

# Design and Development of a Software Defined Radio for the Ecuadorian Army, through the use of a USRP and Simulink tool of Matlab

Hugo A. Angulo and Capt. Manolo Paredes

**Abstract**— This research presents a design of software defined radio using the USRP and Simulink software of Matlab. Based on voice digitization, it was considered necessary to implement the codec G.726 of ITU, in 32Kbps version, to perform the respective comparison with 64 Kbps G.711 codec, which was developed as a Simulink's blockset. To provide sturdiness to the system, it was used an appropriate channel coding. This was achieved through the use of block codes and interlaced or Trellis, such as Convolutional codes.

Finally, both systems are proposed with QPSK modulation and frequency modulation, with the hope of providing a significant contribution and comparative development to SDR systems, through the assessment by the BER.

**Index Terms**— USRP, G.726, QPSK, SDR, BER.

## I. INTRODUCTION

Due to advancements in telecommunication, in order to provide users with higher quality of service and increasing reliability, digital communication systems have emerged, where signal processing is simplified by binary digits field. By providing the communication service through a digital system, some disadvantages arise, such as the infrastructure size, mainly based on the device's dimensions used to give effect to that purpose, the communication.



Fig. 1. Universal Software Radio Peripheral.

As a result, in 1992, the Department of Defense of the United States (DoD)[1], decided to start a development project of Software Defined Radio, which had its inception in 1970, both in Europe and in the mentioned country. The project was named "SpeakEasy" and its initial goal was to operate in 2 to 2000 MHz band. Around the middle of 1995,

Hugo A. Angulo and Capt. Manolo Paredes, Electronics and Telecommunications Engineering, Army Polytechnic School, ESPE, Sangolquí, Ecuador.

the project succeeded by achieving all set goals, which was a strong argument for the second stage development, that began in the same year and ended in 2000. The second stage was to give it a complete radio system's functionality, that complements the first stage, so that had a modem functionality[2].

Software Defined Radio is a relatively new field in both, the industrial and wireless. For this reason in 2000, several radio hams began to generate some development projects focused on this latest technology. One of them was SDR-1000, which was released by FlexRadioSystems in 2003. In 2001, a team led by Eric Blossom, called GNU Radio[3], began a tool development for signals processing [4]. Shortly after, Matt Ettus developed a card capable of providing front-end radio section. Thus the so-called USRP [3], [5], [6] was born, which is detailed in Fig. 1.

Aside from the front-end device used in communication systems, there is also a very important section, which is an initial step used for signal processing, audio in this case, and is the source coding stage, which is responsible for digital encoding signals or information transmitted by a tone.

As for audio signal, and in order to optimize bandwidth use, it must be compressed, this was achieved using a codec like G.711 [7] or G.726 [8], ITU's recommendations. The next stage is a signal encoded processing; this information must be encoded by a particular algorithm in order to reduce errors generated by introduced channel noise, as well as to amend these when they appear. This process is achieved by Convolutional, encoding and decoding, using their respective Viterbi algorithm, for instance. Another way to perform a similar task, but with some constraints, is by using Hamming code, which only detects errors, while the first one has the ability to, not only detect, but also correct them.

The step before sending signals to the channel is modulation of information, for this, different modulation techniques are used in baseband as: phase modulation, quadrature, M-PSK, it also can be done using an analog modulation as frequency modulation or FM.

Consequently, decoding process is run using the same algorithms, in the right order to obtain a signal approximation emitted by source and get proper performance of the system.

## II. DESIGN AND DEVELOPMENT OF SDR

### A. Overview

A communications system has several essential and fundamental processes to assure information reaches the receiver timely and effectively. For the present design have been analysed and selected the best processes according to their performance under conditions the radio equipments will operate, this implies, for instance, to consider frequency band. This, in turn, involves existence of wavelengths in meters, this is HF and VHF, with low frequencies that allow a wide coverage area. Channels, in the same way, suffer interference and noise insertion. As a result, the equipment must be capable to operate in adverse weather conditions on land that's orography is complicated, and other factors that can be added according to location and operating conditions[9].

On this basis, it is considered that the radio equipment will be made in the way detailed in the following diagram.

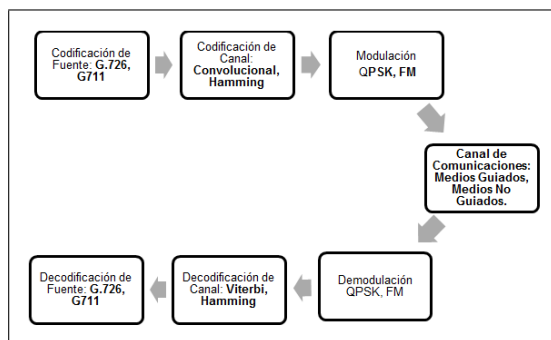


Fig. 2. Digital Communication System.

In Fig. 2, we can see a basic diagram of digital communications system. Under this premise, and considering all mentioned earlier, each communications system's block will be designed, except the channel[10]. Consequently, the system will include the following steps:

- Source Encoder - Decoder:
  - Codec: ITU Recommendations G.711 & G.726.
- Channel Encoder - Decoder:
  - Convolutional - Viterbi.
  - Hamming.
- Modulator Demodulator:
  - QPSK.
  - FM.
- Front End device:
  - USRP, that will be the device that performs link to make effective the communication process.

### B. Source Coding

1) *G.711 Codec*: ITU - G.711 [7] The Recommendation is a speech codec that uses 8 bits to encode each speech signal sample. By being one of the first developed codec to optimize digitization and subsequent voice transportation, it helped to improve developed algorithms.

This codec has three very distinct stages:

- Sampling

- Quantization

- Coding

Furthermore, quantization is an allocation process of certain discrete values, depending on values obtained from samples. For this case there are two kinds of quantizers, Uniforms and Non Uniforms. In the first one, there are assigned values evenly spaced to each of the samples. In the second one, values assignment is governed by complex algorithms or laws of encoding and compression. Within the non-uniform, are:

- A-Law, used in Europe and Latin America.

- $\mu$ -Law, used in U. S. and Japan.

2) *G.726 Codec*: The design of this codec is configured to operate only 32kbps, it has been implemented on Simulink based on the recommendations established by the ITU on its G-REC G.726 [8] published in December 1990. Together, it was taken into account Annex A published in November 1994, Annex B published in July 2003 and the final corrections published in March 2005, which are available on the ITU website. Fig. 3 and Fig. 4 correspond to the encoder and decoder of ITU-G.726 Recommendation.

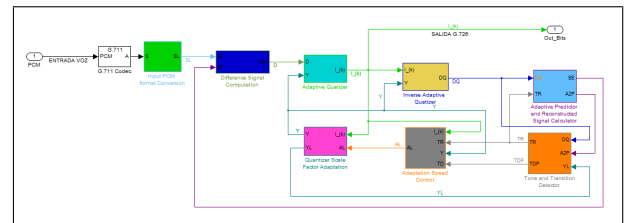


Fig. 3. G.726 Implemented Encoder Scheme.

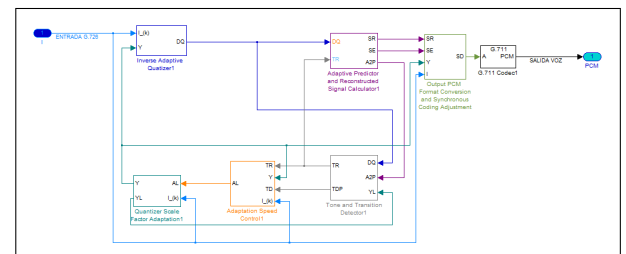


Fig. 4. G.726 Implemented Decoder Scheme.

Below, each of the blocks that were implemented within this research is detailed.

#### • Input PCM format Conversion

This block is responsible for transforming encoded A law signal into uniform PCM format, that means, it partially rebuilds a signal that was encoded using G.711 encoder, with the purpose that following stages can predict future values of signal[7], [8].

#### • Difference Signal Computation

This block is responsible to calculate the difference between the uniform PCM signal  $s_l(k)$ , from the previous block, and estimated signal  $s_e(k)$  that emanates from adaptive predictor. In summary, this

block acts as a differentiator[8].

- *Adaptive Quantizer*

A quantifier is an element that is responsible for allocating certain discrete values, depending on the range of input signal. Thus, this block is a non-uniform quantizer 31, 15, 7 or 4 levels that are used to quantify the signal  $d(k)$ . The indicated levels, operate at 40, 32, 24 and 16 kbps, respectively[8].

It is important to stress this project only took into consideration non-uniform quantizer of 15 levels, that is, 32 Kbps.

To achieve the goal of making a non-uniform quantization, this block converts a signal  $d(k)$  into a logarithmic base of 2. This value is limited by scale factor  $y(k)$ .

- *Inverse Adaptive Quantizer*

In contrast to the previous block, it is responsible for reproducing a quantized version of signal as a function of the scale factor  $y(k)$ . This signal switches to logarithmic domain[8].

- *Quantizer Scale Factor Adaptation*

To determine scale factor  $y(k)$ , this proceed under the principle of bimodal adaptation, which states that there are two adaptation speeds. A fast, for signals with prominent fluctuations, such as voice signals, and a slow, with minimum fluctuations for signals such as data signals using a tone. A combination of both speed results a more efficient adaptation[8].

This is explained as follows. Audio and voice signals have sharp changes, therefore it can separate the signal into sections, one in which the fluctuations are large, so a rapid adaptation is required and others which are undetectable in where slow adaptation is enough. Therefore, a mixture of both factors may be the ideal solution.

- *Adaptation Speed Control*

The control parameter  $a_1(k)$  varies in range between 0 and 1. It approaches zero when tone signals are used for data signals and tends to one when signals are vowels. This means, this factor give it faster or slower speeds to adapt to the system, hence its importance is essential[8].

- *Adaptive predictor and reconstructed signal calculator*

The main task developed by this block is to deliver an estimated signal  $s_e(k)$ , from a quantified signal  $d_q(k)$ . The adaptive predictor consists of two sections that respond effectively to input signals diversity. The first section is a sixth order zeros bank and the second one, a second order poles bank. With this structure, this block aims to have all possibilities, so they can meet the demands raised by various kinds of signals[8].

These blocks were merged, Fig. 3 and Fig. 4, as required by the recommendations. However, this block

serves two purposes, first, as already mentioned, to predict a signal, and the second is to rebuild, that is not enough to predict data, but must also be reconstructed according to their previous states.

- *Tone and Transition Detector*

This block is useful in a tone signal used to transmit information, not necessarily voice. It is helpful, to detect the signal's tone stability, it also helps to slow down adaptive speed control block, and consequently the system could reach equilibrium. Opposite occurs with voice signals.

Based on issues raised in the aforementioned recommendations, the implemented design is shown in Fig. 3 and Fig. 4, which use Simulink tools such as embedded block or embedded code, that covers all the same functions and calculations contained in operating algorithm of the code.

On the decoder side, as shown in Fig. 4, the only difference is that besides all blocks mentioned above, it adds one more, the Synchronous Coding Adjustment. For that reason, only that block will be mentioned, since the others were already described.

- *Synchronous Coding Adjustment*

This block aims to eliminate cumulative distortion produced by cascaded synchronous encodings, that is switched from PCM to ADPCM and PCM again. Consequently, eliminating the chance of hearing errors in the signal at the output of the decoder.

### III. PROOFS AND EXPERIMENTAL RESULTS

It is important to state that in this research, eight communications system models with different variations were developed, as shown in Fig. 2. However, all those models in which Hamming encoder as a channel coder was included, none of their results were acceptable, as indicated below in TABLE II. Next, in Table I, will show those implemented systems with their respective characteristics:

TABLE I  
TABLE OF PROPOSED MODEL FEATURES

	Source Encoder	Channel Encoder	Modulation
<b>Model I</b>	G.711	Convolutional	QPSK
<b>Model II</b>	G.711	Hamming	QPSK
<b>Model III</b>	G.726	Convolutional	QPSK
<b>Model IV</b>	G.726	Hamming	QPSK
<b>Model V</b>	G.711	Convolutional	FM
<b>Model VI</b>	G.711	Hamming	FM
<b>Model VII</b>	G.726	Convolutional	FM
<b>Model VIII</b>	G.726	Hamming	FM

In Fig. 5 and Fig. 6 are displayed best-performing models, in audio signal reception.

As for the simulations using the BERtool of Matlab, the following results were obtained with models that only used base-band modulation, that is, those which used QPSK modulation.

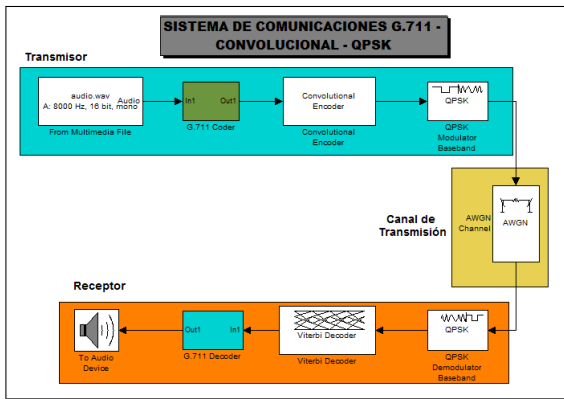


Fig. 5. Proposed Communications System Model I.

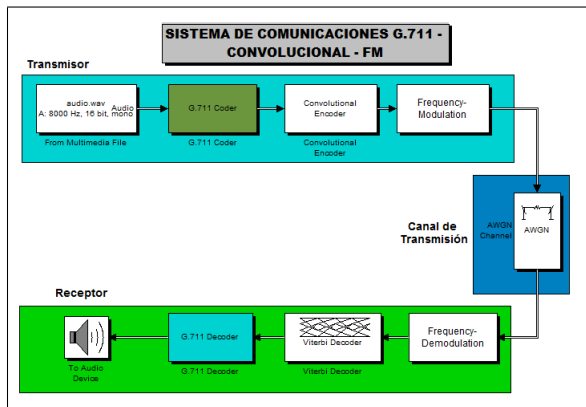


Fig. 6. Proposed Communications System Model V.

In addition, and because of research features, measurements made by use of the Agilent N1996A-506 Spectrum Analyzer are included.

Next, a comparative table of implemented systems and measurements are presented. The assessment of voice tangibility is based on the voice sharpness, considering a scale: Good, Bad and Noise. This according to whether: the signal is recognized without problems, the signal is recognized with difficulties or just noise is heard.

TABLE II  
TABLE OF PROPOSED MODEL FEATURES

	Voice Tangibility	Output Power [dBm]	Transmission Rate [KSps]
Model I	Good	-8	64
Model II	Bad	-10	56
Model III	Good	-10	56
Model IV	Bad	-9	28
Model V	Good	-7	16
Model VI	Noise	-10	14
Model VII	Good	-10	8
Model VIII	Noise	-10	7

Finally, looking at both Table II and Fig. 8 and Fig. 9, power measurements made by Agilent N1996A-506 Spectrum Analyzer, it is concluded that the best performance systems were model I and model V.

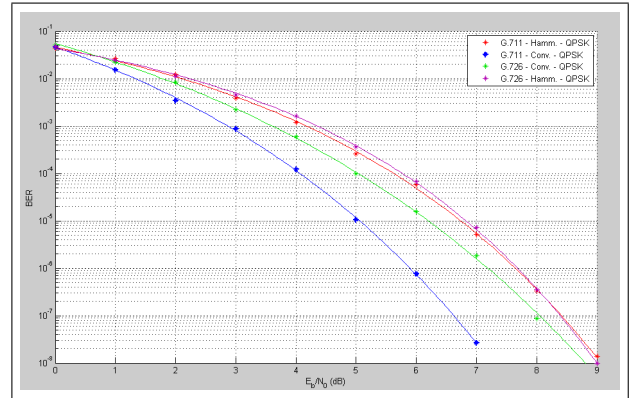


Fig. 7. Comparative BER of designed systems.

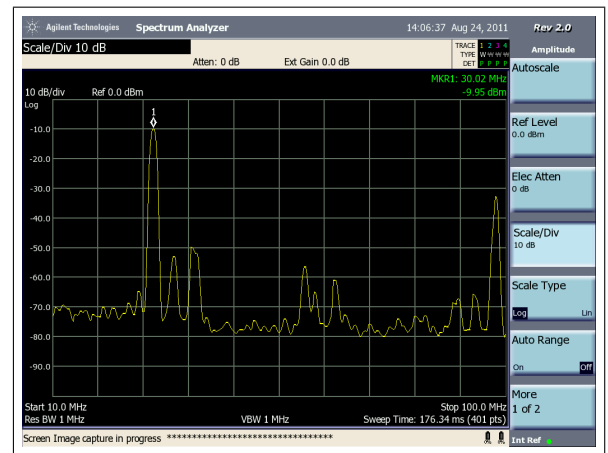


Fig. 8. Transmission Power of the Model I.

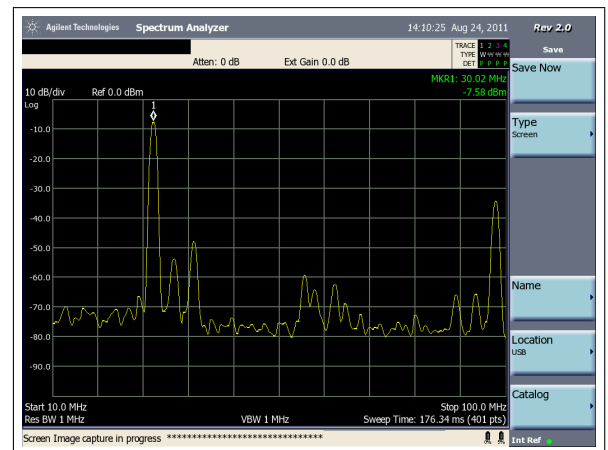


Fig. 9. Transmission Power of the Model V.

#### IV. CONCLUSION

It is important to mention that G.726 codec, implemented within this research, allowed that the process of voice coding and subsequent reconstruction, has better quality compared to the process performed by G.711 codec. According to this, it was possible to transmit the signal through space and obtain, at reception, an acceptable voice quality. Therefore, it is concluded that due to the performance provided by G.726 codec, it is more efficient than G.711 codec.

The goal of channel coding is to detect and correct errors, thereby reducing the error probability, to ensure information that reaches the receiver is similar to that was sent. According to the above, it was found in a practical way that a signal, even with noise induced by the channel, or due to processing carried out by USRP boards, in this case, can be transmitted and the decoder corrects some of these errors, while a signal that has no channel coding and during process, adopt errors, reception is not able to improve or correct these errors and the result will be a damaged signal or completely noisy.

BasicTX daughterboard card, although its specification details that the operating band range is from 1 to 250MHz, presents problems in practice when trying to operate across that band and it was found that actual working range is from 1 to 44MHz, which is the instance in which this card can properly propagate signals through space.

The USRP v.1.0 does not have the ability to guarantee good communication using any base-band modulation, because this, requires phase reaches the receiver accurately, so this can decode the information. As it can be interpreted, cards do not have enough capacity to keep the signal phase, since due to digital analog conversion and conversely, makes this vary sharply in alluded process that is generated by the FPGA together with USRP's converters. Hence, it is concluded that cards do not work well in digital modulation processes.

In regard to voice compression, initial coding should be taken into consideration, that is the transition between the analog section of the voice to digital section, it is run by the microphone Simulink's module, it uses 16 bits per sample to carry out the digitalization. To this must be added that, due to implementation of the 15 levels ITU G.726 codec, it was possible to compress each of these samples to 4 bits, representing a 75% bandwidth usage optimization.

Finally, it is recommended to use a signal filtering stage, before receiving it, since as can be seen in Fig. 8 and Fig. 9, the harmonics generate noise on information that is to be decoded.

#### REFERENCES

- [1] W. Turtlebee, *Software Defined Radio: Origins, Drivers and International Perspectives*, 1st ed. Chichester, England: John Wiley & Sons, 2002.
- [2] J. Bard and V. J. Kovarik, *Software Defined Radio: The Software Communications Architecture*, 1st ed. Chichester, England: John Wiley & Sons, 2007.
- [3] GNU Radio. [Online]. Available: <http://gnuradio.org>
- [4] P. Burns, *Software Defined Radio for 3G*, 1st ed. Norwood, U. S.: Artech House, 2003.

- [5] F. A. Hamza, *The USRP under 1.5X Magnifying Lens! GNU Radio*, 1st ed., Journal of GNU Radio, June 2008. [Online]. Available: [http://gnuradio.org/redmine/attachments/129/USRP\\_Documentation.pdf](http://gnuradio.org/redmine/attachments/129/USRP_Documentation.pdf)
- [6] M. Ettus, *USRP User's and Developer's Guide*, Journal of Ettus Research LLC, 2007. [Online]. Available: [http://www.olifantasia.com/gnuradio/usrp/files/usrp\\_guide.pdf](http://www.olifantasia.com/gnuradio/usrp/files/usrp_guide.pdf)
- [7] *Pulse Code Modulation (PCM) of Voice Frequencies*, ITU Std. ITU-T Recommendation G.711, 1988.
- [8] *40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (APCM)*, ITU Std. ITU-T Recommendation G.726, 1990.
- [9] R. Blake, *Electronic Communication Systems*, 2nd ed. Publishing House of Electornics Industry, 2002.
- [10] J. Proakis, *Digital Communications*, 4th ed. New York, U. S.: McGraw-Hill, 2008.