Fall 2016 6.341 Project: RF Spectrum Sensing with Software Defined Radio

Part 1: Spectrum Characterization

Released: Thursday, November 10, 2016

Due: 11:59 pm on Saturday, November 26, 2016

Introduction

Spectrum sensing is an important element of future wireless systems that will collaboratively share precious radio-frequency (RF) spectrum resources and adapt to changes in real-time. Techniques for spectrum sensing and the subsequent optimization of wireless communication and sensing system operations are in relatively early stages of research and development. The terms *cognitive radio* and *cognitive RF sensing* are sometimes used to describe such future systems. The availability of inexpensive software-defined radio (SDR) systems with very capable wideband digital RF transceivers is enabling much of this research.

For the 6.341 project this year you will have the opportunity to develop and implement signal processing approaches to perform spectrum sensing with data from a capable commercial SDR. You are given a dataset collected by the Ettus Universal Software Radio Peripheral (USRP) B200 device. The data represents 6 MHz of a portion of the radio frequency spectrum that has been down-converted and stored as 32-bit binary representations of the in-phase and quadrature components of the signal. Part 1 of the project involves doing a coarse characterization of the spectrum (Part 1a) and implementing a downsampling system to capture a single signal of interest (Part 1b). Later this term, you will implement a filter bank to analyze the time-varying spectral characteristics of this or a similar dataset.

Project Description

Part 1a: It is known that there are *at least* 4 narrowband signals of varying power levels present in the dataset. Your job is to develop a signal processing system whose output is a list of the center frequencies of the narrowband signals, their bandwidths, and their powers. Possible approaches that may be employed include DFT-based spectral analysis, parametric modeling, filter banks, and others.

You may assume that in addition to the narrowband signals, that there is also additive white noise. This noise is dominated by thermal noise in the radio receiver and possibly also by any ambient sky noise in the recorded RF band. You are to estimate the power P_{noise} of the white noise component of the data. Based on this estimate, let us declare any portion of the spectrum in which the power is less than $(P_{\text{noise}} + 5)$ dB to be an "unoccupied" part of the spectrum. You are also to develop and implement an automated approach for estimating the total amount of unoccupied spectrum.

Part 1b: After determination of the frequency bands that are occupied, a next step in a spectrum-adaptive system could be a signal classification step in which a more detailed characterization of each signal is performed. Such processing might aim, for example, to classify the type of modulation employed, and determine time-frequency structure of the signal. With the results from Part 1a as a cue, you are to produce a high-resolution spectrum of the narrowband signal with the highest power. Implement a filtering and downsampling system to complete this task.

Project Report

Download the LaTeXor Google Document template from the course website and use it, or something similar, to describe the design tradeoffs you encountered and the rationale that led you to your final result.[1] Include block diagrams for each system, as well as any intermediate plots that you feel are important with appropriately labeled (and readable!) axes. Keep in mind that the report should showcase **your understanding** of the techniques you used, including whether or not your results were what you expected.

Dataset Access

The data file `sdr_data.bin` can be downloaded from the following link:

`https://www.dropbox.com/s/t3byu9qkjkyul7u/tenseconds-nov09.bin?dl=0`.

It is recommended that you implement your system using either MATLAB or Python's SciPy library. Copy and paste the code below into your editor to read in the data values.

MATLAB[2] :

```
   rawdata =  fopen('sdr_data.bin','r'));
         L =  80905608.0;
[val,count] = fread(samples,2*L,'float');
   samples =  complex(val(1:2:end),val(2:2:end));
```

Python :

```
samples = scipy.fromfile(open('sdr_data.bin'),
                  dtype=scipy.complex64)
```

[1]Note: The template provided is intended as a guide for the structure of your report. You may change the general format as long as the major sections are still included.

[2]Thank you to James Streitman and Thomas Royster of Lincoln Laboratory for providing this code.