A SOFTWARE-DEFINED RADIO IONOSPHERIC CHIRPSOUNDER FOR HF PROPAGATION ANALYSIS

Pradeep B. Nagaraju*, Eric Koski[¥], Tommaso Melodia*
*Department of Electrical Engineering, State University of New York at Buffalo, NY, USA
[¥]Harris Corporation, RF Communications Division, Rochester, NY, USA

SUMMARY

Advanced users of high frequency (HF) radio communications have long used systems known as chirpsounders[1] to obtain real-time information about the ionospheric propagation conditions of different communication channels between a transmitter and a receiver. Traditional chirpsounders are expensive to realize and lack the flexibility of state-of-the art digital radio platforms. However, the advent of Software Defined Radio (SDR) technology has created opportunities to provide a chirpsounder capability in a flexible, low-cost form. In this paper, we report on our development of a prototype Software Defined Radio (SDR) chirpsounder system based on a commercially-available SDR platform. The accurate real-time picture of ionospheric propagation provided at low cost by such a chirpsounder can be instrumental in providing HF communications users with self-configuring link optimization capabilities that dynamically select the best channels to be used in an HF link, to maximize communications capacity and reliability.

1 INTRODUCTION

Services used by military, marine, aviation, and amateur radio users rely on HF signals that propagate through the ionosphere over long distances. The ionospheric layers are subject to variations caused by factors like absence or presence of sunlight, seasons, sunspot cycle, solar activity, polar aurora, among others. All these factors make ionospheric propagation stochastic in nature and variable over relatively short time scales. For this reason, ionospheric chirpsounders have been traditionally used to provide an accurate estimate of the ionospheric propagation characteristics. Chirpsounders are linearly varying frequency (LVF) modulators that vary between 2.0 MHz to 30.0 MHz, designed to estimate the capacity and reliability of high frequency (HF) links over different channels. A chirpsounder is one kind of *ionosonde*, a device for measuring the characteristics of the ionosphere by transmitting a sounding signal and analyzing the signal returned to earth as a result of ionospheric refraction.

This paper describes our experience in developing what is, to the best of our knowledge, the first software-defined HF chirpsounder developed on a currently-off-the-shelf radio platform. The chirpsounder is realized using the Flex-radio 5000A [1] (F5k) software defined radio [2] (SDR) along with the powerSDR open source software platform. Compared to traditional chirpsounders, the proposed SDR-based chirpsounder provides an inexpensive and reconfigurable solution for automatic frequency management. The chirpsounder system consists of a transmitter and a receiver. The transmitter transmits a 100W swept CW tone [11] across much of the entire HF spectrum, traversing the spectrum upward in frequency at 100 kHz per second, so that a complete sweep of the 2.0 MHz to 30.0 MHz HF spectrum takes 280 seconds to complete.

---

[1] This work was supported by Harris Corporation.

The receiver is time-synchronized to the transmitted chirp and tracks and analyzes it so as to produce an ionogram, i.e., a chart of the received signal strength as a function of frequency and latency. By inspecting the ionogram, experienced users can quickly identify the propagating layers of the ionosphere and their critical frequencies, i.e., the maximum frequencies above which transmitted signals are not refracted back to the earth. The chirpsounder can be coupled with specialized communication software to generate an automated frequency management system that adaptively controls the operation of the HF communication link so as to dynamically allocate reliable channels for HF signals to propagate. The proposed software defined chirpsounder can flexibly vary the sweep rate, receive different chirp sounder formats used in some of the traditional chirpsounder systems, enable the user to choose different chirp sounding formats for transmission, manage the forbidden frequencies and power as per the FCC standards and provide portability on different platforms. To test the chirp sounders for various transmission formats in the indoor milieu, we incorporate Wattersons [3] ionospheric channel modeling at the transmitter with transmit power used less than 500 mW. The user can also incorporate and use alternative channel models. All these functionalities are selectable at run time at user discretion. Finally, we discuss techniques to implement chirpsounders on software defined radio both for indoor testing and field deployment. Extensive experimental performance evaluation results will be reported to assess the performance of the software-defined chirpsounder.

The rest of this paper is structured as follows. In Section 2, we review related work. In Section 3, we discuss the software and hardware architecture of the proposed chirpsounder. In Section 4, we discuss details of the chirpsounder transmitter and receiver.

2 RELATED WORK

Barry Research (BR), now TCI, was the first organization to develop commercially-available chirpsounders. TCI/BR chirpsounders use traditional analog techniques to realize a chirpsounder. For this reason, the TCI/BR-designed chirpsounders cannot be modularized and extended, and cannot be incorporated into automated HF communication systems where frequency management is the key for achieving high quality links over the ionosphere. Moreover, these chirpsounders are very expensive and no longer commercially available.

The Radio Oblique Sounding Equipment [4] (ROSE) 100/200/300 is another series of commercial ionosonde. They use modern digital signal processing techniques to realize ionosondes with less hardware dependency. ROSE ionosondes produce ionograms with higher frequency resolution than those of TCI/BR. However, the hardware is specifically designed for implementing the ionosonde. This makes ROSE to be unsuitable for extending into automated HF communication systems.

Digisondes [5], developed at the UMass Lowell Centre for Atmospheric Research, are mainly used as vertical sounders. Vertical sounders do not propagate along the ionosphere. They are used to determine the characteristics of the ionosphere at a specific geographic location rather than along a signal transmission path of interest. Hence, it is difficult to predict the ionospheric propagation characteristics. Digisondes are characterized by high power consumption since they use pulsed sounding techniques.

To provide a low-power and inexpensive solution, a software-defined-radio-based chirpsounder is proposed in this paper. Due to rapid prototyping and real-time adaptation capabilities, the F5k based chirpsounder can be easily extended to other high frequency ionospheric communication applications.

3 SOFTWARE DEFINED CHIRPSOUNDER

A software-defined radio is used to realize the chirpsounder. A generic SDR platform such as USRP2 (universal software radio peripheral 2) by Ettus Research [6] coupled with the GNU Radio signal processing tool is a flexible platform to implement wireless communication systems in software over a wide range of frequencies. However, chirpsounders are used to analyse ionospheric communication channels in the frequency range between 2.0 MHz to 30.0 Mhz. There are frequency bands within 2.0 MHz and 30.0 MHz that are used for broadcasting, emergency, amateur communications, and defence communications. According to the tactical performance requirements of MIL-STD-188-141B [15], the harmonic suppression should be -40dBc or better. To use generic SDRs, a power amplifier with filters must be used to achieve the required harmonic suppression. At the receiver, an adequate dynamic range for selectivity is required. This ensures that the receiver does not lose sensitivity in the presence of strong out-of-band signals. Tactical desensitization should be 90dB according to the specifications of MIL-STD-188-141B. This requires an automatic gain control (AGC) (with a good trade-off between speed and performance) to get the RF signals to the A/D (analog-to-digital) converter within the dynamic range limits of the receiver. To achieve these performance levels with a general-purpose system made up of off-the-shelf components would be complex and expensive. The F5k is designed to provide a cost-effective RF ‘front-end’ for an HF radio system meeting stringent RF performance requirements such as those prescribed by the MIL-STD-188-141B standards. Thereby, we have chosen to use F5k as the software defined radio platform for realizing the chirpsounder.

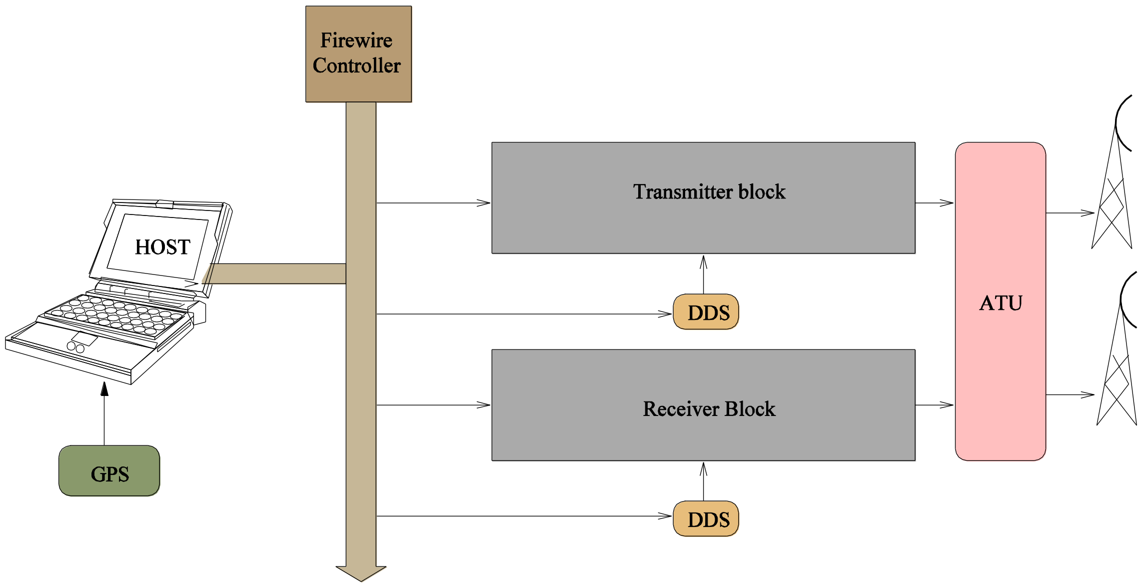

Figure 1. Chirpsounder Architecture.

3.1 SOFTWARE DEFINED CHIRPSOUNDER ARCHITECTURE

The overall architecture of the software-defined chirpsounder is shown in Fig.1. There are two main components in the entire architecture, the host and the F5k.

3.1.1 Host

In the current chirpsounder implementation, the host is a 3 GHz Intel core2-duo processor with 6MB L2 cache and 4GB RAM. The host is responsible for all signal processing activities, controlling the F5k and transceiving the In-phase (I) and Quadrature (Q) samples to and from F5k.

Figure 1 shows the software functionalities performed on the host. The controller is the manager of the chirpsounder application on the host. It manages the PowerSDR [1] firmware, which is designed to access the F5k for controlling it and exchanging the I/Q samples with it. The DSP logic core comprises the signal processing logic of the chirpsounder transmitter and receiver. The samples that are exchanged with the F5k are processed within the DSP logic core, as explained in detail in Section 4.

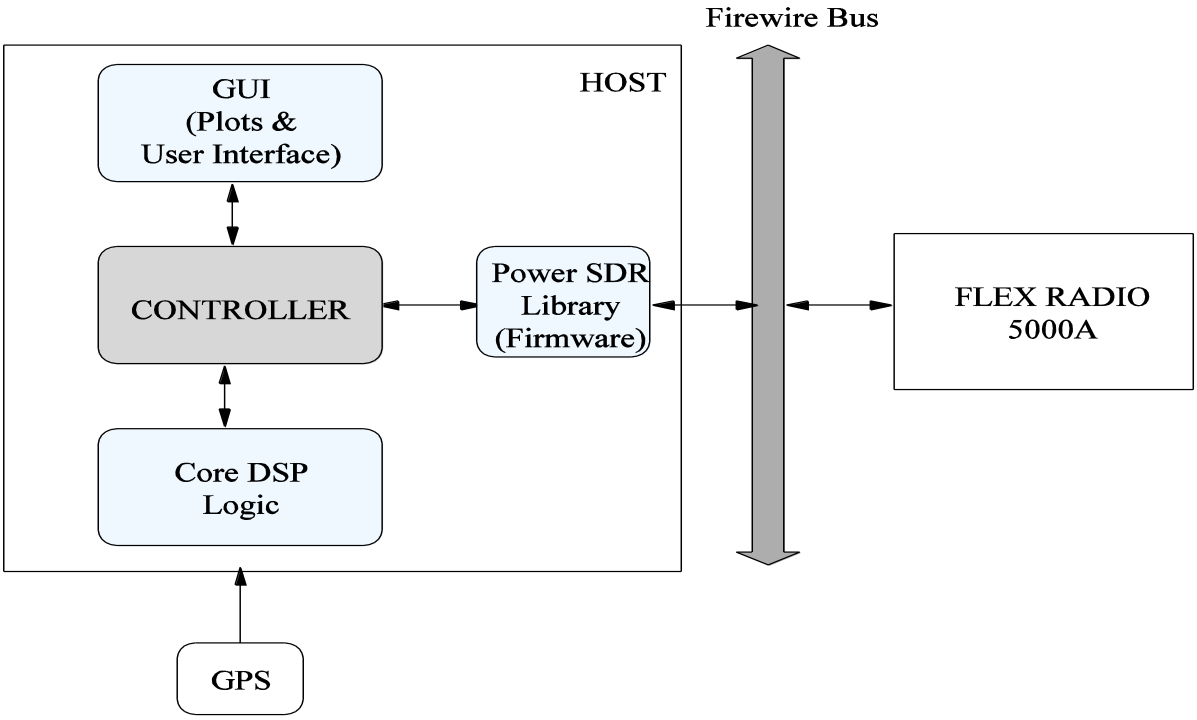

Figure 2. Host Architecture.

A graphical user interface (GUI) is managed by the controller to provide users with access to the DSP logic core and the F5k. The chirpsounder transmitter and receiver may be deployed across the globe. Global positioning system (GPS) is used by the chirpsounder application to synchronize both transmission and reception of the chirpsounder. Both 1 pps and the National Marine Electronics Association (NMEA) messages of the GPS are used to synchronize the sweep time of both transmitter and the receiver. GPS time synchronization with the chirpsounder application is also managed by the controller.

A modular approach, as shown in Fig. 2, is very important to test, debug and add new functionalities to the chirpsounder without having to change the whole structure of the application.

3.1.2 <u>The F5k Radio</u>

As shown in Fig.1, the F5k consists of a transmitter block and a receiver block with full duplex capabilities. There are two Direct Digital Synthesizers (DDS) to tune the operating frequency of the F5k. Each DDS has a resolution of 1 Hz with a total tunable bandwidth of 500 MHz. The F5k includes an antenna tunable unit (ATU) for coupling the antenna without loss for varying frequencies. The chirpsounder bypasses this ATU, since tuning the ATU might take significant time and the chirpsounder may not be able to sweep at the desired rate. However, a broadband antenna is used to couple with the F5k to minimize the losses due to antenna load mismatch and thereby achieve the desired sweep rates. The front end filters of the F5k are designed to suppress the harmonics to match the tactical performance requirements of MIL-STD-188-141B.

The F5k uses a firewire interface to both control the hardware and exchange I/Q samples between the host and itself. However, the techniques adopted to address these two functionalities are different as explained in Section 3.2. The F5k uses the Digital Interface Communication Engine (DICE) controller chip from TC technologies [16] to provide firewire interface to the host. DICE firewire chip communicates to the host through a physical abstraction layer (PAL) firmware as explained in Section 3.2.1. However, Flex-Radio has wrapped the PAL firmware with F5k specific functionalities. Hence, the F5k hardware functions are controlled through this Flex-Radio-DICE-PAL firmware.

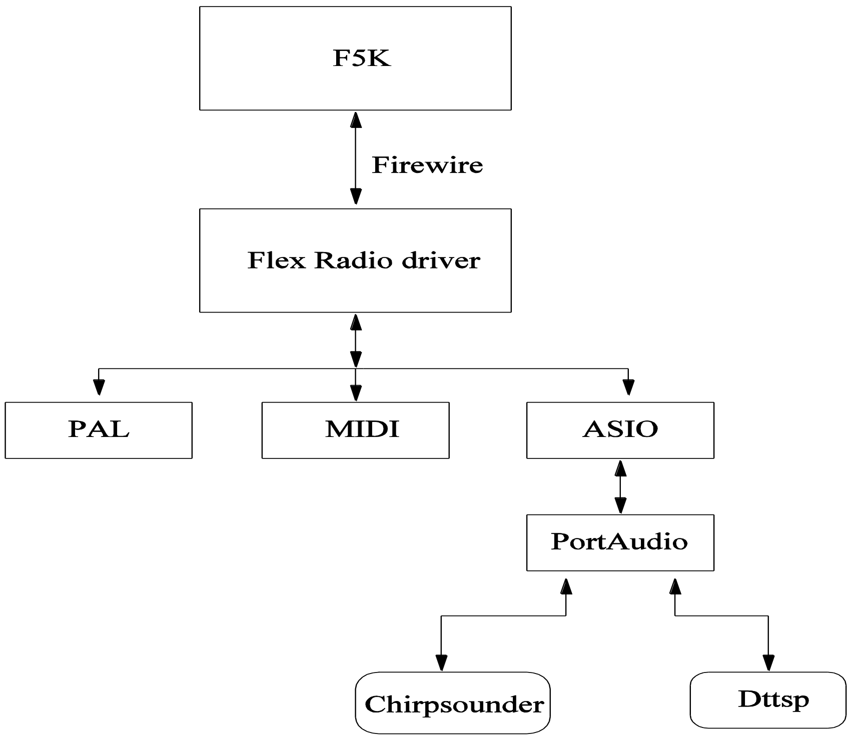

Figure 3. Firmware Diagram.

3.2 F5K FIRMWARE

The powerSDR library provided by Flex Radio includes hardware control functions like adjusting the DDS for frequency, enabling the QSE/QSD [7] (Quadrature sampling exciter/detector), enabling the DAC/ADC [8], and exchanging data samples between the host and the F5k. As a result of our exploration of F5k and powerSDR, we have been able to reconstruct a block diagram of the existing modular hierarchies in the F5K firmware.

The block diagram shown in Fig. 3 describes the relationship between the Chirpsounder and the hardware, and the different software modules that makes it possible to access the radio hardware. As discussed before, the F5k interfaces to the host through a firewire connection. The Flex-Radio driver is an interface between the computer and the hardware through firewire. The application that needs access to the F5k interacts with the Flex-Radio driver in two different ways: control the hardware functions through the F5k physical layer abstraction (PAL) firmware and exchange the I/Q samples through the audio drivers on the host.

3.2.1 Hardware control

The basic hardware controls such as setting up the transceiver, hardware filters, frequency tuning, enabling ADC/DAC, full duplexing, and accessing EEPROM are achieved through the platform abstraction layer (PAL). The F5k also uses a musical instrument digital interface (MIDI) interface for hardware control, but only for a limited set of functions such as manual tuning.

3.2.2 Transceiving the digital samples

The IQ [9] samples are exchanged between the computer and the F5k through three different interfaces. The Chirpsounder interacts with the port audio for sending/receiving the samples, since port audio manages samples between various audio drivers within the computer. Unlike the windows audio driver, the audio streaming input output (ASIO) bypasses many layers of the windows operating system to access the F5k. This helps in achieving high data rates between the host computer and the F5k. Hence, the ASIO driver provided by Flex Radio is configured to access F5k as its hardware device. ASIO interacts with the Flex Radio driver to access the samples to/from the F5k. Thus, the Chirpsounder interacts with the ASIO to send/receive from the F5k through port audio.

4 CHIRPSOUNDER DESIGN

This section represents the DSP core logic of the chirpsounder application on the host, and illustrates the design of the transmitter and receiver.

4.1 TRANSMITTER

The chirp signal is a LVF signal as shown in Fig. 4. There are at least two possible ways to achieve the chirp signal at the transmitter and the receiver. The F5k radio uses a DDS to tune the frequency of the carrier wave. The DDS can be tuned in steps of 1 Hz up to 500 Mhz with a speed of 250 MHz/s. The DDS tuning is phase continuous, and hence it can be used in high-speed varying-frequency applications like the chirpsounder. However, our experimental evaluation showed that, with the current F5K firmware, the DDS can be retuned at most 55 times per second. This is attributed to the fact that Flex-Radio wrapped PAL code has a overhead in accessing the DDS as fast as 250MHz/s. Hence, only retuning the DDS is not enough to realize a chirpsounder sweeping at a rate of 100 kHz/s.

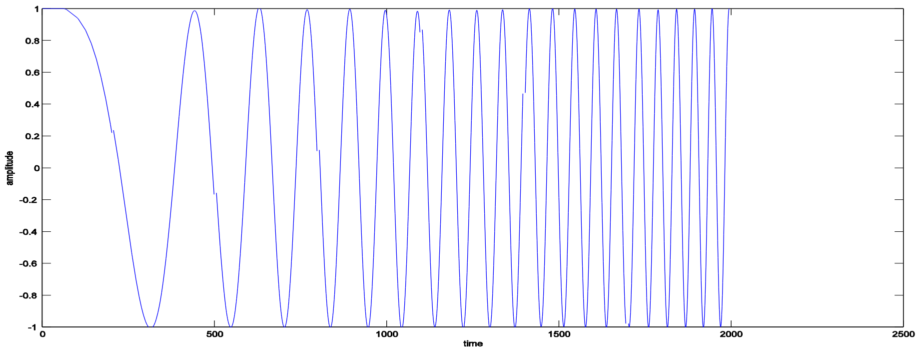

Figure 4. Chirp Signal.

The second alternative is to generate the LVF samples on the host and then send these across the firewire interface to the F5k. However, the front-end DAC of the F5k can process only 192000 samples/s [1]. This limits the bandwidth of the F5k to a maximum of 96 kHz. Since the DDS can be retuned 55 times every second, in our design it is retuned in steps of 96 kHz every 0.960 seconds to achieve a 100 kHz/s sweep rate. Figure 5 shows the spectrum that is plotted as frequency vs time. Chirp samples are obtained for sweep rates of 100 kHz/s with a sampling rate of 192000 samples/s. It can be noticed that the chirp signal runs out of samples beyond 96 kHz. Hence, we need to tune the DDS every 0.960s in steps of 96 kHz. For sweeping at higher rates than 100 kHz/s, the DDS must be re-tuned in less than 0.960s. The

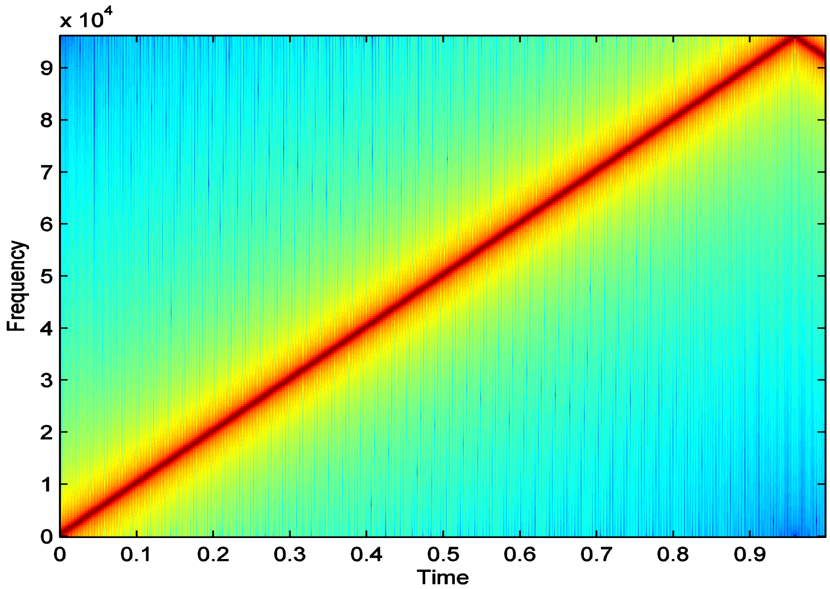

Figure 5. Sweep Spectrum.

upper limit of DDS retune rate depends on the computing platform. To determine the upper limit of the DDS retuning performance, a platform performance analysis routine is implemented at the start-up of the chirpsounder program. This includes tuning the hardware 1000 times and calculating the time taken for each retune.

4.2 RECEIVER

Figure 6 shows the receiver operations. The samples from the F5k are received by the host via firewire using the firmware, as explained in Section 3. The received digital samples are time stamped before they are processed. Since the constant-rate sweep tone is known at the receiver, a matched filter technique is implemented to detect the received swept tone. The matched filter produces peaks at locations where the correlation between the received sampled and the swept frequency sample at the receiver is maximized. By performing a maximum likelihood ratio, the samples whose peaks are at places with maximum correlation are extracted. An FFT is performed on these samples to determine the frequency of the received swept samples. FFTW (FFT wisdom), an optimized C/C++ library, is used to perform the FFT. For each of those frequencies received, we calculate its signal strength and time lag. The time lag is a measure of the time elapsed when the receiver has swept that frequency and the same frequency component is received by the receiver. Using this information, a plot of the signal strength as a function of propagation delay and frequency, known as an ionogram, is plotted. This ionogram provides accurate information on the effects of the ionosphere on various frequencies that are swept between 2.0MHz and 30.0MHz.

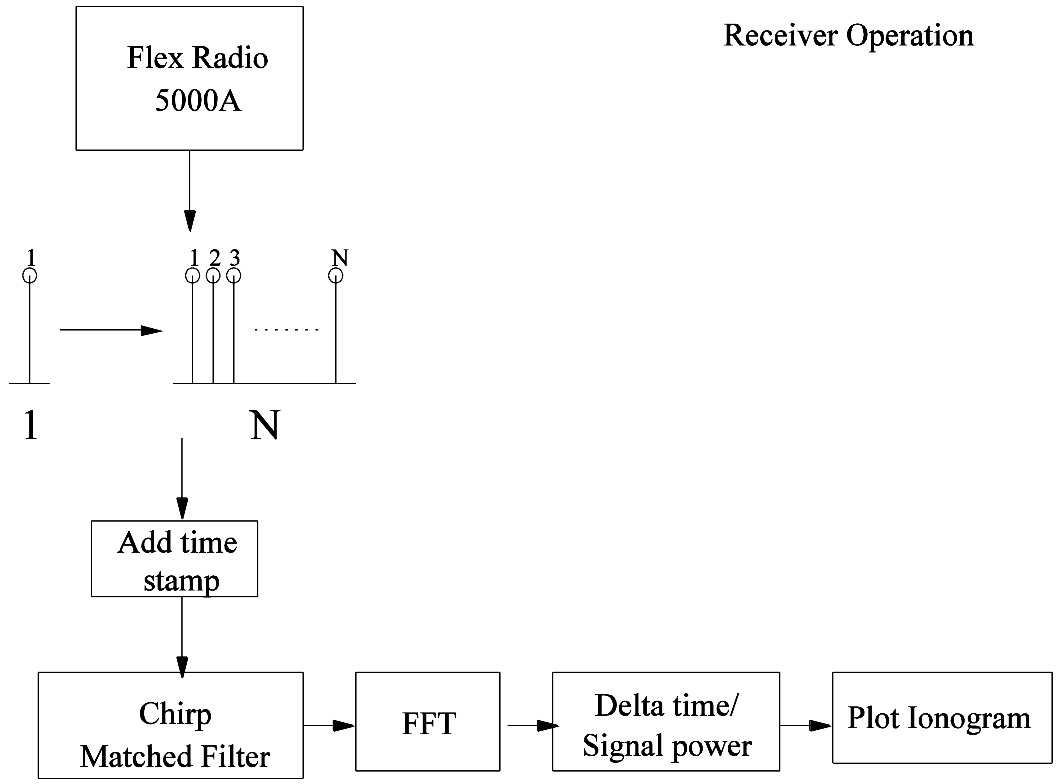

Figure 6. Receiver Operation.

4.3 TESTING THE CHIRPSOUNDER

The chirpsounder design is made testable by including the ionospheric channel models. To test the chirpsounder under indoor conditions, the baseband samples at the transmitter are applied to the Watterson ionospheric channel model [3] [14]. The Watterson model represents the simulated Doppler and multipath spreads. The main assumption in the Watterson model is that the HF channel is non-stationary in both frequency and time for bandwidths less than 10 kHz and for time less than 10 minutes. The output power of the transmitter is kept less than 500mW. The final version of this paper will include plots evaluating the performance of the overall chirpsounder design.

5 CONCLUSIONS AND FUTURE WORK

In this paper, the design of an ionospheric chirpsounder based on a software defined radio was discussed and illustrated. The F5k radio is used to implement the chirpsounder because of its very rigid front end which strictly adheres to the regulations in using the frequency channels between 2.0MHz to 30.0MHz. A modular architecture is illustrated to implement the chirpsounder with GPS interface to synchronize the transmitter and the receiver. The design of the chirpsounder transmitter and receiver architecture was also briefly discussed.

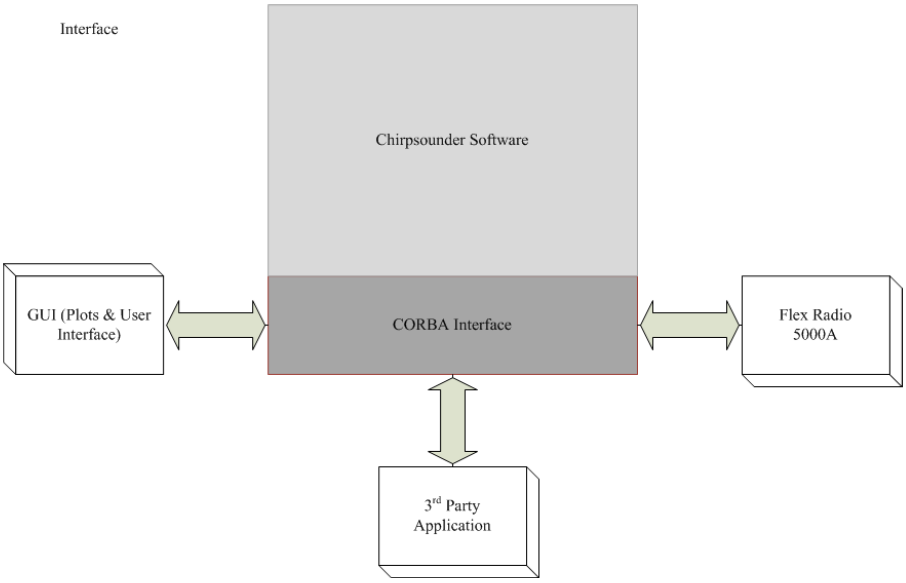

Figure 7. Interface Diagram.

In the future, we plan to expose the chirpsounder functions using the Common Object Request Broker Architecture [10] (CORBA) interface as shown in Fig. 7. This adds to the capabilities of the chirpsounder to be extended into an automatic link establishment (ALE) [12], automated HF communication, cognitive networking [13] in high frequency communication.

REFERENCES

[1] Flex-Radio, *Software Defined Radio*, http://www.flex-radio.org.

[2] J. Mitola (Ed), *Software Defined Radio*, IEEE Communications Magazine, May 1995

[3] C. C. Watterson, J. R. Juroshek, and W. D. Bensema, *Experimental confirmation of an HF channel model*, IEEE Trans. Commun. Technol., vol. COM-18, no. 6, Dec. 1970

[4] Paul C.Arthur and Paul S.Cannon, *ROSE:a high performance oblique ionosonde providing new opportunities for ionospheric research*, Annali di Geofisica, Vol.37, May. 1994

[5] Digisonde, *UMASS, digisonde*, http://ulcar.uml.edu/digisonde_dps.html.

[6] Marc Ettus, *Building Software Defined Radios: the USRP product family*, Product Brochure, July. 2009

[7] Gerald Youngblood, *A Software-Defined Radio for the Masses, part 1*, QEX, July. 2002

[8] Gerald Youngblood, *A Software-Defined Radio for the Masses, part 2*, QEX, September. 2002

[9] Gerald Youngblood, *A Software-Defined Radio for the Masses, part 3*, QEX, November. 2002

[10] CORBA, *Common Object Request Broker Architecture*, http://www.corba.org/

[11] G.H.Barry and R.B.Fenwick, *Extraterrestrial and ionospheric sounding with synthesized frequency sweeps*, Hewlett-Packard J., vol.16, July 1965, pp. 8-12

[12] Johnson, E.E, *et al.*., *Advanced High-Frequency Radio Communications*, Artech House, Boston, 1997.

[13] E. Koski, W. N. Furman, *Applying Cognitive Radio Concepts to HF Communications,* IET IRST Conference, Edinburgh, U.K. 2009.

[14] ITU-R Rec. F.1487. *Testing of HF Modems with Bandwidths of up to About 12 kHz Using Ionospheric Channel Simulators*, International Telecommunication Union, Radiocommunication Sector. Geneva, 2000.

[15] Department of defense interface standard, *Interoperability and performance standards for medium and high frequency radio systems.*

[16] Digital Interface Communication Engine,*TC technologies*, http://www.tctechnologies.tc/