do you see a BW of 200 KHz?	
2	Part 2	Make note of each stations program type, stereo or HD	
2	Part 2	Discuss what the difference is in the spectra of stereo and an HD broadcast.	
0	Part 3	Setup for 70 MHz FM modulated tone (as shown in fig. 13)	
2	Part 3	Verify reception using the SDR. Can you hear the tone?	
2	Part 3	Reduce carrier amplitude verify it eventually fades away, try turning on and off the AGC and/or manually adjust the gain to compensate for the reduced signal received. Can you get the signal back?	
2	Part 3	Switch from FM to AM of the modulation function generator and verify reception, which has a better audio tone quality? (Be sure to change the SDR to AM demodulation)	
		A brief introduction SDR signal processing using Python	
22		Weather Channel custom receiver (inside the sample notebook ZIP as `wx_custom_receiver.ipynb` and in PDF). (1) Capture 10s of complex IQ waveform data using the RTLSDR tuned to 162.475 MHz with sampling rate 2.4 Msps. (2) Design two custom lowpass filters using fir_design_helper to satisfy the needs of NBFM demodulation as described in the block diagram. (3) Characterize the filters in terms of the frequency response magnitude in dB versus frequency in Hz. (4) Process the captured IQ samples through the Python code cell to ultimately recover the message waveform at 48 ksps.	

		(5) Adjust the spectrum centering variable `f_lo` (6) Listen to the recovered audio and compare it with results you obtained using SDR# in Part 1 above. Save the audio array to a .wav file and include in your report as an attached file, e.g., use `sk_dsp_comm.sigsys.to_wave()`.	
45		Total Points	