Abstract
This article analyzes channel coding and equalization stages in the 8-level Vestigial Sideband transmission for the Digital Television broadcast system developed by ATSC, the Advanced Television Systems Committee. We present bit error rate versus C/N performance for a simulator developed by the authors for the purpose of assessing the robustness of the ATSC 8-VSB coding and equalization stages in the presence of multipath propagation, additive Gaussian noise and of channel impulsive noise surges. Implementation aspects for the developed simulator are discussed.

I. Introduction
The ATSC 8-VSB digital system [1] was proposed aiming to replace the veteran NTSC analog system for terrestrial television broadcast. Using the same 6MHz channel of the NTSC system, the 8-VSB system is conceived for superior performance, with strong immunity to interference, noise and multipath effects. As comparison, at 34dB above the noise floor the NTSC video has a performance considered just marginal, while the 8-VSB digital signal could drop to only 15dB above the noise floor before any video or audio degradation would be noticed [12]. To a great extent, this performance of the ATSC 8-VSB digital system is due to the coding and equalization stages, or data processing stages, which are the scopes of this study.

Before we start to analyze the data processing stages, it is instructive to briefly describe the other stages in the ATSC 8-VSB system. Figures 1 and 2 respectively show the 8-VSB transmitter and receiver general block diagrams. The blocks inside the dashed rectangles in both figures represent the data processing stage. MPEG-2 packets (188 bytes) that stem from the transport layer of a previous MPEG-2 encoder [3][4] compose the information stream at the input of the 8-VSB transmitter. This MPEG-2 encoder, which precedes the 8-VSB transmitter (not shown in Figure 1), has previously compressed the audio and video data so that the information rate at the transmitter input is 19.39Mbps [1][2].

Figure 1: 8-VSB transmitter block diagram.

The data stream at the output of the Convolutional Encoder is composed of a sequence of 8-VSB symbols, each one with 3 bits. One 8-VSB symbol can assume one of the values of the set {1, -1, 3, -3, 5, -5, 7}. For each incoming 188 bytes MPEG-2 packet at the Encoding Stage input, a sequence of 828 symbols is generated at the Encoding Stage output. Upon being processed by the Multiplexer, each 828 symbols sequence is pre-appended by the symbol sequence [S, -5, -5, 5], named Segment Sync. The Segment Sync plus the Field Sync followed by the 312 Data Segments is called a Field, therefore each Field is constituted by 313 sequences of 832 symbols.

Figure 3 is the two-dimensional representation of the one-dimensional sequence of 2x313x832 8-VSB symbols which constitutes a pair of fields. Figure 4 shows the first 200 symbols of a typical Data Segment in an 8-VSB filed at the Encoding Stage output.

The purpose of the Segment Sync is to synchronize the transmitter and receiver clocks. At the receiver, a correlation filter in the Sync Restorer block recovers the transmitter original clock using the periodicity of the Segment Sync signal. The Field Sync provides the Equalizer with known symbol sequences (PN511 and PN63...
which are previously inserted in the Field Sync generated at the transmitter. These sequences are used as references for the Equalizer and allow it to adaptively minimize multipath effects [2].

The Pilot Insertion block digitally adds the DC level 1.25 to all 8-VSB symbols at the Multiplexer output. This results in a small pilot carrier at the lower portion of the channel spectrum, which allows a PLL (phase-locked loop) in the receiver Synchronous Detector to establish a phase reference between the receiver and the transmitter. This is necessary in order to assign a time reference to the Field Sync and Segment Sync signals in the receiver.

The Pilot Insertion block digitally adds the DC level 1.25 to all 8-VSB symbols at the Multiplexer output. This results in a small pilot carrier at the lower portion of the channel spectrum, which allows a PLL (phase-locked loop) in the receiver Synchronous Detector to establish a phase reference between the receiver and the transmitter. This is necessary in order to assign a time reference to the Field Sync and Segment Sync signals in the receiver.

The Pilot Insertion block digitally adds the DC level 1.25 to all 8-VSB symbols at the Multiplexer output. This results in a small pilot carrier at the lower portion of the channel spectrum, which allows a PLL (phase-locked loop) in the receiver Synchronous Detector to establish a phase reference between the receiver and the transmitter. This is necessary in order to assign a time reference to the Field Sync and Segment Sync signals in the receiver.

Figure 2: 8-VSB receiver block diagram. Following the receiver, an MPEG-2 decoder reconstructs the audio and video signals [4]. The 188 bytes packets needed for the MPEG-2 transport layer are obtained by re-inserting the sync byte in each 187 bytes sequence at the Derandomizer output.

The NTSC Filter rejects any interference signals from strong nearby NTSC stations. The NTSC filter is a temporary feature in the ATSC 8-VSB system [1][2]. It will be eliminated by the end of the transitional period from the NTSC system to the 8-VSB system [2]. Thus, this study will analyze the ATSC 8-VSB coding stage assuming the NTSC Filter as nonexistent.

The Decoding Stage input is the input of the Viterbi Decoder. At this point of the receiver, the information stream has the same structure shown in Figures 3 and 4. Specifically, if no signal degradation has been occurred in the channel, the information flow at this point is a replica of the information flow at the Convolutional Encoder output in the transmitter.

For each incoming sequence of 828 8-VSB symbols at the Decoding Stage input, a 187 bytes sequence at the Decoding Stage output is generated. By adding the sync byte to each one of these sequences we obtain the MPEG-2 packets for the subsequent MPEG-2 decoder transport layer.

Figure 3: 8-VSB Field structure. The 828 symbol sequence, which follows the 4 symbol Segment Sync in each Data Segment, contains audio and video information from the MPEG-2 data packets. The Encoding Stage FEC (forward error correction) algorithms add redundancy to this information in order to identify and to correct any signal degradation imposed by the channel.

Figure 4: Typical Data Segment of an 8-VSB Field. Notice the randomness of the 8-VSB levels, which contributes to a nearly flat channel spectrum.

II. ATSC Encoder

In this section we analyze the 8-VSB transmitter Encoding Stage, shown in Figure 1. An 8-VSB simulator implemented by the authors for the purpose of studying the ATSC 8-VSB channel coding stage supports the analysis that follows. Thus, the data structure and the processing flow, in those aspects left at designer discretion by the ATSC standard [1], shall follow the implementation conceived by the authors.

For each incoming 188 bytes MPEG-2 packet at the Encoding Stage, the Synchronizer extracts the sync byte (the first byte) and stores the remaining 187 bytes sequence in a buffer with a capacity of 58344 bytes. Ahead in the transmitter block diagram, the Multiplexer replaces the sync
byte by the Segment Sync. The buffer is considered full when it stores 312 sequences, each one with 187 bytes. In this situation the buffer stores the number of bytes that enables the Encoding Stage to generate a complete 8-VSB Field at its output

Once the buffer is full, the Randomizer performs a bitwise exclusive-or (xor) logic operation between each buffer byte and each output byte from the pseudo-random sequence generator shown in Figure 5. This procedure assures a flat channel power spectrum, maximizing the channel occupation efficiency.

![Figure 5: Pseudo-random sequence generator. It uses a 16 bits shift register [6], which is initialized with the hexadecimal value $F180h$ at the beginning of each Field [1].](image)

The Reed-Solomon Encoder independently processes each one of the 187 bytes sequences stored in the 58344 bytes buffer. At the end of each one of them, the Reed-Solomon Encoder adds a 20 bytes sequence and stores the resulting 207 bytes sequence in a buffer with a capacity of 64584 bytes (for implementation purposes, this buffer is just an extension of the previous one). This capacity corresponds to 312 sequences, each one with 207 bytes, which is equivalent to a complete 8-VSB Field at the Encoding Stage output. In the context of Reed-Solomon coding, each 187 bytes sequence is called Message and each 207 bytes sequence is called Codeword. The 20 bytes added at the end of each Message is called Parity, which is the redundant information added to the Message for error correction purposes.

There is a univocal mapping between each Message and its Parity [10]. Therefore, if one Codeword is received in error due to signal degradation in the channel, the error can be detected and eventually corrected in the receiver, since the decoder "knows" all possible Codewords [6].

Reed-Solomon codes are a sub-class of the block code class called BCH (Bose-Chaudhuri-Hocquenghem) [5]. A Reed-Solomon code $RS(n,k)$ is characterized by $n$ – the number of symbols per Codeword, by $k$ – the number of symbols per Message and by $m$ – the number of bits per symbol [10]. Thus, the 8-VSB-ATSC encoding/decoding stage utilizes an $RS(207,187)$ block code with $m=8$ bits (1-byte) per symbol.

An $RS(n,k)$ code is considered a systematic code [5], because the Message symbols are not transformed – the $n-k$ parity symbols are just appended to the Message. The Code Rate, which measures the information transmission efficiency, is $k/n$ for an $RS(n,k)$ code. The maximum number of symbols in a Codeword received in error which an $RS(n,k)$ code is able to correct is given by $(n-k-1)/2$ for $(n-k)$ odd, and by $(n-k)/2$ for $(n-k)$ even. Therefore, an $RS(207,187)$ code with $m=8$ is able to correct up to 10 bytes (10 symbols) in a Codeword received in error, no matter which of the 207 bytes are wrongly received.

For a block code whose symbols are just bits ($m=1$), as is the case for the binary Hamming codes [5], if the number of bits received in error exceeds the code correction capacity, the received Codeword is summarily corrected to a different one from that which was originally transmitted. That does not happen with a Reed-Solomon code. For an $RS(207,187)$ – $m=8$ code, if the number of error exceeds 10 bytes, the received Codeword will not be corrected. However, the Berlekamp error correction algorithm [6] used in this work, even so, labels the received Codeword as uncorrectable.

The great advantage of the Reed-Solomon code becomes apparent when the information to be decoded stems from a continuous stream of bits, with no block delimitation, such as in the case of a bit stream generated by a Viterbi Decoder [5][6][9]. In this situation, the error correction capacity of the concatenated system Viterbi/Reed-Solomon is even higher because the Reed-Solomon code is able to correct the symbols as a whole, independently of which bits in the symbols received in error have been corrupted.

The 64584 bytes buffer at the Reed-Solomon encoder output, when totally filled with the 312 Codewords of 207 bytes, is submitted to a "shuffling" process of its bytes by means of the Interleaver action. Two kind of Interleavers are utilized in the ATSC 8-VSB system. The first one is a Convolutional Interleaver [6][7], shown in Figure 6, which shuffles those bytes associated with symbols that may pertain to distinct Data Segments along one 8-VSB Field. The second one is a Block Interleaver [6][7], which shuffles bytes associated with symbols that pertain to the same Data Segment. Although both Interleavers are located prior to the Convolutional Encoder in Figure 1, the data flow between them cannot be described in such a simplistic way.

![Figure 6: Convolutional Interleaver, composed of 52 banks of byte shift registers. This interleaver introduces a 10608 bytes delay, called End-To-End delay [2], which is compensated in the receiver. The position of the rotating switches with respect to the bank order follows the](image)
sequence $[0, 1, ..., 31, 0, ...]$. The position 0 is synchronized with the first data byte of the Field.

Figure 7 shows the data flow between the Convolutional Encoder and the Block Interleaver. Actually, the Convolutional Encoder is composed by a group of 12 parallel individual encoders, each one with the architecture shown in Figure 8.

Figure 7: Data flow diagram between the Block Interleaver and the group of 12 parallel Convolutional Encoders.

Figure 8: Internal diagram of one of the 12 identical Convolutional Encoders shown in Figure 7. A 2 bits shift register with feedback [5][6] composes each encoder. For each 2 incoming bits respectively at the input nodes ($X_2, X_1$), the encoder generates 3 bits at the respective output nodes ($Z_2, Z_1, Z_0$). The bit assigned to the input node $X_2$ is directly applied to output node $Z_2$. During the transitional period from the NTSC system to the 8-VSB system, the nodes $X_2$ and $Z_2$ are to be interconnected via the NTSC Interference Filter Pre-Encoder [1][2].

Based on Figures 6, 7 and 8, the operation of the Interleaver in the Figure 1 can be described as follows. The Convolutional Interleaver processes the 312 RS Codewords applied to its input and stores the result in its output buffer (Figure 6). Depending upon which segment $\Psi$ the symbol belongs (Figure 7 - 8-VSB symbol matrix), the Pre Block Interleaver selects the couple of bits $(b^u, b^v)$, $u,v = \{0, 1, ..., 7\}$, at the byte $B$ of the Convolutional Interleaver output buffer, $B = \{0, 1, ..., 64583\}$, and assigns $(b^u, b^v)$ to the input nodes $(X_2, X_1)$ of the Convolutional Encoder $T$, $T = \{0, 1, ..., 11\}$. Then, the encoder $T$ yields the trio of bits $(Z_2, Z_1, Z_0)$ at its output, which is assigned by the Block Interleaver to $M_{2, \Psi}$, where $M$ represents the 8-VSB symbol matrix of Figure 7.

The analytic relationship that defines $B$ as a function of $\Omega$ and $\Psi$ is determined by equations (1) to (6):

$$B = 12 \beta + (\lambda \mod 48) \mod 12 + \Delta$$  

$$\beta = \left\lfloor \frac{\lambda}{48} \right\rfloor$$  

$$\lambda = 828 \Omega + \Psi,$$

where the operator $\left\lfloor . \right\rfloor$ returns the integer part of the argument, the operator $p \mod q$ returns the remainder of $p/q$, and

$$t_1 = g((48\beta) \mod 828, 1), \quad \text{if} \quad t_1 \neq -1$$

$$t_2 = g((48\beta) \mod 828, 2), \quad \text{if} \quad t_2 \neq -1$$

$$t_3 = g((48\beta) \mod 828, 3), \quad \text{if} \quad t_3 \neq -1$$

$$-1, \quad \text{if} \quad \text{none of the above conditionals is true}$$

and

$$g(t, n) = \begin{cases} p, & \text{if} \quad (12p + l) \mod 828 = 0 \\ -1, & \text{if} \quad (12p + l) \mod 828 \neq 0 \end{cases}$$

The analytic relationship that defines $T$ as a function of $\Omega$ and $\Psi$ is determined by equations (7) and (8):

$$T = S_{[2 \mod 4] \lfloor \lambda \mod 12 \rfloor},$$

where $\lambda$ is defined by equation (3) and $S$ is the $3 \times 12$ matrix given by

$$S = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 8 & 9 & 10 & 11 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix}$$

The analytic relationship that defines $u$ and $v$ as a function of $\Omega$ and $\Psi$ is determined by equations (9) to (11):
\[
v = \Phi_{\lfloor (\lambda \mod 48)/12 \rfloor}
\]
(10)

where \(\lambda\) is defined by equation (3) and \(\Phi\) is the 4x2 matrix given by

\[
\Phi = \begin{bmatrix}
7 & 6 \\
5 & 4 \\
3 & 2 \\
1 & 0
\end{bmatrix}
\]
(11)

For instance, in order to generate the symbol \(Y=12\) of the segment \(\Omega=2\) in the 8-VSB symbol matrix \(M\), the Pre Block Interleaver selects the couple of bits \((b, b')\), \(u = 1, v = 0\), at the byte \(B = 412\) of the Convolutional Interleaver output buffer, and assigns \((b, b')\) to the input nodes \((X, X')\) of the Convolutional Encoder \(T=8\). Then, the encoder \#8 yields the trio of bits \((Z_2, Z_1, Z_0)\) at its output, which is assigned by the Block Interleaver to \(M_{2,12}\).

In order to optimize the distance properties of the 8-VSB symbol constellation [8], each element \((Z_2, Z_1, Z_0)\) of \(M\) is transformed into a new value given by Table 1.

| Table 1: \((Z_2, Z_1, Z_0)\) \(\Rightarrow\) 8-VSB Level mapping |
|-----------------|-----------------|-----------------|-----------------|
| \(Z_2\) | \(Z_1\) | \(Z_0\) | 8-VSB Level |
| 0    | 0    | 0    | 7             |
| 0    | 0    | 1    | -5            |
| 0    | 1    | 0    | -3            |
| 0    | 1    | 1    | -1            |
| 1    | 0    | 0    | 1             |
| 1    | 0    | 1    | 3             |
| 1    | 1    | 0    | 5             |
| 1    | 1    | 1    | 7             |

Once transformed, matrix \(M\) is sent to the Multiplexer as a one-dimensional vector \(V\) given by

\[
V_i = M_{\lfloor i/828 \rfloor \mod 828}, \quad i = 0, \ldots, 258335
\]
(12)

where the symbols \(V_0\) and \(V_{258335}\) are respectively the first and the last symbols sent to the Multiplexer. Then, the Multiplexer inserts the sync sequences as described before and generates a full 8-VSB symbol Field.

### III. ATSC Decoder

In this section we analyze the 8-VSB receiver Decoding Stage, shown in Figure 2.

In this work, the Phase Tracker output is stored in a vector \(V\) of 258336 8-VSB symbols. As in the transmitter case, the receiver 8-VSB symbol matrix \(M\) (Figure 9) is obtained according to equation (12). Then, each element of \(M\) is transformed into the trio of bits \((Z_2, Z_1, Z_0)\) by means of the inverse mapping of Table 1. The group of 12 Viterbi Decoders (Figure 10) [5] plus the Block Interleaver decode the 312 x 828 symbols in the 8-VSB symbol matrix \(M\) and store them into the Convolutional Deinterleaver input buffer (64584 bytes), as shown in Figure 9.

The symbol \(Y\) of the segment \(\Omega\) in the 8-VSB symbol matrix \(M\) is selected by the Pre Block Deinterleaver, which assigns the trio of bits \((Z_2, Z_1, Z_0)\) to the input nodes of the Viterbi Decoder \(T\). The Block Deinterleaver assigns the Viterbi Decoder \(T\) output nodes value \((X, X')\) to the couple of bits \((b, b')\) at the byte \(B\) of the Convolutional Deinterleaver input buffer. The analytic relationship that defines \(B, T, u\) and \(v\) as a function of \(\Omega\) and \(Y\) is determined by equations (1) to (11).

![Figure 9: Receiver Block Deinterleaver and the group of 12 parallel Viterbi Decoders.](image)

The Convolutional Deinterleaver is identical to the interleaver shown in Figure 6, except that the rotating switches position follows the sequence [51, 50, ..., 0, 51, ...]. The initial position 51 is synchronized with the first byte of the input buffer. The Convolutional Deinterleaver output buffer is a queue with a capacity of 2x64584 bytes, i.e., the storage capacity for 2 complete data Fields, as shown in Figure 11. The queue output is taken at 10608 bytes with respect to the queue initial position so that the delay introduced by the transmitter Convolutional Interleaver End-To-End delay is compensated.

Then, the Reed-Solomon Decoder decodes and corrects (up to 10 bytes received in error) the 312 RS Codewords that stem from the Convolutional Deinterleaver output buffer, yielding 312 Messages of 187 bytes each. Finally, the Derandomizer performs the bitwise exclusive-or logical operation between each buffer byte and each output byte from the pseudo-random sequence generator shown in Figure 5. Each one of the 187 bytes Messages is sent to the subsequent MPEG-2 decoder transport layer.
Viterbi Decoder:

\[ Z_1 \rightarrow Z_2 \rightarrow X_2 \rightarrow X_1 \]

State transition diagram:

- \( Z_1 \rightarrow Z_2 \rightarrow X_2 \rightarrow X_1 \)
- \( 0/00 \rightarrow 0/01 \rightarrow 0/10 \rightarrow 0/11 \)
- \( 1/01 \rightarrow 1/11 \)

Figure 10: Internal diagram of one of the 12 identical Viterbi Decoders [3] shown in Figure 9. It is shown the state transition diagram that defines the trellis allowed paths. This state transition diagram is associated with the Convolutional Encoder shown in Figure 8 [6].

Field n (64584 data bytes) from the Convolutional Decoder
Field n+1 (64584 data bytes) from the Convolutional Decoder

![Diagram](image)

Figure 11: 10608 bytes compensation for the delay inserted by the Convolutional Interleaver in the transmitter.

The concatenated Viterbi/Reed-Solomon coding with an intermediate Interleaver exhibits better error correction capability (under additive Gaussian noise channel) than any other error correcting system of similar complexity [6]. This is basically due to the fact that the Viterbi and the Reed-Solomon decoding characteristics are approximately complementary. For example, due to the multi-bit nature of its symbols, a Reed-Solomon code achieves maximum decoding efficiency when the errors to be corrected occur in short bursts of bits. However, its efficiency is reduced when the bit errors occur with no correlation in time. For the concatenated Viterbi/Reed-Solomon decoding, this drawback is compensated by the Viterbi Decoder, which is quite suited to this kind of error behavior. On the other hand, the Viterbi Decoder also fails when its error correction capacity is exceeded, generating a long error sequence at its output [6]. This sequence of errors is, in general, much longer than the short bursts of bits that maximize the Reed-Solomon decoding efficiency. In some extreme cases, it could exceed the maximum number of correctable RS symbols. Therefore, when applying the Viterbi Decoder output directly to the Reed-Solomon Decoder input, the latter will suffer an efficiency decrease (or even will fail) because the Codewords at its input present a high correlation in time between symbols received in error [13]. A Deinterleaver inserted between the two decoders, which "shuffles" the Reed-Solomon input sequences, is an efficient solution to this problem. Thus, any eventual correlation between the symbols received in error that could stem from a Viterbi Decoder failure is greatly minimized.

IV - Decision Feedback Equalizers

A very important issue in 8-VSB receivers is equalization. In the ATSC standard any compensation for channel impairments caused by multipath propagation is left mostly to the equalizer. The Decision Feedback Equalizer (DFE) [15] is an efficient structure that has been successfully used in 8-VSB receivers. A prosaic implementation of a DFE would follow the block diagram of Figure 12. More advanced versions would take advantage of certain a priori channel information or improved decision devices that uses decoded output reliability information [16].

![Diagram](image)
Table 2: Channel B used in the laboratory tests in Brazil.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Relative Amplitude</th>
<th>Amplitude dB</th>
<th>Time Delay (us)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Echo 1</td>
<td>0.2512</td>
<td>-12.0</td>
<td>0.30</td>
</tr>
<tr>
<td>Echo 2</td>
<td>0.6310</td>
<td>-4.0</td>
<td>3.50</td>
</tr>
<tr>
<td>Echo 3</td>
<td>0.4467</td>
<td>-7.0</td>
<td>4.40</td>
</tr>
<tr>
<td>Echo 4</td>
<td>0.1778</td>
<td>-15.0</td>
<td>9.50</td>
</tr>
<tr>
<td>Echo 5</td>
<td>0.0794</td>
<td>-22.0</td>
<td>12.70</td>
</tr>
</tbody>
</table>

V. Experimental Results

In this section we analyze the performance of the cascade operation of the ATSC 8-VSB Encoding Stage and Decoding Stage, as well as the DFE equalizer performance for channel B. Initially, an ATSC simulator was implemented in which the Encoding Stage output (Figure 1) is connected to the Decoding Stage input (Figure 2) through an additive Gaussian noise generator. Multipath channel models were then included to obtain a complete baseband equivalent simulator.

Channel Coding Performance

The 8-VSB performance for the Gaussian channel is shown in Figure 12, which presents the Bit Error Rate (BER) after Reed-Solomon decoding as a function of carrier to noise (C/N) ratio. These results in part confirm the proper performance of the implemented ATSC simulator.

In order to obtain the BER, the simulator follows the heuristic proposed by Odenwalder [14], which is suited to the Viterbi/Reed-Solomon concatenated coding. The simulator assumes the following events when the error correction capacity of the RS decoder is exceeded, i.e., a Codeword is erroneously decoded:

1. The simulator adds \((n-k)/2 = 10\) bytes \(= 80\) bits in error to the Codeword (due to the fact that the RS decoder "corrects" the Codeword to an erroneous one).
2. All bits in a byte in error are also in error.
3. All bytes in error in a Codeword occur in the range of those 187 bytes that correspond to the Message.

Notice that the ATSC 8-VSB coding stage performance is obtained at an expense of a 10608 bytes delay, which is inherent to the Convolutional Interleaver operation. Such a delay can be unacceptable for a bi-directional narrowband voice system, for instance. In such a system, the Convolutional Interleaver banks must be flushed and reinitialized at the beginning of each one of the bi-directional data streams, which prohibits a 10608 bytes delay. However, for a TDM (Time Division Multiplex) system, where the channel bit rate is quite high, or for a continuous video system as the ATSC 8-VSB, where 10608 bytes corresponds to a time interval of \(4\)ms, an interleaver End-To-End delay of 10608 bytes is perfectly admissible.

DFE Equalization Performance

The performance of the DFE equalizer for channel B (partially truncated Harbor Apartment channel) has been assessed by using the Decision Directed Least Mean Square (LMS-DD) algorithm. Two different adaptation schemes were used, one with common selection of adaptation parameters (LMS-DD), and the other with added intelligence (LMS-DD1).

The completely closed eye diagram before equalization is depicted in Figure 13 for C/N = 31 dB. In Figures 14 and 15 it can be seen that both algorithms are able to equalize the channel. The evolution of the Mean Square Error for the LMS-DD and LMS-DD1 algorithms is shown in Figure 16, as a function of Field Number. It is clear from this figure that the LMS-DD scheme converges in approximately 4 fields, whereas the LMS-DD1 algorithm converges in only two fields. Also, the residual MSE for the latter is somewhat smaller.

It happens that the signal to noise ratio C/N of about 31 dB, is the value that attains the Threshold of Visibility (TOV) of
3x10^-9 defined for the ATSC standard. It was observed that the cliff effect in the BER versus C/N curve for this channel is extremely abrupt, a characteristic that is also observed in DVB receivers.

It should be noticed that the signal to noise penalty of about 15 dB, as compared with the Gaussian channel, is due to noise enhancement in the forward filter and error propagation in the backward filter of the DFE equalizer. If more sophisticated equalizers were used, this penalty would certainly be much smaller. For example, a fractionally spaced equalizer with error propagation mitigation, which can be obtained from decoder output feedback to the decision device, would certainly outperform the present approach.

V. Conclusions
This work analyzes and assesses the implementation of an ATSC 8-VSB simulator for DTV broadcast. The simulation results confirm that the channel coding stage of the ATSC 8-VSB system is robust to additive Gaussian noise. They also show that the 8-VSB receivers are also quite robust to impulsive noise.

It is important to point out that this work assesses only the performance of the ATSC 8-VSB baseband channel coding and equalization stages. In order to assess the whole ATSC 8-VSB system performance, the RF stages should also be considered. However, since the 8-VSB RF stages seem quite robust, assuming perfect receiver synchronization is not likely to make much difference in the final results.

Acknowledgments
The authors would like to thank Profs. José G. Chiquito and João M. T. Romano, from the Department of Communications of FEEC-UNICAMP, for stimulating discussions during the course of this work.
References