

## **AC 2009-350: USING SOFTWARE DEFINED RADIO (SDR) TO DEMONSTRATE CONCEPTS IN COMMUNICATIONS AND SIGNAL PROCESSING COURSES**

### **Sharlene Katz, California State University, Northridge**

Sharlene Katz is a Professor in the Department of Electrical and Computer Engineering at California State University, Northridge (CSUN) where she has been for over 25 years. She graduated from the University of California, Los Angeles with B.S. (1975), M.S. (1976), and Ph.D. (1986) degrees in Electrical Engineering. Recently, her areas of research interest have been in engineering education techniques, software defined radio, and neural networks. Dr. Katz is a licensed professional engineer in the state of California.

### **James Flynn, California State University, Northridge**

James Flynn is a part time faculty member in the Department of Electrical and Computer Engineering at California State University, Northridge (CSUN). He holds a B.S. (1977) degree in Electrical Engineering from the Illinois Institute of Technology and a Master of Fine Arts (1981) degree from Northwestern University. He is a partner in a consulting firm specializing in electronics for television and film production. Currently he is developing education tools involving software defined radio (SDR).

# Using Software Defined Radio (SDR) To Demonstrate Concepts In Communications and Signal Processing Courses

## Abstract

The fundamental course in communications systems included in most electrical engineering programs introduces the concepts of basic analog and/or digital modulation techniques. Teaching modulation theory is mathematical in nature and can be an abstract concept for students. This may often result in low student motivation and comprehension of the subject matter. This paper describes a set of classroom demonstrations developed with software defined radio (SDR) to illustrate the concepts presented in an analog communications course.

Based on feedback from the students in this course, the demonstrations developed increased their understanding and motivation. Cost was minimal and could be nearly zero with available free software and downloadable signals.

## I. Introduction

This paper presents a set of classroom demonstrations developed for use in the senior level analog communications course that is common to most electrical engineering programs. The demonstrations are intended to provide motivation to students with little or no practical experience with communications systems. By using software defined radio (SDR), communication systems are demonstrated with signals that are familiar to students. The demonstrations can be used in any classroom or laboratory with minimal cost.

Section II of this paper provides background on some of the issues that faculty currently face when teaching analog communications theory and explains the need for demonstrations in the course. Section III gives an overview of SDR and the features that make it an ideal platform for classroom demonstrations in communications and signal processing. Section IV describes the particular SDR platform (hardware and software) that was used by the authors. Section V presents the demonstrations created. Section VI describes the results of using these demonstrations in a classroom and section VII presents conclusions and plans for future work in this area.

## II. Background

Courses on communications theory rely heavily on mathematical models. The abstract nature of the mathematical treatment of modulation/demodulation can be difficult for students to understand. This is particularly true now since most of our students do not have prior hobby and/or work experience with electronics as students in the past have had.

Lacking practical experience, our typical students have more of a need to see that the material they are learning in class is useful in the “real world”. In addition, practical applications of communications theory provide an overall structure in which to place what could be seen as unconnected equations and concepts. Unfortunately, classroom access to commercial

communications systems is impractical. Classroom demonstrations illustrating the theoretical concepts as they are introduced can be vital in motivating students and helping their understanding<sup>1,2</sup>. These demonstrations can be particularly interesting if they use signals that are familiar to the students (i.e. AM/FM radio, TV).

Many undergraduate communications courses have an accompanying laboratory to give students a better understanding of the hardware implementation of the techniques they are studying<sup>3-7</sup>. Due to the complexity of communications circuits, the laboratory experiments typically deal with simple circuits and rarely provide an opportunity for students to observe actual communications signals. It would be desirable to provide demonstrations that construct an entire communications system and follow an RF signal from the antenna until it is output as an audible signal from a speaker. One solution has been the move to simulation. Unfortunately, this does not satisfy the need to experiment with “real” signals and systems.

In addition to affecting their performance in class, the abstract nature of the material taught in communications courses often deters undergraduate students from pursuing a course of study in the communications area. Students completing a typical communications course may not be motivated to take additional courses in the area because they do not see the practical application of what they have learned. At California State University, Northridge, three communications courses are available to undergraduate students. The first is a course in analog communications and has an average of 20 undergraduate students enrolled each semester (40 per year). Students can follow this course with a course on digital communications and/or a course on communications networks. These higher level courses are usually taken by graduate students with undergraduate enrollment on the order of only 5 per year in the digital communications course.

A set of communications systems demonstrations that process real world signals has been created for the senior level analog communications course. These demonstrations use SDR to implement receivers and/or transmitters for various modulation schemes.

SDR offers a multitude of unique and effective tools to teach signals and communications, providing the ability to create and manipulate fairly complex systems without the hardware problems cited above. Put in simplest terms, SDR is the direct implementation of the mathematics of signal processing on real world signals. Instructors can go directly in the classroom from a set of equations for a demodulation scheme to a program to apply the mathematics to a stream of sampled data from a live or recorded real signal or a test waveform generated in SDR. They can immediately demonstrate the results of the mathematical processing in the time or frequency domain and experiment with parameters on the same signal over and over, without resorting to expensive signal generators or large antennas. Students can even hear or see the demodulated signal. SDR software can change a laptop computer into thousands of dollars worth of spectrum analyzers, oscilloscopes and signal generators – at a mouse click. The students benefit from immediate and tangible experience in applying the complex theories and principles they are trying to master.

Perhaps the most attractive aspect of SDR is the availability of signal databases<sup>8-10</sup>. In SDR large sections of the RF spectrum can be digitized and recorded in computer files. These files

allow students to experiment with real signals without any hardware component. This also allows faculty to assign projects to students without requiring expensive hardware. Section III of this paper gives an overview of SDR for those unfamiliar with this relatively new technology.

### **III. What is Software Defined Radio(SDR)?**

Software Defined Radio (SDR)<sup>11-14</sup> is an emerging technology built on one of the great developments of the 20<sup>th</sup> Century: fast and cheap microcomputers. It represents a new method in processing radio signals that is a quantum leap over the older, original method of analog processing.

Since the invention of radio in the late 1800's, signals have been processed through a single method: the signal is picked up by an antenna that turns the radio wave back into a fluctuating voltage signal. This signal is passed on to various components in the receiver which react with it to eventually produce the sound, video or data that is desired.

The function of these components and systems has been modeled mathematically in well known formulas. Student engineers spend years studying and mastering these mathematical concepts. Communications engineers think and design with these formulas, but the ultimate implementation is still in the physical components that make up a radio.

SDR moves the mathematical realm of communications theory into the real world. In SDR, the signal is still picked up by an antenna. However, almost immediately, the signal is converted digitally to a sequence of numbers representing the value of the signal at regular time intervals. These digital values are then processed in software, using the very formulas that student engineers have studied for the last century. The resulting output can then be converted back into audio, video or remain data. In a sense, the SDR is given a signal and it "solves for" or "calculates" the intelligence riding on it.

Even before the development of microprocessors in the 1970s, engineers had been making efforts to move signal processing out of the analog domain and into the more precise and flexible realm of digital signal processing (DSP). Early systems were crude, often converting the input signal with a single bit analog to digital converter (ADC) and integrating the number of 1 counts or triggering a counter with the input and processing the counter output for timing detection. Applications included LORAN and OMEGA navigation systems which relied on the relative timing between received signals to fix location.

Early systems were limited in a number of ways. Fixed designs hard-wired the signal processing algorithm into a pattern of logic gates. The systems were hybrid, a combination of analog and digital processing, with the bulk of the preconditioning of the signal, i.e. amplification, frequency shifting, filtering, etc., in traditional analog circuits. In addition, the vast majority of applications were strictly receiver systems.

Engineers and users alike were hardly satisfied with the status quo. Designers found that traditional analog design was becoming inadequate for modern signals and the complex features often desired by users. Tantalizing on the horizon was what engineering students had often

heard in a communications or signals course: “This system is theoretically possible, but not physically realizable.” In other words, the math was there but the electronics were not.

Forays into DSP began with main frame computers working on previously recorded signals that had been already converted to digital samples. Outputs from such systems gave the results days or even weeks after the signal was actually received. With the advent of inexpensive microprocessors and faster analog to digital convertors, both the speed and capabilities of DSP advanced rapidly. Systems became less and less hybrid as the conversion from analog signals to digital numbers moved ever closer to the antenna. From the outset, however, one of the true advantages of SDR became apparent: radio systems could be changed quickly by reprogramming.

By the middle 1990’s, SDR had advanced to the point where systems could be fabricated with a minimum of signal preconditioning and the bulk of the signal processing in software.

Most of this development has been propelled by the military and commercial interests. However, groups of experimenters and hobbyists began building their own simple systems, often using simple mixers to convert the signal down to audio frequencies and digitizing the converted signal using a sound card. Entrepreneurs took advantage of inexpensive large-scale integrated circuits and began selling radio frequency interfaces to the general public. In recent years radio amateur electronics manufacturers have marketed the first SDR systems with transmit and receive capability for the radio amateur community.

With the advent of inexpensive RF interface hardware, SDR enthusiasts began setting up websites to share signal processing code. GNU Radio<sup>15</sup> is one of the most popular repositories of free, open-source SDR code. Together with RF interfaces, such as the Ettus Universal Software Radio Peripheral (USRP)<sup>16</sup>, those interested in experimenting or working with SDR have ready access to hardware and software.

Among those taking advantage of the inexpensive systems and software available are a small number of universities beginning to develop courses in the area of SDR<sup>17,18</sup>. Generally these are at the graduate level.

Currently, a number of SDR platforms are available. Section IV describes the SDR platform (hardware and software) that the authors have used for this project along with references to some of the other systems.

#### **IV. Implementation of SDR**

The Ettus Research USRP was chosen by the authors because of its performance and price. The basic USRP offers a 12 bit ADC running at 64 megasamples per second (MS/s) on receive and a 14 bit DAC running at 128 MS/s on transmit. The architecture allows sampling or generation of signals from 0 to 32 MHz, the entire HF band.

The digital information is processed by a field programmable gate array (FPGA), which performs down conversion of the signal of interest to near zero frequency and any decimation.

The FPGA passes the signal information to the USB interface for transmission to the PC for the bulk of the signal processing. The software involved is described later in this paper.

The transmit side is basically the process in reverse. Transmit signal information is passed from the PC through the USB port. In the USRP, the FPGA performs the interpolation and necessary up conversion to place the near zero frequency signal information at the desired frequency in the HF band. The output of the FPGA is passed to the DAC and the analog output is boosted to 100 milliwatts before leaving the USRP.

Almost all USRP specifications are programmable, offering maximum flexibility and experimental possibilities. Decimation and interpolation rates, transmit and receive frequencies, receive gain and transmit power are all controllable through software. It is capable of full duplex operation; simultaneously transmitting and receiving through separate transmit and receive connectors. Half duplex operation is also a programming option.

For other frequency bands, Ettus Research also manufactures daughter boards to plug into the basic USRP, covering VHF, UHF and microwave bands, currently up to 5.9 GHz. Analog mixers and programmable frequency synthesizers convert signals of interest down to the HF band, where they are processed by the basic USRP as described above. Similarly, transmit signals are up converted and amplified to 100 milliwatts. Like the main board, full and half duplex are available.

SDR developers can write their own code to process signals and control the USRP. However, there is an enormous body of pre-written, free software in GNU Radio.

GNU Radio is a community of programmers who have written blocks of code in C++ to handle a wide range of signal processing functions. These blocks can be combined and compiled using scripts written in PYTHON. PYTHON and GNU Radio blocks all run under a LINUX based system such as Ubuntu. The beauty of GNU Radio is in the “connect” function which allows programmers to string together the prewritten processing blocks, much the same as a radio design engineer connects subsystems to make a transmitter and/or receiver.

Some of the available blocks perform as low pass filters, using finite impulse response (FIR) or infinite impulse response (IIR) structures. The filters can be configured using other blocks of code which generate a vector of coefficients to program these filters based on a desired cutoff frequency, stop frequency and passband ripple. High pass and band pass structures are also included.

In addition to filter functions, modulation and demodulation blocks are provided for analog modes such as FM and the digital modes, including PSK, QPSK and m-QAM. Modulation and demodulation in other modes, such as AM and SSB are accomplished using more basic blocks such as multipliers and adders.

A variety of interface blocks are also included. These control the USRP and interface with the audio card on the PC or with files. This last feature is a distinct advantage in the authors' application. Signals can be pre-recorded off air and saved, unprocessed. Sample signals can be

stored this way and various processing schemes can be applied to the same signal, even if the USRP, external antennas or the signal are not available.

Finally, there are GUI blocks available which provide for the creation of windows with buttons and controls. Also FFT, oscilloscope and constellation displays are available and can be “wired in” to any point in the signal stream, providing built-in test equipment for development or displays for the end user. This feature is vital in the demonstrations created by the authors, allowing students to experience immediately and first-hand the principles of filtering and other signal processing concepts.

Programming in C++ and Python can be daunting and cumbersome. The authors were often frustrated in debugging the code created by the PYTHON scripts. Josh Bloom of GNU Radio has created a program, GNU Radio Companion (GRC)<sup>19</sup>, which allows someone with almost no experience in programming to create and manipulate complex routines using every standard block available in GNU Radio.

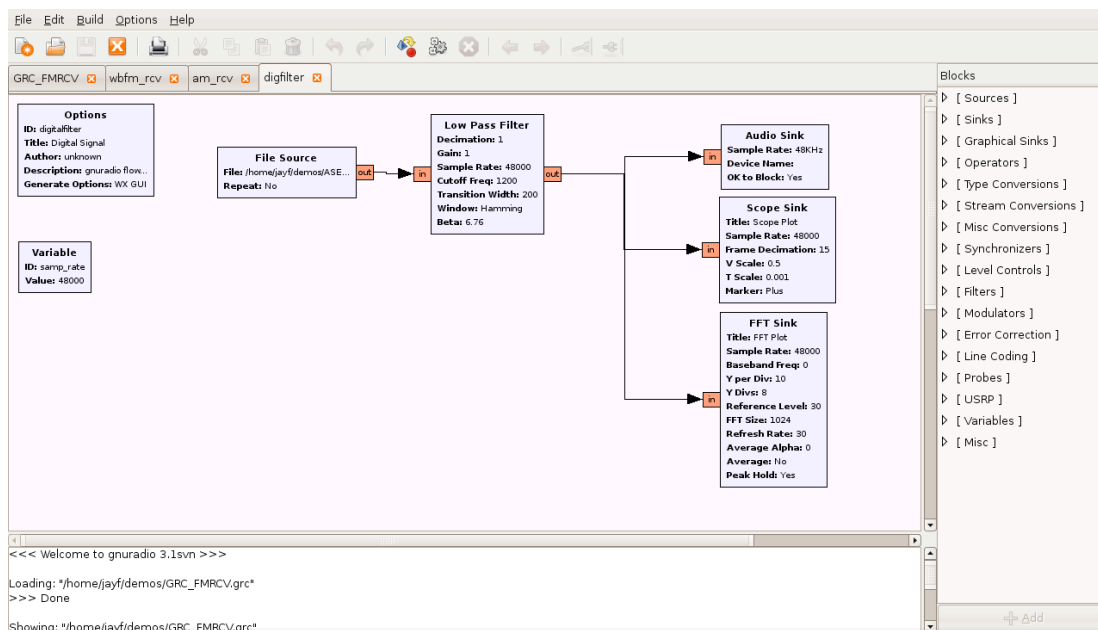


Figure 1. Screen shot of GRC

Using a GUI window (see figure 1) similar to MATLAB’s SIMULINK, the user can easily configure and interconnect GNU Radio blocks into working systems in minutes. The process is intuitive, with error identification and prompts directing the user to possible solutions. For example, if the user attempts to configure a low pass filter with the cutoff frequency higher than the stop frequency, the block title turns red and an error message appears pointing out the error. This feature allows users to rapidly locate and correct problems.

GRC includes all of the GUI blocks in GNU Radio, which were perhaps the most intricate to code, but the most desirable for the demonstrations.

GRC allows the user to run the programs inside the application or create a PYTHON script which can be run from the Ubuntu command line. The program can also be saved as a .grc file, which GRC can edit and manipulate.

Using GRC, the authors were able to create working programs, complete with GUIs, for such complex systems as an AM receiver, wideband FM receiver and an HF SSB receiver – all within the space of an hour. The capture of signals for demonstration, such as chunks of the AM broadcast band, HF bands and various baseband signals was equally straightforward.

A number of other platforms are available for implementing SDR systems. These include Soft Rock<sup>20</sup>, Flex Radio Systems<sup>21</sup>, High Powered Software Defined Radio<sup>22</sup>, and others<sup>23-26</sup>.

## **V. Demonstration Descriptions**

The authors developed six classroom demonstrations for a senior analog communications course, ECE 460, at California State University, Northridge. The goal was to give students a tangible, in-class experience with such concepts as the relationship between time varying signals and bandwidth, common baseband signals, linear systems and filters, amplitude modulation and angle modulation. The demonstrations simultaneously illustrated signals and processing in both the time and frequency domain, constantly reinforcing the connection between these important views of signals in the communications world.

### Demonstration 1: Relation of Time and Frequency Domains

It is vital that students understand the fundamental relationship between the time and frequency domains before proceeding deeper into any communications course. This demonstration allowed the students to see and hear this relationship.

A simple program was created in GRC connecting a white noise source to a low pass filter. The output of the low pass filter was routed to the computer's speaker to give students an audible impression, an oscilloscope sink to show the signal in the time domain and an FFT sink to give the frequency domain. The GNU Radio oscilloscope and FFT displays are shown in figure 2. The bandwidth of the low pass filter was then varied (see figure 3). The students could hear the change in the sound of the signal as the bandwidth was reduced. They could also observe that the signal varied more slowly as the spectrum shrunk.



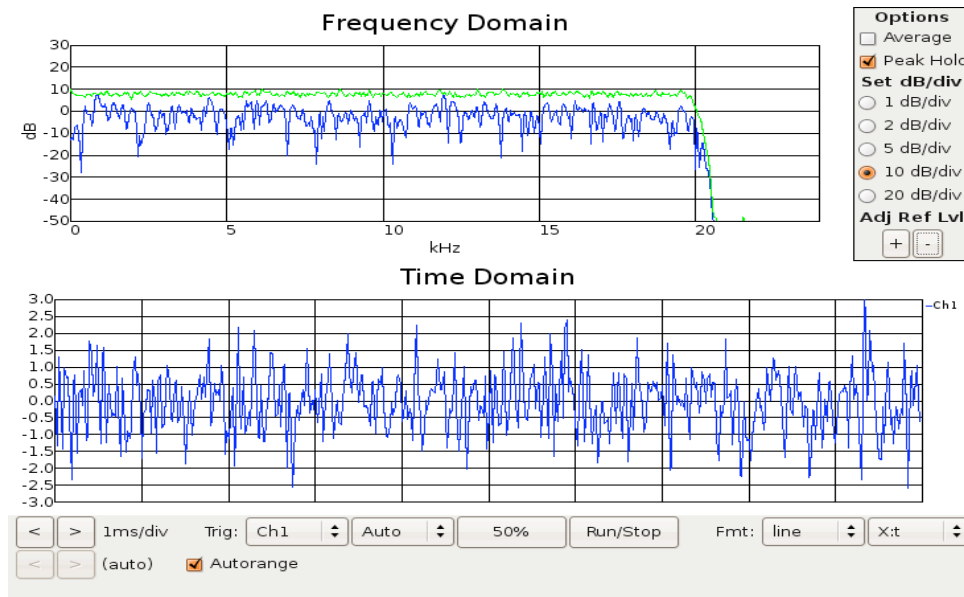


Figure 2. Wideband Signal Display

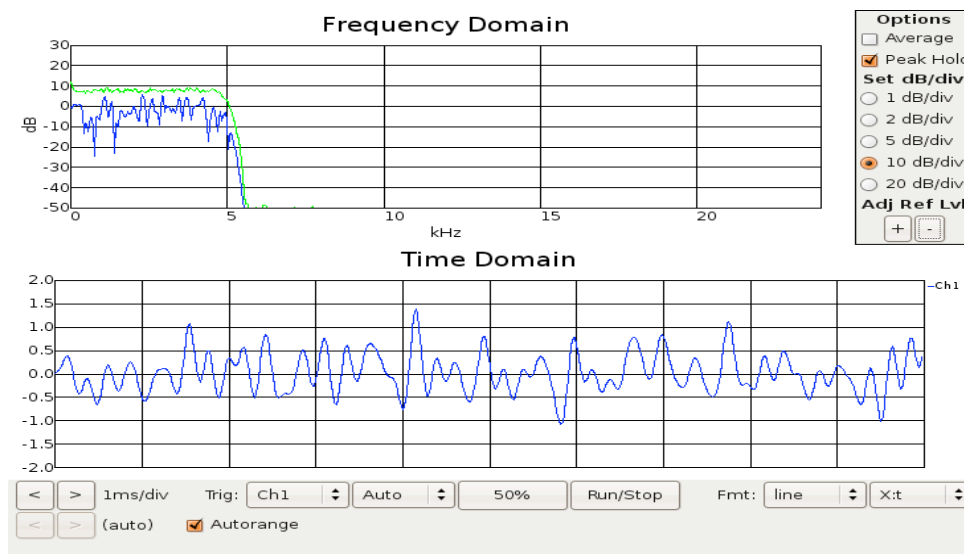


Figure 3. Narrowband Signal Display

Students could also observe the ancillary concept of the reduction of noise amplitude as the bandwidth was reduced.

In the second part of the demonstration, pre-recorded files of various common baseband signals were played back in GRC, with the output connected to the speaker and an FFT sink (see figure 4). These included a male voice, a female voice, music and low speed data (see figure 5). A NTSC analog video spectrum was also displayed. Students could observe the spectra and the differences in energy content at various frequencies while gaining an understanding of the signal sources that will be applied to the communications systems that they will study in the course.

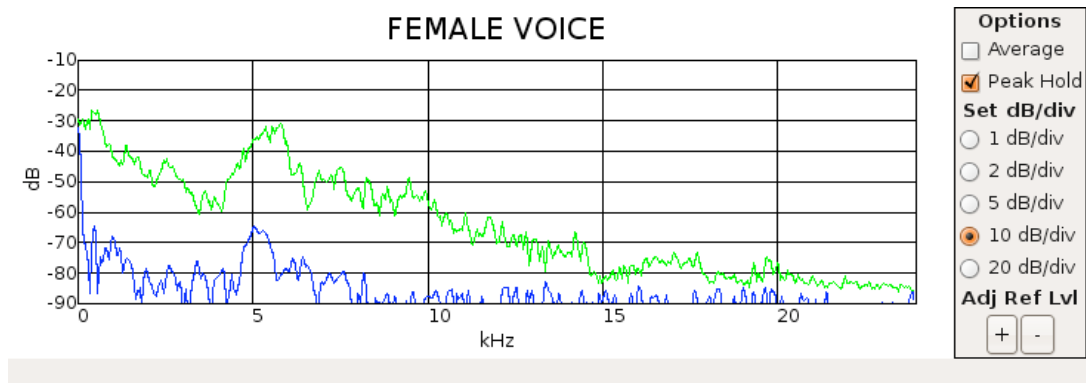


Figure 4. Female voice spectrum

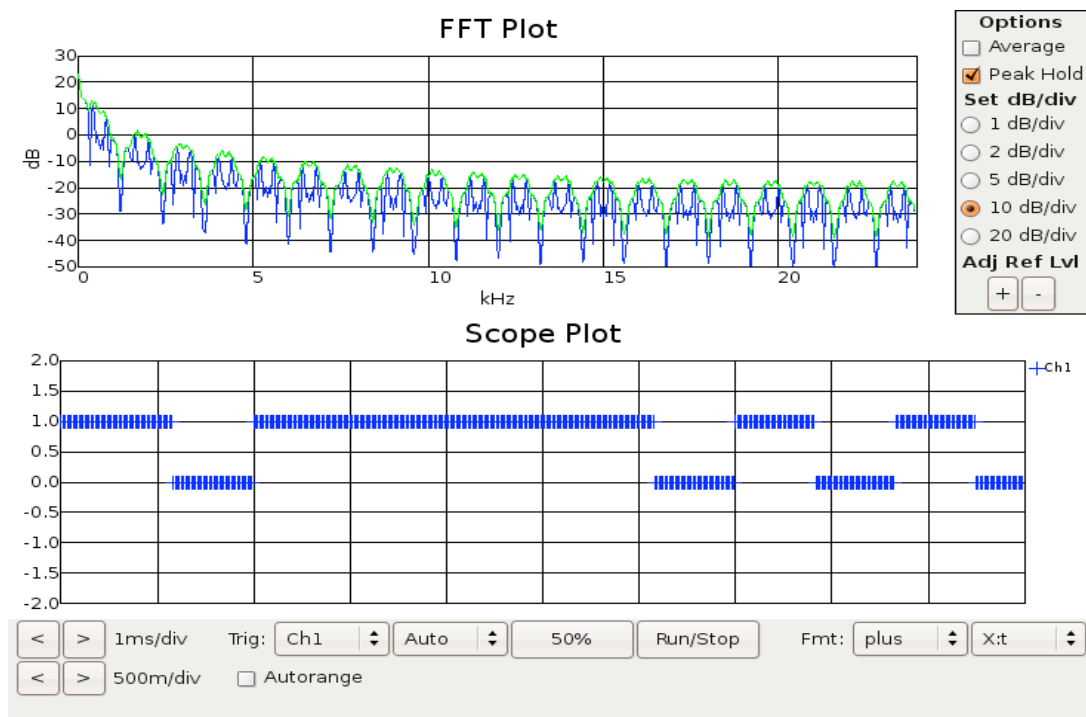


Figure 5. 1200 baud Data Signal

In the data signal, students could directly observe the position of the nulls in the spectrum and their relation to the data rate.

### Demonstration 2: Linear Systems and Filters

In this demonstration students observed the effects of filtering on the baseband signals encountered in Demonstration 1. High-pass, low-pass and band-pass filters with various bandwidths and cutoff frequencies were applied to the recorded voice samples. For example, students could hear how the signals remained easily intelligible even when the bandwidth of the voice signals was reduced to only 2.8 KHz, the nominal bandwidth transmitted over most communications circuits. In a second example, the 1200 baud baseband data signal was bandwidth limited to the first null. Students could observe that, although the signal edges slowed considerably, the original data could be recovered by sampling at specific points 1/1200 seconds apart (see figure 6).

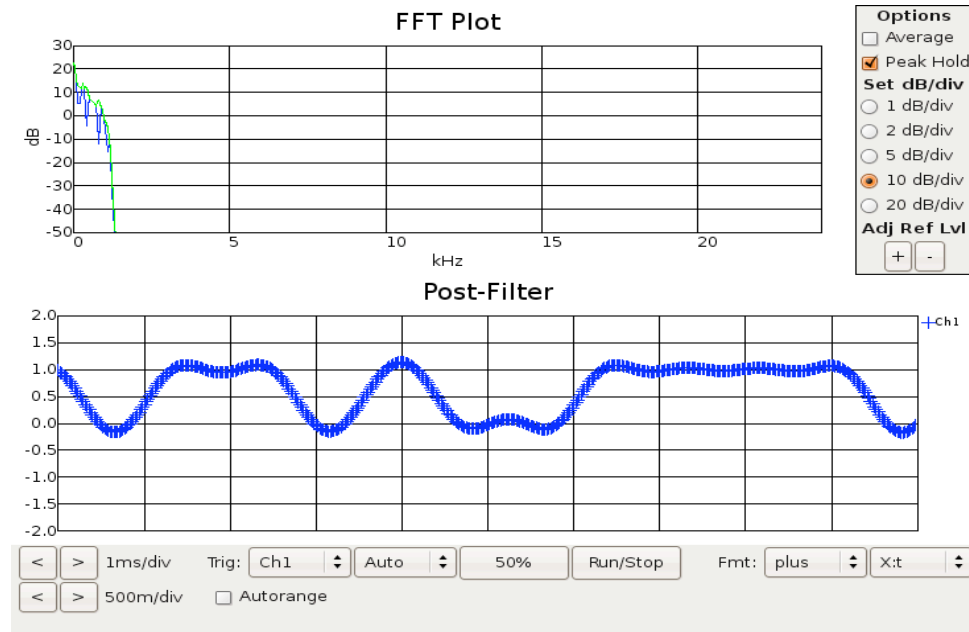


Figure 6. Bandwidth-limited Data Signal

The filters used in the demonstrations to this point were essentially near-ideal filters. In GNU Radio, it is possible to “look under the hood” of the filter process and see that the software can easily implement a 700<sup>th</sup> order filter with a 700 tap finite impulse response (FIR) filter. A line was added to the PYTHON script to print out the filter order on compilation.

SDR also illustrated the phase linearity of FIR filters when compared to infinite impulse response (IIR) filters. In this part of the demonstration, a square wave was generated and passed through an FIR filter and an IIR filter in parallel. Both filters had identical cutoff and stop frequencies. Students could readily observe the effects of non-linear phase response in the distortion of the square wave, an important concept to remember when dealing with data signals.

The demonstration concluded with the extraction of a communications signal from a band of other signals, the most common application of a filter. A pre-recorded section of the AM broadcast band containing multiple signals was played back and a band pass filter used to isolate the signal of interest for further processing (see figure 7). Students could hear the difference between the demodulated unfiltered signal and the demodulated filtered signal.

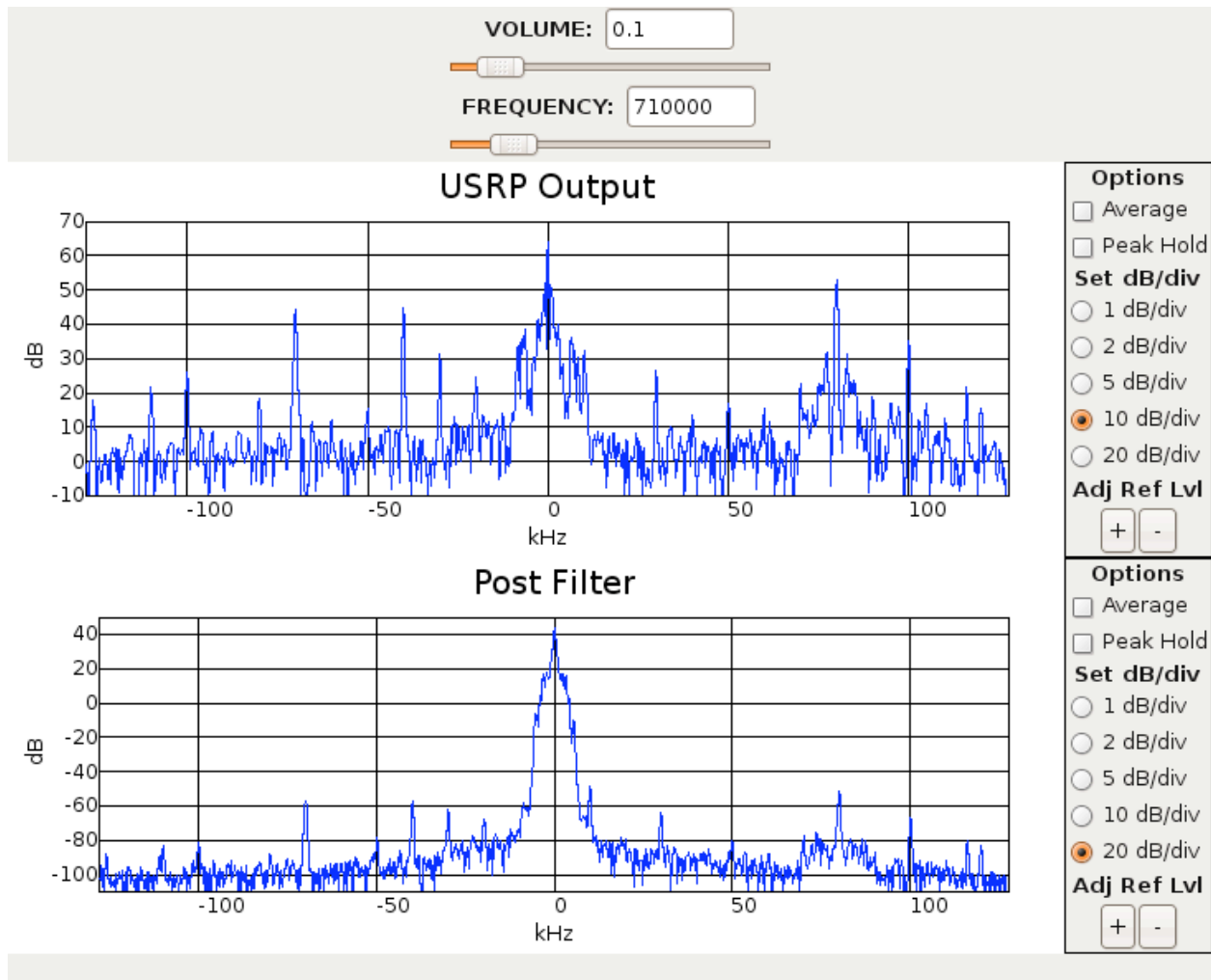


Figure 7. AM spectrum Before and After Filtering

### Demonstration 3: Amplitude Modulation

Picking up where Demonstration 2 left off, this demonstration dealt with the most basic form of modulation: amplitude variation of a carrier signal, including amplitude modulation (AM), double sideband (DSB) and single sideband (SSB). Both the generation and demodulation of these signals was covered in detail. Students could observe the relationship between the baseband bandwidth and that of the modulated signal.

This demonstration illustrates one of SDR's greatest powers when used in education: the direct application of mathematical equations to signal processing. The basic equation of an AM signal is:

$$s_{AM}(t) = (1 + ax_m(t))\cos(2\pi f_c t)$$

where:  $s_{AM}(t)$  is the modulated AM signal

$x_m(t)$  is the baseband signal

$f_c$  is the carrier frequency

Figure 8 shows the implementation of this equation in GRC. The upper signal source is the baseband signal at 1500 Hz with amplitude 1. A constant, 1, is added to the baseband signal before it is multiplied by the other signal source, 22.5 KHz, acting as a carrier. The output of the multiplier, the amplitude modulated signal, is sent to both an FFT sink and an oscilloscope sink to display the frequency and time domains, respectively.

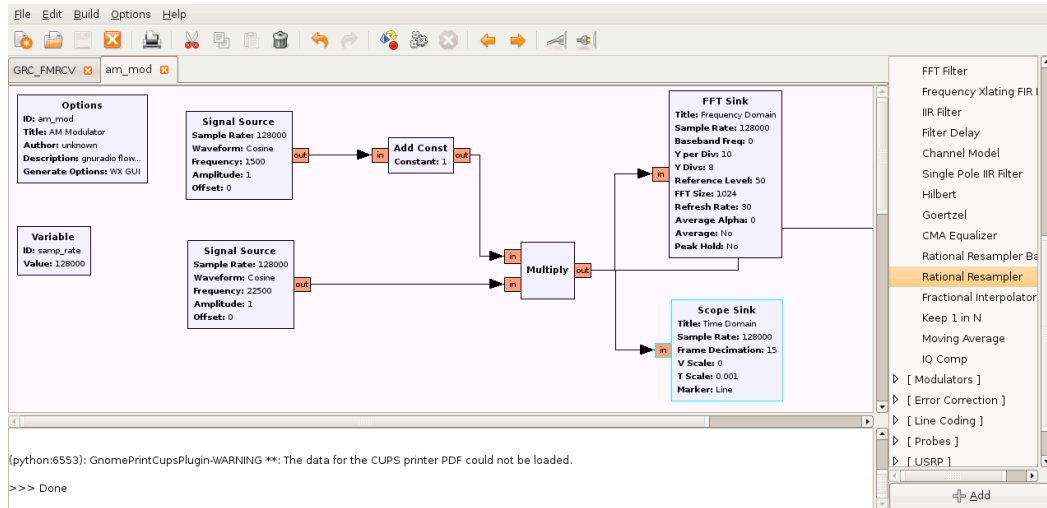


Figure 8. Amplitude Modulation Implementation

Students can readily follow the signal flow, with the function of each block clearly marked and relate it to the classic AM equation.

The output of the modulator (see figure 9) gives the satisfying result of the sine wave modulated carrier on the oscilloscope and, in the frequency domain, the two sidebands 6dB down, flanking the carrier 1500 Hz above and below it.

Back in the GRC window, an AM demodulator is added. Students can hear and see the output of the demodulator.

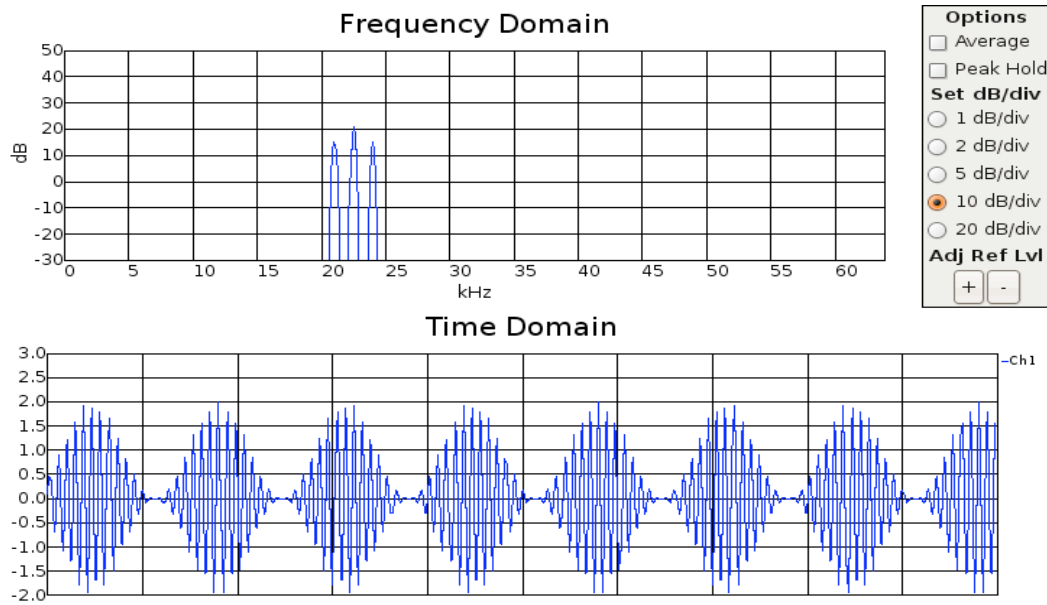


Figure 9. AM Modulator Output Display

In the next part of the demonstration, the signal flow was reconfigured “live” in front of the students with the removal of the constant being added to the baseband signal. A DSB signal was generated and the output in both the time and frequency domain was examined (see figure 10). Students could easily see the phase reversal of the carrier during the negative part of the baseband signal. They could also see that the AM demodulator will not produce the original signal but rather the full wave rectified cosine wave. They heard the distorted signal at twice the frequency as the original baseband.

A local beat frequency oscillator at the carrier frequency was added as part of the demonstration, replacing one of the inputs to the multiplier in the detector. Students could see the original baseband signal recovered (see figure 10).

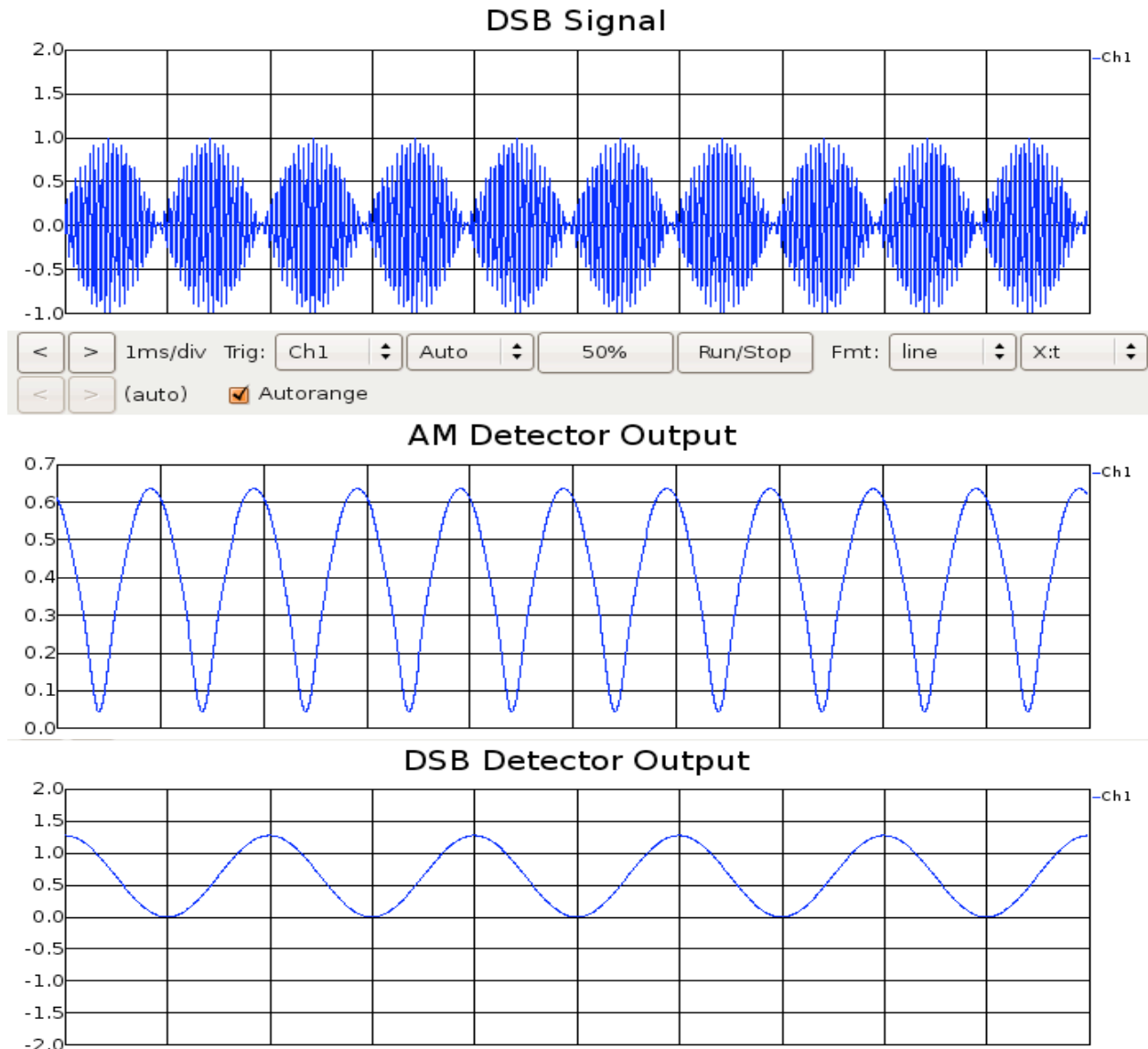


Figure 10. DSB Signal and Demodulator Outputs

The demonstration is then repeated with the recorded voice signal, bandwidth limited to 3 KHz, acting as the baseband signal. Students could hear the characteristic “Donald Duck” sound as the AM detector tries to recover the audio. With the beat frequency oscillator added, they clearly heard the original baseband signal.

The demonstration concluded with an illustration of SSB. With one of the sidebands of the DSB signal removed by filtering, the students could see the spectrum of the transmitted signal. Detection with the AM demodulator provides no output at all with a simple cosine signal source and the “Donald Duck” sound with voice modulation. Addition of the beat frequency oscillator produced the original intelligence. Here the authors also varied the frequency of the local oscillator, giving the students the experience of an off-frequency SSB signal.

At a later date, the class met in the communications lab, where the USRP was connected to an outdoor antenna. Students were able to see the signal flow in a SSB receiver implemented in GRC using Weaver's Method. Weaver's Method is highly complex and involves such concepts as analytical signals using in-phase(I) and quadrature(Q) components. Signals in the 20 meter HF amateur radio band were received (see figure 11). This demonstration could have also been carried out using pre-recorded signals captured by the USRP.

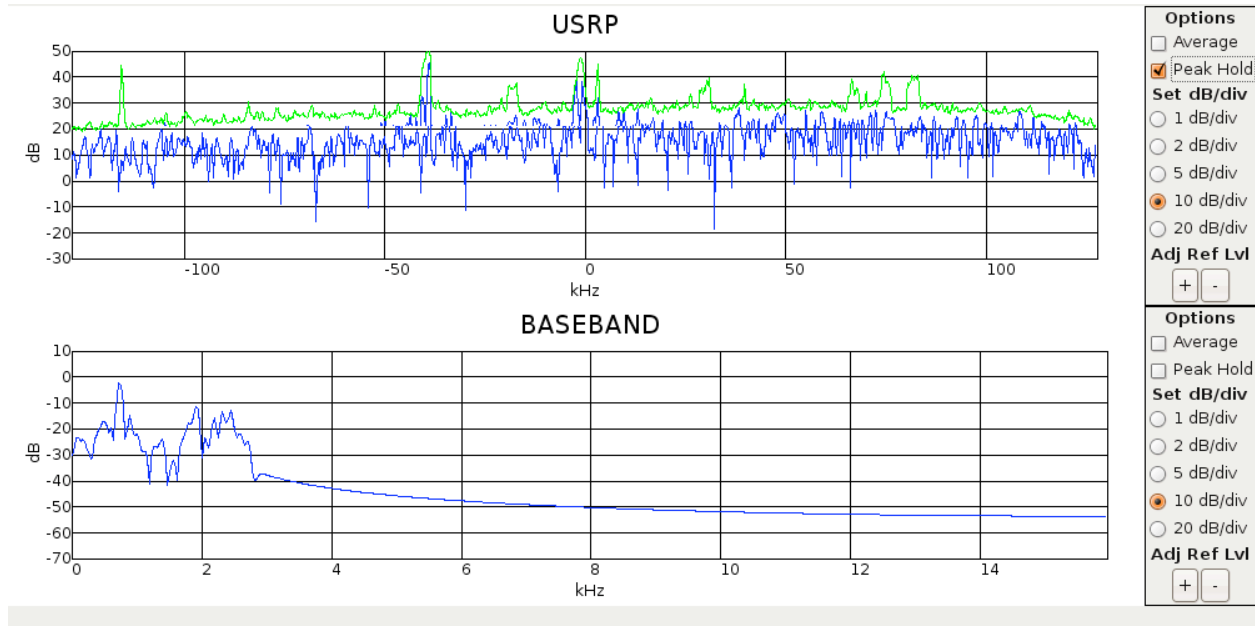


Figure 11. Spectrum of 20 Meter HF Amateur Radio Band

#### Demonstration 4: Angle Modulation

The final demonstration of the series dealt with the most complex form of modulation, angle modulation. At this point in the semester, students had learned about the mathematics behind phase and frequency modulation. They had been exposed to Carson's Rule and Bessel Functions for determining the bandwidth of an FM signal.

In GNU Radio, there are pre-written blocks to generate and demodulate phase and frequency modulated signals. Implementing either form of modulation with basic blocks was considered beyond the scope of the course and the authors decided to limit the demonstration to the bandwidth/modulation index relationship in FM.

Using the FM modulator block and a simple 1 KHz cosine signal, the spectrum of the modulated signal was observed. Students could observe that the amplitude of the FM signal did not change in the time domain, but the bandwidth of the signal increased as the modulation index increased as predicted by the Bessel function analysis. Confirmation of the Bessel functions for relating carrier amplitude to modulation index was demonstrated using a simple narrowband FM modulator and a baseband signal of 2079 Hz. The carrier null (50dB down) occurred, as predicted, at a deviation of 5 KHz (see figure 12).



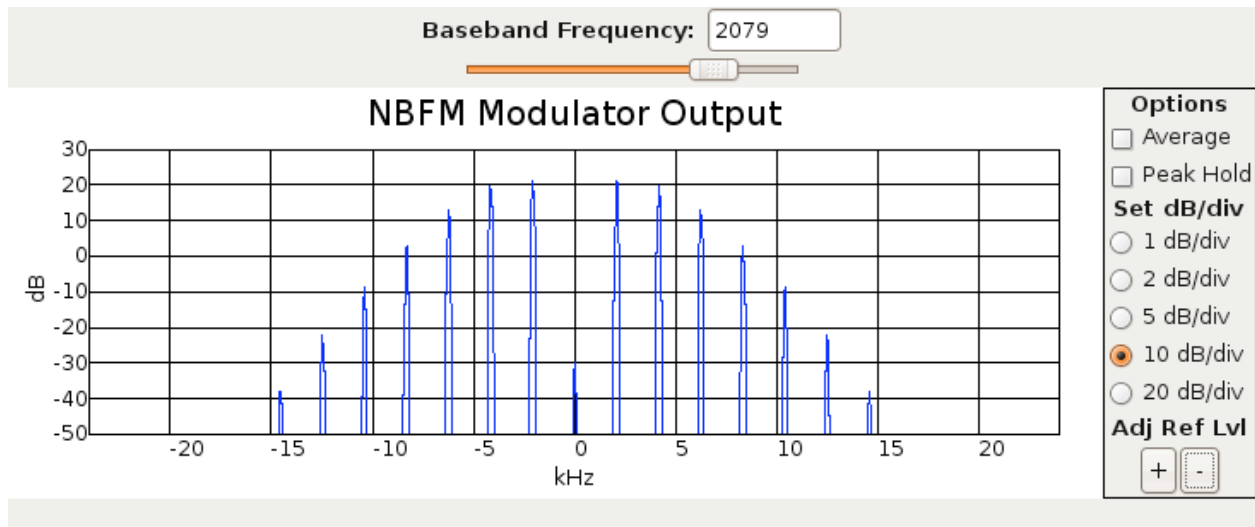


Figure 12. Sidebands and Carrier Null in NBFM Signal

The demonstration included some off-air commercial broadcast signals. A wideband FM receiver was implemented in GRC and students could hear the distortion increase in the demodulated signal as the bandwidth of the receive filter was reduced. Students were also exposed to the various subcarriers riding on the broadcast signal, including the stereo information and stereo pilot tone, subsidiary communications authorization signal (SCA) and radio data system signal (RDS) (see figure 13).

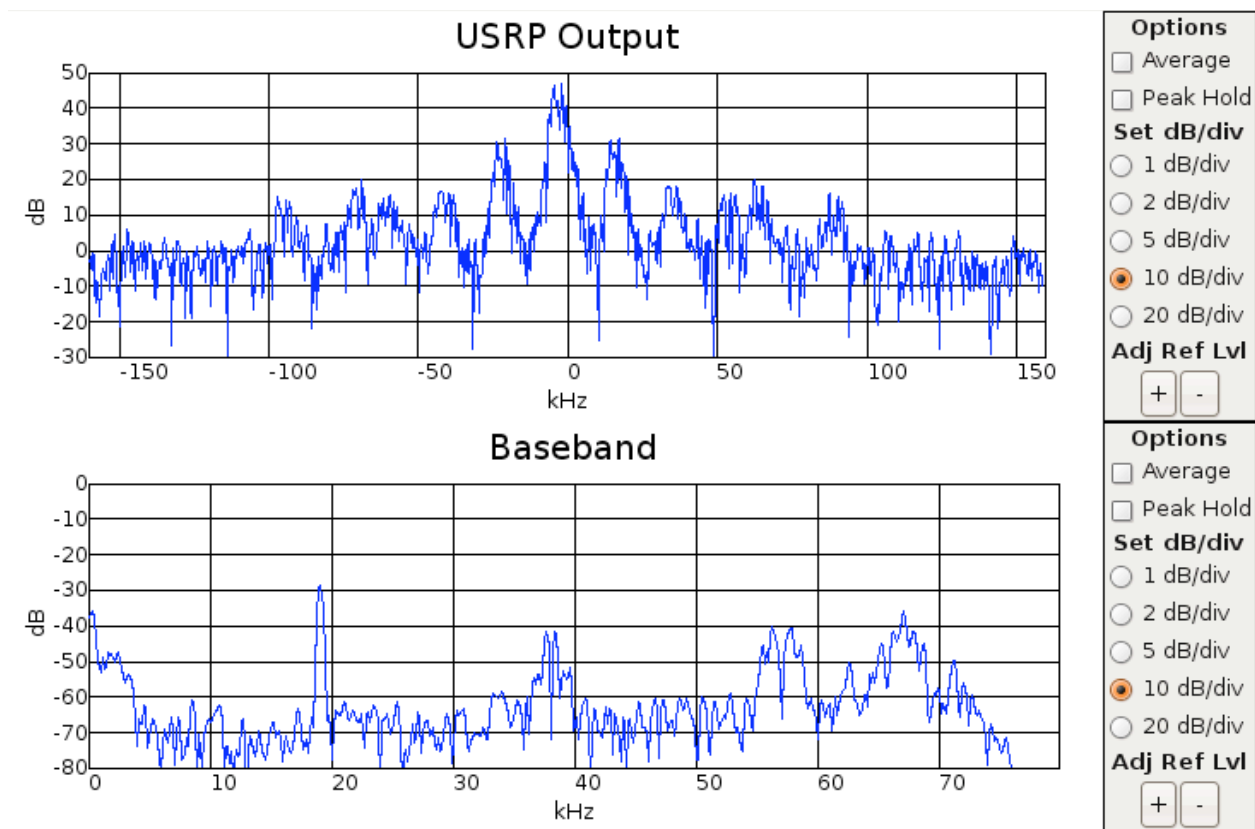


Figure 13. Composite FM Broadcast Signal

Finally, using a narrow band FM receiver implemented in GRC, the authors demonstrated the quieting and threshold effects associated with an increasing signal to noise ratio. The authors used a commercial handheld radio and a variable attenuator on antenna input to the USRP. The students could hear and observe the demodulated signal to noise ratio and compare it to the incoming carrier to noise ratio using FFT sinks placed just after the USRP and after the FM demodulator.

## **VI. Results**

In the Fall 2008 semester, the demonstrations described above were presented in the ECE 460 course (analog communications) at California State University, Northridge. The class was composed of 20 students, primarily undergraduates. This course covers a review of linear systems and signals followed by material on AM, DSB, SSB, FM, PM and system performance. Throughout the semester, the demonstrations were presented to the class as the appropriate topics were covered in class. It was clear after the first demonstration that they created student interest in the subject. This was demonstrated by an increase in participation during the lecture. This gave the authors encouragement to continue to develop subsequent demonstrations.

There is also evidence to support the claim that the demonstrations increased student interest in communications as a field of study among undergraduate electrical engineering students. Of the 16 undergraduate students in this course, all but two continued on to the digital communications course offered the following semester. Generally less than 5 undergraduate students continue on to this second level course.

Because of their increased interest in the field of communications, a large number of students expressed a desire to complete their senior design projects in the area. As a result, this semester the authors are supervising 12 undergraduates on senior design projects that utilize SDR to implement complete communications systems.

Using SDR to create these demonstrations provides the additional advantage of introducing students to an emerging interdisciplinary technology. Furthermore, the entire system with all its capabilities can easily migrate to the lab. Instructors can even extend the use of this demonstration platform to courses in digital communications and digital signal processing.

## **VII. Conclusions and Plans for Future Work**

Based on the experience with these demonstrations, we conclude that the use of demonstrations in communications courses can improve student learning, motivation, and interest in the area of communications systems. SDR is an excellent platform for creating demonstrations since it allows the creation of complex systems without the complex hardware constructions and debugging.

Due to the success of this project, the authors are continuing this work with the creation of a set of SDR demonstrations for the digital communications course. These will be developed and used in the spring 2009 semester at California State University, Northridge.

## Bibliography

1. Goodman, Peter, "A Software-Defined Radio Project for First Year ECET Students," Proceedings of the 2007 ASEE Annual Conference.
2. Goodman, Peter, "Geek Civilization: Amateur Radio and First Year Projects to Improve Recruitment and Retention in an ECET Program," Proceedings of the 2006 ASEE Annual Conference.
3. Kubichek, Robert, et al, "A Comprehensive Suite of Tools for Teaching Communications Courses," Proceedings of the 2006 ASEE Annual Conference.
4. Dunne, Bruce, and Cooke, Melvin, "Design of a Hardware Platform for Analog Communications Laboratory," Proceedings of the 2008 ASEE Annual Conference.
5. Frolik, Jeff, "A Comprehensive, Laboratory-Enhanced Communications Curriculum," Proceedings of the 2004 ASEE Annual Conference.
6. Gonzalez, Virgilio, and Mehdi, Shadaram, "Development of a Communications Course Integrating a Virtual Laboratory and Complex Simulations," Proceedings of the 2005 ASEE Annual Conference.
7. Frolik, Jeff, "Laboratory Enhancement of Digital and Wireless Communications Courses," Proceedings of the 2005 ASEE Annual Conference.
8. Hofbeck, Joseph, and Melton, Andrew, "RF Signal Database for a Communication Systems Course," Proceedings of the 2006 ASEE Annual Conference.
9. Hofbeck, Joseph, "Using Real RF Signals such as FM Radio to Teach Concepts in Communication Systems," Proceedings of the 2008 ASEE Annual Conference.
10. KD7LMO.net: [http://www.kd7lmo.net/ground\\_gnuradio\\_baseline.html](http://www.kd7lmo.net/ground_gnuradio_baseline.html)
11. Bard, John and Kovarik, Vincent, "Software Defined Radio: The Software Communications Architecture," Wiley Series in Software Radio, 2007.
12. Kenington, Peter, "RF And Baseband Techniques for Software Defined Radio," Artech House, 2005.
13. Fette, Bruce, "Cognitive Radio Technology," Newnes, 2006.
14. Reed, Jeffrey, "Software Radio: A Modern Approach to Radio Engineering," Prentice Hall, 2005.
15. GNU Radio, the GNU Software Radio: <http://gnuradio.org/trac>
16. Ettus Research LLC: <http://www.ettus.com/>
17. Kragh, Frank, et al, "Education in Software Defined Radio Design Engineering," Proceedings of the 2008 ASEE Annual Conference.
18. Bilen, Sven, "Implementing a Hands-On Course in Software Defined Radio," Proceedings of the 2006 ASEE Annual Conference.
19. GNU Radio Companion: <http://www.gnuradio.org/trac/wiki/GNURadioCompanion>
20. Softrock and other SDR Radios: <http://www.softrockradio.org/>
21. Flex Radio Systems, Software Defined Radio: <http://www.flex-radio.com/>
22. High Performance Software Defined Radio: <http://hpsdr.org/>
23. Youngblood, Gerald, "A Software-Defined Radio for the Masses, Part 1," QEX, Jul/Aug 2002.
24. Youngblood, Gerald, "A Software-Defined Radio for the Masses, Part 2, QEX, Sep/Oct 2002.
25. Youngblood, Gerald, "A Software-Defined Radio for the Masses, Part 3, QEX, Nov/Dec 2002.
26. Youngblood, Gerald, "A Software-Defined Radio for the Masses, Part 4, QEX, Mar/Apr 2003.