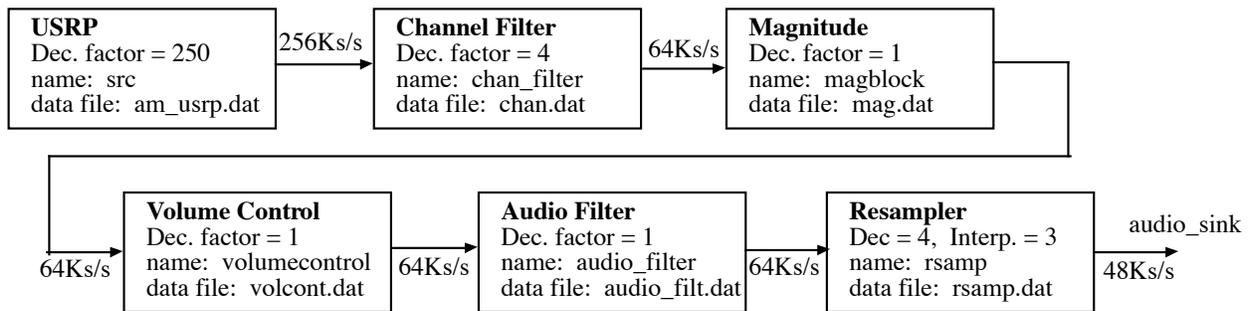


Receiving the AM Broadcast Band with the USRP and GNU Radio

The USRP along with the Basic RX or LFRX daughterboard can be used to receive strong AM broadcast stations. Since neither of these daughterboards has a pre-amplifier it is important to use a good antenna or to add a pre-amp. The antenna is connected to the RX-A input on the daughterboard.

This document describes the GNU radio script that is used to receive and demodulate a station. This script is a modified version of the `am_rcv.py` script that can be found in the `gnuradio-examples` directory. The script used here is named `am_rcv3.py`.

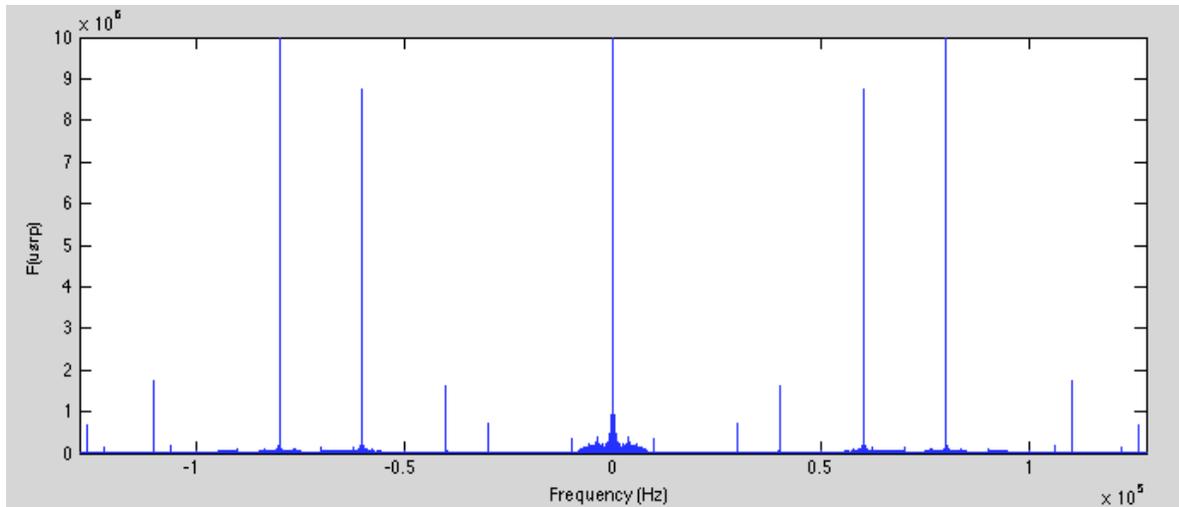


The block diagram of the receiver is shown above. Each of the blocks along with their outputs are discussed below. Refer to the `am_rcv3.py` file for the Python script.

USRP

The ADC in the USRP samples the input signal at 64Ms/s. It is set to the frequency of a broadcast station. The mixer in the digital downconverter is set to this frequency and thus, the selected station is shifted to be centered around 0 Hz. The decimation factor in the FPGA is set to 250, resulting in an output data rate from the USRP of 256 Ks/s ($64\text{Ms/s} / 250$). A sample of this broadcast was stored in a file (`am_usrp.dat`) that contains 993280 samples. At a data rate of 256Ks/s (or a sampling interval of 0.0039 mS) this file contains $993280 \cdot (0.0039\text{m}) = 3.88$ seconds of signal. Using MATLAB's `fft` command, the DFT of the real part of these complex samples was found and plotted using the following MATLAB commands:

```
>> fs=256000;
>> ts=1/fs;
>> N=993280;
>> Fusrp=fft(real(usrpam));
>> deltaf = 1/(N*ts);
>> plot([0:N-1]*deltaf, abs(Fusrp))
>> Fusrp2=fftshift(Fusrp);
>> plot([-N/2+1:N/2]*deltaf, abs(Fusrp2))
```



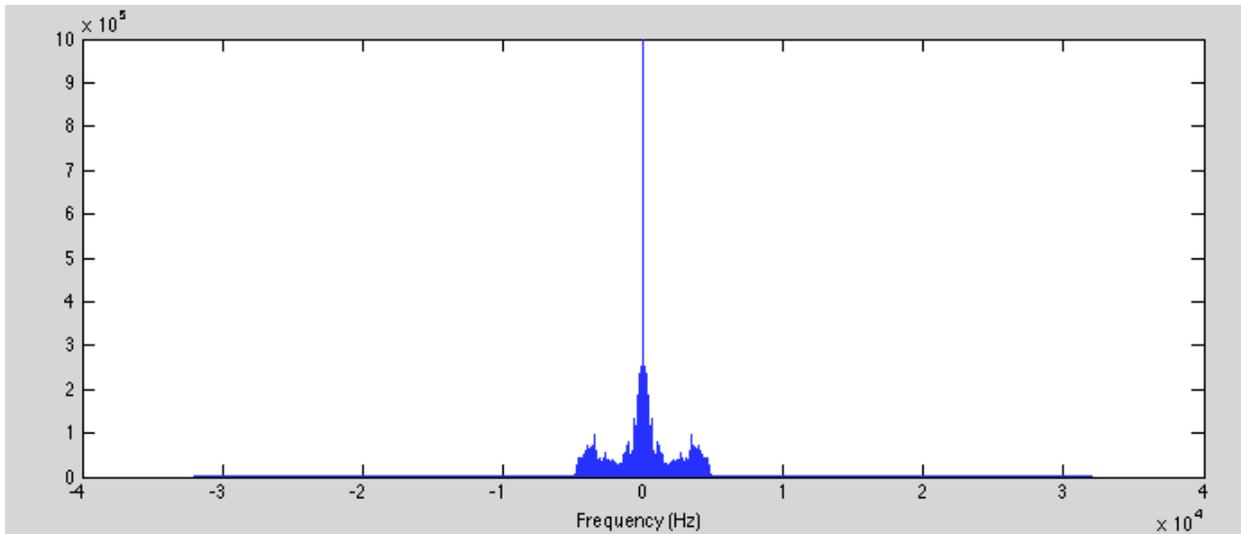
Note that a sampling rate of 256Ks/s produces frequencies up to 128 KHz. The center component is the signal that is being received. The other spikes represent other AM stations above and below the received station. In this example, the receiver was tuned to 870 KHz. Thus, the spectrum shown represents the range from 742 KHz to 998 KHz. Only strong stations will be seen due to the lack of a pre-amp.

Channel Filter

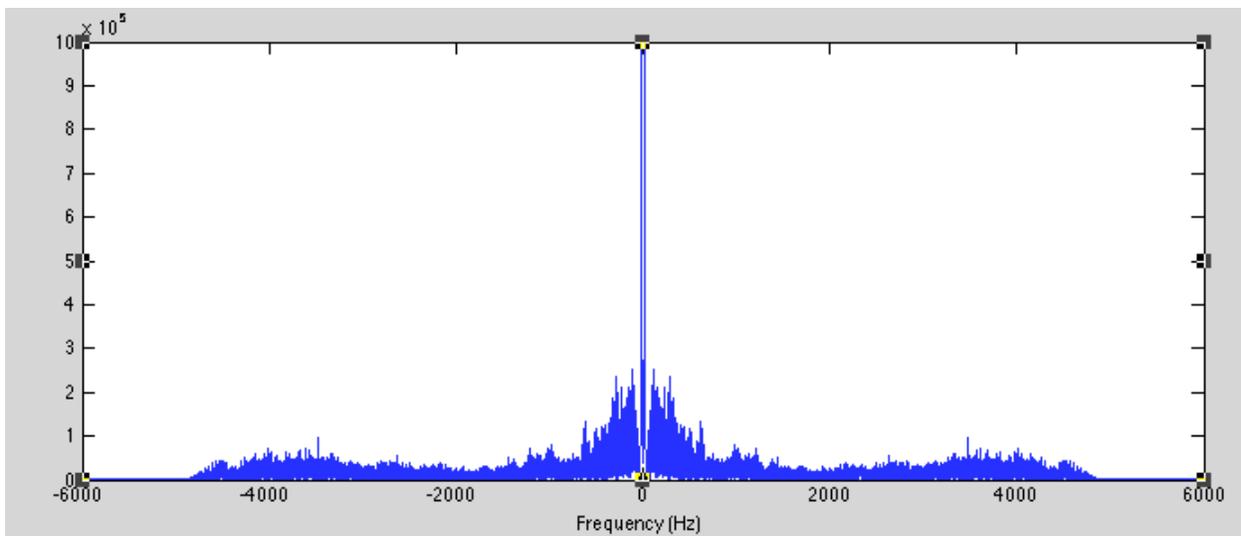
The channel filter was designed to pass frequencies below 4500 Hz and block frequencies above 5000 Hz on either side of the center frequency. The channel filter was specified to have a decimation factor of 4, bringing the sample rate down to 64Ks/s. Correspondingly, the number of samples in this file is 248320, or 1/4th of the number of samples in original file. Using the following MATLAB code:

```
>> fs=64000;
>> ts=1/fs;
>> N=248320;
>> Fchan=fft(real(chanam));
>> deltaf=1/(N*ts);
>> Fchan2=fftshift(Fchan);
>> plot([-N/2+1:N/2]*deltaf, abs(Fchan2))
```

The following spectrum results:

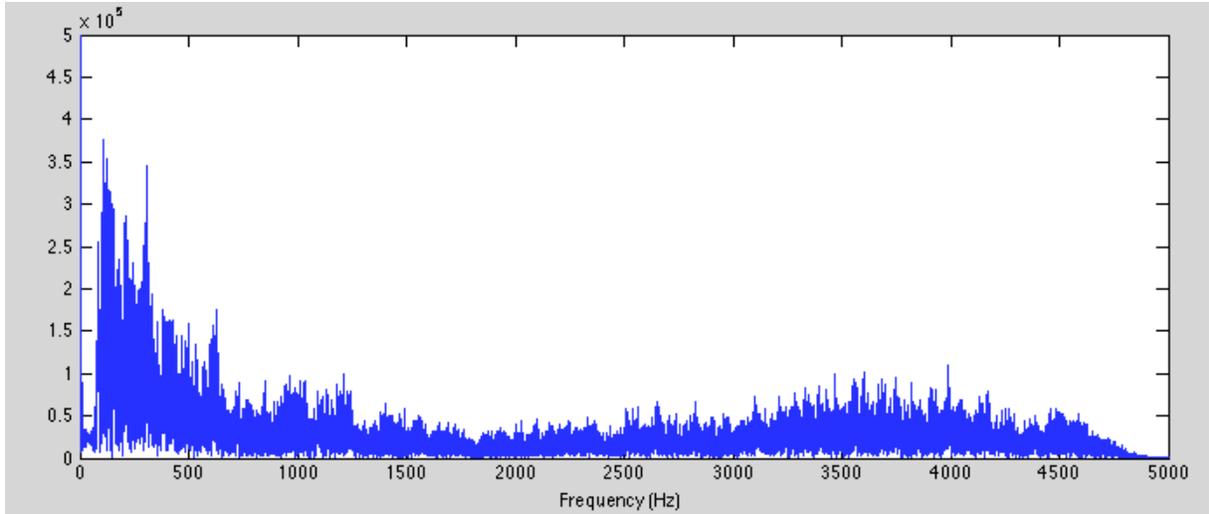


Note that only the selected station remains and the others have been filtered out. If we expand the scale to show the frequency range from -6 KHz to 6 KHz, we verify that frequencies more than 5 KHz away from the center have been eliminated by the channel filter.



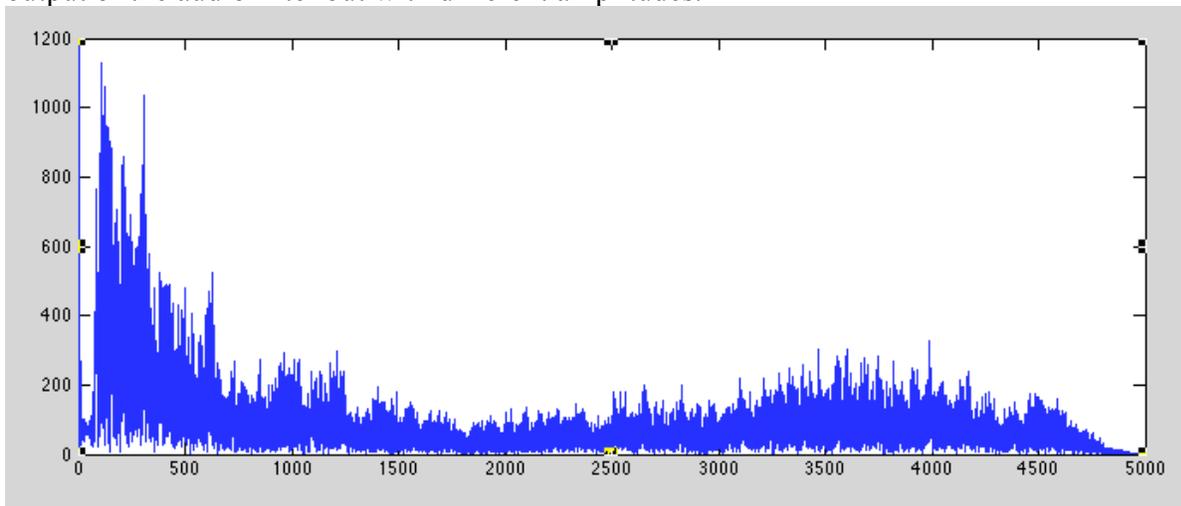
Magnitude Block

Passing the signal through an envelope detector demodulates an AM signal. In mathematical terms, the envelope detector finds the envelope or magnitude of the AM signal. The MATLAB code to plot the spectrum of the magnitude block output is similar to that used above. However, the values in the output file are now real values. The resulting spectrum is shown below.



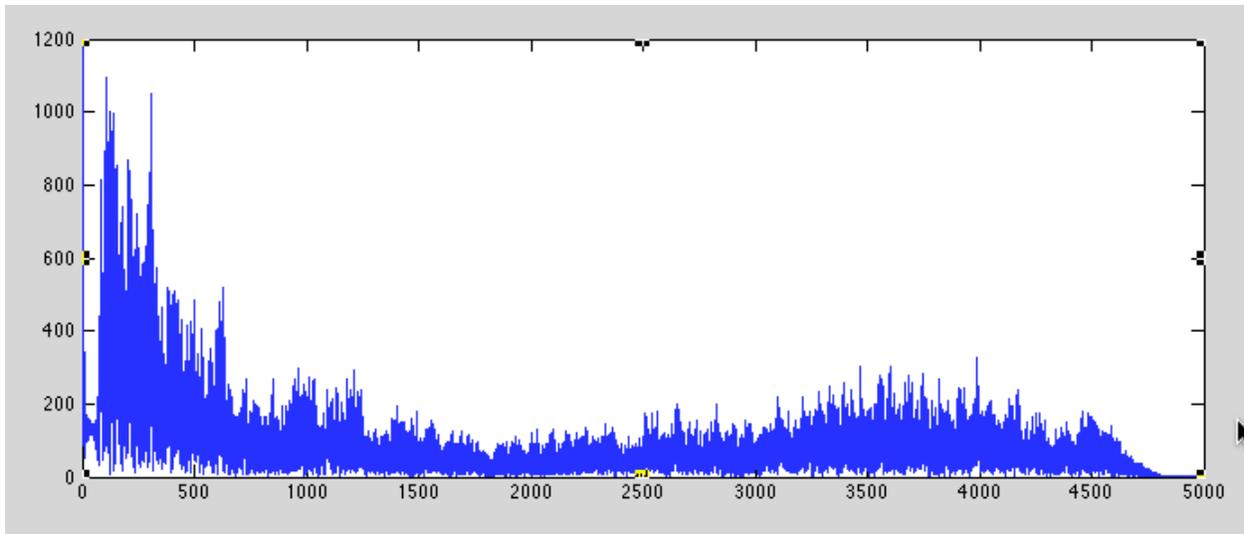
Volume Control

The volume control is used to attenuate the values. Note that its spectrum is the same as the output of the audio filter but with different amplitudes.



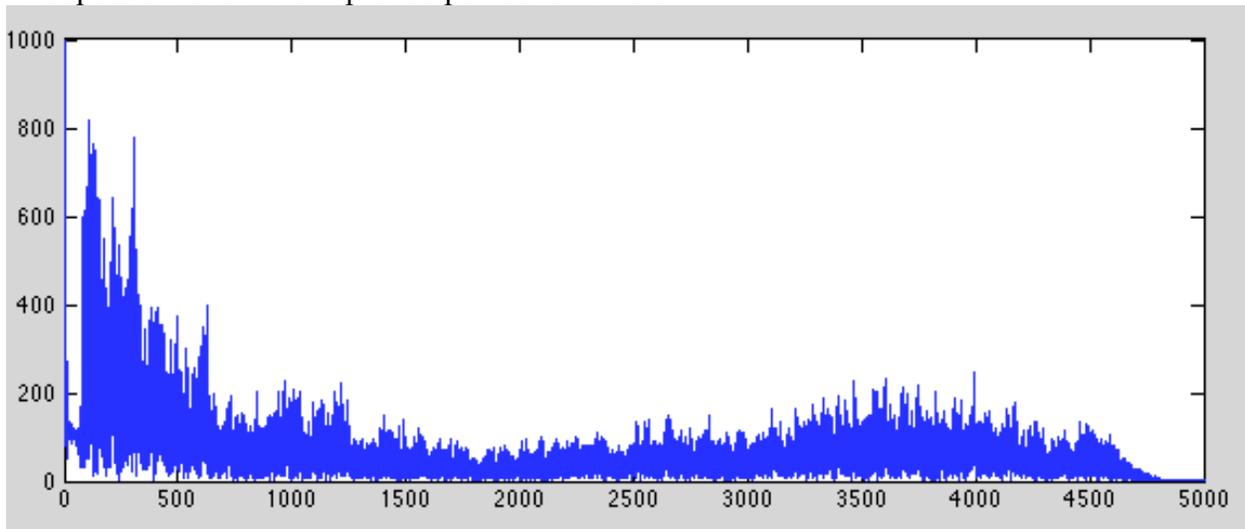
Audio Filter

The spectrum of the audio filter output is shown below.



Resampler

The spectrum of the resampler output is shown below.



Try this:

Change the final audio filter to a bandpass filter that eliminates the components below about 100 Hz. How does this change the output?