Simulation Software for the ISDB-T\textsubscript{B} Modulation System

Cristiano Akamine and Yuzo Iano

Abstract—This paper presents a simulation tool for the ISDB-T\textsubscript{B} modulator, to perform complex analyses in each processing stage. From a Broadcast Transport Stream (BTS) file you can follow the processing performed in each modulator block to the final transmission stage. It is possible to make real transmissions with the RF files generated in the simulator with the use of a vector signal generator. A mathematical and practical approach to the operation of each modulator block is performed starting with the description of the BTS signal up to the final transmission stage.

Index Terms—Broadcast Transport Stream – Orthogonal Frequency Division Multiplexing (BST-OFDM), Integrated Digital Broadcasting System – Terrestrial (ISDB-T), Modulator, Simulation.

I. INTRODUCTION

The Integrated Services Digital Broadcasting - Terrestrial (ISDB-T) modulation system \cite{1} is designed to operate with various types of services occupying a TV channel with a bandwidth of 6, 7 or 8 MHz. The system is very flexible and allows the combination of multiple services, such as portable reception, mobile, and fixed with unequal protection errors on the same channel. The system uses the Band Segmented Transmission - Orthogonal Frequency Division Multiplexing (BST-OFDM) modulation in which each segment uses a wide band corresponding to $6/14\ \text{MHz} = 428.57\ \text{kHz}$ for 6 MHz bandwidth. The thirteen segments may be combined in up to three hierarchical layers called Layer A, Layer B, and Layer C. Recently, a modified version of the ISDB-T modulation system called ISDB-T version B (ISDB-T\textsubscript{B}) \cite{2} was developed and adopted in Brazil and many other countries. The main difference of the ISDB-T\textsubscript{B} modulation system is the RF channel and transmission mask \cite{2}, \cite{3}. The ISDB-T\textsubscript{B} modulation system can be divided into three stages composed of the re-multiplexor, the channel encoder, and the modulator as can be seen in Fig. 1. The ISDB-T\textsubscript{B} modulator cannot be configured locally and depends on the multiplexer/re-multiplexer to control and configure all the processing stages. To illustrate the operation of these stages, this article presents a software simulation of the ISDB-T\textsubscript{B} modulator.

From the Broadcast Transport Stream (BTS) that is generated in the multiplexing/re-multiplexing stage, all blocks that compose the modulator are described and analyzed using this computational tool. This simulator allows the monitoring of the input and output of all blocks in supported file formats with Matlab, C/C++ and support for Field Programmable Gate Array (FPGA). In addition, RF vectors can be created that can be transmitted by arbitrary signal generators. Fig. 2 shows a screenshot of this software, in which it is possible to select the file format, processing block, signal generator type and simulation time.

Thus, a review of the re-multiplexing stage is carried out in Section II. In Sections III and IV, the coding and modulation stages are presented. In Section V, the simulation tool is evaluated using an arbitrary signal generator and spectrum analyzer with an ISDB-T\textsubscript{B} demodulator. Finally, in Section VI the conclusion is presented.

II. RE-MULTIPLEXING

To synchronize all the layers between the system source coding and modulation, the MPEG-2 TS 188 byte \cite{4} from the source coding scheme, data carousel, etc. is multiplexed and re-multiplexed. The output of re-multiplexer is formed by a single TS with a size of 204 bytes and constant bit rate of four
times the sampling frequency of the Inverse Fast Fourier Transform (IFFT) modulator, or \((4 \times 512/63) = 32.5079\text{Mb/s}\) to 6 MHz bandwidth. Due to this characteristic the output signal of the re-multiplexer is called BTS [5]. The process of re-multiplexing is the positioning of each Transport Stream Packets (TSP) for each Layer and null TSP in a synchronized order with the demodulator of the receiver. The null TSP insertion maintains a constant BTS bit rate signal independent of the modulation parameters and channel coding. The packet order is required to secure the hierarchical transmission in a single TS and to minimize processing at the receiver [6].

The BTS signal is structured in a multiplexing frame in which the number of TSP depends on the mode and guard interval (GI) as can be seen in Table I [1], [2].

Additional the dummy byte, or 16 bytes in each BTS TSP is used to indicate the hierarchical layer that each TSP will be transmitted, TSP counter, frame headers, and the auxiliary data drivers, etc. Optionally, a Reed Solomon (RS) block code shortened (204,196,4) is applied and has correction capability of up to 4 bytes in a BTS TSP.

Fig. 3 shows an example of a TSP and multiplexer frame.

The segment transmission order of the OFDM at the end of the modulation must be fully synchronized to the BTS signal frame multiplexer at the output of the re-multiplexer. Problems in the formation of the BTS signal multiplexer frame and clock and can generate a signal transmission error.

Additionally the Modulation Configuration Control Information (MCCI) is sent in a package called ISDB-T Information Packets (IIP). The IIP is only transmitted once in the BTS signal multiplexer frame in which it has two descriptors that are called MCCI and Network Synchronization Information (NSI). The MCCI sets the modulation parameters and channel encoder as the size of IFFT, IG, modulation method, code rate, and the number of segments, etc. The NSI is used in the synchronization of the single frequency network in which the Synchronization Time Stamp (STS), maximum delay, control equipment, and product number, etc. are entered in this field.

Fig. 4 shows an example of an IIP for a BTS signal.

### III. Encoder Channel

The channel encoding process starts with IIP detection. The IIP can be identified by PID 0x1FF0 or by Layer 0x8 indication that can be obtained in the first four bits of byte 190. Reading descriptor IIP MCC, the TMCC generated is responsible for the configuration and control of all coding and modulation stages. In [1] and [2] it is possible to obtain the details of each bit for the MCCI and TMCC descriptors.

Fig. 5 shows the extracted IIP modulation parameters from Fig. 4 and Fig. 6 shows a simplified block diagram of the channel encoder. The detail of each of these blocks is presented in the following sub-sections.

### A. Layer Separator

The layer separator has the purpose of directing each BTS TSP to its respective layer. The separator reads BTS byte 190 and redirects each TSP according to the information contained in Table II. Null TSP and IIP are not transmitted in any of the hierarchical layers. The resulting output of the layer separator is an MPEG-2 TS, 188 bytes in size. Fig. 7 shows a Layer A TSP at the output of the separator.

### TABLE I

<table>
<thead>
<tr>
<th>Mode</th>
<th>Guard interval reason</th>
<th>1/4</th>
<th>1/8</th>
<th>1/16</th>
<th>1/32</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (2k)</td>
<td></td>
<td>1280</td>
<td>1152</td>
<td>1088</td>
<td>1056</td>
</tr>
<tr>
<td>2 (4k)</td>
<td></td>
<td>2560</td>
<td>2304</td>
<td>2176</td>
<td>2112</td>
</tr>
<tr>
<td>3 (8k)</td>
<td></td>
<td>5120</td>
<td>4608</td>
<td>4352</td>
<td>4224</td>
</tr>
</tbody>
</table>

Fig. 3. BTS Frame Multiplexer.

Fig. 4. IIP parameter identification.

Fig. 5. Extracted IIP modulation parameters.

Fig. 6. Simplified block diagram of the channel encoder.

Fig. 7. Layer A TSP at the output of the separator.
Fig. 6. Diagram of the channel coding stage.

Using (2) in GF(256) it is possible to calculate the polynomial generator \( g(x) \) in (3).

\[
g(x) = (x + a^0)(x + a^1)(x + a^2) \ldots (x + a^{2t-1})
\]  

(2)

\[
g(x) = x^{16} + 59x^{15} + 13x^{14} + 104x^{13} + 189x^{12} + 68x^{11} + 209x^{10} + 30x^9 + 8x^8 + 163x^7 + 65x^6 + 41x^5 + 229x^4 + 98x^3 + 50x^2 + 36x^1 + 59
\]  

(3)

Considering the message to be coded \( U(x) \), the encoded signal \( V(x) \) can be obtained using (4) and (5) where \( q(x) \) is the quotient and \( r(x) \) is the remainder of the division.

\[
\frac{u(x)x^{n-k}}{g(x)} = q(x) + r(x)
\]  

(4)

\[
V(x) = U(x) \times x^{n-k} + r(x)
\]  

(5)

Fig. 8 shows the TSP in Fig. 7 in the output of RS code. The addition of 16 bytes and the displacement of 1 byte can be seen. This displacement is necessary to maintain the synchronism with the OFDM frame.

C. Energy Dispersal

In one TS, there are many null packets or bytes with values 0xFF and 0x00 that are used to adjust the bit rate or complement the number of bytes in a TSP. These packet sequences or bytes are often present at the source coding output and multiplexing.

To disperse these sequences, an interconnected energy dispersal at the RS output is used to reduce interference between symbols generated by the repetitive transmission of the same information.

This dispersal consists of a Pseudo Random Binary Sequence Generator (PRBS) and an adder module 2. The PRBS generator uses the polynomial generator \( g(x) = 1 + x^{14} + x^{15} \). This generator consists of 15 shift registers and an adder module 2 that is connected to the output registers 14, 15.
and input register 1 as can be seen in Fig. 9. The sequence length of this generator is $2^{15} - 1 = 32767$, and is synchronized with the IIP and the TS clock of each layer. The disperser is initialized by a frame multiplexer with the word “100101010000000” and is disabled by a TSP synchronizing byte.

Fig. 9. Energy dispersal circuit.

Fig. 10 shows the TSP of Fig. 8 at the energy disperser output. It can be seen that the repetitive sequence of ones (0xFF) are dispersed.

D. Byte interleaver

Interleavers are used in a wide variety of communication methods. An interleaver is a device that receives an alphabet of symbols and produces identical output symbols in a different order without increasing the bit rate. The interleaver disperses the bit sequences in order to minimize the block error effect in the middle of a data stream [9], [10]. These block errors are created from the impulsive noise of some equipment and selective fading in the communication channel. The error-correcting codes cannot correct errors that are concentrated in a sequence. Using interleavers, the sequence of symbols to be transmitted is scrambled and at the time that an accumulation of errors in the communication channel occurs, the decoder unscrambles the sequence spreading the errors. At that point, the error correcting codes can detect and correct errors.

The byte interleaver is a convolutional interleaver that uses time multiplexing and shift registers [11]. Each input multiplexer has a set of shift registers. The registers delay the bit sequence and then these bits are grouped again. In Fig. 11, it is possible to see how this interleaver is built. It has 12 branches (B) and 17-byte shift registers (M). The B branches are cyclically connected to the energy disperser output and transfer 1 byte at a time via each branch. The first branch $(B_i=0)$ has no memory, and the symbols are immediately transferred to the output. The byte interleaver aims to spread the packets from the Reed Solomon and the energy disperser to increase their efficiency before block errors occur.

The maximum delay by the interleaver branch can be calculated by (6).

$$N = (B - 1) \times M$$

(6)

The delay generated by the convolutional interleaver can be calculated by (7) in which the equivalent of 2244 bytes is represented by 11 TSP consisting of 204 bytes each.

$$L_0H$$

(7)

However, to keep all blocks synchronized with the frame multiplexer, a delay adjustment is inserted to supplement the number of TSP in a multiplexer frame. It is possible to calculate the number of TSP used in a multiplexer frame with the use of (8). This way, the synchronization delay value can be calculated using (9), where $N_s$ is the number of segments, $N_c$ is the number of data carriers, $b_{QAM}$ is the number of bits per symbol in the primary modulation and $R_{cc}$ is the rate of the convolutional encoder.

$$L_0H$$

(8)

$$L_0H$$

(9)

Fig. 12 shows the packet in Fig. 10 at the byte interleaver output after 48 packets.

E. Convolutional Encoder

The convolutional codes were introduced by Elias in 1955 as an alternative to the use of block codes [12]. The substantial difference between block codes and convolutional codes lies
in the fact that convolutional codes make use of memories, thus a given output at a certain time depends on not only some inputs at that very time, but also a time in the past \( m \), where \( m \) is the number of memories [8]. A convolutional encoder \((n, k, m)\) with \( k \) inputs, \( n \) outputs, and \( m \) memories may be implemented as a combinatorial sequential circuit which makes use of registers comprised of flip-flops, adder module 2 (or exclusive), and multiplexers/re-multiplexers. The rate of the convolutional encoder \( R = k/n \) depends on the input \( k \) and the output \( n \). Fig. 13 shows the combinational circuit used by ISDB-TB [1], [2].

![Convolutional encoder (2,1,6) rate ½.](image)

This convolutional code has mother rate \( R = 1/2 \), 64 states and its impulse response is expressed in (10).

\[
g^{(0)} = 171_{oct}(1 1 1 1 0 0 1) \]
\[
g^{(1)} = 133_{oct}(1 0 1 1 0 1 1) \]

(10)

Considering the input message \( u \), \( X \) and \( Y \) outputs can be obtained using (11), where \( \otimes \) represents the discrete convolution.

\[
X = u \otimes g^{(0)} \]
\[
Y = u \otimes g^{(1)} \]

(11)

Additionally, puncturing is used to discard some of the bits at the convolutional encoder output. This technique allows the bit rate to be varied, and the puncturing pattern can be seen in Table III. The \( P \) field indicates the bits that must be discarded and are represented by 0. The minimum free distance (\( d_{free} \)) of this convolutional code [13] is used as a performance parameter. In (12) it is possible to calculate the asymptotic gain of this code using a Viterbi decoder hard decision in relation to the uncoded modulation [14].

\[
N = \frac{120}{B-1} \]

(13)

The bit interleaving is completed by a serial/parallel converter with variable size according to the modulation method. Thus, the input bits are converted from serial to parallel in dibit (2 bits), quadbit (four bits) or sixbit (6 bits) for the QPSK, 16-QAM, and 64-QAM modulations, respectively. The number of branches \( B \) of the interleaver is equal to the number of bits in each modulation method, and each branch has a specific delay. Because of these features, this interleaver is classified as a multiplexed convolutional interleaver [15]. The maximum branch delay is calculated using (13) and the total delay using (7).

![Diagram of the bit interelaver used in the 64-QAM modulation.](image)

IV. MODULATION

The ISDB-TB modulation stage can be seen in Fig. 15 and will be detailed in the following subsections.

A. Layer Combiner

The set of bits entering the layer combiner are grouped, and in this article are called symbols. It may be noted that our proposal is different from [1] and [2] and aims to reduce memory usage. Thus, the layer combiner performs the concatenation of the data symbols in all segments. For example, Mode 3 has 384 data carriers in each segment, this block groups all the segments amounting to 4992 symbols (carriers). Table V shows the distribution of the pilots in the three ISDB-TB modes.
B. Time Interleaver

After the layer combiner, the signal is interlaced. The time interleaver is formed by a convolutional interleaver that aims to interlace the carriers within various OFDM symbols. The time interleaver acts separately on each OFDM data segment and is cyclically combined at the output. The interleaving size can be adjusted by varying parameter $I$. In (14) it is possible to calculate the delay for each branch of the time interleaver, where $i$ represents the number of each branch $i = 0, 1, 2, ... B - 1$ and mod is a function that returns the remainder of the division. The number of branches $B$ of the time interleaver is equal to the number of data carriers.

$$D_i = \text{mod}(54 \times i, 96) \times 1$$  \hspace{1cm} (14)

Table VI indicates the maximum time interleaver delay and the delay adjustment that is required to synchronize symbols within an OFDM frame. Fig. 16 shows a diagram of the interleaver and in Fig. 17 a scatter plot of the time interleaver of a segment in Mode 3. The time interleaver spreads the symbols from the modulation in each layer in which the length can be chosen, for the approximate values 0, 100, 200 and 400ms. The time interleaver increases the robustness of the system against impulsive noise and improves performance of mobile reception [16].

C. Frequency Interleaver

The frequency interleaver can be considered a block interleaver. In this interleaver symbols are written in a memory and read in a certain order. The frequency interleaver is performed in one OFDM symbol and is divided into three parts, as can be seen in Fig. 18.
The first type of interleaving performs carrier interleaving between the segments. If the partial reception option is enabled, the zero data segment is not scrambled. Fig. 19 shows the scatter plot of the interleaver used in the configuration shown in Fig. 5. It can be seen that symbols in the first segment enter and leave in the same order.

In Fig. 20, the carrier rotation within each segment can be seen. In Fig. 21, the carrier dispersion pattern within each segment can be seen. In [1] and [2] it is possible to obtain the tables and parameters of these interleavers. The final result, of frequency interleaving, can be seen in Fig. 22. Frequency interleaving is applied to the OFDM symbol that increases the system robustness against selective frequency fading [16]. Both the time interleaver and the frequency interleaver increase the error correction stage efficiency.

D. Mapper and Spectrum Adjustment

After frequency interleaving, these symbols are modulated in QPSK, 16-QAM or 64-QAM through the use of a table.
Table VII shows the bit values, and the symbols for the QPSK modulation standardized to the power of 1 Watt. The tables for modulations 16-QAM and 64-QAM can be extracted from [1], [2].

<table>
<thead>
<tr>
<th>bit[1], bit[0]</th>
<th>In-phase</th>
<th>Quadrature</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>ξ</td>
<td>ξ</td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>-ξ</td>
</tr>
<tr>
<td>1.0</td>
<td>-ξ</td>
<td>0</td>
</tr>
<tr>
<td>1.1</td>
<td>ξ</td>
<td>-ξ</td>
</tr>
</tbody>
</table>

Fig. 23 shows the constellations of the QPSK, 16-QAM, and 64-QAM modulations, respectively.

The spectrum adjustment is required to place the segments in the specific order established in [1] and [2]. Thus, the zero segment is positioned at the center of the spectrum and odd and even segments are distributed according to Fig. 24.

![Segment positions utilized after the spectrum adjustment.](image)

**E. Frame Adaption**

The transmission signal is organized into frames. Each frame has a duration T_f and consists of 204 OFDM symbols. Each OFDM symbol has 13 segments and consists of a number of carriers, in Mode 1 K=1405 carriers (2k), K=2809 carriers in Mode 2 (4K), and K=5617 carriers in Mode 3 (8k) in which all are transmitted with the duration T_s. Fig. 25 shows an OFDM frame for a coherent modulation. Within the OFDM frame, Scattered Pilots (SP) and Continual Pilots (CP) are inserted and modulated in Binary Phase Shift Keying (BPSK) with a 33% power increase to guarantee the estimation and synchronization. The TMCC and Auxiliary Channel (AC) are inserted in the frame and modulated in Differential BPSK (DBPSK) with a 33% power increase for the purpose of signaling the modulation parameters, channel coding, frame synchronization, and transmission of auxiliary data. In total, 157 carriers out of 1405 carriers are used as pilots in Mode 1, 313 carriers out of 2809 carriers are used as pilots in Mode 2, and 625 carriers out of 5617 carriers are used as pilots in the Mode 3. In Fig. 26, the constellation of the frame adaption output can be seen. It can be seen that the SP pilots, TMCC, and AC are modulated in BPSK and possess greater potency than the QPSK and 64-QAM constellations.

![Frame output adaptation constellation.](image)

**F. OFDM Modulation**

The OFDM modulation appeared around the 1960s when Chang published his article on transmission synthesis with limited multi-channel bands [17]. He introduced the concept of transmitting messages across limited multiple channel bands without causing Inter-Carrier Interference (ICI) and Inter-Symbol Interference (ISI). In 1971, Weinstein and Ebert [18] used the Discrete Fourier Transform (DFT) to improve the performance of modulation and demodulation. Due to the computational complexity of DFT and inverse DFT (IDFT), the Fast Fourier Transform algorithm (FFT) and Inverse Fast Fourier Transform (IFFT) discovered by Cooley and Tukey in 1965 [19] are used in modulation and demodulation, respectively. However, zeroes are added to the adapted frame output, to obtain the number of samples necessary to use IFFT and generate the useful part of the OFDM symbol (Tu). This technique is called zero padded.
and the OFDM modulator block diagram can be seen in Fig. 27, where \( N_c \) represents the number. The IFFT size can be configured to one of three values, Mode 1 = 2k; Mode 2 = 4k; and Mode 3 = 8k.

G. Guard Interval

An important contribution to the OFDM modulation was made by Peled and Ruiz in 1980 [21], which introduced the cyclic prefix or cyclic extension, solving the orthogonality problem. Instead of using an empty guard space, they filled this space with a cyclic extension of the OFDM symbol.

The duration of the guard interval (\( \Delta \)) is obtained by the duration \( T_u \times k \), where \( k = 1/4; 1/8; 1/16; \) or \( 1/32 \). Thus, an OFDM symbol with duration \( T_s \) is composed of a guard interval with the duration \( \Delta \) and \( T_u \).

Fig. 28 shows an OFDM symbol with guard interval.

V. SIMULATION SOFTWARE

The simulator test was conducted with an arbitrary waveform generator and spectrum analyzer with an ISDB-T\(_B\) signal demodulation option. The RF vector from the simulator was installed in the generator and Bit Error Rate (BER), Modulation Error Rate (MER), and Mask Intermodulation tests were performed. Fig. 30 shows the equipment used in the tests.

In the first test, it was verified that the ISDB-T\(_B\) signal generated by the simulator is within the mask transmission. Fig. 31 shows the result of this test.

In the Fig. 32, the measured bit error rate (BER) shows that the error is null before and after the RS in the two hierarchical layers. With this result, it can be concluded that all the stages of channel coding and simulated modulation function correctly. Furthermore, it can be seen that the modulation parameters detected through TMCC are the same as those in Fig. 5.

Figs. 33 and 34 show the MER measurement results. The value of MER is above 30 dB showing that any distortion in the modulation stage was introduced by the simulator.
VI. CONCLUSION

This paper presented a simulation tool that allows the analysis of all the blocks that make up an ISDB-TB modulation to be completed. Numerical and graphical results were presented in order to illustrate the operation of an actual modulation system. Practical measures using an arbitrary waveform generator and spectrum analyzer were performed to demonstrate the compatibility of this simulation tool with the ISDB-TB system.

ACKNOWLEDGMENT

The authors would like to thank RH-TVD CAPES and their colleagues at Mackenzie Digital TV Laboratory.

REFERENCES

Cristiano Akamine received his B.Sc. degree in Electrical Engineering from Mackenzie Presbyterian University, São Paulo, Brazil, in 1999. He received his M.Sc. and Ph.D. degree in Electrical Engineering from the State University of Campinas (UNICAMP), São Paulo, Brazil, in 2004 and 2011 respectively. He is a professor of Embedded Systems, Software Defined Radio and Advanced Communication Systems at Mackenzie Presbyterian University. He is a researcher in the Digital TV Research Laboratory at Mackenzie Presbyterian University since 1998, where he has had the opportunity to work with many digital TV systems. His research interests are in system on chip for broadcast TV and Software Defined Radio.

Yuzo Iano graduated and obtained a doctorate at Universidade Estadual de Campinas (Unicamp) where he has acted as a professor since 1973. He is responsible for the Laboratório de Comunicações Visuais de Departamento Comunicações da Faculdade de Engenharia Eléctrica e de Computação de Unicamp (LCV/Decom/FEEC/Unicamp). His research covers telecommunications fields including digital signal processing, audio, video, and images.