The Hunt for Red Herring
Project 1 — Due Octoberl 15, 2011

The Fraternal Order of Eccentric Spies (FOES), an organization of retired secret agents, uses en-coded wireless communications to relay social messages among its members, a method preferred over email by former field agents. One of FOES members, however, has lost his demodulation decoding key, just as he was expecting an important message from another retired agent, code name Red Herring. He has come to you to ask for help in recovering the message from a snippet of a digitized data. Your mission is to apply your newly learned frequency-domain signal analysis skills to recover the message from Red Herring. But, beware! There may be other kinds of red herrings among the messages, so be sure to recover all messages in order to authenticate the one from the true Red Herring.

There are roughly 440,000 data samples, taken at a sampling rate of 44100 samples per second (about 10 seconds total), of the baseband signal which are captured in a Matlab-formatted data file project1_data.mat, located on the class internet site. Reference material for this lab may be found in the class notes and in class demo files example_data.mat (a data file) and example_processing.m (an associated Matlab demonstration processing script that provides guidance for the type of processing expected to complete this assignment).

It is known that multiple messages are encoded for simultaneous transmission by FOES using single-sideband modulation in an unorthodox frequency-division multiplexing (FDM) scheme. A conventional FDM signal is created by multiplexing channels of uniform bandwidth stacked at uniform frequency intervals. FOES, however, uses non-uniform channel bandwidth and non-uniform frequency spacing. Due to the loss of the decoding instructions, you will first need to apply your signal analysis skills with Fourier transforms to determine the bandwidths and frequency locations of all message channels in the baseband FDM sampled signal before you can proceed to recovery of the agent messages in each channel.

Three steps are required to perform this lab assignment. Please note the expected responses for each step that are to be included with your submitted report documentation in order to receive full credit. This assignment must be performed using Matlab. The 3 steps are:

1. Step 1 — Analysis (2 points)
   Due to the missing decoding key, an evaluation of the frequency content versus time using fast Fourier transform (FFT)-based analysis is needed to determine bandwidth and frequency location of each message channel. Demonstration script example_processing.m, shown in class, illustrates how to create a two-dimensional (2-D) time-vs-frequency spectrogram for plotting in by the MATLAB function, plot_gram.m. Both are on the course web page.

   Electronically export (use EXPORT under the Matlab FILE menu on the Figure window) the time-vs-frequency gram that you create for importing into a report document; specify the parameters used to create the gram (e.g., the analysis window duration, the interval between analysis windows). Answer the following questions:

- Based on the time-vs-frequency gram, how many channels (messages) are encoded in the data?
- For each observed message channel, what is the lowest and highest frequency extent associated with each channel; therefore, what is the message bandwidth of each channel?

2. Step 2 — Filter Design (3 points)
   Based on the procedures given in class, you will need to design a pair of filters per message channel in order to recover the audio message from each channel — i.e., a bandpass filter to ex-tract out and isolate an individual channel, and a lowpass filter to filter out undesirable byproducts of channel demodulation. Use the design techniques in the text and class notes, as applied by the Filter Design and Analysis Tool (FDATool) of the Matlab Signal Processing Toolbox. FDATool provides an interactive Matlab-based graphical interface for filter design. You may select any IIR or FIR digital filter. Please note that quality (sharp transition) Butterworth and Chebyshev IIR bandpass digital filters may not be realizable (error conditions re-ported out by the design tool) when transition band frequencies occur within 1 KHz of either 0 or the Nyquist frequency (22050 Hz). If realizability issues are encountered, either switch to a bandpass FIR filter design, or use a realizable lowpass IIR filter design (if near 0 Hz) or a high-pass IIR filter design (if near Nyquist) rather than attempt a bandpass IIR design. For each filter pair designed for a given channel discovered in Step 1, provide the following in your report:

   - Lowpass filter specifications: filter type (IIR Butterworth, IIR Chebyshev, FIR, etc.), filter order, bandwidth of passband, transition bandwidth, stopband attenuation (in dB relative to passband) [snapshot copy of the FDATool window showing entries will suffice].

   - Bandpass filter specifications: filter type (IIR Butterworth, IIR Chebyshev, FIR, etc.), filter order, pass band edge frequencies, transition bandwidths on either side of passband, stopband attenuation (in dB relative to passband) [snapshot of FDATool window is OK].

   - Frequency response plots of each filter design (a feature provided by FDATool); use EXPORT under FILE menu in a Figure window to create an importable figure for your report document.

3. Step 3 — Recovery of Messages (2 points)
   Using the filter pair designed in Step 2 for each message channel, apply these together with a demodulation operation to recover each message embedded in the data of project1_data.mat. Use the Matlab sound command to listen to each demodulated message. Provide the following in your lab report:

   - Text and sound descriptions of each message.
   - Which message was from Red Herring? (Hint: He likes to explain DSP to students.)
   - Include a Matlab listing, or listings, for all Matlab scripts applied to steps 1, 2, 3.