Lab02- Decoding Signals

In this lab, we will analyze a data file that has FM RF data contained within. Your mission is to load the data file and decode the audio that's contained within. As a process of researching FM demodulation, you'll find a great deal of information on the various controls that are needed to demodulate FM signals. In addition, we'll look at some simple methods of demodulating digital data from an RF capture.

Note: for this lab, since it requires audio, you can't use the IA Lab. The lab's web interface does not allow the routing of audio, so you'll need a mechanism to run things locally.

Part I: Demodulate Audio

Use the following information about the capture for your demodulation:

- The data is wideband frequency modulated
- The capture is already centered on the broadcast band
- The sample is captured at 2 million samples/second
- You can find the capture in the D2L dropbox.

Instructions

Perform the following tasks using the rfToolkit VM or hardware on your own system (or just make your own VM). Unfortunately, the IA Lab will not work as audio isn't transmitted from the virtual machines.

1. Setup a system with GNU Radio and GNU Radio Companion or use the VM provided in D2L
    a. GNU Radio is typically happiest on Linux
    b. If you fell in love with BSD in 840, please consider breaking up
2. On your canvas, load the data file and demodulate it using the demodulate module.
    a. You're welcome to investigate demodulating it manually as well
    b. Many tutorials exist online for demodulating the signal of a Software Defined Radio. You can use those, just swap the file source for the RTL-SDR source (you may need to add a throttle with the signal rate too)

Tips for Linux VM users!

Running GNU Radio in a Linux VM is a great way to do this without having to tinker much with your computer. However, there's a good chance these days that your Linux instance will suffer from audio buffer underruns (sort of the opposite of an overflow), especially if you're using Alsa Audio (you probably are). Pop this config into your `~/.gnuradio/config.conf` file (if the file doesn't exist, create it):

```
[audio_alsa]

default_input_device = default

default_output_device = default

#period_time = 0.010           # in seconds (default)

period_time = 0.100           # in seconds

nperiods = 4           # total buffering = period_time * nperiods

#verbose = false
```

verbose = true

Part II: Demodulate Digital Data

Audio is fun and all, but the world around us functions through the transmission of digital data. Rarely are these common devices using any type of encryption—or even basic obfuscation. With the exception of modern-ish car keys and garage door openers, most small/portable electronic devices transmit simple amplitude modulated transmissions (referred to on-off keying) as the spectral use is very quick and efficient.

What to do: Part IIA

We have two captures in question! One is already captured, the other needs to be discovered. Using SDR# (to capture the signal) and other various tools (to analyze the signal), work through the following scenario and answer the questions based on it

For the mystery device:

- A device is transmitting a signal on an ISM band nearby our radio
- Using SDR#, connect to one of the radios that we used in the first lab (remember, the SDR# address is: sdr://138.247.12.20:5555)
- We know the FCCID of the device: M3N-A2C31243300. Important: this particular device uses amplitude modulation.
- Based on the FCCID, take a screenshot showing that you've discovered the frequency that the device would be transmitting on
- Save the transmission!

For both transmissions:

- Opening the transmission in an appropriate tool, identify the transmission's waveform. Include a screenshot showing you've found the actual waveform
- What digital data is in the transmission? Convert the binary to HEX data and include it as a screenshot or text

*Pro Tips for both when using SDR's:

- Pick something you know works and make sure you can tune into it (103.1mHz is a radio station in town, if you can't hear that, you won't hear a little key fob transmitting)
- Note that the signal is transmitting intermittently. It's not constant (it's every couple seconds).
- When the class video/demo was done, there's a good chance that the signal strength will be very high—my SDR was right next to the radio I was receiving with. The signal for this will be a little lower, but still strong enough to detect.
- Check your bandwidth when recording: if you only grab 50% of the signal, you won't have enough to see what's really transmitted. SDR# is a little goofy with these settings; AM modulation will be great; bandwidth of 30,000 should be enough. If you don't have a high enough bandwidth, it's like looking at a low resolution image. You can make out what the signal should look like, but you can't be precise

- Once you’ve recorded the signal & opened it in Audacity you may notice your recordings may have background noise. You can identify the actual signal with the preamble-- it should stand out. If it doesn’t, check signal strength, bandwidth, etc

What to Submit

Submit a screenshot of your GRC Canvas that successfully decodes the signal, as well as a screenshot showing your method/tools for decoding the digital data. Submit these as a single PDF to D2L & answer the following questions based on the top-secret data you’ve recovered:

- Who is the sweepstakes open to which age groups?
- There’s a commercial with a big-wig CEO, who is it or what is he the CEO of?
- What is the digital data was sent in each transmission in part II?