Matlab/Simulink wireless HDMI model and simulation

Ruben Cabral¹

¹Instituto Superior Técnico, Lisboa, Portugal

The growing availability of High Definition video contents is driving the development of technologies capable of multi-gigabit per second throughput, like High-Definition Multimedia Interface and Display Port. As wireless communication systems have become common in everyday life, video transmission is also pushing wireless technologies.

The 60 GHz band is extremely appealing due to the huge continuous bandwidth available, up to 9 GHz of continuous available bandwidth, which provide enough throughput to transmit uncompressed FullHD video. Several standards have been developed to take advantage of this frequency band like ECMA-387, IEEE 802.15.3c and IEEE 802.11ad, and proprietary specifications like WirelessHD and Wireless Gigabit.

This work's goal is to model in Matlab/Simulink a transceiver for assisting the hardware implementation, in FPGA, of an uncompressed HD video transceiver.

Index Terms— Wireless uncompressed video transmission, 802.15.3c, AV HRP, millimeter-wave, Simulink modeling.

I. INTRODUCTION

The recent increase in High Definition (HD) video contents has brought the need to develop communication standards capable of multi-gigabit per second throughput, like High-Definition Multimedia Interface (HDMI) [1] and Display Port [2]. Consumer electronics (CE) users also want the flexibility provided by wireless connections to set up and reconfigure multimedia systems, and to eliminate wired connections required by HD multimedia systems, like home theatres. Driven by these needs, the CE industry is developing formats capable of delivering uncompressed video, at the necessary data rates, via wireless connections. Simultaneously, 802.11 devices have become ubiquitous and, in the latest specification (802.11n [21]), are capable of net data rates up to 600Mbps. This data rate is still insufficient for streaming uncompressed HD video or for transferring HD contents, like a HD film, as fast as would be desirable. Uncompressed HD video transmission requires very high bit rates, up to 3 Gbps for Full HD video.

Uncompressed HD video transmission avoids compression at the transmitter and decompression at the receiver, therefore providing: lower latency which permits timing sensitive applications like multimedia applications and gaming; higher interoperability between devices, because, unlike compressed video transmission, the receiver device just displays the video content and does not need to be able to decoded the video codec; and no degradation in picture quality due to compression losses in the transmission.

To address these needs several specifications have been created and several other are still being developed. The development efforts have been focused in basically in two frequency bands: the 2.4 to 10 GHz and 60 GHz bands. The former frequency band is used by specifications like Wireless Home Digital Interface (WHDI) [11] and Intel’s Wireless Display (WiDi) [12], WirelessUSB (WUSB) [9], and a draft version of IEEE 802.11ac [6], while the later is used by IEEE 802.15.3c [3], WirelessHD (WiHD) [4], ECMA-387 [5], Wireless Gigabit (WiGig) [8] and a draft version of IEEE 802.11ad [7]. Technologies using the lower frequency band have better range and are not limited to in room transmissions. The propagation characteristic of the 60 GHz band limits the transmissions to a maximum distance of 10 meters and to in room transmissions. However, the 60 GHz band has more available bandwidth and therefore higher bit rates can be achieved using this frequency band.

The technologies specifically designed for wireless video transmission, like WHDI, IEEE 802.15.3c, ECMA-387, WiHD, IEEE 802.11.ad (draft) and WiGig, organize the source devices (transmitters) and the sink devices (receivers) into a wireless video network that allows for example:

- Point to point uncompressed video transmission.
- Point to multi point uncompressed video transmission.
- Office desktop, allowing to wirelessly transmitting a laptop/computer desktop to a HD display.

As of the time of this writing there are some WHDI, WiGig, WiHD and IEEE 802.15.3c compliant products commercially available ranging from external HDMI adapters to built-in solutions on HD displays and high-end laptops. These consumer electronics devices are implemented with ASIC because it significantly reduces costs per unit, significantly saves power, is smaller and works in higher speeds compared to FPGA. However, the costs involved in creating an ASIC are very high and are only economically viable once the devices starts to be massively produced. For prototyping, FPGA technology offers more flexibility to test, analyze and correct the hardware implementation. A vast majority of designs intended for ASICs are originally prototyped in an FPGA. Prototyping before definitive specifications creates conditions to reduce time-to-market.

Even though developing hardware using FPGA technology is faster than developing using ASIC, it is still a time consuming process and a lot of development has to be done before a full system test can be executed. System modelling and simulation can help reduce the development phase. By using already developed blocks and high level programming languages the development time of the hardware
prototype can be greatly reduced by enabling an early detection of design problems, helping determining block specifications, producing and validating system wide tests as well as individual block tests.

Our goal is to model and simulate an OFDM base band transceiver unit capable of achieving throughputs high enough to transmit uncompressed HD video. The model will aid the future development of a hardware prototype using FPGA technology by generating test vectors for each transceiver block and validating the results.

To achieve the main objective, specific objectives are defined:
- To analyse the available standards for wireless uncompressed HD video transmission.
- To choose an appropriate standard to use.
- To model the transmitter, receiver and channel.
- To define the numerical representation of the complex coefficients generated by the system mapper and IFFT blocks.
- It will also be used to determine the specifications of the hardware blocks, specially the individual blocks in which the implementation details are not addressed by the respective standard, e.g., the required numeric precision for the IFFT and FFT blocks.

To achieve the main goal we defined several intermediate goals: determining the requirements for transmitting uncompressed HD video and selecting a suited specification for the task.

The preliminary study conducted to determine the bandwidth requirements for wireless transmission of uncompressed HD video and audio, the current capabilities and applications of wired HD video/audio interfaces, and the current technical solutions for wireless transmission of video/audio is presented on Chapter 2.

The selected specification is described in detail in Section 3 and in the Matlab/Simulink transceiver model is presented in Section 4. The results and conclusions are presented in Sections 5 and 6 respectively

II. WIRELESS HD VIDEO TRANSMISSION

This section is dedicated to research the requirements for transmitting uncompressed video/audio; to identify the existing wired and wireless specifications for transmitting HD video; and to select the most appropriate specification for wireless transmission of uncompressed HD video/audio.

A. Digital video and audio signals

Characterization of digital video and audio signals is important to assess the net data rate requirements. A digital video signal data rate is defined by: resolution, i.e., the total number of pixels of each image, normally referred as number of horizontal pixels by vertical pixels on the screen; color depth, i.e., the number of bits used to represent each of the three colors of a pixel; refresh rate, i.e., number of times per second the image is completely reconstructed on the screen; progressive or interlaced formats, i.e., the way lines of an image are displayed in the refreshing cycles, progressive formats display all the lines on all the refresh cycles, but interleaved ones display even and the odd lines in alternated refresh cycles.

Digital audio data is defined by; number of audio channels; sampling rate; number of bits used to quantify each audio sample.

Currently available HD video formats use designations as “720i”, “720p”, “1080i”, and “1080p”. These terms indicate the number of lines and the display method used. The images used in HD video formats have a 16:9 (image length : image width) aspect ratio resulting in wider images than the conventional 4:3 aspect ratio used in Standard Definition (SD) video. The number of pixels, np, in each HD image can be calculated from (1) using the number of lines, nl, mentioned by the video format designation:

\[ np = (16/9) \times nl^2 \quad (1) \]

From (1), it is possible to calculate that 720p and 720i images are formed by 1280x720 pixels; 1080i and 1080p images are formed by 1920x1080 pixels.

The bit rate required to transmit “Full HD” video, vbr, with progressive display can be calculated from (2), where np is the number of pixels, ncchannels is the number of color channels, cdepth is the number of bits used to represent each color and rfreg is the display refresh frequency. The audio bit rate, abr, can be calculated from (3), where nac is the number of audio channels used, srate is the sampling rate and sdepth is the number of bits used per sample.

\[ vbr = np \times ncchannels \times cdepth \times rfreg \quad (2) \]
\[ abr = nac \times srate \times sdepth \quad (3) \]

The net bit rate for Full HD video and audio is shown in (4) and (5).

\[ vbr_{fullHD} = 1920 \times 1080 \times 3 \times 8 \times 60 = 2.99 \text{ Gbps} \quad (4) \]
\[ abr_{fullHD} = 8 \times 192 \times 24 = 36.8 \text{ Mbps} \quad (5) \]

B. Wired HD video standards

One of the most widespread HD interfaces is HDMI (Home Digital Multimedia Interface). It was introduced in December 2002 and was designed to transmit uncompressed video, up to 8 channels of audio and control data. HDMI is electrically compatible with the DVI (Digital Visual Interface) [24], introduced in 1999 as a digital alternative to the analogue VGA (Video Graphics Array) interface. The latest version, HDMI 1.4 [1], supports video resolutions up to 4096x2160 at 24 Hz or 3840x2160 at 30Hz with 12 bits for each color channel, requiring a maximum bit rate of approximately 7.64 Gbps and 8.96 Gbps respectively.

Display Port is a HD video and audio interface presented in 2006 by Video Electronics Standards Association (VESA). In the latest specification it is capable of delivering digital video
up to resolutions of 3840x2160 at 30 Hz with 10 bit for each color channel and 8 audio channels at 192 kHz with 24 bit samples [2]. Display Port version 1.2 supports resolutions up to 2560x1600 at 120Hz and 3840x2160 at 60 Hz with 10 bits for each color channel. These new video formats demand a bit rate of up to 14.9 Gbps.

C. Wireless video transmission technologies
In this section several wireless video transmission technologies will be compared to determine the best solution to transmit uncompressed HD video. The technologies will be divided into two categories: technologies using frequencies up to 10 GHz and technologies in the 60 GHz band.

1) Under 10 GHz technology summary
Table 1 presents the maximum data rates and frequency band of the wireless technologies working below 10 GHz. At the present moment WHDI has the highest date, 3 Gbps, and is, according to its promoters, able to transmit uncompressed HD video. Future data rates in the frequency band below 6 GHz will reach approximately 7 Gbps with the completion of IEEE 802.11ac specification.

<table>
<thead>
<tr>
<th>Data rate (Mbps)</th>
<th>WUSB</th>
<th>WHDI</th>
<th>WDi</th>
<th>IEEE 802.11ac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency Band (GHz)</td>
<td>3.1 GHz to 10.6 GHz</td>
<td>5 GHz</td>
<td>2.4 GHz</td>
<td>&lt; 6 GHz</td>
</tr>
</tbody>
</table>

2) 60 GHz technology summary
Table 2 presents a summary of the 60 GHz technologies presented on the section. WiGig is not included in Table 2 because of the small amount of information available on this technology. It is interesting to note that all technologies, except WiHD, have single carrier modes. In spite of this, only ECMA-387 requires all devices to support single carrier modes.

ECMA-397 is the only technology with the capability of aggregating frequency channels enabling it to reach data rates in excess of 25 Gbps.

3) Conclusions on wireless HD video transmission
High Definition video and audio require high bit rates. As shown in Section 2.1, the minimum data rate for transmitting 1080p uncompressed video and audio is approximately 3 Gbps. Newer video formats, like HD 3D and 4k resolutions, require even higher dates, close to 15 Gbps.

Current wireless technologies in the 2.4-10 GHz band are unable of accommodating such high throughput. The only exception is WHDI, a proprietary specification which is able of 3 Gbps equivalent video data rates. Future version of the IEEE 802.11 standard, IEEE 802.11ac, will reach data rates of approximately 7 Gbps, using MIMO technology with 7 spatial streams and higher order carrier modulation constellations, but will also be incapable of supporting the newer video formats.

### Table 2: Comparison of the 60 GHz technologies

is available about it.

Table 3: Comparison criteria

<table>
<thead>
<tr>
<th></th>
<th>WUSB</th>
<th>WIDI</th>
<th>802.11ac (draft)</th>
<th>802.15.3c</th>
<th>WHDI</th>
<th>ECMA-387</th>
<th>802.11ad (draft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Rate (Gbps)</td>
<td>0.48</td>
<td>3</td>
<td>6.9</td>
<td>5.39</td>
<td>3.8</td>
<td>6.35</td>
<td>6.8</td>
</tr>
<tr>
<td>International Standard</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Specifications available</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes (draft)</td>
<td>No3</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Specific video modes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Taking into consideration the mentioned criteria, ECMA-387 is the more appropriate standard on which to base the implementation of a wireless uncompressed HD video transmission system. However, if the implemented system is to be based on OFDM modulation, then ECMA-387 would no longer be the most suitable standard because it requires that all devices support a SCBT mode resulting in additional required hardware and the maximum throughput of the OFDM modes is similar to IEEE 802.15.3 HSI and AV HRP modes and inferior to IEEE 802.11ad OFDM modes. In this context IEEE 802.15.3c AV mode is the best alternative because the specification is already published and is specifically designed for transmission of HD video; the IEEE802.11ad specification was discarded because it is still being developed.

III. IEEE 802.15.3c AV MODE

The HRP mode was designed for uncompressed HD video transmission. Table 4 shows the data rates and coding for HRP mode. Modes 0 to 2 use EEP on both MSB and LSB; modes 3 and 4 use UEP to provide different protection levels to the MSB and LSB, thus providing a more appropriate protection for streaming video contents.

The OFDM modulation is performed by a 512 point IFFT/FFT and the FFT period must not exceed 202 ns.

Each HRP channel can include three LRP channels, but only one can be used at a time. All the LRP modes use BPSK modulation and the data rates vary from 2.5 Mbps to 10.2 Mbps.

The focus will be on the HRP mode because it’s the actual mode used to transmit video and has more demanding and challenging requirements than the LRP.

A. HRP PHY frame structure

Each MAC frame received at the HRP PHY layer is converted into a HRP data unit (HRPDU). The HRP PHY layer adds the HRP preamble, the HRP header and the header check sequence (HCS) to the Extended MAC header. The MAC frame body is sent as HRPDU payload. The HRPDU format is shown in Figure 1.

Table 4: HRP data rates and coding [3]

<table>
<thead>
<tr>
<th>HRP mode index</th>
<th>Coding mode</th>
<th>Modulation</th>
<th>Code rate</th>
<th>Raw data rate (Gbps/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>EEP</td>
<td>QPSK</td>
<td>1/3</td>
<td>0.952</td>
</tr>
<tr>
<td>1</td>
<td>QPSK</td>
<td>2/3</td>
<td></td>
<td>1.904</td>
</tr>
<tr>
<td>2</td>
<td>QPSK</td>
<td>4/7</td>
<td>4/5</td>
<td>3.807</td>
</tr>
<tr>
<td>3</td>
<td>QPSK</td>
<td>4/7</td>
<td>4/5</td>
<td>3.804</td>
</tr>
<tr>
<td>4</td>
<td>MSB-only</td>
<td>QPSK</td>
<td>1/3</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td>Retransmission</td>
<td>QPSK</td>
<td>2/3</td>
<td>1.904</td>
</tr>
</tbody>
</table>

Figure 1: HRPDU frame format

The HRP preamble, HRP header, MAC header and HCS fields are transmitted using HRP mode 0. The HRP payload is divided into up to seven blocks, called HRP sub-frames. Each sub-frame can use a different HRP mode and has a maximum length of $2^{20} \times (1.048.576)$ octets, allowing a maximum HRPDU frame size of approximately 7MB.

IV. IEEE 802.15.3c AV HRP MODE TRANSCEIVER SIMULINK MODEL

The HRP transmitter is presented in Figure 2. The incoming octet stream undergoes some initial processing in the “Frame Conditioning” block, like splitting and scrambling the MSBs and the LSBs into upper and lower branches octet streams. In the “Frame Encoding” block both streams are encoded with a Reed-Solomon code, interleaved and divided into 4 parallel streams that are convolutionally encoded and punctured. The multiplexer block interleaves and serializes all eight parallel bit streams. In he OFDM block the bit stream is converted to a complex coefficients stream by the system mapper and pilot, DC and null coefficients are inserted to form an OFDM symbol. The tone interleaver shuffles the order of the all coefficients and the IFFT block converts the OFDM symbol to the time domain.

On the receiver side, the data received by the radio block is delivered to the receiver input, labelled “receiver in”, and processed by the “OFDM demodulation” block, where the
time signal is converted to the frequency domain, by the FFT block, the pilots, DC and null carriers are removed and the system demapper creates a bit stream from the receiver data coefficients. The demultiplexer block deinterleaves and separates the bit stream into eight parallel streams that are decoded by eight Viterbi decoders, deinterleaved and decoded again by 2 Reed-Solomon decoders. In the final block the two bit octet streams are de-scrambled and merged into a unique octet stream and provided to the upper layer.

A. Transceiver blocks

1) Transmitter and receiver controller
The configuration of all the transmitter blocks and sub blocks is performed by the transmitter controller block. It decodes the information on the HRPDU header and configures the system accordingly.

The receiver controller block is responsible for configuring all the blocks on the receiver model. It decodes the transmitted HRPDU header and generates all the signals to configure appropriately all the remaining blocks.

2) Frame processing blocks
The frame conditioning block is formed by three sub blocks: the bit stuffer block, the splitter block and the scrambler block. The bit stuffer is responsible for ensuring that the transmitter produces an integer number of OFDM symbols; the splitter only processes payload data; separates and reorders each incoming octet in the 4 MSBs and 4 LSBs; the scrambler scrambles all data bits using a polynomial scrambler.

The reciprocal block in the receiver is the also called frame conditioning block and its main blocks are the destuff bits block; the desplitter block and the descrambler block.

3) Frame encoding and decoding blocks
The frame encoding block has three main sub blocks: incoming octets are encoded by the Reed-Solomon encoding block, and then the outer interleaver block groups, reorders and distributes the data by the eight convolutional encoders that make up the convolutional encoder block.

On the receiver side, the data undergoes the reverse processing: first the eight Viterbi decoders recover the data, which is then deinterleaved and regrouped by the outer deinterleaver and then decoded by the Reed-Solomon decoding block.

4) Bit interleaver and deinterleaver blocks
The bit streams generated simultaneously by the convolutional encoders are serialized and interleaved in the multiplexer/bit interleaver block.

Similarly to the multiplexer/bit interleaver block the main blocks are the header demultiplexer and data demultiplexer. This block deinterleave the data stream received from the system demapper and separate it into parallel bit streams.

5) OFDM modulator and demodulator blocks
The OFDM modulator includes the following blocks: system mapper, preamble, pilots insert blocks, tone interleaver, the IFFT block and some additional routing blocks. The serial data stream is mapped to complex coefficients by the system mapper and the insert pilots block adds the null, dc and pilots. The carriers are interleaved by the tone interleaver block and the IFFT block converts the OFDM symbol to the time domain and the signal is sent to the radio block. Each OFDM symbol has 512 sub carriers: 336
modulated by data, 157 null carriers, 16 pilot carriers and 3 DC carriers.

The OFDM demodulator block performs the reverse processing: the received signal is converted to the frequency domain by the FFT block, the complex coefficients are deinterleaved and the pilots, null and DC coefficients are separated from the data coefficients and the system demapper block converts the complex coefficients into a bit stream.

**B. Complete Matlab/Simulink model**

To simulate the transceiver operation two additional blocks were introduced: the HRPDU frame generator block and a channel block. The HRPDU frame generator block creates the data that will be transmitted and the channel block simulates several transmission channel models.

1) **HRPDU frame generator**

The frame generator block is not enclosed in transmitter block and is responsible for generating the data to be transmitted. This block reproduces the frame structure described in Figure 1. The parameters on the HRP PHY control and sub frames headers can be configured through the HRPDU frame settings block.

2) **Channel block**

The channel block implements three different channels:

- Ideal channel;
- Random noise channel, which adds random noise to the amplitude and phase of the sub carriers;
- Additive White Gaussian Noise channel;
- Configurable attenuation and phase channel, which allows defining the attenuation and phase values for each individual sub carrier.

3) **Model settings**

The model top level view has four different settings blocks that allow changing configuration parameters in the Matlab/Simulink model.

The HRPDU frame settings block allows the configuration of the following parameters:

- The number of HRPDU frames to generate;
- The scrambler seed to use in the data scrambler;
- The mode and length of each payload sub-frame.

The transmitter settings block allows modifying the numeric representation of the complex coefficients used in the IFFT block the between the default Matlab double precision and customizable fixed-point representation.

The receiver settings block enables configuring:

- The number of soft bits to use in the system demapper, demultiplexer and Viterbi decoders blocks;
- The numerical representation used in the FFT block: either double precision or customizable fixed point representation.

The channel settings block allows choosing and configuring the type of channel to use in model.

- Ideal channel;
- Random noise channel;
- Additive White Noise Channel;
- Configurable attenuation and phase channel, which allows defining the attenuation and phase values for each individual sub carrier.

**V. TESTS AND RESULTS**

In order to verify the correct function of the model, a series of tests were conducted: a HRPDU frame configured as shown in Table 5 was transmitted over an ideal channel, the Random noise channel and the AWGN channel. The maximum available numeric precision was used in the IFFT and FFT blocks and the system demapper used sixteen levels (four soft bits). The frame structure was chosen to provide an overview of the performance of the main HRP modes when transmitted over noisy channels. The random noise channel was configured to have 3 dB of maximum noise amplitude and 0.2 radians of maximum phase variation, and the AWGN channel was configured to have a SNR of 43 dB.

The test was executed three times and the average bit error rate (BER) and the average number of bit errors at the output of the system demapper (1) and at the output of the receiver (2) are presented in Table 6. The BER was measured at the output of the system demapper and at the overall system output to measure the effect of the decoding block on the performance of the system.

**Table 5: Test HRPDU frame structure**

<table>
<thead>
<tr>
<th>Subframe</th>
<th>Mode</th>
<th>Length (octets)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subframe 1</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>Subframe 2</td>
<td>1</td>
<td>1000</td>
</tr>
<tr>
<td>Subframe 3</td>
<td>2</td>
<td>1000</td>
</tr>
<tr>
<td>Subframe 4</td>
<td>3</td>
<td>1000</td>
</tr>
<tr>
<td>Subframe 5</td>
<td>4</td>
<td>1000</td>
</tr>
<tr>
<td>Subframe 6</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>Subframe 7</td>
<td>1</td>
<td>1000</td>
</tr>
</tbody>
</table>

The results show that in, these conditions, modes two and four are the most sensitive to channel noise. This result was expected because 16-QAM modulation is less permissive to channel noise than QPSK and, in the case of sub-frame 5, the use of lower rate convolutional code rate. These results also show that the decoding blocks were able to correct some of the errors introduced by the channel.

To determine the minimum numerical precision to use in the IFFT block a sub-frame using mode 4 was transmitted over the ideal channel while varying the numerical representation used in the IFFT block. The numerical precision used on the FFT block and the number of soft bits remained constant and set to maximum and four respectively. These tests conditions
were chosen because they represent the worst case scenario verified in the previous test.

The results are presented in Table 7 and the chosen threshold criterion was that no errors were detected at the output of the system demapper, i.e., BER=0. This criterion guarantees that no errors are introduced in the system due to the numerical representation used on the IFFT block.

| Table 7: Determining minimum numerical precision for the IFFT block |
|-----------------|-----------------|-----------------|
| Ideal channel (BER) | Random noise channel (BER) | AWGN channel (BER) |
| (1) | (2) | (1) | (2) | (1) | (2) |
| Subframe1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Subframe2 | 0 | 0 | 0 | 0 | 0 | 0 |
| Subframe3 | 0 | 0 | 0.26% | 0.44% | 0.16% | 0.11% |
| Subframe4 | 0 | 0 | 0 | 0 | 0 | 0 |
| Subframe5 | 0 | 0 | 2.86% | 1.96% | 0.16% | 1.41% |
| Subframe6 | 0 | 0 | 0 | 0 | 0 | 0 |
| Subframe7 | 0 | 0 | 0 | 0 | 0 | 0 |

The same procedure and criterion was used to determine the minimum numerical precision required in the FFT block: the IFFT block precision was fixed at the maximum value and varying numerical precision of the FFT block. Again the subframe used mode 4 and the same Random noise channel as the previous tests.

The results are presented in Table 8. Both results indicate the fixed point representation 3.8 as the minimum in order not to introduce errors to the Viterbi decoders.

To determine the influence of the number of soft bits on the overall system BER, the numeric precision on both IFFT and FFT blocks was set to one signal bit, two integer bits and eight fractional bits; the same HRPDU sub-frame was transmitted over the random channel, which on the previous tests proved to introduce more errors than the AWGN channel, while varying the number of soft bits used.

The results are presented in Table 9 and show that the influence of using fixed point representation on the IFFT and FFT blocks can be compensated by the increase in the number of soft bits used by the system demapper and Viterbi decoders. These results show that the use of five soft bits and the minimum fixed point representation determined previously yields overall BER similar to the BER achieved with hard decoding on the system demapper and floating point representation on the IFFT and FFT blocks.

| Table 8: Determining minimum numerical precision for the FFT block |
|-----------------|-----------------|-----------------|
| Integer Bits (+1 signal bit) | Fractional bits | BER at system demapper output |
| 2 | 10 | 1.522% |
| 3 | 5 | 5.018% |
| 3 | 6 | 1.414% |
| 3 | 7 | 0.023% |
| 3 | 8 | 0 |
| 3 | 9 | 0 |
| 3 | 10 | 0 |
| 3 | 11 | 0 |

| Table 9: Influence of the number of soft bits |
|-----------------|-----------------|
| Number of soft bits/numeric precision | BER at receiver output |
| 1/Floating point | 0.85% |
| 1/ 3.8 | 8.90% |
| 2/ 3.8 | 4.56% |
| 3/ 3.8 | 2.44% |
| 4/ 3.8 | 1.41% |
| 5/ 3.8 | 0.67% |
VI. CONCLUSION

Our goal was to implement a Matlab/Simulink model for an uncompressed HD video transceiver. To accomplish this, we started by analysing the characteristics of video and audio signals and establishing the requirements for transmitting uncompressed HD video and audio. The next step was to analyse the state of the art in wireless video transmission to choose the standard to use. The analysed technologies are separated into two different frequency bands: the 2.4 GHz to 10 GHz band which is used by WHDI, WUSB, WiDi and IEEE 802.11ac; and the 60 GHz band used by IEEE 802.15.3c, ECMA-387 and IEEE 802.11ad. On the 2.4-10 GHz band only WHDI and IEEE 802.11ac fulfil the requirements for Full HD video transmission but WHDI is a proprietary specifications and IEEE 802.11ac is still in draft. Due to the very high amount of continuous bandwidth all the standards on the 60 GHz band provide enough throughput to transmit Full HD video.

We choose a standard from the 60 GHz band because the amount of bandwidth available enables all the standards to achieve higher bit rates than the 2.4 GHz - 10 GHz band standards/specifications. Among the three possible options, we choose IEEE 802.15.3c because it provided enough throughput to transmit uncompressed HD video and has a dedicated OFDM mode with a forward error protection scheme more appropriate for video transmission.

A Matlab/Simulink model was developed to simulate the end-to-end transmission system. The model includes the transmitter base band processor, the receiver base band processor, a frame generator block and several channel models. The frame generator emulates the MAC layer output and generates data frames with the structure described in [3] which can be used to perform end-to-end simulations. This is important because it enables to evaluate the behaviour of the system in the presence of channel noise, and in future developments, can also be used to evaluate the performance of the hardware blocks in the same channel conditions.

Simulations using this model permitted to determine the minimum numerical precision of the complex coefficients in the OFDM modulation/demodulation blocks. A fixed point representation with 11 bits, one for the signal, two for the integer part and eight to represent the fractional part, proved to be the minimum numerical representation in which no errors were detected at the output of the system demapper, meaning that the quantization error introduced by this numerical representation was small enough as to not interfere with the performance of the system demapper. The influence of the number of bits in each soft bit generated by the system demapper and used by the Viterbi decoder was also analysed. The simulations show that using 5 soft bits and the 3.8 fixed point representation on the transmitter and receiver OFDM blocks, yields a lower BER in the overall communication than the BER obtained when using floating point representation and hard decision in the OFDM blocks.

We expect the IFFT/FFT, Viterbi and Reed-Solomon decoders to prove the most difficult to implement in hardware because of the complex algorithms they implement and the very high amount of data that most be processed in these blocks. We also expect the system mapper/demapper and HRP multiplexer/demultiplexer to be challenging to implement in hardware because these blocks must process the bit stream resulting from the serialization of the eight parallel convolutionally coded bit streams.

The model is a useful tool to guide a future hardware implementation of a transceiver’s base band processor, because it enables the behavioural validation of the hardware blocks, and also enables evaluating the influence of different hardware implementations on the overall system performance for any hardware block. Finally the model provides a platform for testing the overall performance influence of different solutions from the specified in the IEEE 802.15.3c standard, e.g.: different block codes; different polynomial generators for the convolutional encoders/decoders; different sub-carrier modulations schemes.

The model was simulated using channels that add random noise to the transmitted signal, however, TG3c recommends, a more realistic channel model which includes the typical use case scenarios for residential and office environments. Therefore, future work includes the introduction of this model in the developed transceiver model. Also, as future work LRP mode be used. The inclusion of the LRP mode would enable the exchange of control signals between the transmitter and receiver enabling for example, the transmitter to choose the HRP mode according to the channel’s noise characteristics.

This model was developed as an initial step for the hardware implementation of the transceiver. Therefore, the work will continue: defining hardware architectures that can fulfill timing requirements for each block, with special attention to the OFDM modulator/demodulator, Viterbi and Reed-Solomon decoders, mapper/demapper and HRP multiplexer/demultiplexer. Future work also includes the validation of the implemented hardware blocks and the evaluation of the overall system performance, i.e. transmitter’s baseband processor, channel and receiver’s baseband processor.

REFERENCES


[5]. “High Rate 60 GHz PHY, MAC and PALs”, ECMA–387.