Musical Acoustics
HL4: Radiance estimation

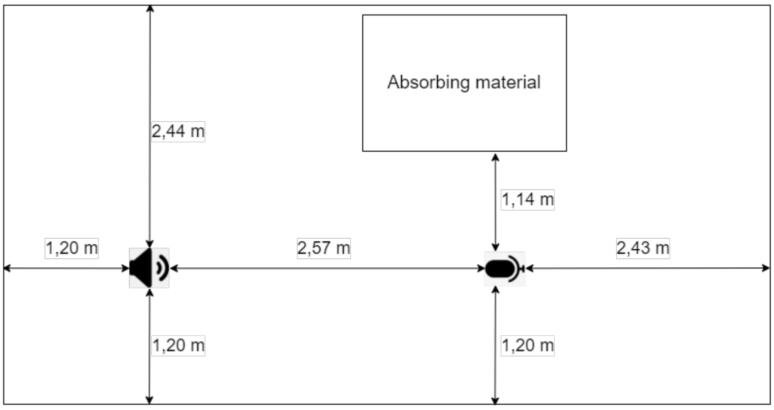
Marin Pasin Davide 10799610
De Bortoli Gian Marco 10805035

A.Y. 2021/2022

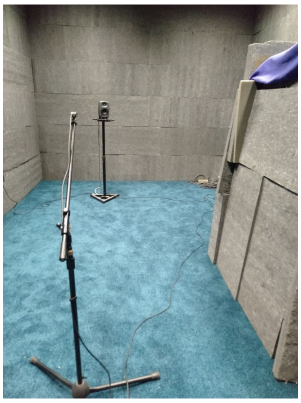
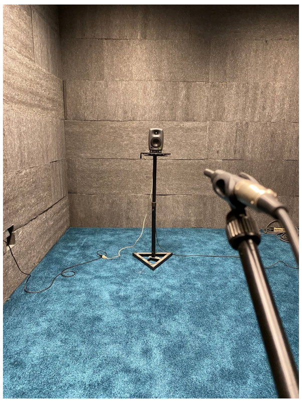
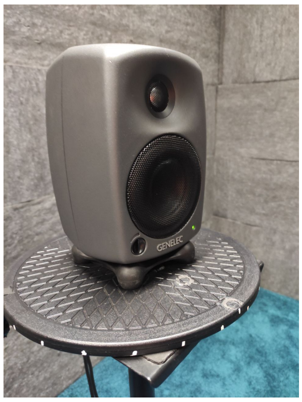
POLITECNICO
MILANO 1863

Introduction and setup

The aim of this homework is to estimate the radiance pattern of a loudspeaker, starting from measurements taken in a lab session. The measurements were taken in a semi anechoic chamber belonging to a branch of Politecnico di Milano located in Cremona. The loudspeaker under test, a Genelec speaker, is fed a source signal. A measurement microphone picks up the output. The two elements are placed in the room as shown in the picture below which describes a top view of the room.



Both the microphone and the center point of the diaphragm of the speaker are placed at 1.25 m from the ground.
Two different types of signal sources are taken into considerations, a white noise signal and a sine sweep ranging from $f_1 = 50Hz$ to $f_2 = 22kHz$. Both the source signals and the measurements are sampled with a sampling frequency $f_s = 48kHz$.
The loudspeaker is placed on a support able to rotate 360 degrees, and for both source signals 24 measurements were taken, at regular intervals of rotation, 15° each.

It is noticeable the presence of some absorbing panels to the right of the microphone, which may cause increased amounts of reflections and limited our freedom in choosing an ideal position for the setup.

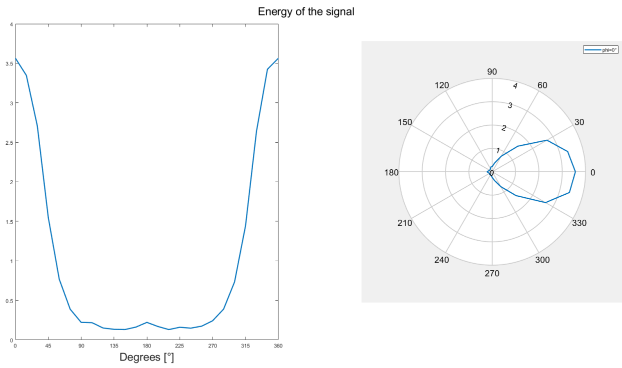
1 Signal observation

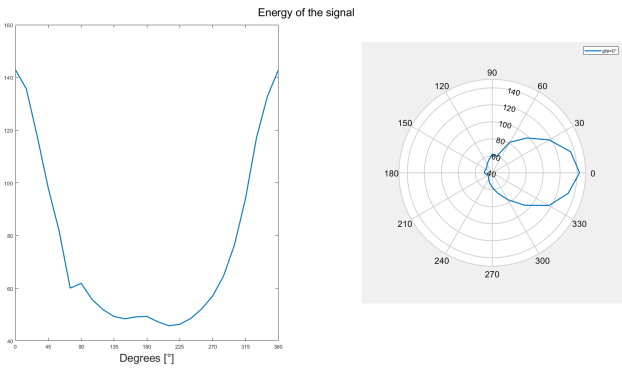
We want to observe the directivity of the loudspeaker in terms of the energy of the signal captured by the microphone.
After loading the files y_i, with $i = 1...24$ in Matlab, we calculate the energy of each

one of them as

$$E = \sum_{n=1}^{N} |y_i(n)|^2 \tag{1}$$

where N is the number of samples of the signal.
Since for an angle of rotation $\theta = 0°$ the loudspeaker is directed towards the microphone, we expect to find here the maximum energy, which should decrease for measurements made at increasing angles, until reaching a minimum for $\theta = 180°$. We expect the energy to then increase again, until reaching a value close to the maximum when completing the rotation.
As it can be seen from the plot below, the results are quite close to what was expected, both for the white noise case (first plot) and for the sine sweep case (second plot)





We didn't take a measure for the angle of 360°, we copied the value obtained for 0° in order to close the curve.

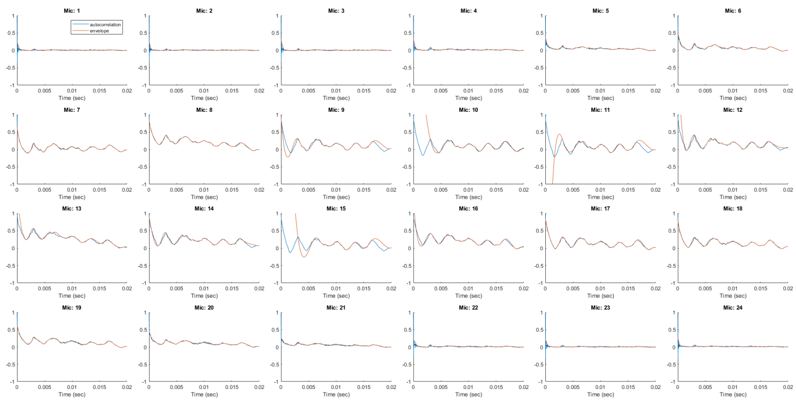
2 Room reflection analysis using autocorrelation

We want to estimate the time at which the first reflection occurs studying the autocorrelation of the captured signals.
The autocorrelation of a signal $x(n)$ can be defined as

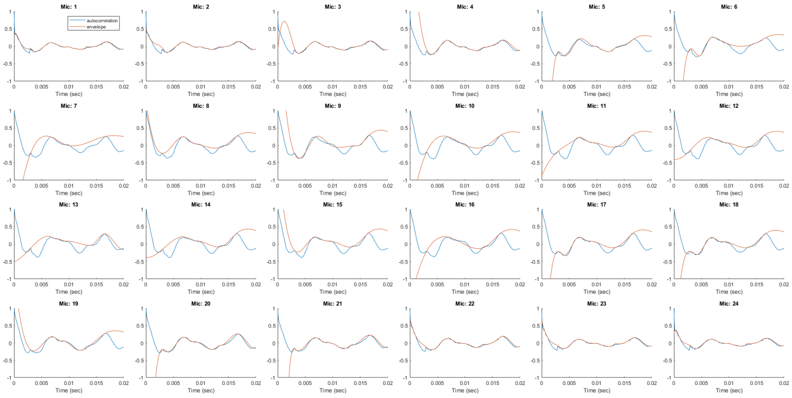
$$R_{yy}(n) = \sum_{m=-\infty}^{\infty} y^*(m)y(n+m) \tag{2}$$

For both the cases of white noise and sine sweep source signal, we perform the autocorrelation of all the signals measured by the microphone, using the Matlab function *xcorr(x)*.
We then observe the peaks of the autocorrelation. What we expect to observe is a peak reaching the value 1 at the beginning of the autocorrelation, indicating that the

signal is identical to itself when not shifted. More interesting is the second peak: it should have a high value (lower than 1), and hopefully be quite separated from the first. This second peak corresponds to the first reflection that happens in the room, and by observing where it appears in the time axis of the autocorrelation and assuming the speed of sound in air to be 343.8 m/s, we should be able to estimate where it was generated in the setup of our room.
In both the case of the white noise input and the sine sweep input, the autocorrelations are characterized by noise, in order to easily find the peaks of the signals' pace we considered the function *envelope()* inside matlab. The function is fed with a signal and returns the envelope of it computed by peaks interpolation. We decided to compute the envelope considering one every 10 peaks, hence obtaining as second highest peak in the shown plots the approximated time at which the first reflection reaches the microphone. This method is conditioned by the little amount of measurements we were able to record, the higher the number of the measurements and the better the approximations.
For the case of the white noise source, the results are similar to what is expected, and we can calculate the delay with which the first reflection reaches the microphone with respect to the direct path, as well as how much longer is the path it followed.



Considering the geometry of the setup we are able to say that the most probable first reflections are due to the floor, which is at distance $d = 1.25\,m$, to the lateral wall, $d = 1.20\,m$ and to the object of absorbing material placed inside the room, $d = 1.14\,m$. Considering a isosceles triangle with two vertices in correspondence of the speaker and the microphone and one in correspondence of the point in which the first reflection occurs, the average value of one of the sides of the triangle (which corresponds to the distance from the point of reflection to the microphone) is $l = 1.7562\,m$. But the results obtained by the white noise indicates that the distance between the first reflection point and the mic is circa $\hat{l}_n = 0.968733\,m$. This probably means that some part of the surface of the object made of absorbing material are not parallel to the walls of the room, and so the idea of using an iscosceles triangle to find the reflections is not the best idea.
For the case of the sine sweep source signal, instead, it is not possible to follow this approach: since the sine sweep is made of a pure tone (even if the frequency is continuously increasing) the autocorrelation easily finds correspondence between the signal and its shifted versions.



The result obtained with the sine sweep autocorrelation is $\hat{l}_s = 1.6641\,m$.

3 Room reflection analysis using impulse responses

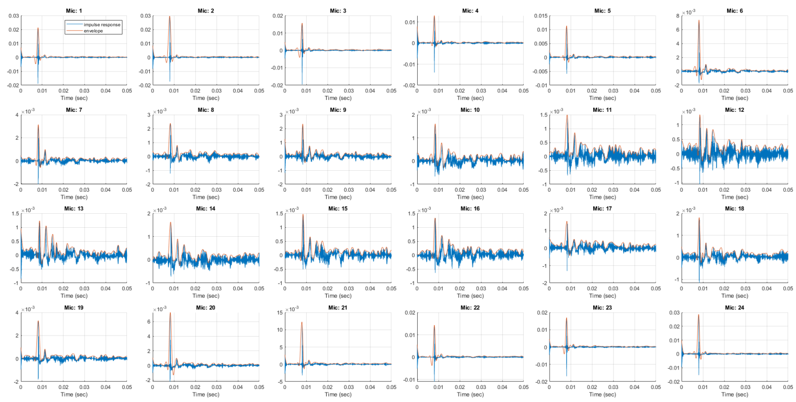
Another method we can use to find the delay of the first reflection is by analyzing the impulse response of the room.

3.1 White noise signal

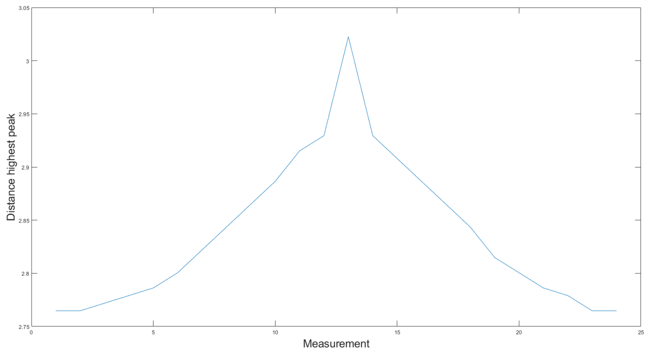
In case of a white noise source signal, we need to load in Matlab both the original signal, x, and the ones recorded from the microphone, y_i, $i = 1...24$. We then compute the Fourier transform of both signals, X and Y, using the *fft* (fast Fourier transform) function of Matlab, using as number of fft points a value equal to the sampling frequency. The frequency response function is then computed as $H = \frac{Y}{X}$. The impulse response is finally obtained by returning to the time domain with an inverse Fourier transform applied to the frequency response function:

$$ir = Re\big(ifft\{H\}\big) \tag{3}$$

These computations are done using the provided function *extractirnoise*.



The position of the first (and highest) peak describes the moment at which the direct sound reaches the microphone, knowing this temporal value and the speed of the sound we can compute the distance for the direct path. As we can see from the following plot, the computed distance for the first position of the speaker is quite similar, but not same, to the one we measured during the setup phase of the experiment: all the values are higher than the measured one. The values obtained for the different recording though follow the pace we expected: the minimum values are computed for the recordings corresponding to when the speaker points directly to the microphone, while, increasing the angle, we reach a maximum in the thirteenth position (180°).



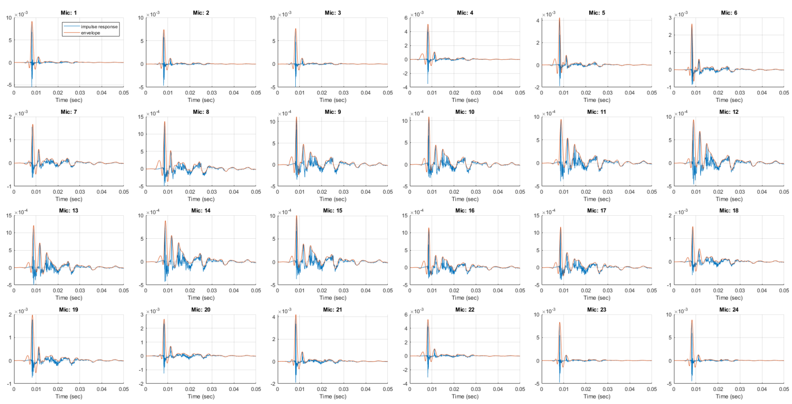
The second peak in the impulse responses corresponds to the time of arrival of the first reflection. In order to easily obtain its temporal value we applied again the envelope to these data. Knowing the time of arrival of the first reflection we can compute the distance between the reflector and the microphone. The average value of the difference between the time of arrival of the direct sound and the time of arrival of the first reflection is $\Delta t = 0.002981\,s$ and so the average distance is $\hat{d}_n = \Delta t \times c = 1.0248\,m$, while the values for each position of the speaker can be seen in detail in the Matlab script *exercise3a*.

3.2 Sine sweep signal

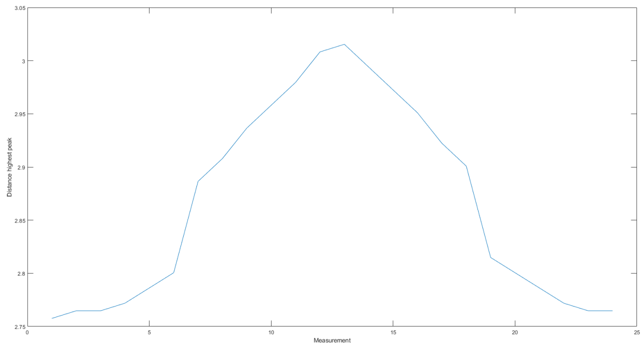
In case of a sine sweep source signal, instead of loading the original file, we use the provided function *synthSweep* to synthesize the sweep and the inverse of its fft, the latter will be called X. The function takes as input the starting frequency f_1, the final frequency f_2, the duration T, and the sampling rate f_s. In our case, $f_1 = 50Hz$, $f_2 = 22kHz$, $T = 10s$ and $f_s = 48kHz$.
To obtain the impulse response, we need to compute the deconvolution of the signal captured by the microphone. To do so we take the *fft* of the recording, Y, multiply it with X, and finally take the real part of the inverse fast Fourier transform (*ifft* function) of the result:

$$IR = Re\big(ifft(XY)\big) \tag{4}$$

These operation are performed with the provided function *extractirsweep*, which also performs a circular shift and a resizing of the signal in order not to lose samples.



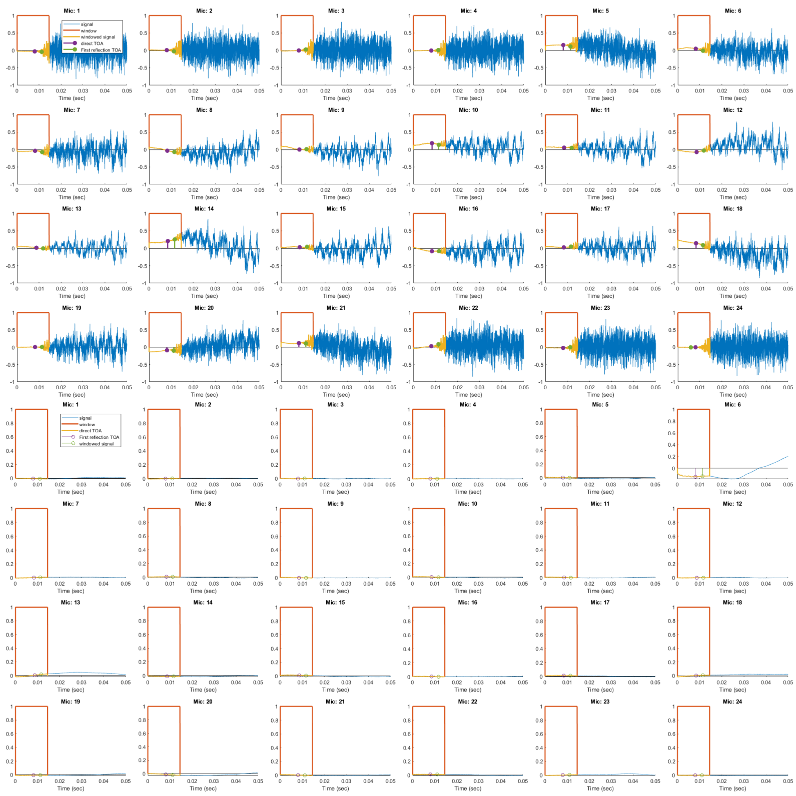
The results for the direct path are slightly different from the ones of the white noise case, as we can see from the plot below, in which the distance increase is less steep in the intermediate angles, but it is quite similar in the first position and in the one at 180°.



As the case of white noise source signal, we can look at the second peak, corresponding to the time of arrival of the first reflection, to compute the distance between the reflector and the microphone. The average value of the difference between the time of arrival of the direct sound and the time of arrival of the first reflection is $\Delta t = 0.003044\,s$ and so the average distance is $\hat{d}_s = \Delta t \times c = 1.0466\,m$, while the values for each position of the speaker can be seen in detail in the Matlab script *exercise3b*.
The results are in line with what we obtained in the previous section with the auto-correlation computed in the case of white noise input.

4 Radiance estimation directly from the signals

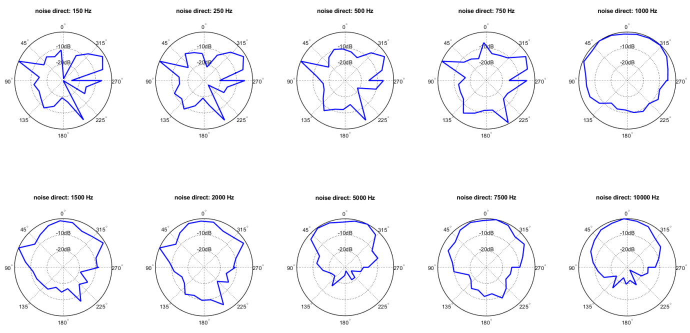
We want now to estimate the radiance of the loudspeaker. At first we try to do so working directly on the recorded signals.
We upload the recordings and we cut the signals to consider only the direct sound using a step window in order to not smooth data. The window is applied starting from the time instant 0.

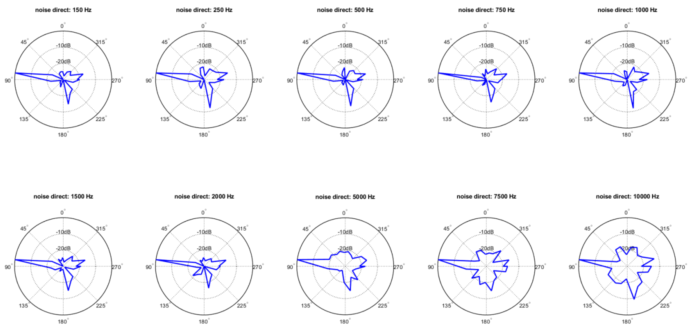


We then compute the Fourier transform Y of the windowed signal, using the *fft* function with a number of points equal to the sampling frequency f_s. The radiance pattern magnitude of the signal can be computed as follows:

$$Rad = \left|RY\right| \tag{5}$$

where R is the distance of the direct path computed for each individual recording as computed in section 3.
We can then plot the radiance pattern of the loudspeaker, centering the frequencies around a different frequency bin for each plot. To do so we use the provided function *radianceplot*, that plots the radiance pattern in polar coordinates at different frequencies.



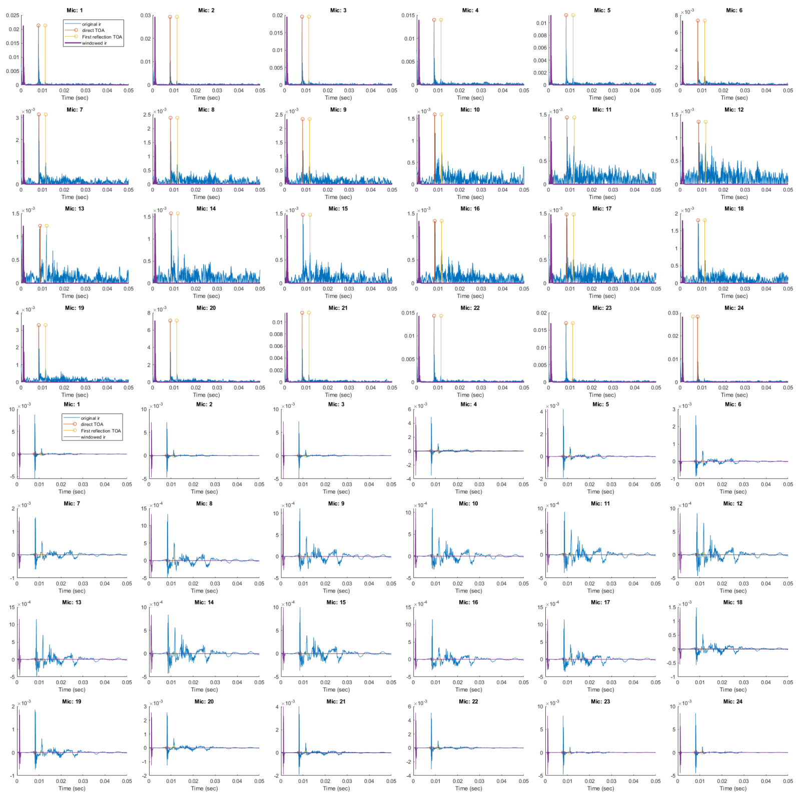


As we can see, this method does not give good results for the estimation of the radiance pattern of the loudspeaker. This is due to the fact that we are working directly on a portion of the recorded signal, with a length of the window of just 701 samples and without smoothing.
Changing the size of the window we still cannot obtain good results for the following reasons:

- by applying the window directly on the signal we are loosing information, windowing should be applied on the impulse response, which is calculated considering the whole signals so all the information at our disposal;
- by applying the window starting from time instant 0 we cannot, in general, center the window at the direct sound time of arrival and still capture only the direct sound;
- the sample length of the window affects in particular the radiation pattern obtained with the sine sweep input, this is because, acting directly on the registered signal, the window captures just the recording of the lowest frequencies. Increasing the window length we are considering more information, but we cannot pass a certain value or we will start capturing also the indirect sound.

5 Radiance estimation from the impulse responses

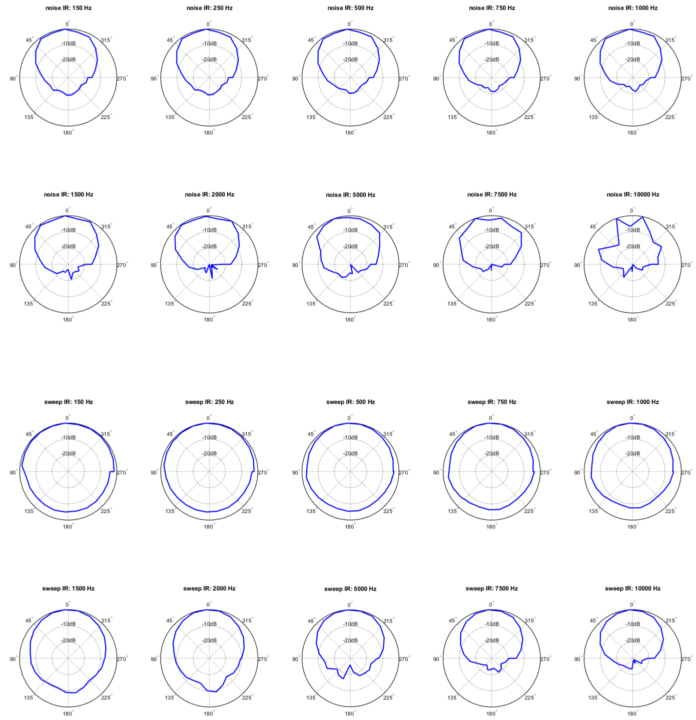
Finally, we want to estimate the radiance pattern of the loudspeaker starting from the impulse responses.
After computing the impulse response like in section 3.1 and 3.2, we need to apply the windowing. In order to take into account windows of arbitrary length, the impulse response is circularly shifted using the Matlab function *circshift* until the max peak is at the middle of the array. This prevents the left boundary of the windows to be out of the vector. We then use the information on the time of arrival of the direct path to find the correct position of the first path in the shifted signal. We allocate one array of the same length of the original impulse response, and then add samples from the windowed signal. Finally, we invert the previously applied shift to return back to the correct impulse response timing.
The window utilized this time is a Hanning window, in order to apply smoothing around the peak, with a length much smaller than the one initially used in section 4 (101 samples rather than 701).



Once we obtained the correctly windowed signal, we compute the radiance, this time as

$$Rad = \left| R \frac{Y}{X} \right| \tag{6}$$

where R is again the distance for the direct path, and X is the Fourier transform of the original signal, in case of white noise, or of the synthesized sweep, in case in sine sweep. We finally plot again the radiance pattern using the function *radianceplot*.



As we can see, the results are greatly improved from those of section 4.
Looking at the polar plots given by the white noise, we can see that the pattern stays quite unvaried at all frequency bins.
Observing instead the plots for the sine sweep, we can see that, for lower frequencies, the speaker behaves almost omnidirectionally, and it becomes more directional with the increasing of the frequencies, until it emits almost only frontally at the highest ones.
Regarding the length of the window, since we centered it on the peak of the impulse response, the only two things we need to take into account are: we need to have enough samples inside it in order to have enough information for the computations; we don't want the window to capture any part of the non direct sound.