Proceedings of WInnComm 2016, Copyright © 2016 Wireless Innovation Forum All Rights Reserved

AN OPEN AND FREE ISDB-T FULL_SEG RECEIVER IMPLEMENTED IN GNU RADIO

Federico Larroca (flarroca@fing.edu.uy)[1], Pablo Flores Guridi (pablof@fing.edu.uy)[1], Gabriel Gómez Sena (ggomez@fing.edu.uy)[1], Víctor González-Barbone (vagonbar@fing.edu.uy)[1], and Pablo Belzarena (belza@fing.edu.uy)[1]

[1]affiliation: Facultad de Ingeniería, Universidad de la República, Uruguay

ABSTRACT

Almost every country in Latin America has adopted the ISDB-T standard for free-to-air television broadcasting. The so-called "analogical blackout" is about to be performed, so broadcast engineers and technicians have to be prepared for such a challenging task. Key to the success of this blackout is a deep understanding of the chosen technology. In this paper we present the first open, free and fully software-based ISDB-T receiver, entirely implemented in GNU Radio. Moreover, all our blocks produce several relevant measurements. This implementation allows broadcasting professionals and researchers to get in touch with a real-time working receiver, avoiding the need of costly Digital Television equipment. Moreover, the block-based architecture of GNU Radio offers the possibility of replacing particular blocks or simulate channel conditions, in order to test different algorithms and implementations. As a toy example of the possibilities brought by our framework, we compare two OFDM synchronization methods in terms of the resulting BER in a typical scenario.

1. INTRODUCTION

Several countries around the world are undertaking the so-called Digital Television (DTV) Transition. That is to say, the replacement of the over fifty years old analog television transmission scheme used by broadcasters, in favor of a digital counterpart. Several reasons may be cited to justify this immensely challenging task. For instance, digital transmissions are much more effective in terms of bandwidth usage (in addition to potentially obtaining higher image and audio quality, several programs may be transmitted simultaneously by means of digital multiplexing in the same channel) and do not require significant guard bands (even adjacent channels may be used in the same region without interference).

However, arguably the most important reason is the Digital Dividend resulting from this transition. That is to say, given the efficient usage of the spectrum mentioned above, important contiguous sections of the spectrum will be freed once the transition is complete. These sections may in turn be auctioned, typically for broadband mobile services.

As it usually happens in these situations, different proposals were made and more than one standard is currently in use for DTV: the American ATSC (Advanced Television Systems Committee) [1], the Chinese DTMB (Digital Terrestrial Multimedia Broadcast) [2], the European DVB (Digital Video Broadcasting) [3] and the Japanese ISDB (Integrated Services Digital Broadcasting) [4]. Of special interest to us is ISDB. When Brazil was evaluating which DTV system to adopt, they decided to take the opportunity and perform a Technology Appropriation. To this end, and based on the ISDB standard, they developed the SBTVD (Sistema Brasileiro de Televisão Digital-Terrestre) [5]. The ensuing improvements over ISDB were later incorporated into the original standard, resulting in the so-called "ISDB-T International" (or ISDB-Tb). The latter was then adopted by most South American countries, in addition to some Central American and Asian ones.

All of these countries are performing the DTV Transition. A first key factor to its success is a deep understanding of the chosen technology. However, no true expertise can be achieved without dealing with actual implementations. Sadly, most South American countries do not have neither analog nor digital TV receiver or transmitter industries, so these devices have to be bought to technology suppliers outside the region. A second important factor in this transition is the ability to measure the DTV signal, both at different points of the receiving chain as well as several locations in the territory. It is imperative for the wide adoption of DTV that the service is (at least) as good as the current analog one. However, measurement equipment specially tailored for DTV are costly (typically in the order of several thousands of dollars), thus limiting the breadth of measurement campaigns.

Software Defined Radio (SDR) is an excellent alternative for, at least, making cheap and flexible measurement equipment and getting to know how things really work. In particular, we have focused on PC-based SDR equipment: USRP (Universal Software Radio Peripherical, from Ettus Research) [6], BladeRF (from Nuand) [7] and HackRF (from Great Scott Gadgets) [8]. In addition to being open to a certain degree (the drivers used by the PC, the code of the FPGA/CPLD in them and even the hardware schematics are open and freely available), they meet the requirements imposed by DTV in terms of sampling rate, and are relatively inexpensive (some hundred of dollars).

Regarding the software that runs on the PC, the most prominent development kit (and by far the most popular) is GNU Radio [9], which in addition to being completely free and open,

Proceedings of WInnComm 2016, Copyright © 2016 Wireless Innovation Forum All Rights Reserved

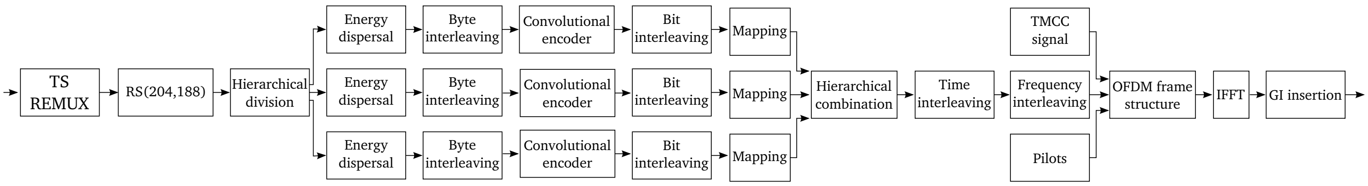

Figure 1: ISDB-T transmission system block diagram.

it also supports all the hardware mentioned above. GNU Radio basically provides a framework where the different blocks that compose a transceiver may be implemented and interconnected with relative ease. Moreover, it includes a growing base of already implemented blocks, ranging from multipliers to demodulators.

This article presents `gr-isdbt`: the first open, free and fully software-based ISDB-T full_seg receiver (https://github.com/git-artes/gr-isdbt), entirely implemented in GNU Radio. The block-based architecture of this framework allows broadcasting professionals and researchers to get in touch with a real-time working receiver. Moreover, it also offers the possibility to replace particular blocks or simulate channel conditions, in order to test different algorithms and implementations.

All blocks in `gr-isdbt` have been designed and implemented with the ability to output the corresponding and relevant measurements. For instance, and just to name two examples, the OFDM synchronization block outputs the estimated carrier frequency offset, and the Viterbi decoder the estimated Bit-Error Rate (BER). Moreover, we have implemented a set of blocks which measure several interesting indicators regarding the resulting constellation points (e.g. Modulation Error Ratio, MER).[1] It is important to highlight that the receiver outputs the corresponding transport stream, which can be fed into any typical video player (e.g. MPlayer or ffmpeg). This possibility further allows our receiver to be used to test the channel effects on the actual decoded audio/video.

The rest of the paper is structured as follows. After a brief overview of ISDB-T in the next section, Sec. 3 presents in more detail GNU Radio, particularly those components we used for our implementation. It also presents relevant past efforts in SDR and DTV. Sections 4 and 5 then discuss `gr-isdbt` with some detail, along with the blocks corresponding to the constellation measurements. As a toy example of the possiblities brought by our framework, we compare two OFDM synchronization methods in terms of the resulting BER in Sec. 6. Naturally, there are several aspects of the receiver which are not complete or may be improved. A discussion on them is presented in Sec. 7, which concludes the article.

[1] Please visit https://github.com/git-artes/gr-mer.

2. THE ISDB-T STANDARD

ISDB-T stands for Integrated Services Digital Broadcasting - Terrestrial, and is the Japanese digital television standard. It is based on DVB-T, and was ratified in the early 2000's, after the European and the American standards were adopted. This delay allowed the designers to take into account the experience gained with the previous digital television systems, resulting in the most complex, but also the most robust and versatile standard (although this is obviously the subject of much debate). As we mentioned in the last section, it was later adopted and adapted by Brazil: most importantly a new interactivity middleware named *Ginga* was defined (instead of the original *BML*) and MPEG-4 replaced MPEG-2 for source coding. The new version of the standard was named ISDB-Tb, and was later adopted by most South American countries.

Figure 1 shows the entire block diagram of the ISDB-T transmission system. Hereinafter we briefly discuss the specifications of each of the blocks, paying special attention to those aspects particular to ISDB-T. For more details, the interested reader should consult [4].

Let us begin by the **modulation scheme**. Orthogonal Frequency Division Multiplexing (OFDM) is used over a 6 MHz bandwidth channel. After the OFDM modulation, a cyclic prefix (CP) is added which length is expressed as a fraction of the active symbol's length, T_s. There are four possible values: $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ or $\frac{1}{32}$. This CP is a copy of the last part of the OFDM symbol which is prepended to it. As we will discuss later, it will be used in reception for symbol alignment and coarse frequency correction. Moreover, over multipath propagation channels, this prepending will eliminate inter-symbol interference and ease equalization (see for instance [10, Sec. 12.4]).

Regarding the number of carriers in one OFDM symbol, it can be either 2^{11}, 2^{12} or 2^{13} (fixed at a power of 2 so as to be able to use the FFT algorithm). However, the sampling rate (termed f_{FFT} in the standard) is always equal to $512/63 \approx 8.126$ MHz. This means that keeping the total data rate constant, the operator may choose to use more carriers but slower symbols in order to immunize the radio signal from multipath propagation effects, or less carriers but faster symbols in order to immunize the signal from Doppler effect. This choice is termed **Transmission Mode**, and may be either 1, 2 or 3, although generally mode 3 is used.

Focusing on mode 3 from now on, not all the 8192 carriers are used, but rather 5617, enough to meet the bit-rate and band-

Proceedings of WInnComm 2016, Copyright © 2016 Wireless Innovation Forum All Rights Reserved

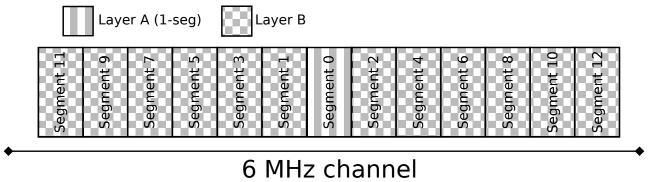

Figure 2: ISDB-T segmented spectrum and the 1-seg layered configuration.

width requirements (where a guard interval and zero-padding is used in the rest of the carriers). This useful spectrum is in turn sub-divided into 13 sub-bands named **segments**, of 432 carriers each. These 13 segments may be used independently from one another, a feature first implemented in ISDB-T and called *Band-Segmented Transmission OFDM* (BST-OFDM). In particular, in this case, they can be combined in up to three so-called *hierarchical layers* (A, B and C), which transmit different Transport Streams. Moreover, these groups of segments can be configured to use different forward error correction (FEC) rates, time interleaving lengths and modulation schemes.

Although up to three different Transport Streams may be thus transmitted in the same channel, the typical configuration uses one segment with very robust transmission parameters (which for instance allows visualization by mobile users), while the rest is used with a configuration resulting in a high data-rate (with HDTV generally in mind). Thus, handheld receivers (such as cellphones, a market which particularly interested the ISDB-T designers) should only tune and demodulate this single segment, making it possible to work with lower sample and data rates, and thus, less CPU requirements. This feature is known as **1-segment** (or 1-seg for short). Receivers capable of tuning and demodulating all segments are known as *full_seg*. Figure 2 illustrates this segment-based spectrum division (and the numbering used by the standard) and the typical 1-seg configuration.

From the 5617 active carriers, there are several which are used as **pilots** to assist the receiver in the equalization process. These so-called scattered pilots (SP) change position from symbol to symbol to avoid pathological situations (such as a permanent deep fade in the spectrum), although their payload is known (which in turn depends on their position). A particularity of ISDB-T is the virtual absence of continual pilots (there is a single CP in the biggest carrier). As we will further discuss later in the article, standard frequency correction algorithms that rely on this kind of pilots are thus not adequate in this case.

In addition to these pilots, several carriers are devoted to transmitting the modulation parameters at use. There are a total of 204 bits to be transmitted: the so-called **TMCC** (Transmission Multiplexing Configuration Control). They occupy fixed carriers in the OFDM symbol, each of them corresponding to the same TMCC's bit, and DBPSK modulation scheme is used. Although they naturally change from symbol to symbol, we will use these carriers with identical information to perform part of the necessary frequency correction as we discuss later. In any case, 204 OFDM symbols are thus required in order to receive the complete TMCC, completing a so-called **OFDM frame**.

Table 1: ISDB-T transmission system parameters available.

Parameters	Values
Total Bandwidth	6 MHz
Number of segments	13
Segments bandwidth	6000/14 $\approx 428.57\ kHz$
Number of active carriers	1405 (mode 1) 2809 (mode 2) 5617 (mode 3)
Active symbol duration	252 μs (mode 1) 504 μs (mode 2) 1004 μs (mode 3)
Guard interval duration	1/4, 1/8, 1/16, 1/32 (of active symbol duration)
Convolutional Code Rate	1/2, 2/3, 3/4, 5/6, 7/8
Time interleaving depth	0, 1, 2, 4 (mode 1) 0, 2, 4, 8 (mode 2) 0, 4, 8, 16 (mode 3)
Modulation schemes	DQPSK, QPSK, 16QAM, 64 QAM

The rest of the blocks are somewhat standard: frequency and a configurable time interleaving of the complex symbols; mapping, bit interleaving and convolutional encoder are applied to bits; byte interleaver, energy dispersal and Reed-Solomon encoding may be regarded as applied to bytes. It is important to highlight that most of these algorithms are applied separately to each layer (thus the three parallel paths in Fig. 1), and each layer may use its own set of parameters.

Table 1 summarizes what was described in this section and adds some other extra information. The parameters used in each layer for the convolutional encoding, time interleaving and modulation scheme for any particular transmission are all specified in the TMCC.

3. GNU RADIO

GNU Radio [9] is surely the most widespread free and open-source software development toolkit that provides signal processing blocks to implement software radios.

GNU Radio is designed to support signal processing on continuous data streams from a source to a sink passing through different signal processing blocks. The stream is a flow of basic types like bytes, integers or complexes. Each GNU Radio block defines input and output signatures which specify the number of input and output streams and their type.

Using the GNU Radio Companion [11] tool, a signal processing application can be graphically created by combining blocks with a simple drag-and-drop user interface.

A software radio system can be constructed by combining some of the general purpose blocks already provided by GNU Radio and its contributors. Common elements typically found in radio systems are available, for instance: filters, channel codes, synchronization elements, equalizers, demodulators, coders, decoders. Moreover, GNU Radio provides a relatively easy way to extend the functionality by writing the specific blocks needed for a particular implementation.

Our implementation also uses the stream tag mechanism provided by GNU Radio which allows to add additional information to a particular sample of a flow. We make use of this feature for instance to signal other blocks the exact point in the stream where one block detects the beginning or ending of a frame. Also after detecting a missing OFDM symbol, a stream tag is used to signal other blocks about the need of a resynchronization process.

As a real time system, our ISDB-T full_seg receiver make extensive use of the *Vector-Optimized Library of Kernels* (VOLK) [12] which improves the performance for some frequently data vector operations required. The VOLK library provides a useful architecture-independent programming tool to enable vectorized mathematical operations using the SIMD (Single instruction, multiple data) capabilities provided by moderns CPUs. The exploit of this data level parallelism improves dramatically the speed of some common operations like the entry-wise multiplication of two arrays.

3.1. Related work

There have been some implementation efforts in the past related to DTV and SDR. In what concerns ISDB-T in particular, and to the best of our knowledge, the only software receiver for ISDB-T was presented in [13]. However, it was not based on GNU Radio, and most importantly, was not released to the public. Moreover, the project seems abandoned, and our attempts to contact the authors have failed. Recently, a video [14] showcasing a ISDB-T modulator developed with GNU Radio and USRP was published. The implementation details are not known because the work has not been shared (nor published) yet. If available, such transmitter would be a very interesting complement to our project.

We have thus turned our attention to DVB-T, which is relatively similar to ISDB-T. The first work in this direction was [15]. However, it shares the same flaws as [13]: it was not released to the public (and was not based on GNU Radio). On the other hand, `gr-dvbt` [16] is an out-of-tree module (a GNU Radio component which is not part of the original source tree, typically developed by third-party programmers) which implements a DVB-T compliant transmitter and receiver, and which is both open and free.

Our ISDB-T receiver includes some generic blocks like filters or FFT and other blocks that had to be created or adapted. Blocks for OFDM symbol acquisition, frequency synchronization, channel estimation, TMCC decoding, frequency deinterleaver, time deinterleaver, Viterbi decoding, Reed Solomon decoding, among others auxiliary blocks are needed for a complete ISDB-T full_seg receiver. Some of them were similar to the ones used in `gr-dvbt` implementation, but others had to be written from scratch.

Please note that a preliminar version of our work (a 1-seg receiver) was presented at [17]. Several improvements were introduced since, which we present here. Notably, the receiver now works for all segments, and we have implemented a much more robust OFDM synchronization block which corrects symbol alignment, sampling and carrier frequency errors (cf. Sec. 4). Moreover, several measurement blocks were developed, effectively making `gr-isdbt` an alternative to costly DTV measurement equipment (cf. Secs. 5 and 6).

4. OFDM SYNCHRONIZATION

Now that we have introduced both GNU Radio and ISDB-T, let us discuss with some detail `gr-isdbt`. Figure 3 presents the complete receiver chain, as shown in the GNU Radio Companion.

The first step is to receive the samples from the corresponding hardware (in this particular example, a USRP). Naturally, the center frequency is dependent on the local broadcasters. On the other hand, the sampling rate is constant and defined by the standard (f_{FFT}, cf. Sec. 2).

However, not all devices are capable of sampling at arbitrary rates. For instance, the USRP model B100, which we used in our tests, is not capable of sampling at exactly f_{FFT} (although several master clocks are available). In such case, a `Rational Resampler` is necessary. Care should be exercised regarding the bandwidth of this block, so as to avoid degrading the 6 MHz signal. In particular, we sampled at 8 MS/s, and the 64/63 `Rational Resampler` block uses the default 0.4 Fractional Bandwidth (i.e. 80% of the incoming bandwidth goes undistorted). Finally, depending on the situation it may be necessary to filter-out neighboring channels. For this purpose, a `Low Pass Filter` block with a cut-off frequency $f_c = 3MHz$ may be used.

The next step, and arguably the most challenging one, is to perform OFDM synchronization. That is to say, finding the samples corresponding to each symbol, and estimating and correcting the carrier frequency offset and the sampling clock offset (and these estimates have to be robust to the channel response). As usual, this is performed in two steps: an acquisition phase (where in this case a one-shot estimate is performed) and a tracking phase (where the estimates are further refined by a feedback mechanism).

This section discusses the two mechanisms used by `gr-isdbt` to perform acquisition and tracking (implemented in the block `OFDM Synchronization`). The next section presents the rest

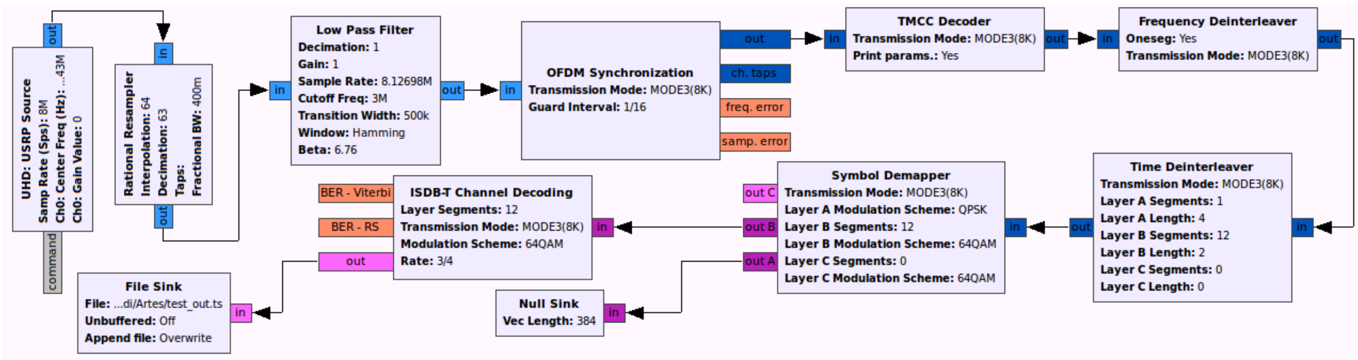

Figure 3: The complete receiving chain of `gr-isdbt`, as shown in the GNU Radio Companion. Note how the Port Labels in each block show the available inputs and outputs.

of the blocks.

4.1. OFDM Synchronization Acquisition

The objective of the acquisition phase is to obtain a complex-valued vector of length 8192 which, although not optimal, may be "usable" by the equalizer and the decision blocks. Then, and as it will be discussed in the following subsection, based on the channel estimated by the equalization block, both frequency and sampling errors are (further) corrected.

An algorithm that may be used to this end is the classic work by van de Beek et al. [18], which considers only two uncertainties: the arrival time of the OFDM symbol (θ) and the carrier frequency offset (ϵ). The basic idea is to use the fact that the last and first L transmitted samples of each symbol (with L being the length of the CP) are the same. Thus, the autocorrelation of the received samples with a lag of $N = 8192$ (and averaged over L samples) should be maximum precisely at the end of the CP. In fact, the likelihood function to estimate θ may be proved to be this autocorrelation plus a term that avoids false maxima due to particularly high samples:

$$\hat{\theta}_{ML} = \arg\max_{\theta} \left| \sum_{k=\theta}^{\theta+L-1} r[k]r^*[k+N] \right| - \rho \sum_{k=\theta}^{\theta+L-1} \frac{|r[k]|^2 + |r[k+N]|^2}{2}, \quad (1)$$

where $r[k]$ are the received samples and $\rho = \sigma_s^2/(\sigma_s^2 + \sigma_n^2) = SNR/(1+SNR)$.

Once the boundaries of the symbol are established, the only uncertainty left is the frequency offset (at least under our present context). Note that the only difference between the first and last L received samples of each symbol are generated precisely by this frequency offset (in addition to noise naturally), which manifests itself as a phase difference. This suggests the following estimate for ϵ (which actually may be proved to be the maximum likelihood estimate):

$$\hat{\epsilon}_{ML} = -\frac{1}{2\pi} \arg \left\{ \sum_{k=\hat{\theta}_{ML}}^{\hat{\theta}_{ML}+L-1} r[k]r^*[k+N] \right\}, \quad (2)$$

Please note that owing to the ambiguity in the arg function, we may arbitrary assume that $0 < \epsilon < 1$. This is thus called *fractional frequency* correction. The integer part will be corrected in a different block.

Our implementation works as follows. When synchronization has not yet been attained (e.g. the first time the flow graph is run), Eq. (1) is calculated for the whole observation period (at least $2N + L$ samples of $r[k]$), its maximum is found, ϵ is calculated through (2), samples are derotated accordingly (as we mentioned before, the performance for these operations was remarkably improved by VOLK), and samples are fed to the equalizer (discussed in the next subsection).

Once synchronization is acquired, the position of the maximum in Eq. (1) should not change significantly from symbol to symbol. In fact, and as discussed for instance in [19, Ch. 5], we actually need to find a θ such that the block outputs N samples of each symbol (i.e. avoiding inter-symbol interference). As long as this is achieved, differences with the actual boundary of the CP will manifest as a simple phase difference, easily absorbed by the equalizer. Thus, the estimation of θ is not changed during tracking.

However, synchronization may be lost (typically, due to samples being dropped by the USRP). We thus calculate (1) for a small interval around the current θ to verify synchronization. If this is not the case (e.g. the obtained maximum is too small when compared to previous values), a re-synchronization is triggered. We first look for a reasonable maximum in the complete observation period (the accumulated effect over several symbols of an uncompensated sampling error may result in the maximum exiting the small interval). If it is not found, we skip some samples and repeat the procedure.

Finally, notice that the Transmission Mode and the CP length that are being used in the transmission (i.e. N and L in the above algorithm) have to be established in advance. To do this, the argument of Eq. (1) may be calculated for all possible values of N and L. It is relatively easy to see that the correct N and L will produce a mostly constant time-series with a marked triangle (whose peak is precisely the solution to Eq. (1)), a wrong N noise, and a wrong L a plateau.

4.2. OFDM Synchronization Tracking

4.2.1 Channel estimation

Once we obtained the 8192 derotated complexes corresponding to one symbol, we calculate its FFT and thus move to the frequency domain (and we thus speak of carriers). Naturally, to this end we used the FFT implementation available in GNU Radio. The next step is to equalize the channel (which as we will see next is crucial in the tracking phase). To this end, we first need to identify the pilots included in every symbol.

The problem is that, for large carrier frequency offsets, in addition to the *fractional* carrier frequency offset ϵ we calculated before (cf. the previous subsection) there is also an *integer* carrier frequency offset Δf_I, being a multiple of the sub-carrier spacing $1/T_s$. The latter, the one we discuss next, represents an ambiguity regarding which of the 8192 carriers output by the FFT block correspond to each of the transmitted ones, as they are now shifted by the unknown Δf_I.

In other broadcast technologies such as DVB-T, the presence of continual pilots (certain fixed carriers which transmit an alternating complex symbol) makes it possible to correlate two consecutive OFDM symbols in order to solve this problem. However, in this case that would not be possible as ISDB-T has no such continual pilots. On the other hand, we may take advantage of the multiple TMCC carriers that are constantly transmitted at fixed frequency positions. Although we still do not know what they are transmitting (i.e. time interleaving parameter, convolutional code rate, modulation scheme), in a given OFDM symbol they are all transmitting the same bit.

Suppose there are M TMCC carriers per OFDM symbol ($M = 52$ for mode 3), denote as $T[i]$ ($i \in \{0, \dots, M-1\}$) their positions when there is no frequency offset (i.e. at transmission), and $Y[k]$ the output of the FFT block. Although the bits transmitted at the carriers corresponding to the TMCC are the same, the modulation used is DBPSK, where the initial value for the differential modulation depends on $T[i]$. This means that, depending on their position, certain symbols are either 4/3 or -4/3. Let us then further denote as $w(T[i])$ this initial value. It should be clear that the following correlation is maximized when $m = \Delta f_I$:

$$\Gamma[m] = \sum_{i=0}^{M-2} w(T[i])Y[T[i]+m].w(T[i+1])Y^*[T[i+1]+m] \tag{3}$$

Once we know which carrier is each, it is time for channel estimation and subsequent equalization. That is to say, if $X[i]$ is the symbol transmitted at the i-th carrier, then $Y[i] = X[i]H[i]$, where $H[i]$ is the channel gain on the corresponding frequency [10, Ch. 12]. We thus have to estimate $H[i]$ for every carrier and divide accordingly.

This task shall be performed by using the scattered pilots (SP), as they have predefined positions and values. Nevertheless, these positions vary with every new symbol cyclically. In particular, there are four different possible carrier configurations. A very similar algorithm to the one discussed before regarding continual pilots is applied to the SPs in order to detect the current arrangement. Then, $H[i]$ is calculated for the carriers corresponding to the SPs. The value for the rest is estimated by a simple linear interpolation.

4.2.2 Frequency and timing tracking

As we mentioned before, the objective of the acquisition system is to obtain a reasonably good estimate of both frequency offset and symbol alignment. However, a small residual error in frequency will still linger (which we will note by ϵ'), which if left unattended will produce Inter-Carrier Interference (ICI). The tracking system is then executed, which will in turn improve this estimate. In our particular case, it will also estimate and correct the sampling errors. That is to say, we will now assume that the received analog signal is sampled every $(1+\delta)T$ instead of the ideal T.

As discussed for instance in [20], the combination of ϵ' and δ produces, in addition to the aforementioned ICI, a time-variant phase rotation. More in particular, the complex symbol in the i-th carrier of the k-th OFDM symbol is multiplied by the phasor $\exp(j2\pi\phi_i(k(N+L)+L)/N)$, with $\phi_i = \epsilon' + \delta i$, which is assimilated to the channel gain. Thus, the phase difference between two consecutive channel gain estimates for the i-th carrier will be proportional to $\epsilon' + \delta i$.

The above result suggests a PLL-like structure as the one shown in Fig. 4, where the calculus of the error signal may be interpreted as adjusting a straight line to the points $(i, \arg\{H_k[i]H_{k-1}[i]^*\})_{i=-I,\dots,I}$ (with $I = (5617-1)/2$ for mode 3). That is to say, the k-th error signal is calculated as follows:

$$e_{\epsilon'}[k] = \arg\left\{\sum_{i=-I}^{I} H_k[i]H_{k-1}[i]^*\right\},$$

$$e_\delta[k] = \arg\left\{\left(\sum_{i=1}^{I} H_k[i]H_{k-1}[i]^*\right)\left(\sum_{i=-I}^{0} H_k[i]H_{k-1}[i]^*\right)^*\right\}.$$

The only way to implement a system with such feedback on GNU Radio is to write it as a single C++ block (in this case, `OFDM Synchronization` in Fig. 3). However, we have used several functions already present in GNU Radio's codebase, most notably the interpolator filter and FFT transforms. Moreover, the above calculations were significantly accelerated by using VOLK's functionality.

Proceedings of WInnComm 2016, Copyright © 2016 Wireless Innovation Forum All Rights Reserved

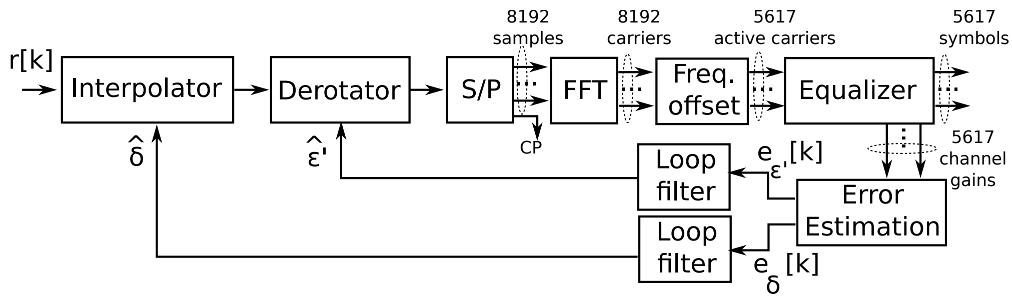

Figure 4: A functional diagram of the OFDM synchronization tracking subsystem. Note that the block that obtains the θ used by the Serial-to-Parallel (S/P) is not shown, as it is part of the acquisition subsystem.

5. FURTHER BLOCKS

5.1. TMCC, demapping, interleaving and decoding

At this point, once we have correctly synchronized and equalized the signal, it is time to decode the TMCC in order to set the appropriate values for channel decoding and symbol demapping blocks. We proceed as follows. For any given OFDM symbol, we demodulate the bit on every TMCC carrier, and perform a majority vote to take the final decision of the corresponding bit.

Moreover, in order to decide whether a complete TMCC was received, we can make use of the TMCC synchronization signal which consists of a 16-bit word that takes either the form 0011010111101110 or 1100101000010001 depending whether it is an odd or even frame. A buffer with the last 204 bits is constantly filled and bits 1 to 16 are compared to the synchronization words presented above. When matched, the BCH parity code syndrome is computed and if no errors were found, a new frame start is triggered and the TMCC can be read. This is signaled downstream by a tag, since other blocks make use of this information (most notably the energy dispersal, which should be reset with every new OFDM frame).

Finally, the `TMCC decoder` block is also responsible for filtering all the control pilots and letting out only the data carriers reordered increasingly by segments number.

The rest of the blocks are relatively standard, so their implementation did not cause major problems. In Fig. 3, after the `TMCC Decoder` block discussed above, frequency and time deinterleaving are performed and then symbol demapping. Here, each layer is separated and a stream is generated for each of them.

Bit deinterleaving is then applied. After the Viterbi decoder there is a byte deinterleaving and an energy descrambler. This last stage may be used to test the correct reception of the signal. As specified in [4, Sec. 3.5], the byte just before the beginning of a new OFDM frame (signaled by the tag we mentioned before) should be the synchronization byte used by the Transport Stream (i.e. 0x47). After that, the Reed Solomon decoder, which is able to correct up to eight corrupted bytes, removes the last 16 bytes of redundancy from every Transport Stream package.

Please note that Fig. 3 shows `ISDB-T Channel Decoding`, a hierarchical block which includes all of the blocks mentioned above. Finally, the stream is saved on the hard drive. Both the Viterbi and Reed-Solomon decoder were taken from [16] as they are identical to those used by DVB-T.

5.2. Quality measurements in DTV systems

As we mentioned on the introduction, one of the objectives of our system was to provide the possibility to measure several important quality indicators. We now briefly discuss this aspect. The goal of a measurement system on a DTV receiver is firstly the evaluation of the reception quality, and secondly characterizing the problems that degrade the reception quality. If possible, another goal is to identify the source of these problems. Frequently, detecting the type of problem is possible, but its source is more challenging.

The main impairments that degrade the reception quality have three different sources: the transmitter, the channel and the digital receiver itself. These are:

- Transmitter-side: power (attenuation, low SNR), IQ amplitude imbalance (attenuation and distortion of the constellation), IQ phase imbalance (Non Orthogonal constellation), and residual carrier.
- In the channel: Gaussian noise (SNR), interference or noise bursts, attenuation (SNR), frequency-selective fading (ISI), and Doppler effect (ICI).
- Receiver-side: time synchronization error, frequency error carrier phase error, and different time bases Rx-Tx.

Usually, the receiver does not totally correct the different impairments of the system and therefore the constellations have noise plus, for example, shifts in phase and amplitude. In a DTV receiver there are three categories of measurements depending on the point where they are performed: RF signal, baseband signal, MPEG frames obtained after the demodulation process. Taking into account the characteristics of our DTV receiver, we focus on measurements performed on the baseband signal.

As we mentioned in the previous section, most blocks already calculate several indicators (like the total frequency error or channel gain), and our implementation outputs them. These may aid in determining the origin of problems. However, and regarding the complex received symbols and the resulting bits, three further important indicators are: MER (Modulation Error Ratio), and BER (Bit Error Rate) before Viterbi and after Viterbi. These three measurements characterize the reception quality of a DTV system [21, 22, 23]. The following sub-section will analyze these three indicators in particular, although the measurement system we implemented in GNU Radio (termed `gr-mer`) can measure other parameters that help to understand the quality problems detected with the MER. These auxiliary parameters are: amplitude imbalance, phase imbalance, phase jitter, system target error, residual carrier and quadrature error.

5.3. MER, BER, definitions and measurement procedure

5.3.1 Modulation Error Ratio (MER)

The modulation error ratio (MER) is measured after synchronization. Let (I_i, Q_i) be the i-th complex sample after synchronization. These samples are mapped to the constellation symbols by the decision block; let (I_i^*, Q_i^*) be the constellation symbol corresponding to the sample i. MER is defined as:

$$MER = 10\log_{10}\left(\frac{\sum_{i=1}^{N} {I_i^*}^2 + {Q_i^*}^2}{\sum_{i=1}^{N}((I_i^* - I_i)^2 + (Q_i^* - Q_i)^2)}\right), \tag{4}$$

where N is a certain number of symbols used in the average. It is important to note that the MER in digital systems is a better indicator than the SNR. The MER coincides with the SNR if the only imperfection of the received signal is noise but the MER further reflects other imperfections of digital systems [24].

There are limits in the values of the parameters like the MER in order to have a good quality reception. However, these values depend on the characteristics of the system, for example the size of the constellation. As a reference for ISDB-T systems, the minimum MER is around 25 to 30 dB. Note that if the system were ideal and in the absence of impairments and channel noise, the value of MER goes to infinity. Obviously the value of MER of an actual receiver is limited by the imperfections of the receptor itself.

5.3.2 Bit Error Rate (BER)

Bit error rate (BER) is defined as the ratio between the quantity of erroneous and total bits received. There are two ways to measure the BER. One is offline; that is, the transmitter sends a known sequence of bits and the receiver count the number of erroneous bits received. The second method is online, using the information of the forward error correction algorithms of the receiver. Naturally, in this work we have used this online BER estimation. It is important to highlight that these measurements will be reliable if the algorithm responsible for correcting errors is not outperformed.

Regarding the pre-Viterbi BER, it measures the errors detected by the Viterbi algorithm. Since the implementation we used (included in `gr-dvbt` [16]) uses hamming distance for the path decision, it was relatively straightforward to include the BER as one of its output. Post-Viterbi BER refers to the errors that remain after the Viterbi algorithm and that the Reed-Solomon algorithm detects and corrects. The way to measure this BER is simply counting the number of errors corrected by the Reed-Solomon algorithm.

6. A SIMPLE CASE STUDY

Several studies can be performed in order to test the receiver's performance: effects such as AWGN, multipath fading, Doppler Effect, and carrier and sampling frequency offset may be taken into account. Taking advantage of the already mentioned block-based architecture of GNU Radio, different implementations and algorithms for each one of the blocks in the receiving chain can be tested in order to define the best one in terms of robustness, complexity and computational requirements.

In this section we compare our receiver using two different OFDM synchronization implementations. The first one, without feedback, i.e. without the implementation of the frequency and timing tracking, described in Sec. 4.2.2. The second one, with feedback, more robust regarding carrier and sampling frequency offset nonidealities.

6.1. Equipment and recordings

In order to test a particular channel effect, the signal should be transmitted though an ideal channel, with no multipath fading or noise. It should also be tuned by an ideal tuner, with no frequency or sampling offset. Later on, nonidealities can be digitally added one at a time in order to see how the receiver performs in each case. A full digital simulation, performed entirely in GNU Radio seems to be the best way to do such analysis; however, to that end, an ISDB-T modulator must be used. Sadly, as was already mentioned in Sec. 3.1, no available GNU Radio implementations were found. We then decided to record an actual broadcasting station's transmission taking into account two things: line-of-sight with the transmitting antenna and a MER above 30 dB.

For the recordings we used a USRP B100 with a WBX daughterboard. We also used a standard personal computer with an Intel Core i5-2300 @ 2.80GHz and 8 GB RAM. The tests described in the following subsection were performed in the same computer. The recorded station was Televisión Nacional Uruguay (TNU for short), the Uruguayan national television network, corresponding to frequency 569 MHz. The constellation obtained after demodulating such recording is shown in Figure 5. The resulting MER was almost 33 dB and the BER measured in the Viterbi decoder was about 1×10^{-9}.

Regarding other relevant indicators, the synchronization system detected a sampling offset of about $-0,29 \times 10^{-6}$; that is to say, our USRP's clock is 0.29 ppm slower that the one from TNU. An offset of $-0.28\Delta f$ in the carrier's frequency was also detected, being Δf the bandwidth of each carrier in the OFDM

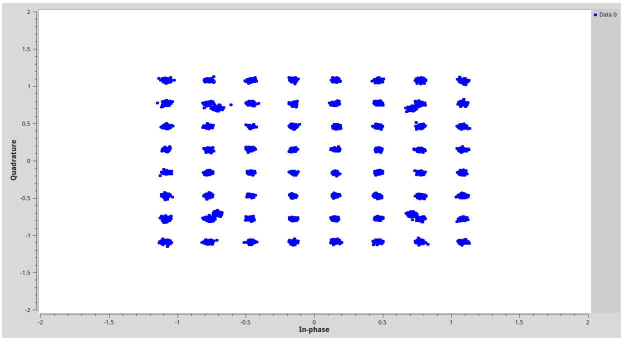

Figure 5: the constellation points for hierarchical layers A and B, corresponding to 64QAM and QPSK modulation schemes, are superimposed.

symbol (125/126 kHz in mode 3).

6.2. Carrier and sampling frequency offset effect

We fist digitally added a $100\Delta f$ extra frequency offset between transmitter and receiver, and found that the system was very robust against this effect, no matter which OFDM synchronization implementation we used. It is worth noting that this similar behavior between both implementations should be expected. The feedback system's benefits will be significant under frequency-selective fading or sampling frequency errors. In this case, where there is no other effect other than the carrier frequency offset, the feedback is not really needed.

The results were different when adding a sampling frequency offset of 100 ppm. As it is illustrated in the upper part of Figure 6, for the synchronization algorithm with feedback, the MER decreased slightly (it is now 29 dB), but the BER at the Viterbi grew up to 1×10^{-5}, although the BER at the Reed Solomon was still negligible most of the time.

Regarding the algorithm with no sampling correction, the results were much worse. For an offset of 100 ppm, the constellation points were almost undetectable. To illustrate how this affects the received signal, in Figure 6 the obtained constellation is shown for a sampling frequency offset of only 10 ppm.

7. CONCLUSIONS AND FUTURE WORK

We have presented `gr-isdbt`, the first fully software-based, free and open ISDB-T full_seg receiver. We have discussed the particularities of ISDB-T with respect to DVB-T, specially those that represent the biggest challenge to the receiver (such as the absence of Continual Pilots).

We have also presented a very simple measurement study to illustrate how our framework may substitute more costly equipment. Indeed, there are several measurement equipment specially designed for digital TV, but their prices range roughly from 5.000 USD to 100.000 USD. Just to cite an example which considers only software, MaxEye Technologies' DVB-T/H sig-

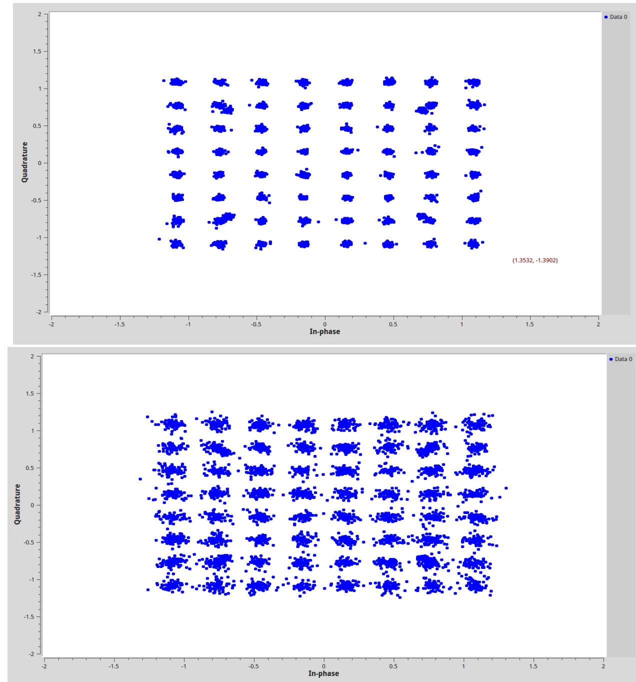

Figure 6: The resulting constellation under a sampling error. Above, the sampling error is 100 ppm and the feedback system is in use. Below, the sampling error is 10 ppm and the feedback systems is not used.

nal analysis and monitoring toolkit, a third-party add-on to National Instrument's LabView, which may be used together with the USRP, has a listing price of 6.000 USD [25]. In addition to cost, these measurement equipments present the negativity of being closed. This means that in case of a doubt, the user counts only with the technical support and/or the manual. Moreover, adding features (such as another type of measurement, or a different measurement technique) will imply even more cost (or will simply not be possible).

Naturally, there is room for improvement. For instance, regarding coarse synchronization, we have used the classic algorithm by van de Beek et al. [18] to perform coarse time and frequency synchronization. However, there have been several proposals since (see for instance [26]), and an evaluation of possible substitutes is in order. The same applies for other blocks, such as the tracking sub-system or the channel equalization (see for instance [27] for a recent survey on the latter subject). It is important to highlight however, that complexity is a crucial factor here, and that any such substitute should not increase it significantly. Another aspect we would like to highlight is that the modular architecture of the receiver (inherited from GNU Radio) makes these substitutions and evaluations a relatively simple task.

We consider this work to be a further step in the technological appropriation that the region is performing regarding digital television. We believe that implementations based on the free and open software paradigm are key to an effective appropriation. Moreover, in our particular case, the Software Defined

Radio technology plays a fundamental role, assuring generality (by means of frameworks such as GNU Radio) and ease of distribution (with the exception of the general-purpose hardware, the different applications are simply downloaded from the Internet).

ACKNOWLEDGEMENTS

This work was partially funded by Uruguay's national research agency (ANII) and the telecommunications and audiovisual media services (DINATEL) under grant FST_1_2013_1_13179, and also by "CSIC Grupos I+D 2014" program.

REFERENCES

[1] "A/53: Atsc digital television standard." [Online]. Available: http://atsc.org/standard/a53-atsc-digital-television-standard/

[2] "Framing Structure, Channel Coding and Modulation for Digital Television Terrestrial Broadcasting System," Chinese National Standard GB 20600-2006, in Chinese, 2006.

[3] "EN 300 744 V1.6.1. Framing structure, channel coding and modulation for digital terrestrial television. ," 2009. [Online]. Available: www.etsi.org/deliver/etsi_en/300700_300799/300744/01.06.01_60/en_300744v010601p.pdf

[4] "STD-B31. Transmission System for Digital Terrestrial Television Broadcasting." 2014. [Online]. Available: www.arib.or.jp/english/html/overview/doc/6-STD-B31v2_2-E1.pdf

[5] "Normas Brasileiras de TV Digital." [Online]. Available: http://forumsbtvd.org.br/acervo-online/normas-brasileiras-de-tv-digital/

[6] Ettus Research, A National Instruments Company, "Universal Software Radio Peripherical." [Online]. Available: http://www.ettus.com/

[7] Nuand LLC., "bladeRF." [Online]. Available: http://nuand.com/

[8] Great Scott Gadgets, "HackRF." [Online]. Available: http://greatscottgadgets.com/hackrf/

[9] "GNU Radio. The free & open software radio ecosystem." [Online]. Available: http://gnuradio.org

[10] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[11] "GNU Radio Companion." [Online]. Available: http://gnuradio.org/redmine/projects/gnuradio/wiki/GNURadioCompanion

[12] T. Rondeau, N. McCarthy, and T. O'Shea, "SIMD Programming in GNU Radio: Maintainable und User-Friendly Algorithm Optimization with VOLK," in *2013 Wireless Innovation Forum European Conference on Communications Technologies and Software Defined Radio (SDR'13 - WInnComm - Europe)*. [Online]. Available: https://gnuradio.org/redmine/attachments/download/422/volk.pdf

[13] H. Sugano, R. Miyamoto, and M. Okada, "Fully Software-based real-time ISDB-T 1 segment receiver," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2011 IEEE International Symposium on*, June 2011, pp. 1–5.

[14] Y. P. Maciel, "ISDB-T Modulator in GNU Radio Companion." [Online]. Available: https://www.youtube.com/watch?v=p5SMmBVFfMk

[15] V. Pellegrini, M. Di Dio, L. Rose, and M. Luise, "A Real-Time, Fully-Software Receiver for DVB-T Signals based on the USRP," in *6th Karlsruhe Workshop on Software Radios*, 2010.

[16] B. Diaconescu, "gr-dvbt." [Online]. Available: https://github.com/BogdanDIA/gr-dvbt

[17] F. Larroca, P. Flores-Guridi, G. Gómez-Sena, V. González-Barbone, and P. Belzarena, "gr-isdbt: An ISDB-T 1-segment Receiver Implementation on GNU Radio," in *XLI Conferencia Latinoamericana en Informática (CLEI 2015)*, 2015.

[18] J.-J. van de Beek, M. Sandell, and P. Borjesson, "ML estimation of time and frequency offset in OFDM systems," *Signal Processing, IEEE Transactions on*, vol. 45, no. 7, pp. 1800–1805, Jul 1997.

[19] T.-D. Chiueh and P.-Y. Tsai, *OFDM Baseband Receiver Design for Wireless Communications*. John Wiley & Sons, 2007.

[20] G. F. Michael Speth, Stefan Fechtel and H. Meyr, "Optimum Receiver Design for OFDM-Based Broadband Transmission–Part II: A Case Study ," *IEEE Transactions on Communications*, vol. 49, no. 4, pp. 571–577, Apr 1997.

[21] "Digital video broadcasting (DVB); measurement guidelines for DVB systems," European Telecommunications Standards Institute, ETSI Technical Report, 1997. [Online]. Available: http://www.etsi.org/deliver/etsi_etr/200_299/290/01_60/etr_290e01p.pdf

[22] "Critical RF measurements in cable, satellite and terrestrial DTV systems," Tektronix, Tektronix Application Note, 2008. [Online]. Available: http://www.tek.com/dl/2TW_17370_2_HR_0.pdf

[23] R. Hranac, "BER and MER fundamentals," Cisco System presentation, 2007. [Online]. Available: http://www.gcscte.org/presentations/2008/Ron.Hranac_Presentation-BER%20+%20MER%20Fun.pdf

[24] "Digital transmission: Carrier-to-noise, signal-to-noise and modulation error ratio," Cisco, Tech. Rep., 2012. [Online]. Available: https://www.broadcom.com/collateral/wp/CMTS-WP101-R.pdf

[25] National Instruments, "DVB-T/H Signal Analysis and Monitoring Toolkit - MaxEye Technologies." [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/en/nid/212594

[26] M. Morelli, C.-C. Kuo, and M.-O. Pun, "Synchronization Techniques for Orthogonal Frequency Division Multiple Access (OFDMA): A Tutorial Review," *Proceedings of the IEEE*, vol. 95, no. 7, pp. 1394–1427, July 2007.

[27] Y. Liu, Z. Tan, H. Hu, L. Cimini, and G. Li, "Channel estimation for ofdm," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 4, pp. 1891–1908, Fourthquarter 2014.