**Calculation of the inverse filters for the Hyundai listening room**

The listening room is equipped with a 6.1 surround loudspeaker rig, as shown in this figure:

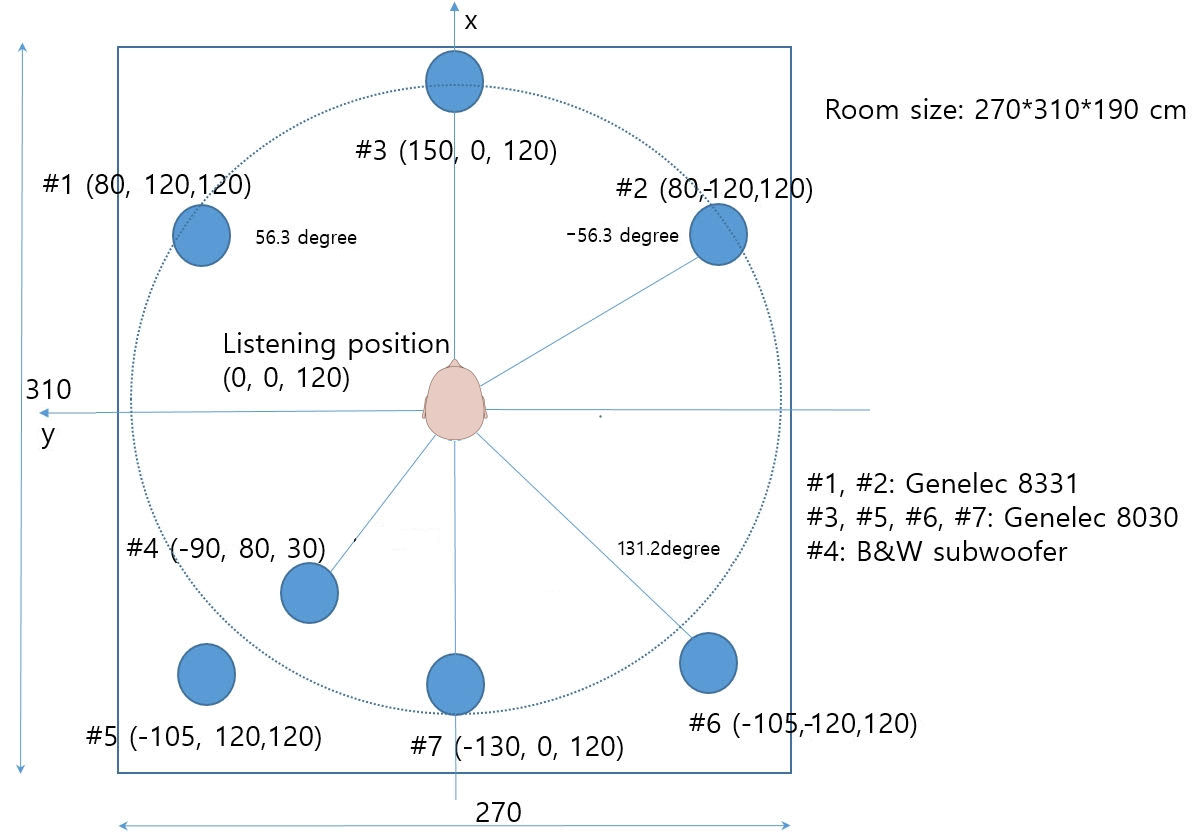


Figure 1 – plan of the listening room with loudspeaker positions.

Hence the loudspeaker numbering, their angular position and their distance from the center is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| N. | Name | Azimuth (°) | Distance (m) |
| 1 | Left | 56.3 | 1.44222 |
| 2 | Right | -56.3 | 1.44222 |
| 3 | Center Front | 0 | 1.5 |
| 4 | Subwoofer | 138.4 | 1.5 |
| 5 | Left Surround | 131.2 | 1.59452 |
| 6 | Right Surround | -131.2 | 1.59452 |
| 7 | Center Back | 180 | 1.3 |

The impulse responses of the system were measured playing an 8-channels WAV file containing an exponential sine sweep in each of them, 15s long, ranging from 22 Hz to 22 kHz, with a silence gap of 5s after each sweep. Channel 8 contains 7 sweeps, as this channel is employed for the electrical loopback connection as shown in the following schematic:



Figure 2 – schematic of electrical connections.

The 8-channels WAV file being played is shown in the following figure:

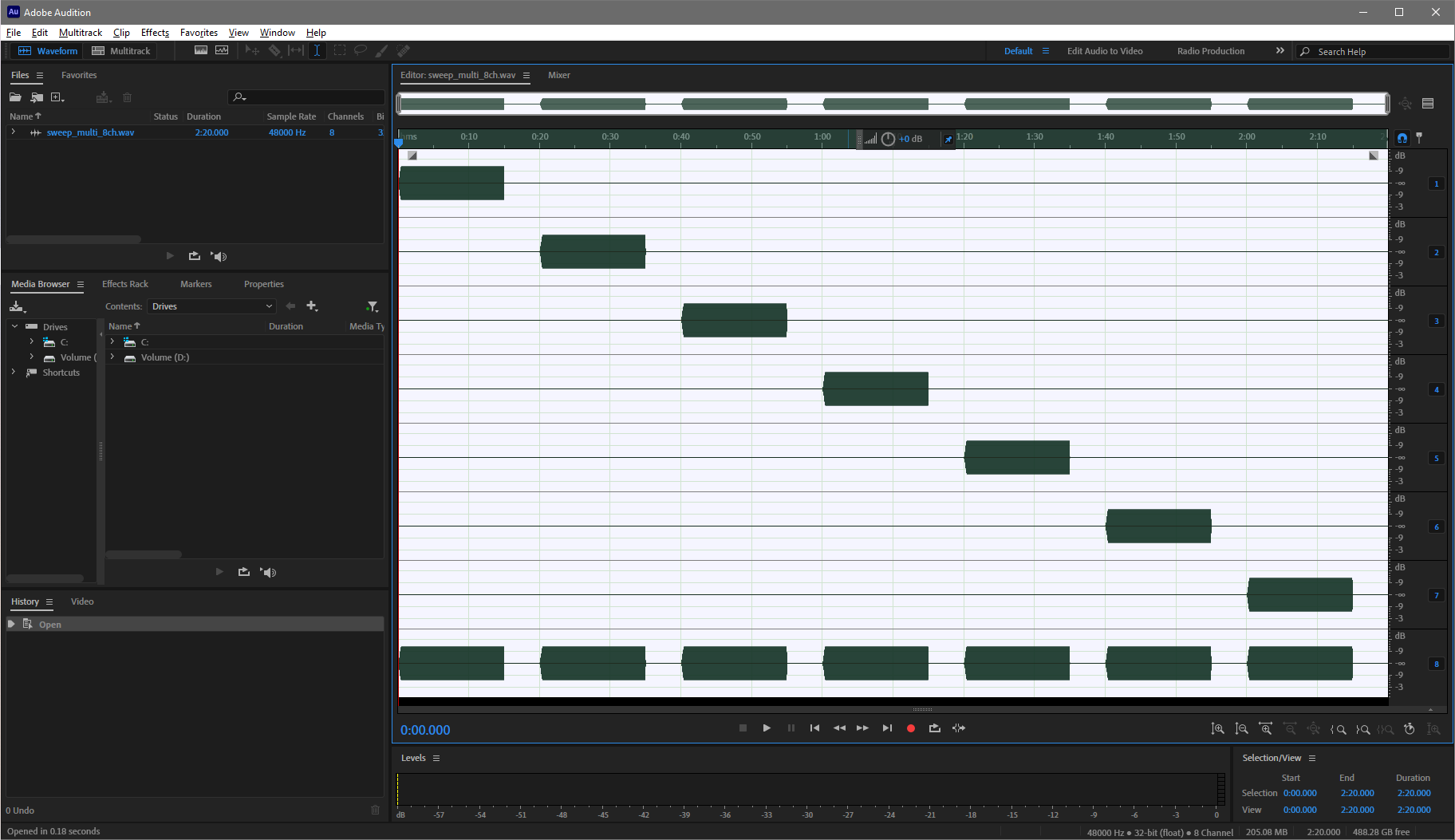


Figure 3 – multichannel test signal

The measurement is performed playing the test signal and simultaneously recording the first two input channels, connected with the measurement microphone, placed at the listening position, and with the electrical loopback cable, as shown in figure 2.

Plogue Bidule was employed for performing the recording, as shown here:

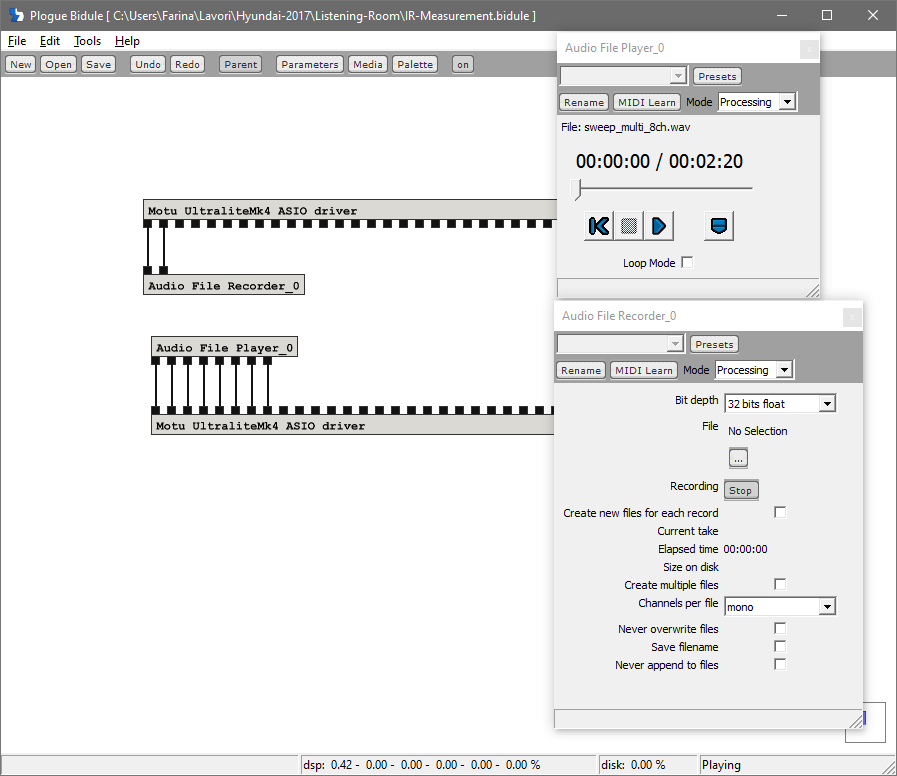


Figure 4 – measurement patch in Bidule

Of course the WAV player and recorder were placed in sync through the automation parameters of Bidule.

The result of the playback&record is a 2-channel file, containing the microphpone signal con channel 1 and the loopback electrical signal on channel 2, as shown here:

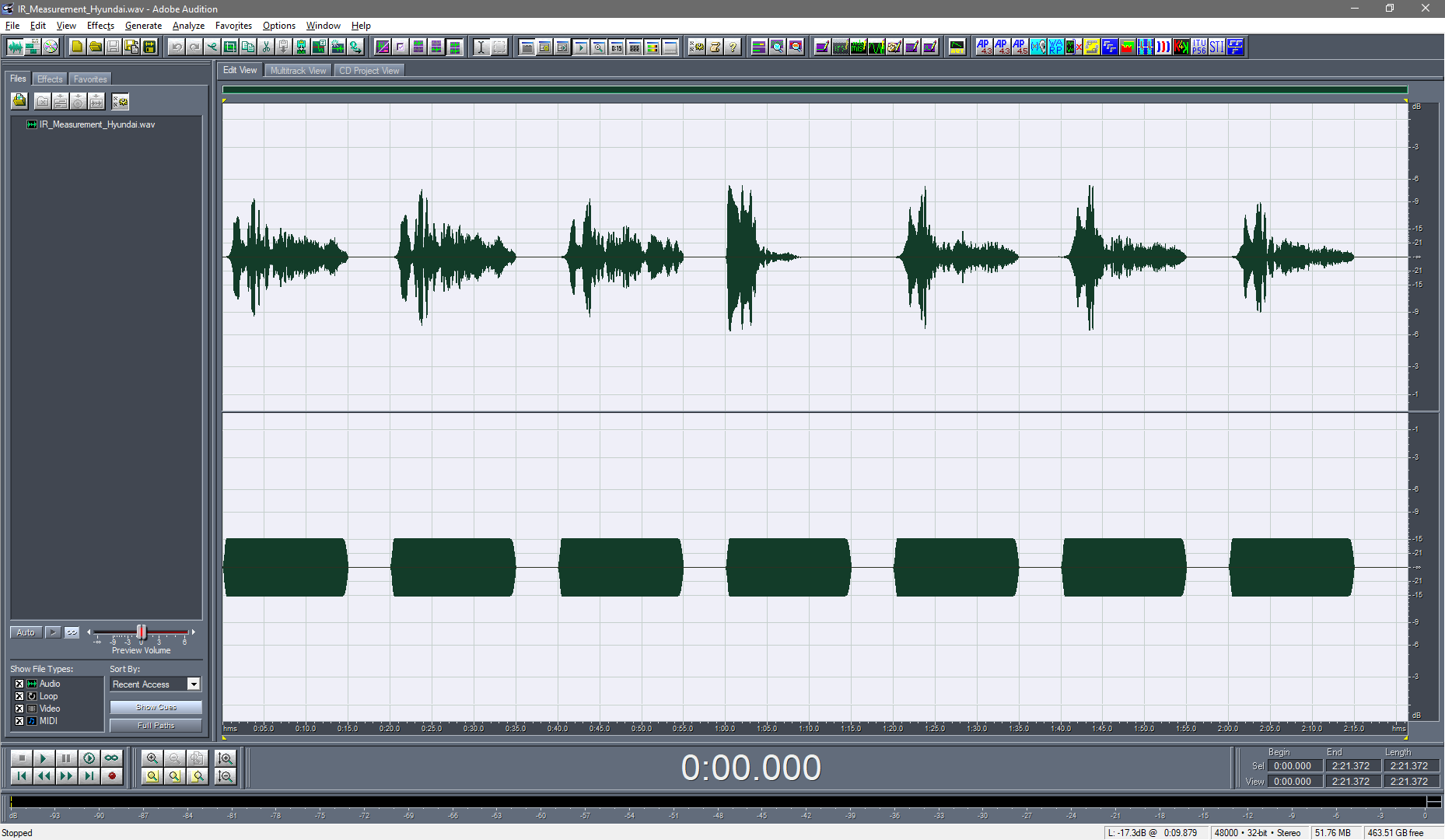


Figure 5 – raw recorded signals

For getting the impulse response, the Inverse Sweep is needed, as shown here:

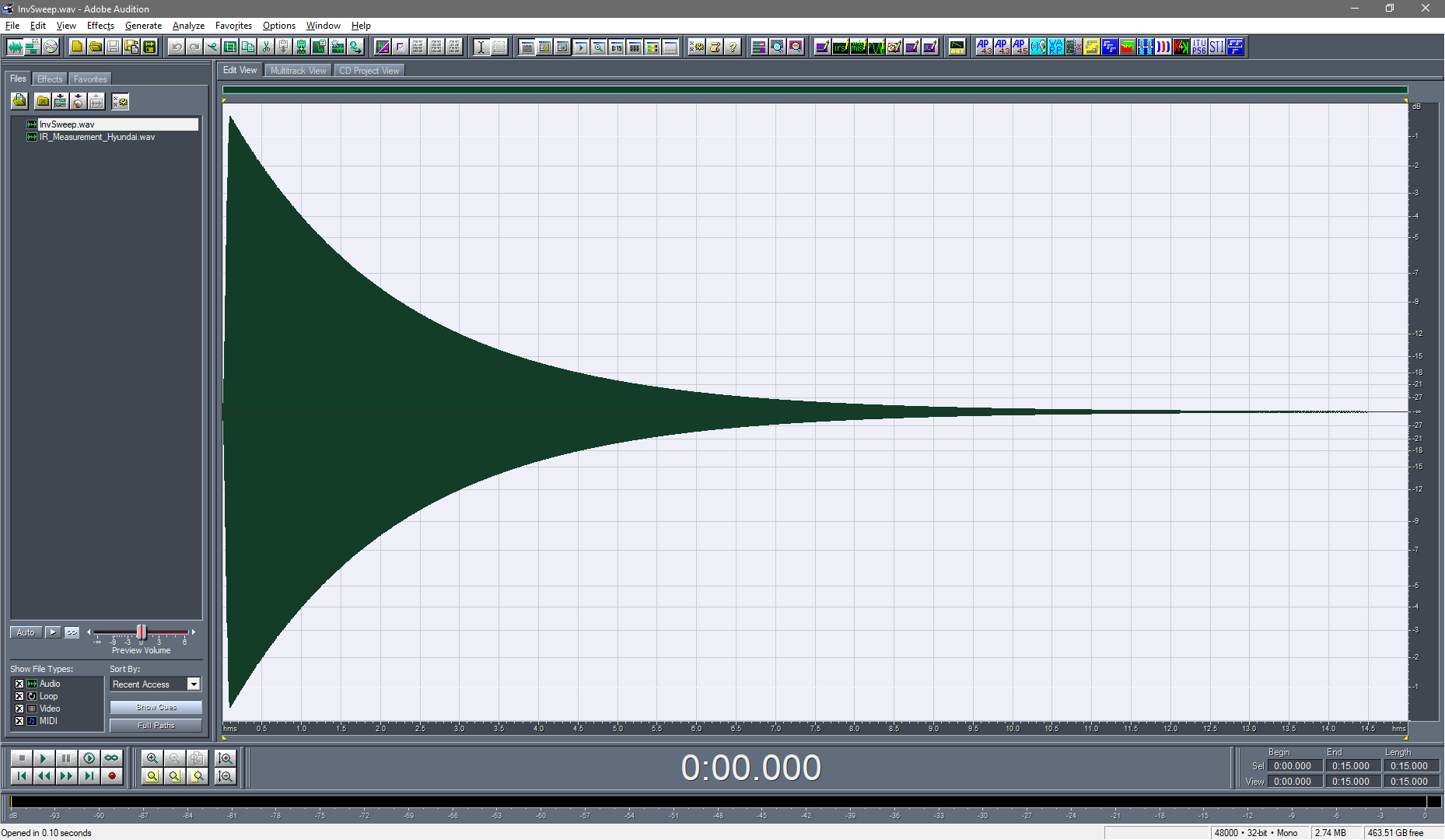


Figure 6 – Inverse Sweep

This signal is copied on the Windows clipboard, and then the stereo signal shown in fig. 5 is “convolved with clipboard” using the Aurora plugin with the same name:

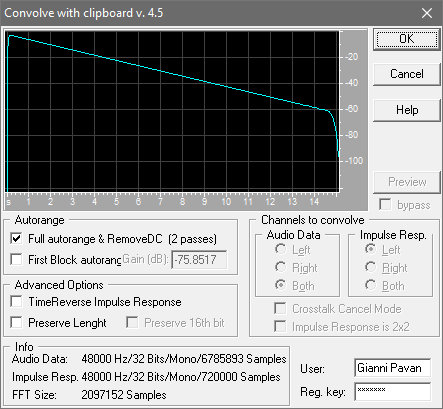


Figure 7 – convolution of the recorded signal with the Inverse Sweep

This is the result of the convolution:

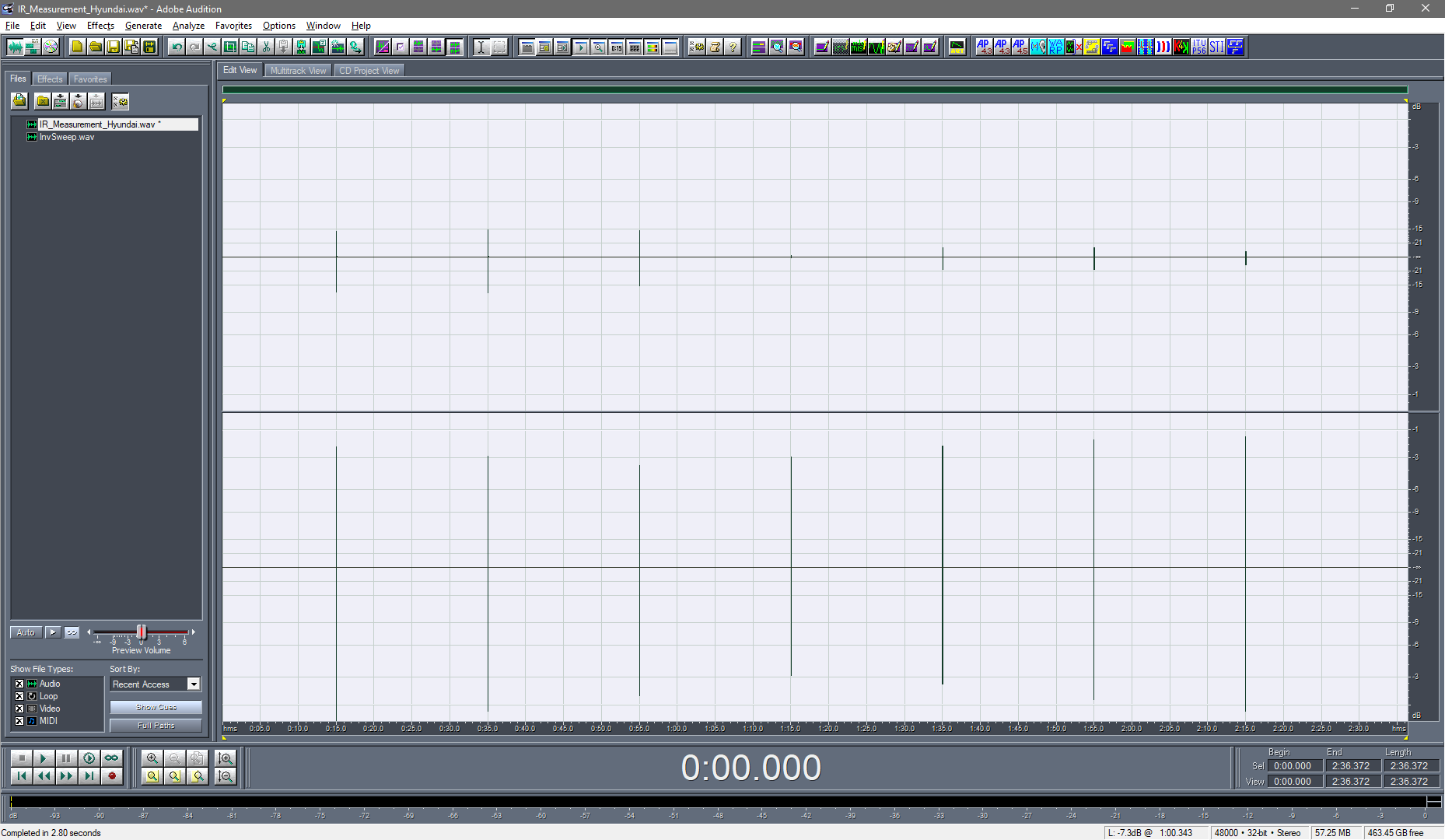


Figure 8 – convolution of the recorded signal with the Inverse Sweep

We now increase the gain of the acoustical impulse response on channel Left, by selecting just it and normalizing its amplitude. Here the result:

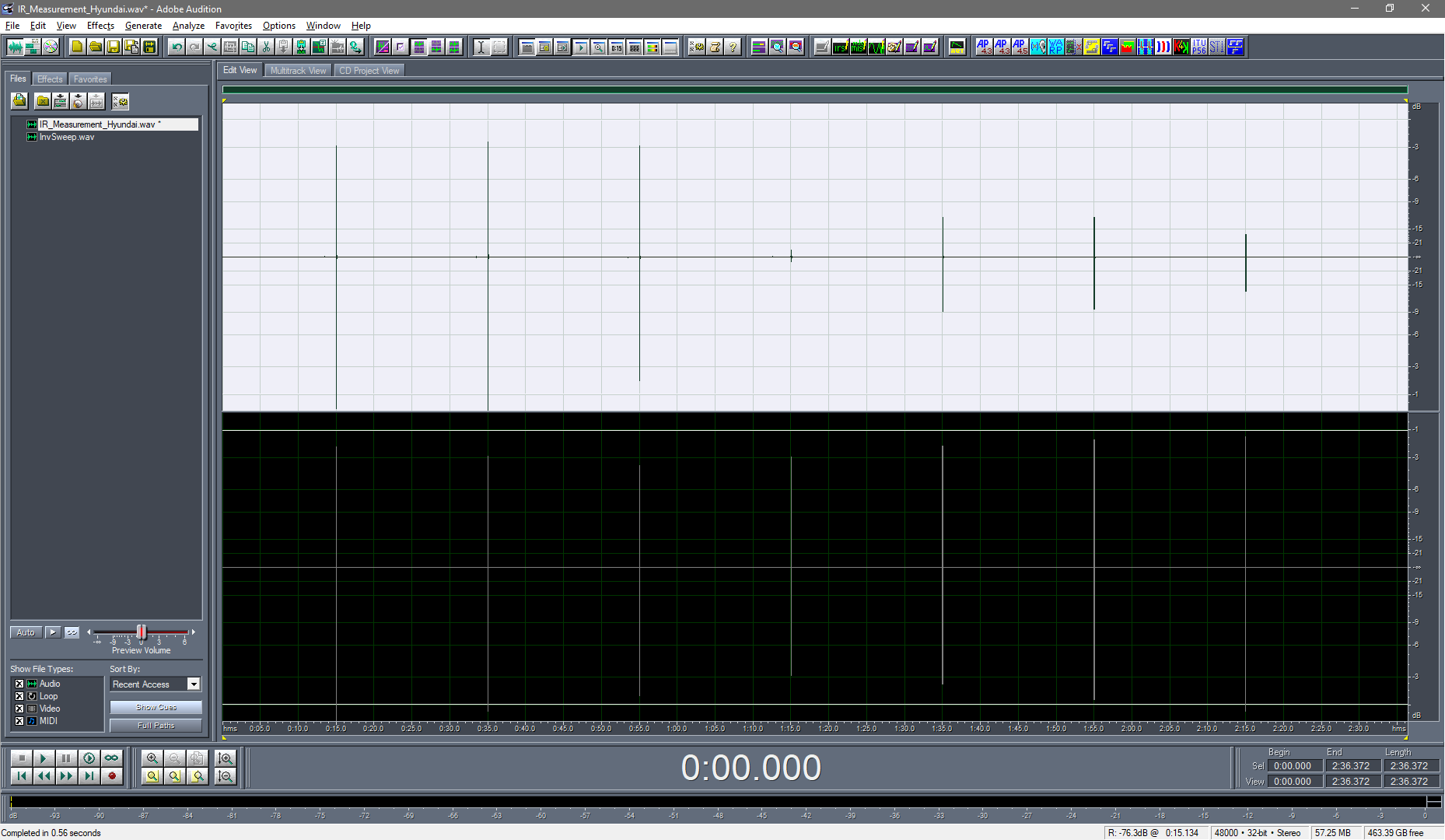


Figure 9 – amplitude normalization of channel Left

Now we want to cut each impulse response much shorter, keeping just 32768 samples of each of them. This task is performed automatically by the Aurora Plugin called IR Select:

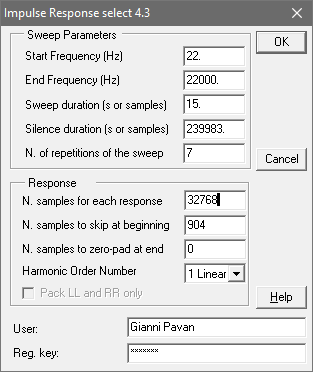


Figure 10 – Impulse Response select

Please note the “strange” numbers entered in the fields “Silence duration” and “N. samples to skip at beginning”. In the first case, thee expected value was 5s (or 240000 samples). But, due to a “clock mismatch” problem between playback and recording, the effective silence duration was shorter by 17 samples.

The second number, 904 samples, is the “electrical latency” of the system, which was obtained by first performing the IR Selection, and then observing that the electrical loopback signal, on channel Right, has its first peak at sample 904 instead of at sample 1.

This is the result:



Figure 11 – Sequence of 7 “short” IRs

After checking that the right channel is showing the peak of the electrical loopback at the expected samples (1, 32769, 65537, etc.), meaning that the temporal alignment is correct, we can now remove the right channels, as it is not needed anymore, after having been useful for measuring the electrical latency. This is done using the command Edit – Convert Sample Type:

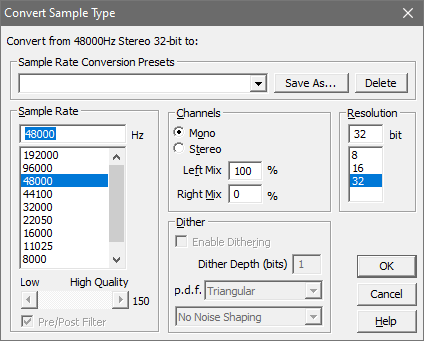


Figure 12 – Removal of right channel

The result is the following sequence of 7 impulse responses, which are the starting point for computing the inverse filters:



Figure 13 – Sequence of 7 Impulse Responses

The problem now is that the room is exhibiting some string reflection, which must be excluded during the calculation of the inverse filters, for avoiding nasty artefacts. See here for example a zoom on the IR of channel 3 (Center):

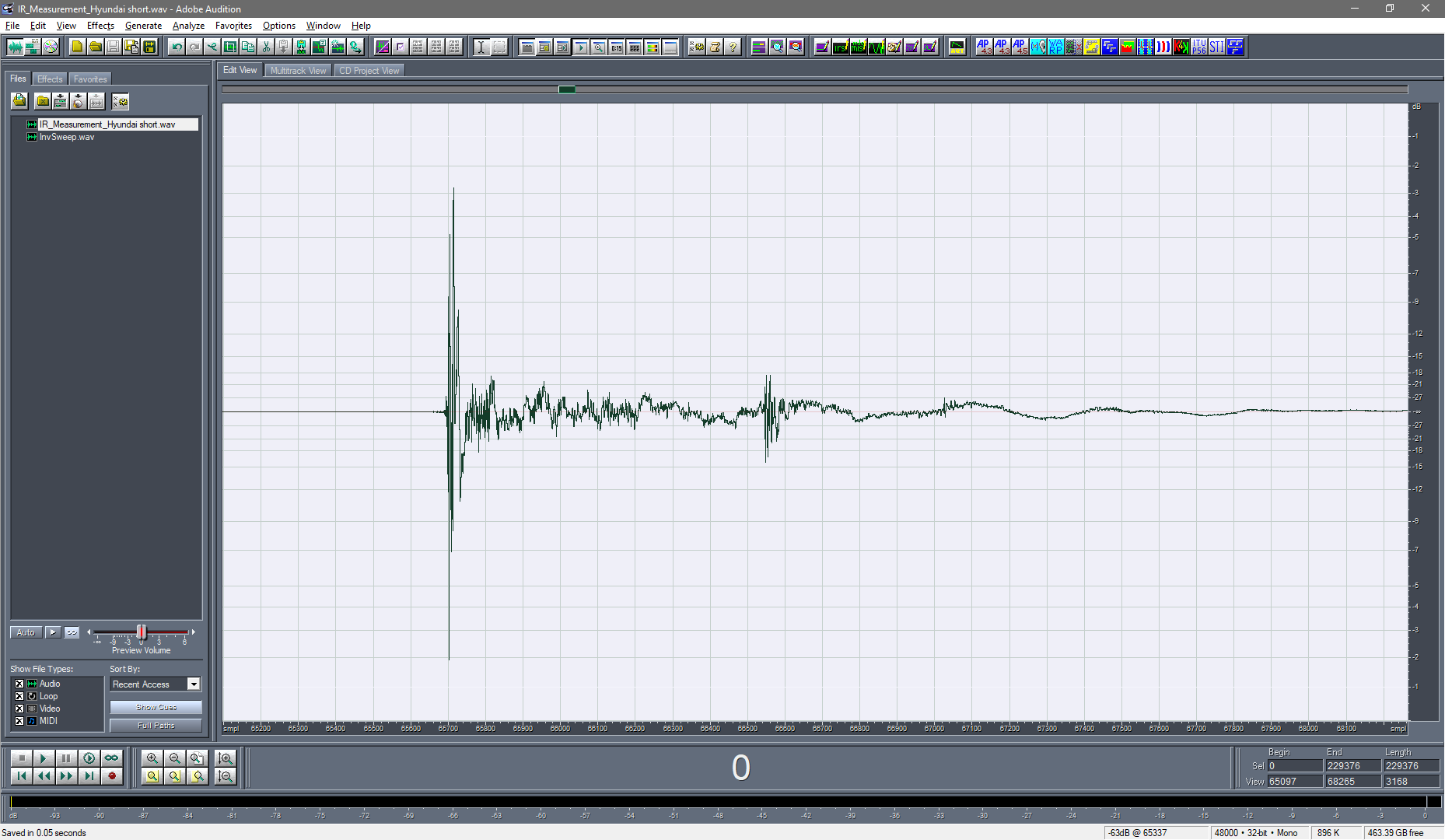


Figure 14 – Parasitic reflection after 19ms on channel 3 (Center)

Here instead what happens on channel 6 (Right Surround):



Figure 15 – Parasitic reflection after 8ms on channel 6 (Right Surround)

The parasitic reflection here has a delay of just 8 ms.

It should be advisable to treat these reflecting surfaces with proper absorbing materials, and to repeat the measurements.

Despite these problems, we attempted to still use these measurements for computing a filter matrix.

There are two approaches for this:

1. a complete inversion approach, attempting to invert the matrix by a “brute force” numerical approach
2. a “hand-made” approach, where temporal alignment, magnitude equalization and spatial information are addressed separately with proper tools, and the results are assembled in the final filter matrix

As already evidence in the AES-52 paper[[1]](#footnote-1), the first approach is superior only in theory, requiring very clean impulse response measurements, without any clock mismatch problem, nor spurious reflections. In the real world, the second approach is usually much more robust, hence in the following we employed it.

We start correcting for the latency. We measured the time-of-arrival of the direct sound from the time-aligned impulse responses shown in fig. 13:

Table I – latencies and distances for the 7 loudspeakers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N. | Name | Azimuth (°) | Distance (m) | Latency (samples) | Distance (m) |
| 1 | Left | 56.3 | 1.44222 | 334 | 2.505 |
| 2 | Right | -56.3 | 1.44222 | 333 | 2.4975 |
| 3 | Center Front | 0 | 1.5 | 200 | 1.5 |
| 4 | Subwoofer | 138.4 | 1.5 | 461 | 3.4575 |
| 5 | Left Surround | 131.2 | 1.59452 | 209 | 1.5675 |
| 6 | Right Surround | -131.2 | 1.59452 | 212 | 1.59 |
| 7 | Center Back | 180 | 1.3 | 171 | 1.2825 |

It can be seen how the measured latency correspond very well with the nominal geometrical distance of each loudspeaker, except for the subwoofer (which will be treated separately) and for the two main loudspeakers 1 & 2. These are very high quality Genelec 8331 SAM studio monitors. These loudspeakers include a powerful DSP unit, which ensures very good equalization. But the DSP also causes an additional latency, making the sound radiated by these two loudspeaker to “arrive later” then the other 4 smaller ones, which are Genelec 8030 monitors, without any DSP.

So, for correcting the additional latency of channels 1&2, it is necessary to add a delay of 140 samples to channels 3,5,6,7.

In practice, what we want is to add some latency to every channel, so that the total latency (acoustical plus the additional one computed here) gives a total latency which is the same for all channels. The total latency was set to 512 samples, hence the latency to be applied to each channel is the following:

Table II – additional latencies in samples for each of the 7 loudspeakers

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N. | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Additional latency | 178 | 179 | 312 | 51 | 303 | 300 | 341 |

We address now the gain of the individual loudspeakers, performing an octave-band analysis of the 7 measured impulse response.

This is performed by selecting exactly the 32768 samples of each IR, and then invoking the Aurora STI plugin, which is employed for performing the octave-band spectrum analysis. The following figure shows this plugin being used for analysing the IR of loudspeaker n.1:

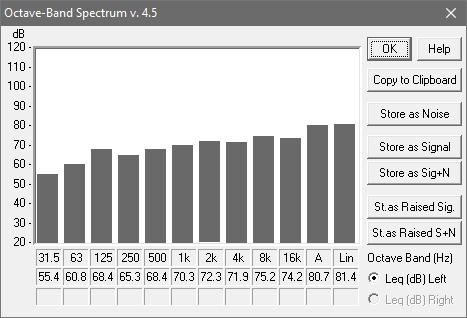


Figure 16 – Octave band analysis of the IR of loudspeaker #1

It must be observed that the ideal response is NOT a flat response, as the test signal of an impulse response is a Dirac’s Delta function, which has a “white” spectrum. In an octave band analysis this result in a spectrum which increases by 3.01 dB for each octave. The following figure shows the octave band spectra of the 7 loudspeakers, and the “white” target spectrum:



Figure 17 – Octave band analysis of the 7 loudspeakers

This means that the loudspeakers are not properly equalized, nor properly gain-matched. So we computed a table of gain correction factors, to be applied for getting proper gain matching and spectrum equalization of the 6 main loudspeakers, as shown here:

Table III – gain correction factors for spectrum equalization and level alignment

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | F [Hz] | 31.5 | 63 | 125 | 250 | 500 | 1000 | 2000 | 4000 | 8000 | 16000 |
| Ls. N. | Target | 50.00 | 53.01 | 56.02 | 59.03 | 62.04 | 65.05 | 68.06 | 71.07 | 74.08 | 77.09 |
| 1 | Gain | -5.42 | -7.75 | -12.42 | -6.23 | -6.31 | -5.28 | -4.21 | -0.87 | -1.14 | 2.92 |
| 2 | Gain | -6.02 | -8.26 | -14.08 | -7.01 | -8.48 | -6.80 | -5.65 | -1.40 | -2.17 | 4.17 |
| 3 | Gain | -0.22 | -6.68 | -12.46 | -4.22 | -5.16 | -7.04 | -1.44 | -4.03 | -1.01 | 1.66 |
| 4 | Gain | -13.74 | -15.47 | -7.29 |  |  |  |  |  |  |  |
| 5 | Gain | 3.41 | -10.75 | -13.51 | -2.05 | -0.04 | -2.83 | 1.61 | 1.05 | 3.42 | 7.48 |
| 6 | Gain | 2.56 | -11.28 | -13.96 | -2.65 | -3.80 | -1.71 | 1.28 | 0.31 | 4.11 | 6.59 |
| 7 | Gain | 6.13 | -8.23 | -12.65 | -2.35 | -3.11 | -0.80 | 4.12 | 4.91 | 7.39 | 10.02 |

The subwoofer is treated differently, equalizing it only in the three lower octave bands, as it will be fed with low-passed signal, hence what happens at 250 Hz and above is irrelevant.

An EQ filter for each loudspeaker is generated starting form a Dirac’s Delta function, and applying the gains shown above by means of the Graphical Equalizer effect of Adobe Audition. The next figure shows the reference Dirac’s Delta employed, having the peak at sample n. 4096. Of course this peak is first delayed of the amount specified in table II, and then the Graphical Equalizer plugin is employed for applying proper gain at each frequency, as specified in table III. The following figure shows the Graphic Equalizer plugin:

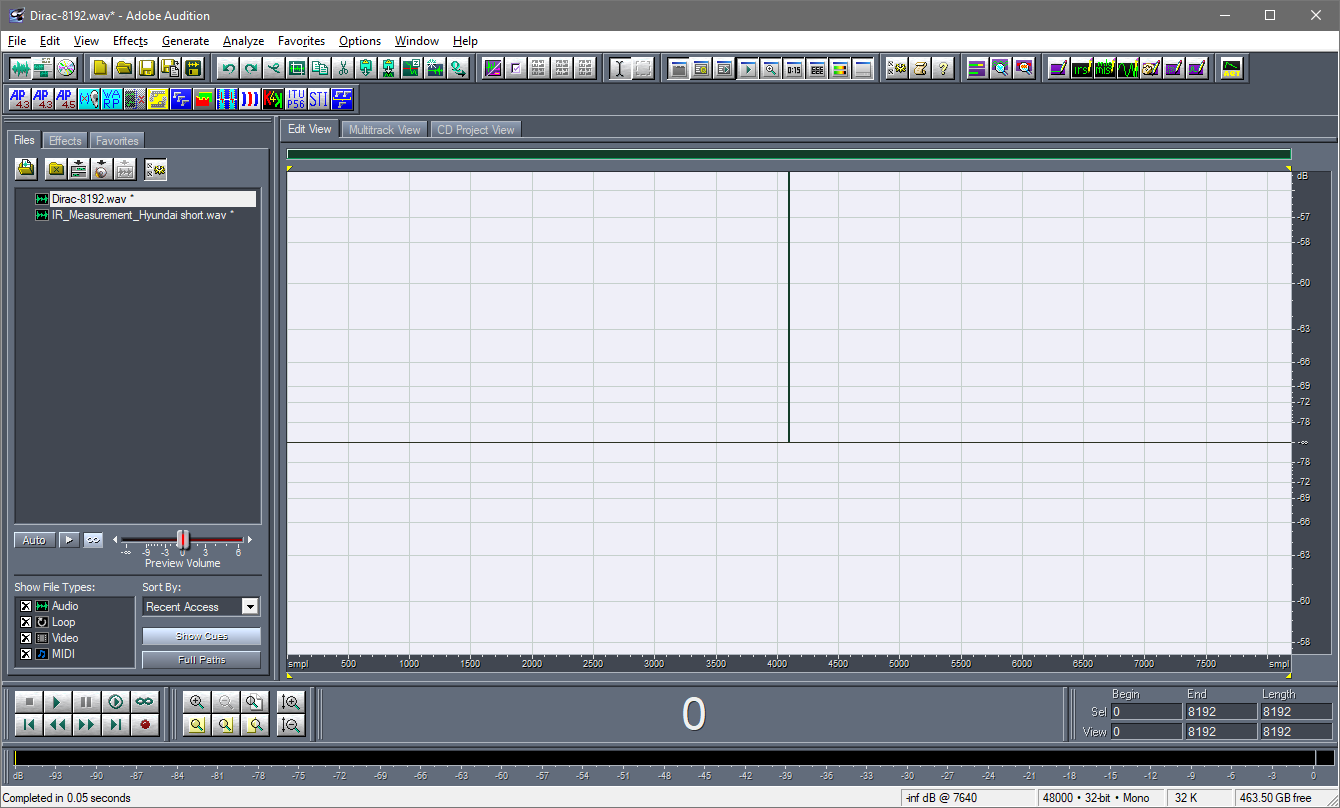


Figure 18 – Dirac’s Delta with a latency of 4096 samples

For each channel, the EQ filter is generated by applying the Graphical Equalizer with the proper gain at each octave-band centre frequency, as shown in the following figure for loudspeaker n.1:

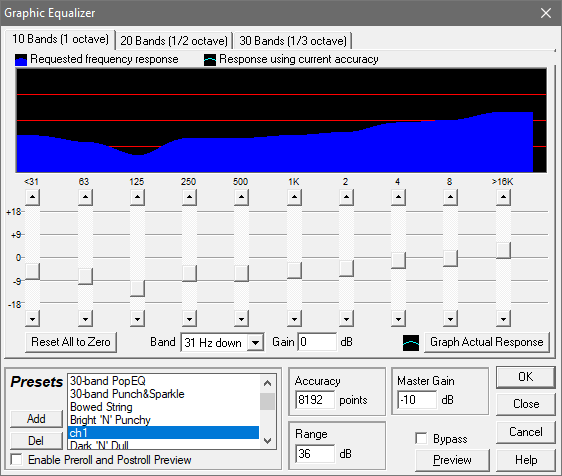


Figure 19 – the Graphical Equalizer plugin processing loudspeaker n.1

The result is a linear-phase FIR filter (symmetrical in time domain around the position of the Dirac’s Delta peak), as shown here for channel n. 1:

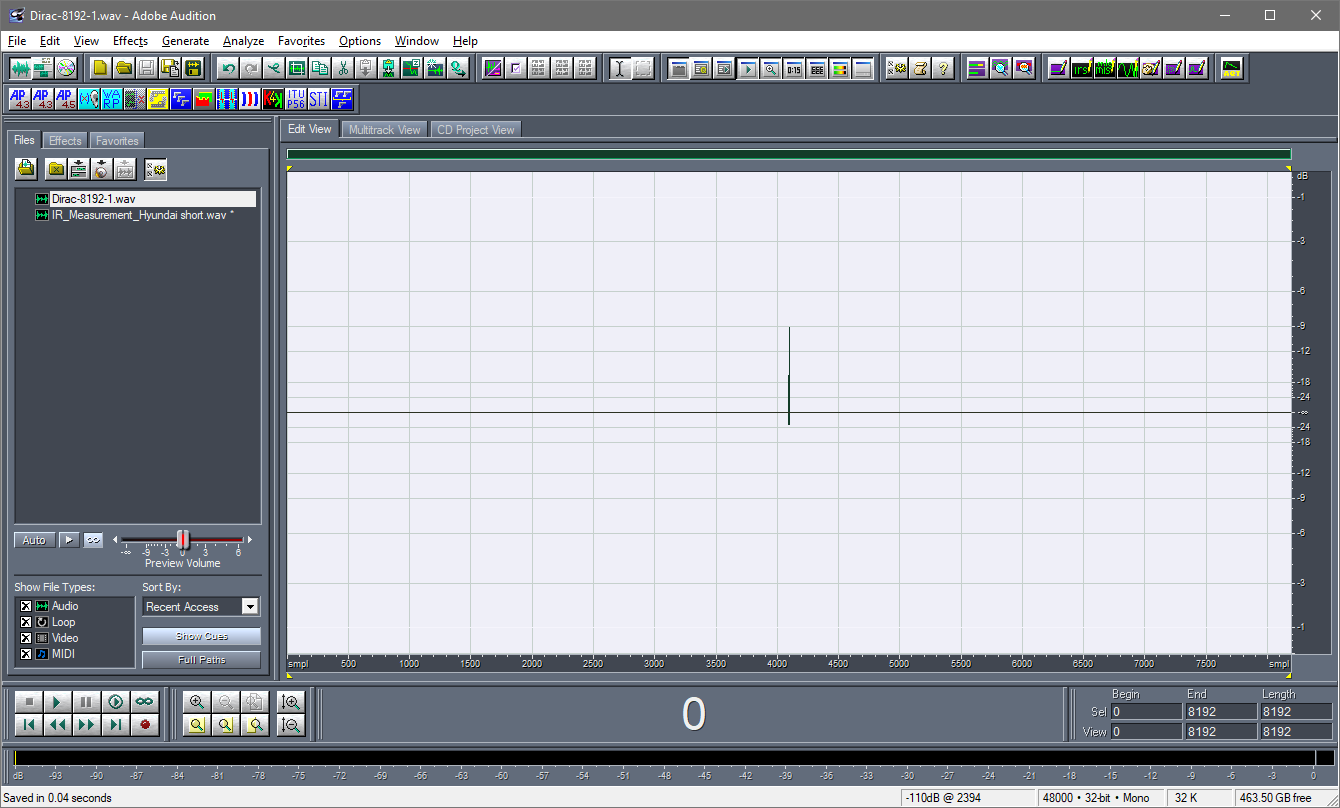


Figure 20 – EQ FIR filter for channel n. 1

We now check that the filter is working properly, by loading it on the Windows clipboard, and applying it by convolution to the IR of channel 1:

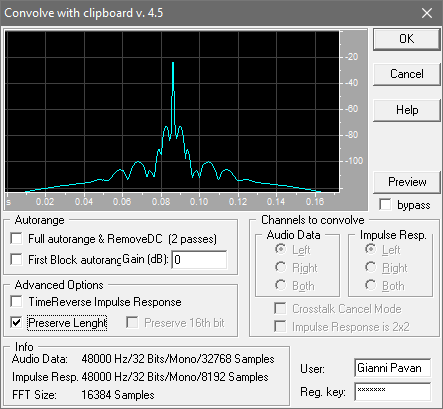


Figure 21 – convolution of the IR of channel 1 with its EQ FIR filter

It must be noted how the gain has been locked at 0 dB and the “preserve length” option has been enabled.

Here the result:

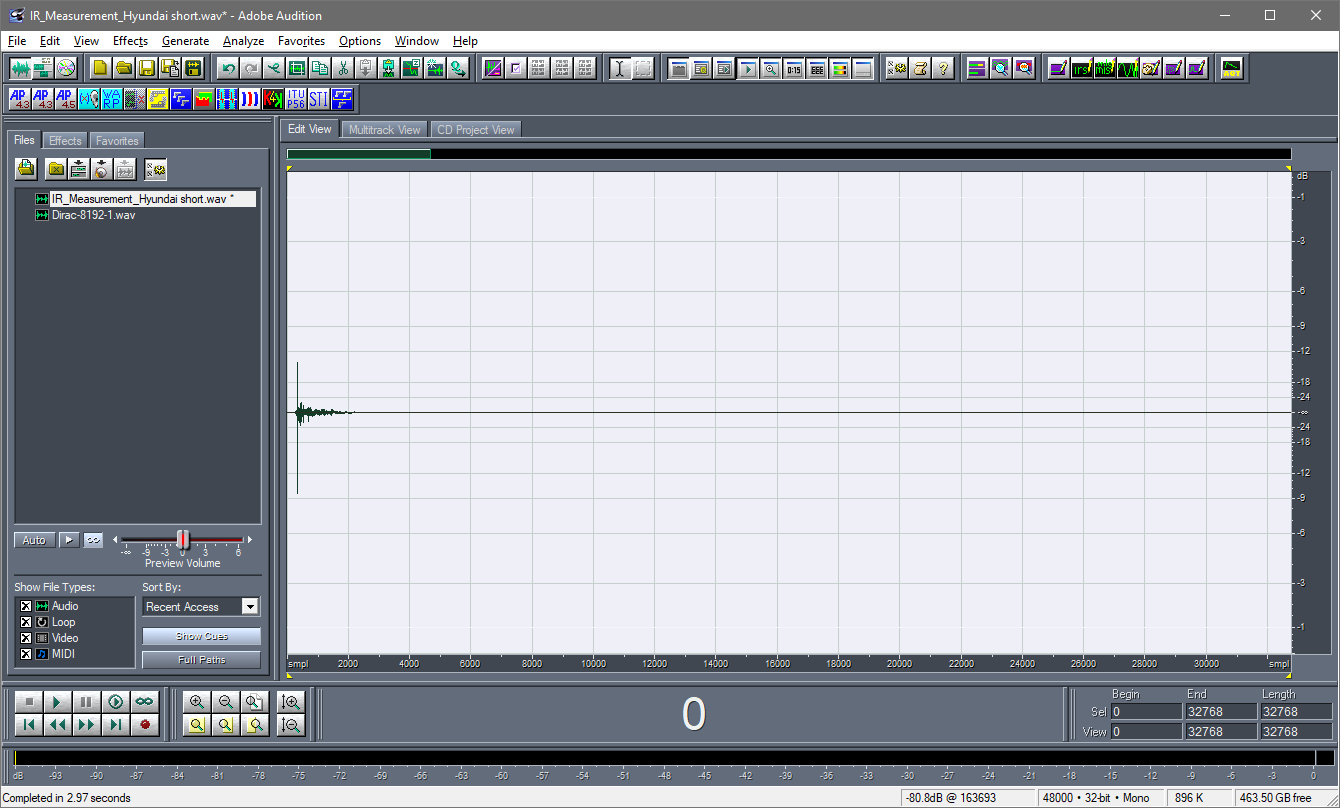


Figure 22 – Impulse Response of ch.1, equalized

By applying the Aurora STI module, we perform again the Octave Band analysis, and we check that the results are close to the “target” spectrum shown in table II:

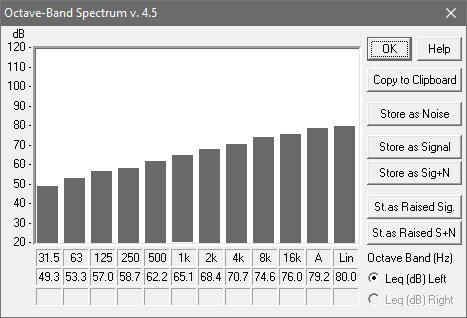


Figure 23 – Octave Band spectrum of ch.1, equalized

The results are reasonably close to the target spectrum, so we proceed computing the EQ FIR filters for all the other channels, except for channel 4 (subwoofer) which will be treated separately later.

After checking the correct equalization for all 6 channels, we obtain the following chart:



Figure 24 – Octave Band spectra of ch.1,2,3,5,6,7, equalized

It can be seen that there are some deviations from the target spectrum, particularly at low frequency.

These are not important, as these frequencies will be partially routed to the subwoofer, as explained later.

The 7 FIR filters are placed along the diagonal of a 7x7 filter matrix, by adding proper chunks of silence, 8192 samples each, before the FIR filter. Such a leading silent shifts the FIR by 8192 samples more for each channel.

The 7 delayed FIR filters are prepared in Adobe Audition, and then assembled into a single 7-channels WAV file using Audacity, as shown here:

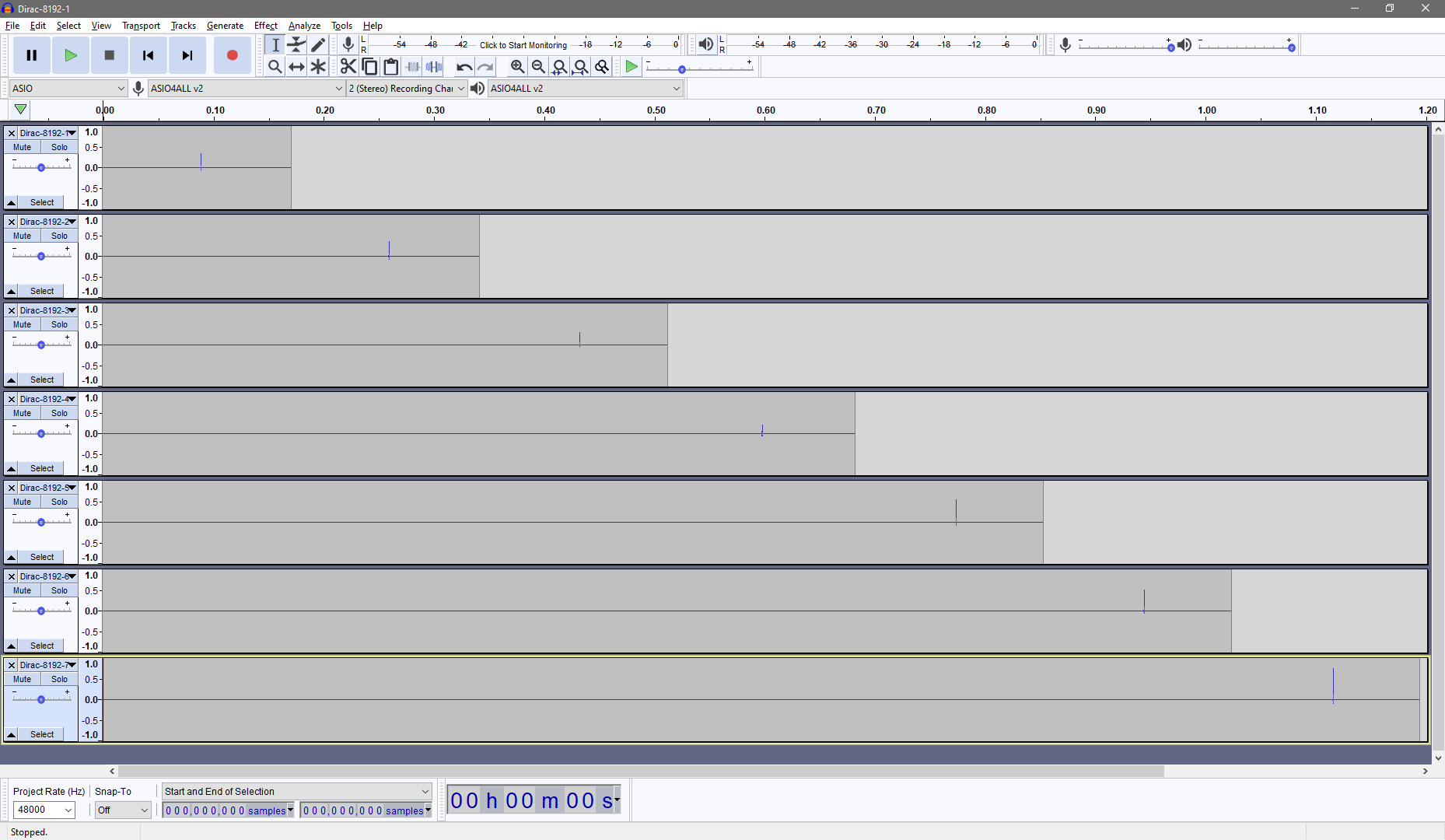


Figure 25 – the 7 FIR filters are assembled in Audacity

Exporting this multichannel session, a single 7-channels WAV file is created, ready to be used in a multichannel convolver such as MCFX Convolver.

At this point we can insert this matrix of FIR filters instead of the IEM Distance Compensator plugin in the patch for Plogue Bidule performing Ambisonics decoding. Here the original patch using Distance Compensator:

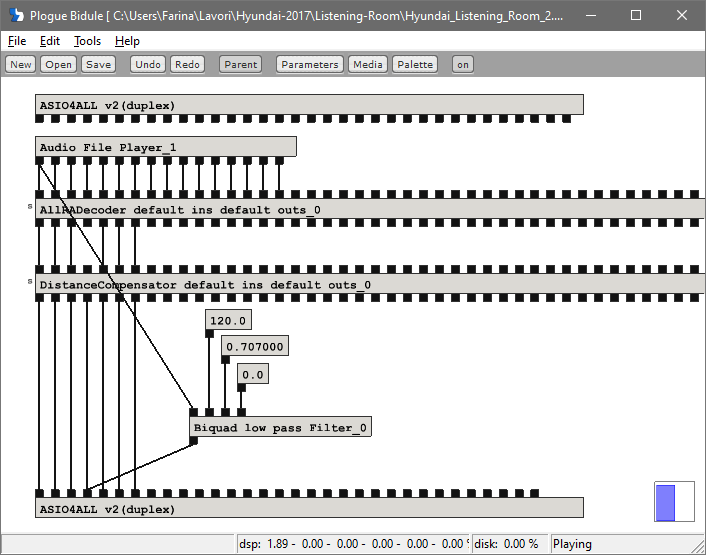


Figure 26 – Ambisonics decoding with IEM plugins

Here the same patch where MCFX Convolver, loaded with the filter matrix shown in fig. 25, has replaced Distance Compensator:

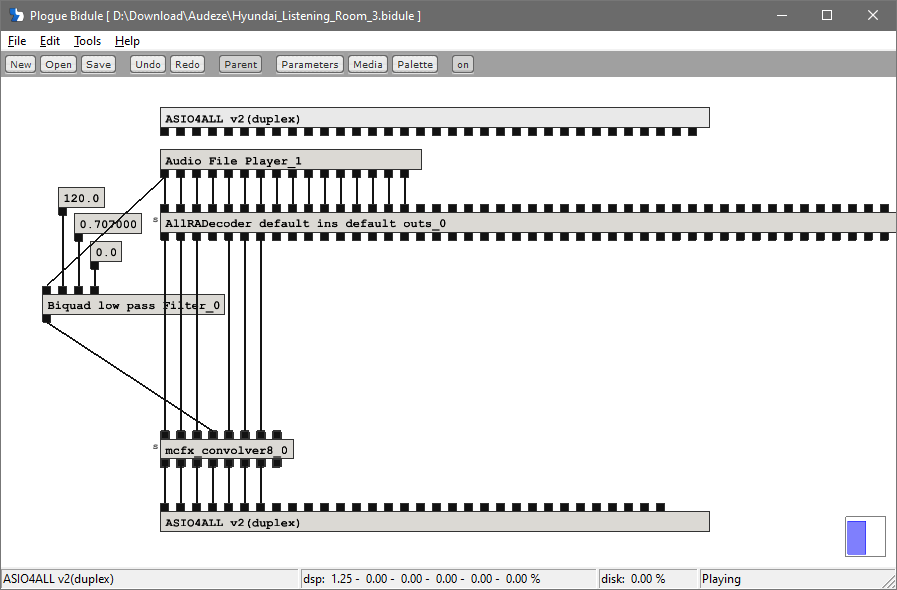


Figure 26 – Ambisonics decoding with the FIR filter matrix

We can now pack the spatial information provided by the IEM Allradecoder plugin and the temporal-gain-spectral information provided by the diagonal FIR filter matrix into a single full FIR filter matrix.  
This is feasible by “sampling” the whole processing performed by the patch shown in fig.26, feeding it with a sequence of 16 Dirac Delta pulses, spaced 8192 samples one after the other, as shown here below:

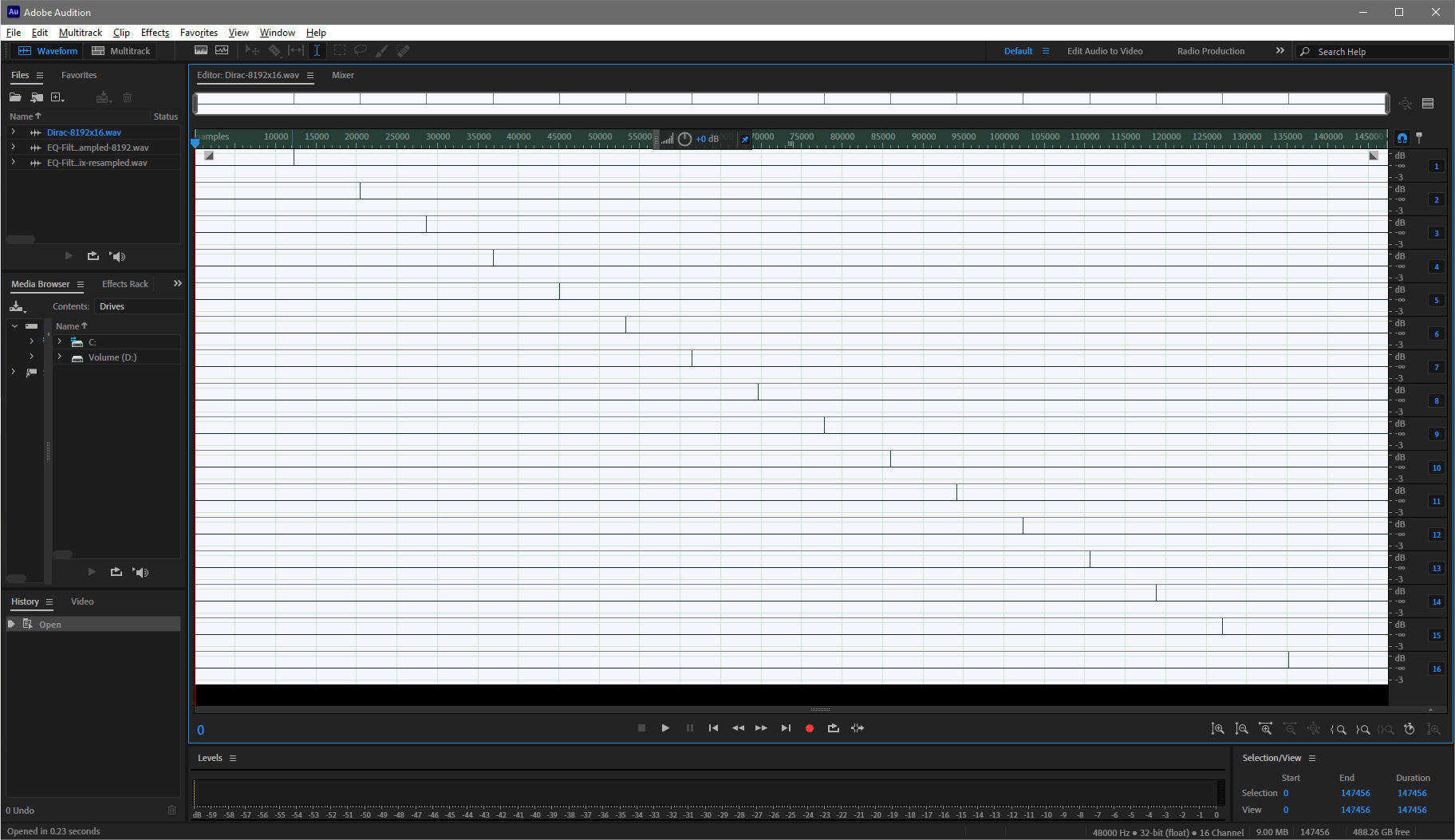


Figure 27 – Sequence of 16 Dirac’s Delta pulses

The processed output is saved in a 7-channels WAV file, which contains the whole filter matrix. After cutting away the leading zeros and the zeros at the end, a full FIR filter matrix of 16x7 filters is obtained, each 8192 samples long, as shown here:

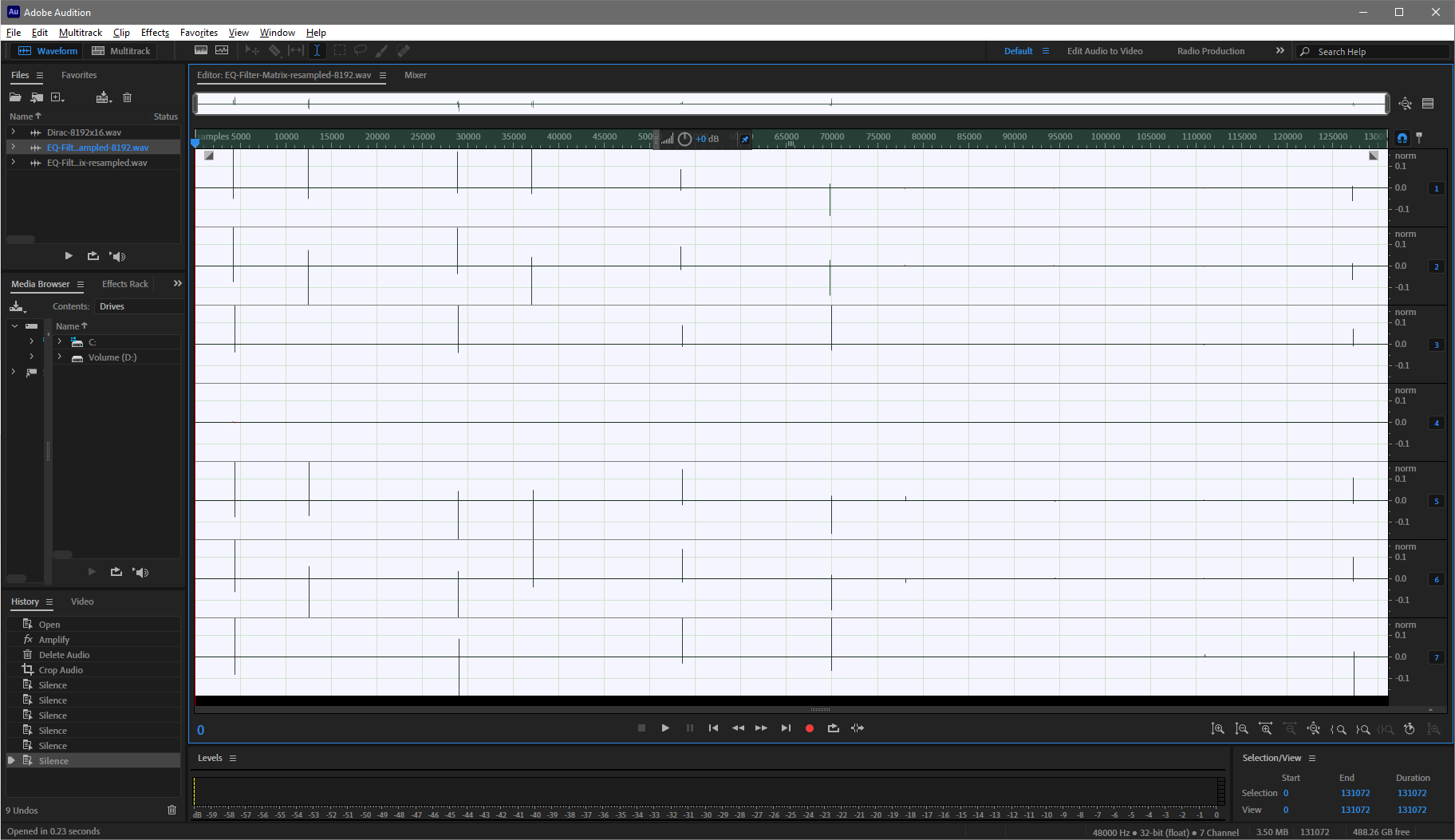


Figure 28 – The complete FIR filter matrix, 16 inputs, 7 outputs

From fig. 28 it is clear that the only 2 of the last 7 inputs (n. 10..16) contribute modestly to the 7 speaker feeds. This is due to two factors:

1. All the loudspeakers are on the horizon plane, hence the spherical harmonics providing vertical information are ignored
2. 6 loudspeakers in a ring are really few for rendering 3rd-order components, in practice only the last one maps significantly to the loudspeaker positions.

This means that the sound reproduced by tis loudspeaker array will not provide the same detail in terms of spatial accuracy as a direct binaural rendering over headphones with head tracking, or employing a more massive loudspeaker array, as the 24-channels system recently installed at Ask industries.

However, thanks to the availability of the complete FIR filter matrix shown in fig.28, it becomes possible to perform the Ambisonics decoding employing just one instance of MCFX Convolver, as shown here:

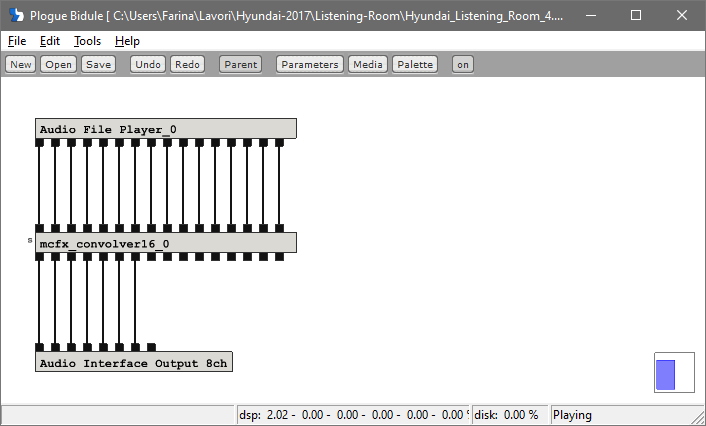


Figure 29 – Ambisonics decoder employing just one instance of MCFX Convolver

Of course for using in MCFX Convolver the FIR filter matrix shown in fig. 28, the WAV file must be associated with a proper .CONF file, with the same name, which must be placed in the same folder as the other MCFX Convolver CONF files, for example in:

**C:\Users\farin\AppData\Roaming\mcfx\convolver\_presets**

The names of the two files to be placed in this folder are:

**EQ-Filter-Matrix-resampled-8192.conf**

**EQ-Filter-Matrix-resampled-8192.wav**

The same can be employed also on the Virtual Tuning Tool, sending the 16-channels Ambisonics output of the convolver running the SPK2SH filters of the car (either measured or simulated) to another instance of MCFX Convolver, implementing the FIR filter matrix shown above, which can be considered a SH2SPK filter matrix.  
In practice the VTT will “transcode” from the speaker feeds inside the car to the speaker feeds inside the listening room. This transcoding will be in some way “lossy”, as the 12 loudspeakers of the car have to be rendered using just 6+1 loudspeakers in the listening room, and so some spatial blur will be unavoidable.

Parma, 24 June 2020

Prof. Angelo Farina



1. <http://www.angelofarina.it/Public/Papers/273-AES52-2013.pdf> [↑](#footnote-ref-1)