RF Signal Database for a Communication Systems Course

Abstract

Students in communication systems courses respond to the material better when it is related to real world examples and systems. In order to strengthen the link between the theory and real systems, the radio frequency (RF) signal was recorded from several commercial communication systems and stored in a database which is available from the first author. Some of the signals included in the database are AM and FM radio, high definition AM and FM radio, analog and digital TV, Bluetooth, WiFi, GPS, WWV time signal, garage door opener, remote control for toy cars, wireless thermometers, and a wireless serial cable replacement system. The recordings, which were made with a Tektronix RSA3408A Real Time Spectrum Analyzer, can be used to illustrate several important concepts such as various modulation methods, frequency division multiplexing, frequency hopping, direct sequence spread spectrum, and noise. The signals can also be used in assignments and projects such as having the students identify the parameters of the signals (such as the bandwidth, type of modulation, baud rate, etc.), or having the students write a computer program to decode the signals. Using software to decode these signals illustrates the operations required in a software-defined radio receiver, and it makes the demodulation and decoding process more interesting than it would be with simulated signals. This paper describes how the signals in the RF signal database were recorded and how these signals can be used to enhance a communication systems course.

Background

When teaching communication systems, it would be very useful to have access to real signals to illustrate various concepts. For example, when teaching the concept of frequency division multiplexing (assigning different signals to different frequencies), it would make this concept very clear to have a recording of the entire AM radio band, so the students could see that in order to tune in a particular station, the receiver must isolate one of the stations and demodulate it. When teaching frequency hopping systems, it would be interesting to examine the signal from a frequency hopping system such as Bluetooth to witness the signal jumping from one frequency to another. In teaching software-defined radio, it would be interesting to test receivers with real RF signals instead of simulated ones.

There are advantages in having the students work with real signals in laboratory courses[1,2,3]. But for those students without access to the expensive test equipment or the time to set up the experiments, a database may be the only way to get access to a wide variety of RF signals. The RF signals in the database described in this paper can be used for examples, projects, and homework assignments in a communication systems course to strengthen the link between the concepts and real systems. A copy of the database can be requested by sending email to the first author.

Experimental Setup

Most of the RF signals in the database were recorded using the setup shown in Figure 1 consisting of a Tektronix RSA3408A Real Time Spectrum Analyzer and an AOR SA7000 Wideband Antenna (30 KHz - 2 GHz).

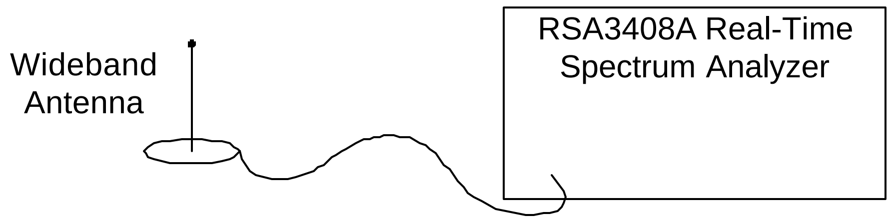

Figure 1: Experimental Setup for Most of the Database Recordings

The Tektronix RSA3408A Real Time Spectrum Analyzer captures the inphase component x(n) and quadrature component y(n) of the complex envelope[4] of the signal as shown in Figure 2. The value of the center frequency w_c in Figure 2 is determined by the center frequency setting of the spectrum analyzer. The decimate operations reduce the amount of data required to represent the signal.

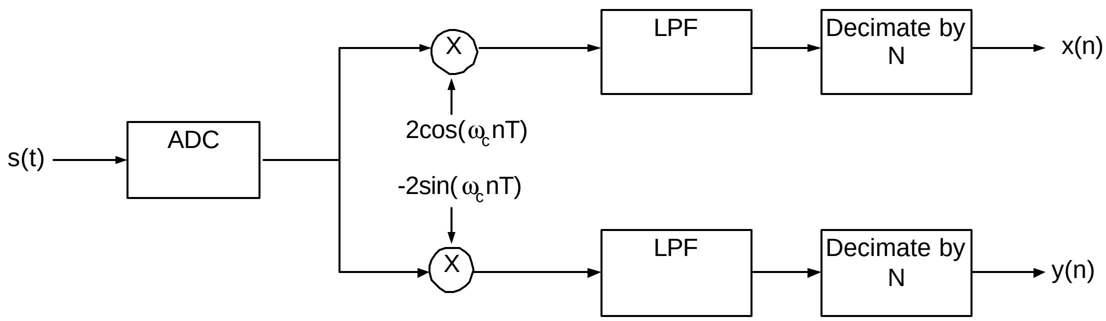

Figure 2: Down-Converter in the RSA3408A Real Time Spectrum Analyzer

The spectrum analyzer stores the inphase and quadrature components as binary files (filename.iqt). The signals can be moved from the spectrum analyzer to a PC using either a USB drive or a computer network. The binary files were converted to text files using a free program from Tektronix called IQTRead, which also corrects for variations in the IF filter response. The resulting files, which are stored in the database, are comma separated value text files (filename.csv) where the first column represents the inphase component x(n), and the second column represents the quadrature component y(n). An example of this format is shown in Table 1.

Table 1: First Few Lines of a CSV File (AM_Radio_HD_1.csv)

-0.00110801991035873,-0.00191097592353516
-0.0013188563394699,-0.00225190291528938
-0.00117979401388594,-0.00242685229263694
-0.000852324666543067,-0.00213526999705767

A parameter text file (filename.txt) was also created for each recording to document the parameters of the recording such as sampling frequency, center frequency, etc. The first few lines of a parameter text file are shown in Table 2.

Table 2: First Few Lines of a Parameter File (AM_Radio_HD_1.txt)

Sampling Frequency (Hz): 64000 Center Frequency (Hz): 1.19e+006 Span (Hz): 50000

If desired, the original bandpass signal can be reconstructed from the inphase and quadrature components as shown in Figure 3, but usually the complex envelope is processed directly because its data rate is lower.

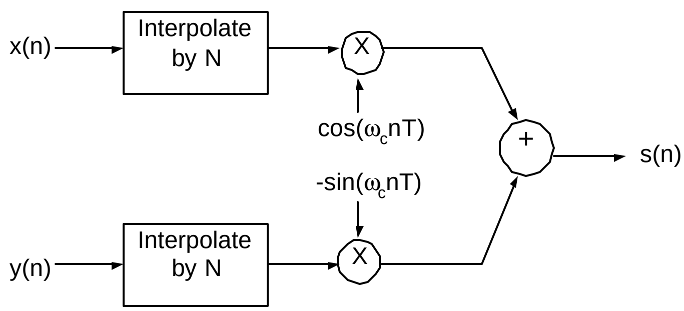

Figure 3: Reconstructing the Bandpass Signal from the Quadrature Components

Description of Recordings in the Database

The recordings that were selected for the database include both analog and digital RF signals with a variety of modulation types and bit rates. This section describes some of the signals and shows how to load and process them using the computer language MATLAB. Although the examples below use MATLAB, the files in the database can be processed by virtually any software program because the data is stored in easy-to-read text files. Some AM radio stations were recorded as examples of existing analog systems that are easy to demodulate. Some of these AM radio stations have just an analog signal, and some have a digital high definition (HD) radio signal that is transmitted along with the analog signal[5].

The following MATLAB (version 7.0) commands can be used to load one of the recordings of an AM radio station and plot its spectrum (see Figure 4). Everything on the line after the '%' sign is a comment that is ignored by MATLAB.

```
x = csvread('AM_Radio_HD_1.csv');        % load recording
x = x(:,1) + j*x(:,2);                   % convert to complex numbers
fs = 64000;                   % sampling frequency in Hz (see AM_Radio_HD_1.txt)
pwelch(x,[],[],256,fs)        % compute power spectral density using Welch's Method
```

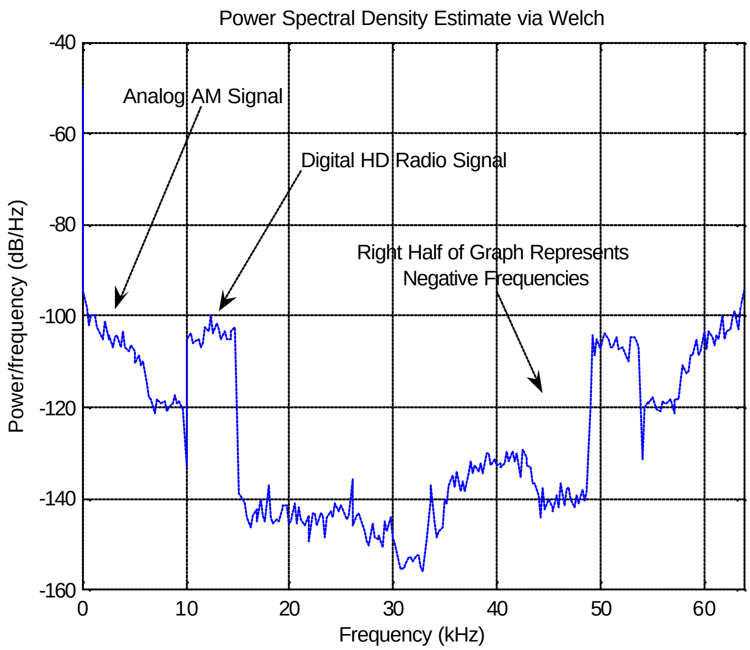

Figure 4: Baseband Spectrum of AM1190 KEX with Digital HD Radio Signal

The AM radio signal for station AM1190 is centered at 1190 kHz, but the spectrum analyzer represents the signal by its complex envelope, which is a complex baseband signal centered at 0 Hz. Since the spectrum of a discrete time signal is periodic, the right half of the complex envelope's spectrum (32 - 64 kHz in Figure 4) is the same as the negative frequencies (-32 to 0 Hz). So the right half of the baseband spectrum actually represents the negative frequencies of the complex envelope.

It can be seen from the spectrum in Figure 4 that this AM radio station broadcasts both the standard analog AM audio signal and the digital HD Radio signal. Unless the digital signal is filtered out, it will cause a high frequency noise to be heard in the demodulated analog audio signal. The digital HD Radio signal can be removed with a digital lowpass filter before demodulating the signals and sending it to the speakers as shown below.

```
% Continued from previous code snippet
b = fir1(100, 7000/(fs/2));    % Use fir1 from the Signal Processing Toolbox  to design
                               % a 100th order lowpass FIR filter with cutoff at 7 kHz
xf = filter(b,1,x);            % remove digital HD Radio signal
y = abs(xf);                   % demodulate AM signal
soundsc(y,fs)                  % send the audio signal to speakers
```

There is also a recording in the database (AM_Radio_Whole_Band.csv) that covers the entire AM band (540 to 1700 kHz), which is very useful in demonstrating the concept of frequency division multiplexing. The spectrum of the entire AM band is shown in Figure 5, where each spike represents one of the twenty AM radio stations in the Portland, Oregon area, and a few of the radio stations have been labeled. Note that the spectrum of this signal is not symmetrical, which is possible because the complex envelope is a complex signal.

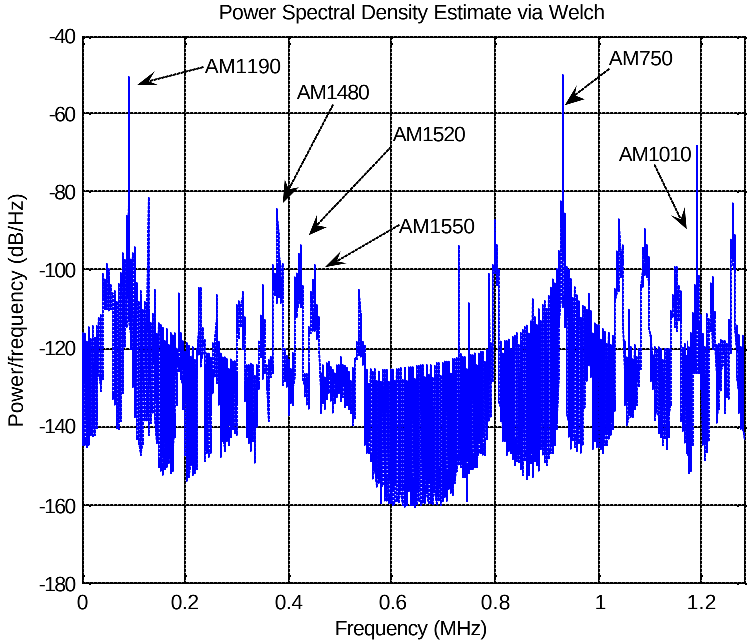

Figure 5: Baseband Spectrum of Entire AM Band

The stations can be identified as follows. For the spikes in the left half of the figure (i.e. 0 to 0.64 MHz), we add the complex envelope frequency to the center frequency of the recording, which is 1.1 MHz. The spike labeled AM1190 appears at 0.09 MHz in Figure 5, so its original frequency is 0.09 + 1.1 MHz = 1.190 MHz or 1190 kHz. For a spike in the right half of the figure (i.e. 0.64 to 1.28 MHz), we add the complex envelope frequency to the center frequency and subtract the sampling frequency, which is 1.28 MHz. The spike labeled AM750, for example, appears at 0.93 MHz and corresponds to 0.93 + 1.1 - 1.28 MHz = 0.75 MHz or 750 kHz.

The following MATLAB commands can be used to isolate and demodulate radio station AM1190 (1190 kHz) for example.

```
fc = 1190000;                                   % AM Carrier Frequency of interest (in Hz)
x = csvread('AM_Radio_Whole_Band.csv');         % load recording
fs = 1.28e+006;              % Sampling Freq in Hz (see AM_Radio_Whole_Band.txt)
centerFreq = 1.1e+006;       % Center Freq in Hz (see AM_Radio_Whole_Band.txt)
x = x(:,1) + j*x(:,2);                          % convert to complex numbers
f0 = fc - centerFreq;                           % compute complex envelope freq (in Hz)
BW = 10000;                                     % bandwidth of AM signal (in Hz)
                    % use cfirpm from the Signal Processing Toolbox to
                    % design a 750th order complex bandpass filter centered at f0
b = cfirpm(750,[-fs/2 f0-BW f0-BW/2 f0+BW/2 f0+BW fs/2]/(fs/2),[0 0 1 1 0 0],'none');
y = filter(b,1,x);           % filter the signal to isolate one station
z = abs(y);                  % demodulate AM signal
soundsc(z,fs)                % send audio signal to speakers
```

The database also includes FM radio stations, some of which transmit digital HD Radio signals in addition to the analog frequency modulated signal[6]. The following MATLAB commands can be used to load the recording and plot the spectrum (see Figure 6).

```
x = csvread('FM_Radio_HD_1.csv');        % load recording
fs = 640000;                             % sampling freq in Hz (see FM_Radio_HD_1.txt)
x = x(:,1) + j*x(:,2);                   % convert to complex numbers
pwelch(x,[],[],256,fs)                   % compute power spectral density
```

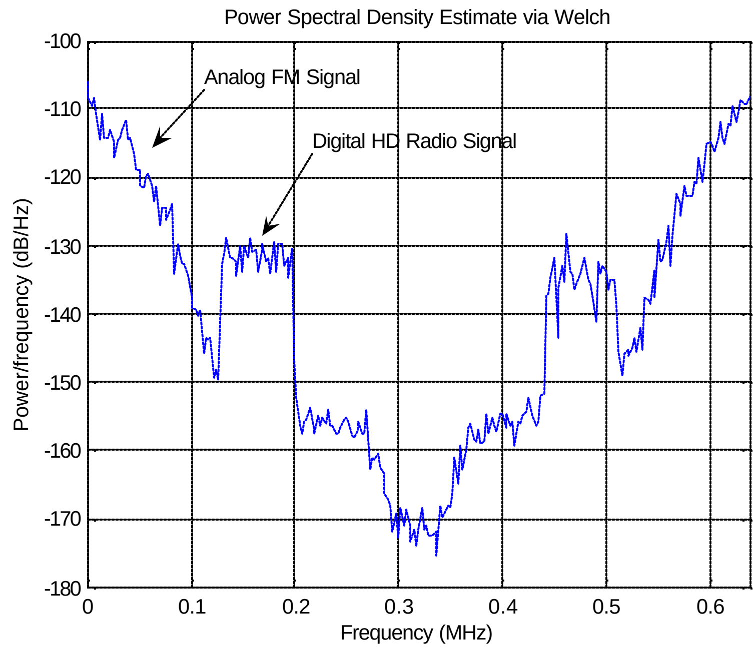

Figure 6: Baseband Spectrum of FM Radio Station with Digital HD Radio Signal

As can be seen in Figure 6, this FM station transmits digital HD Radio signal along with the analog audio signal. Using the commands below, the analog FM signal can be demodulated, and the spectrum of the signal AFTER demodulation is shown in Figure 7.

```
% Continued from previous code snippet
b = fir1(100,100000/(fs/2));             % design LPF to remove HD Radio signal
y = filter(b,1,x);                       % remove digital HD Radio signal
z = diff(unwrap(angle(y)));              % demodulate FM audio signal
pwelch(z,[],[],32768,fs)                 % plot spectrum AFTER demodulation
```

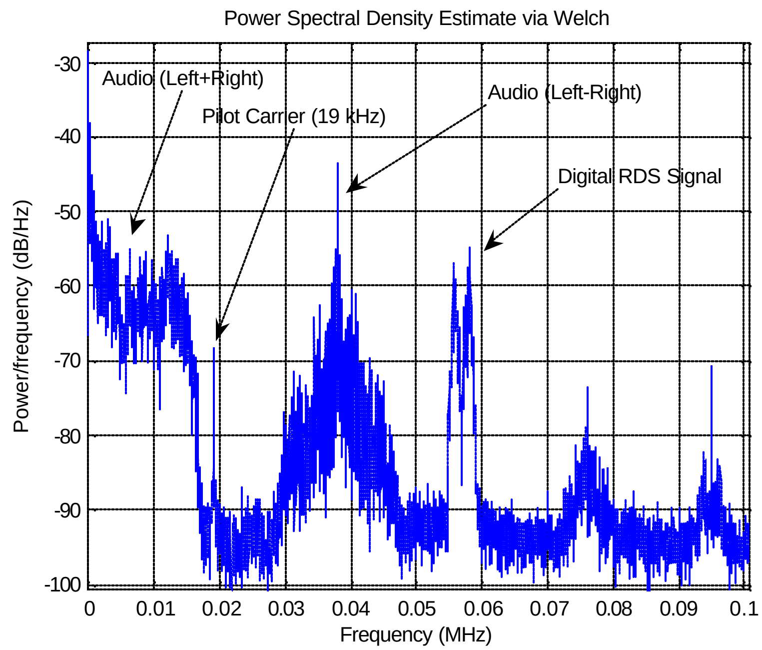

Figure 7: Baseband Spectrum of FM Message Signal AFTER Demodulation

Finally, the "Left + Right" audio signal can be isolated and sent to the speakers as follows:

```
% Continued from previous code snippet
b = fir1(400,15000/(fs/2));        % design LPF to pass only left plus right audio
zf = filter(b,1,z);                % remove everything except left plus right audio
soundsc(zf,fs)                     % send left plus right audio signal to speakers
```

The quality of the FM audio signal can be improved by adding a deemphasis filter to remove the effect of the preemphasis filter that is used in FM radio stations. Also included in the database is a recording of a part of the FM band that includes three FM stations which can be used as another example of frequency division multiplexing.

The database includes some recordings of analog TV signals. The spectrum of the complex envelope of one of these stations (Channel 10) is shown in Figure 8. The recording can be loaded and the spectrum can be plotted using the following MATLAB commands:

```
x = csvread('TV_Analog_1.csv');        % load recording
fs = 1.28e+007;                        % sampling frequency in Hz (see TV_Analog_1.txt)
x = x(:,1) + j*x(:,2);                 % convert to complex numbers
pwelch(x,[],[],16384,fs)               % compute power spectral density
```

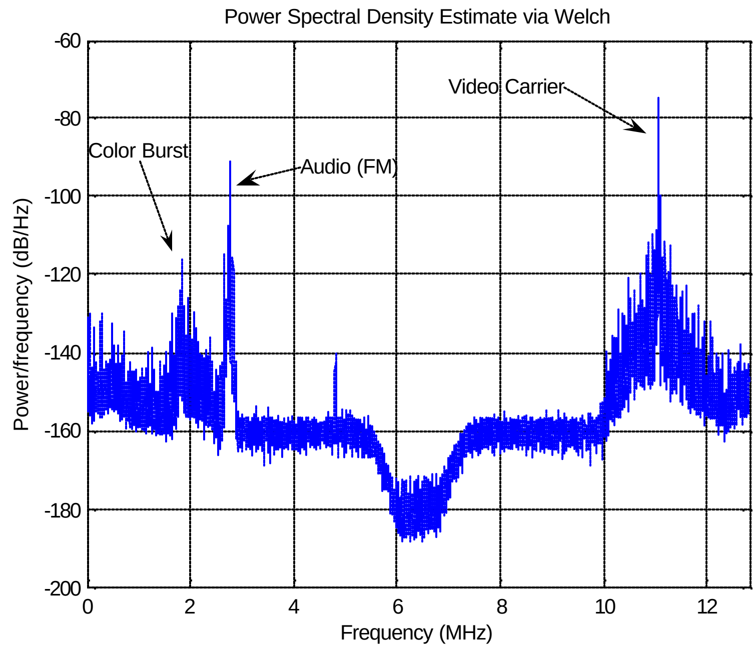

Figure 8: Baseband Spectrum of an Analog TV Station

The center frequency of the recording is 195 MHz and the sampling frequency is 12.8 MHz. The video carrier appears at 11.05 MHz in the right half of Figure 8, which corresponds to 11.05 + 195 - 12.8 = 193.25 MHz. The color burst signal appears at 1.83 MHz in Figure 8, which corresponds to 1.83 + 195 = 196.83 MHz. The audio signal appears at 2.75 MHz, which corresponds to 2.75 + 195 = 197.75 MHz.

The spectrum of a digital TV station is shown in Figure 9 and the spectrum of a WiFi signal is shown in Figure 10.

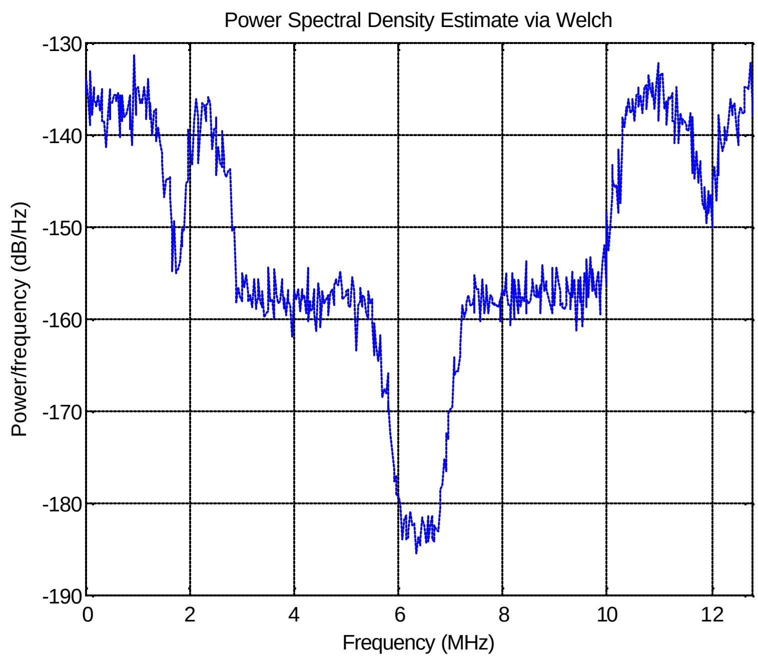

Figure 9: Baseband Spectrum of a Digital TV Station

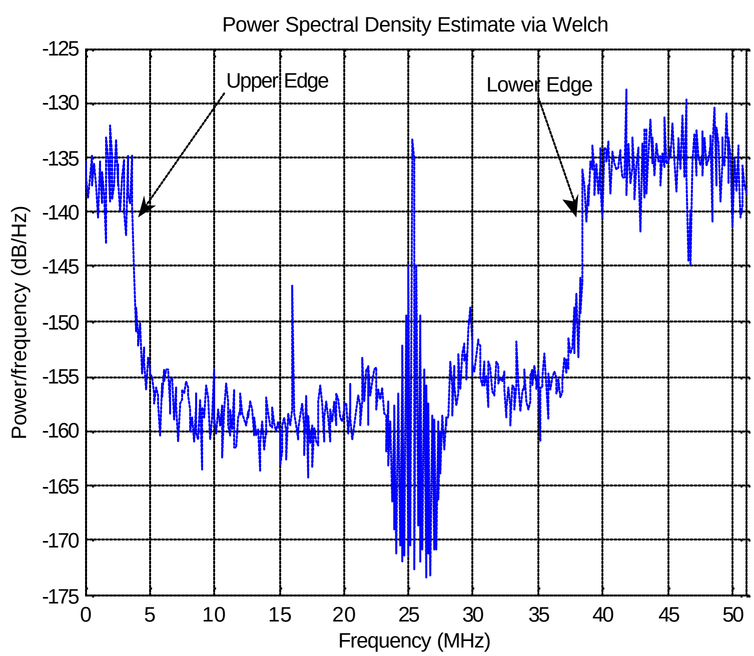

Figure 10: Baseband Spectrum of WiFi

One of the recordings is from a MaxStream 9xStream 900MHz Wireless OEM Module, which is a wireless serial port replacement system that uses frequency hopping. In the spectrogram shown in Figure 11, the horizontal axis is frequency, and the vertical axis is time. The color of the graph indicates the power at a particular frequency and a particular time, with brighter colors representing higher power levels. As shown in Figure 11, this system transmits on a particular carrier frequency for approximately 56 mS, then it jumps to a new frequency.

```
x=csvread('wireless_serial_port.csv');       % load recording
fs = 1.28e+007;                   % sampling freq (in Hz) see wireless_serial_port.txt
x = x(:,1) + j*x(:,2);                 % convert to complex numbers
spectrogram(x,[],[],2048,fs)           % compute spectrogram
```

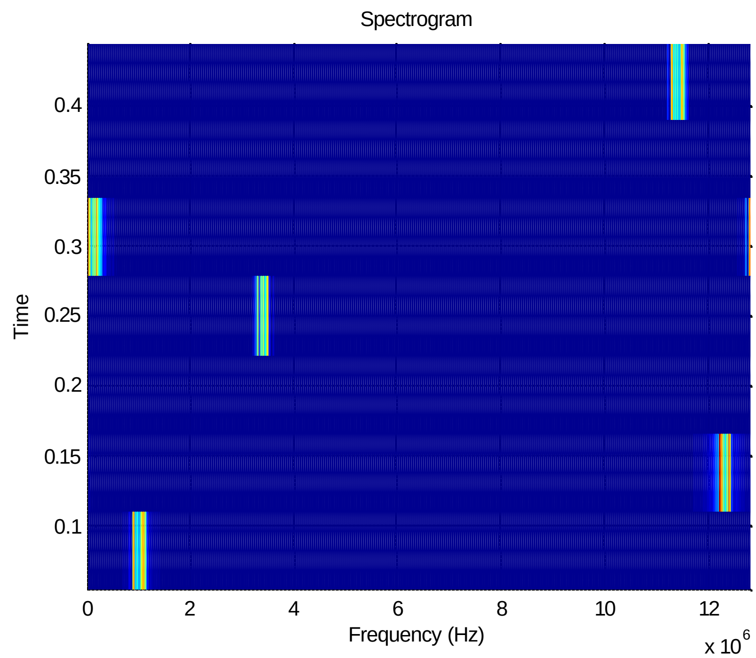

Figure 11: Spectrogram of Frequency Hopping System

In addition to the signals described above, the database currently includes Bluetooth, which is another frequency hopping system, GPS that uses direct sequence spread spectrum, NIST radio station WWV which broadcasts accurate time and frequency signals, remote control for toy cars, garage door opener, and wireless thermometers. All these signals were recorded using the setup in Figure 1, except for WWV and GPS. A 10 MHz bandpass filter was used to eliminate interference from strong AM radio stations while recording the WWV signal. To record the GPS signal, a Rojone A-GPSA95NS antenna, Rojone AMA-061B amplifier, and a DC block were used. The authors are interested in receiving suggestions for additional signals to add to the database.

Educational Uses for the RF Database

If all students had access to equipment capable of recording RF signals, instead of using an RF database, it would probably be better if the students recorded the signals themselves so they knew exactly how the recordings were made. However, for those students who either lack access to this expensive equipment or lack the time required to set up the devices and make the recordings, this database will allow them to easily explore a wide variety of RF signals used in real commercial systems without requiring any specialized equipment or setup time.

Communication system concepts such as AM, FM, bandwidth, noise, filtering, frequency division multiplexing, frequency hopping, and direct sequence spread spectrum can be illustrated using the signals in the database. The signals can be used to check theoretical values such as bandwidth against measured values from a real system. The students can also be asked to identify the type of modulation, baud rate, or other parameters from the recordings. These values are more meaningful when they relate to real-world systems.

The signals can also be used as a type of puzzle where the students are asked to demodulate the signals to uncover the message signals. (The analog signals are much easier to demodulate than the digital signals.) This type of problem gives the students immediate feedback because they can judge the quality of the recovered signal. For example, they can listen to audio signals or look at images to determine the quality of the signal. Furthermore, the process of demodulating the signals illustrates the signal processing algorithms needed in a software-defined radio receiver.

The recordings can also be used in a digital signal processing course to illustrate the real-world tradeoffs between computational complexity of a filter and the resulting audio or visual quality. The students can experiment with filters of various orders and hear or see the effect on the demodulated signal. Also, these signals could provide excellent examples for spectral analysis because some of the spectral features can only be seen if the parameters of the estimation algorithm are set correctly.

Conclusion

The RF database is a useful tool in teaching a communication systems course. It allows students to explore a wide variety of RF signals from real commercial systems without any specialized equipment. Since the database files are text files, they can be processed by virtually any software system, and they are available from the first author.

Acknowledgements

The authors would like to thank Tektronix, Inc., whose generosity helped make this project possible.

References

[1] The Incredible Hulk and Other Techniques for Teaching Waveform Demodulation, Thad B. Welch and Robert F. Kubichek, Proceedings of the 2005 ASEE Annual Conference, Session 3420.

[2] Caller ID: A Project to Reinforce an Understanding of DSP-Based Demodulation, Thad B. Welch, Cameron H. G. Wright, and Michael G. Morrow, Proceedings of the 2005 ASEE Annual Conference, Session 3420.

[3] Using Real Signals with Simulated Systems, Joseph P. Hoffbeck, Proceedings of the 2002 ASEE Annual Conference, Session 2220.

[4] Digital Communications, 4th Edition, John G. Proakis, McGraw-Hill, 2001.

[5] The Structure And Generation of Robust Waveforms For AM In Band On Channel Digital Broadcasting, Steven A. Johnson, iBiquity Digital Corporation, www.ibiquity.com/technology/papers.htm.

[6] The Structure And Generation of Robust Waveforms For FM In-Band On-Channel Digital Broadcasting, Paul J. Peyla, iBiquity Digital Corporation, www.ibiquity.com/technology/papers.htm.