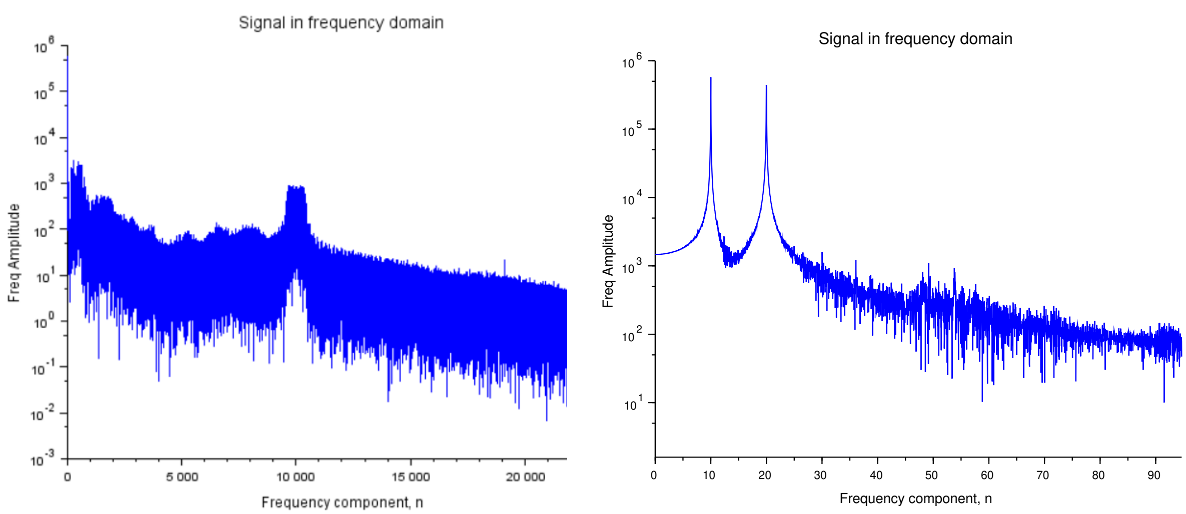
1 FIR filter design

1.1 The Task

The task was to design the filter which will suppress the noises at 10 Hz, 20 Hz and somewhere around 10000 Hz.

Given: Sample track with noise.

1.2 Inspecting the audio with noise



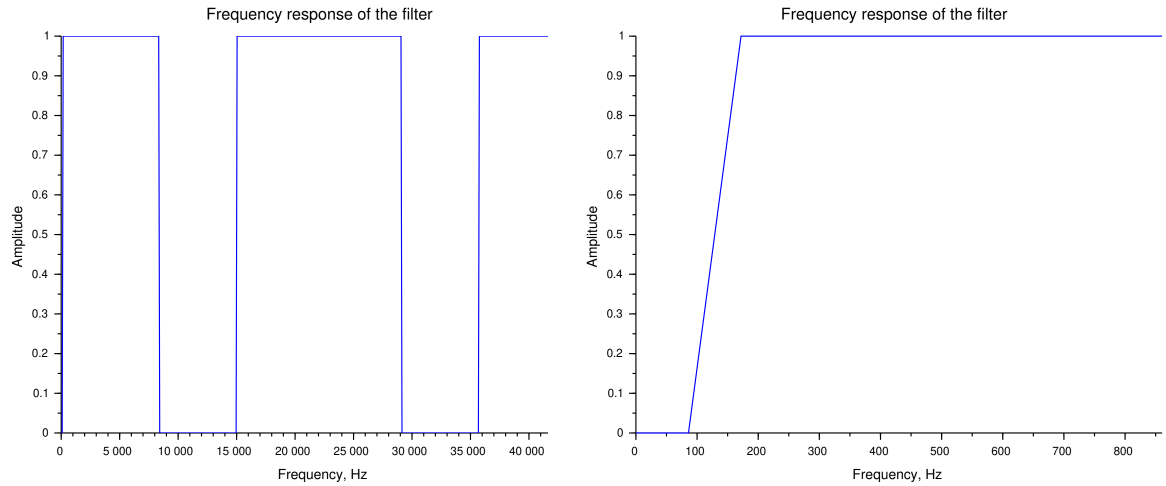
(a) Entire spectrum (b) Spectrum zoomed to 10 Hz and 20 Hz

Figure 1: Spectrum of audio with noise

We can see on Figure 1a there is a noise around 10 kHz. Also, there is a noise on 10 Hz and 20 Hz shown on Figure 1b.

1.3 Creating the desired frequency response

I decided to suppress the bands from 1 Hz to 50 Hz and from 8.5 kHz to 15 kHz.



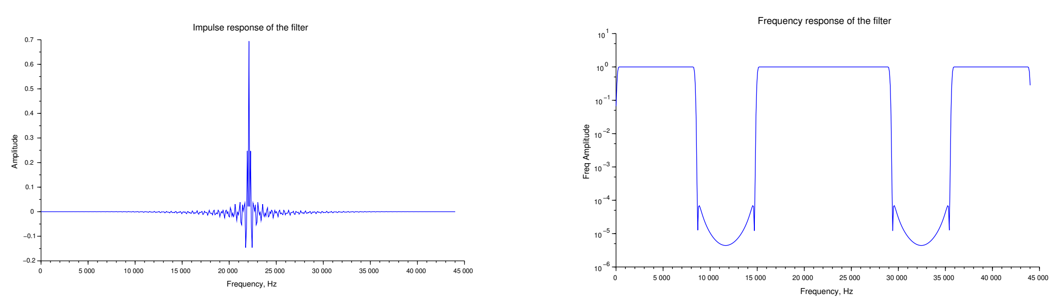
(a) Entire response (b) Response zoomed at beginning

Figure 2: Filter frequency response

1.4 Making the filter

We convert the frequency response to the time domain, shift the impulse and apply Kaiser window function with $\alpha = 8$. The filter is shown of Figure 3.

Note: graphs are in vector format so you can zoom it as much as you want.

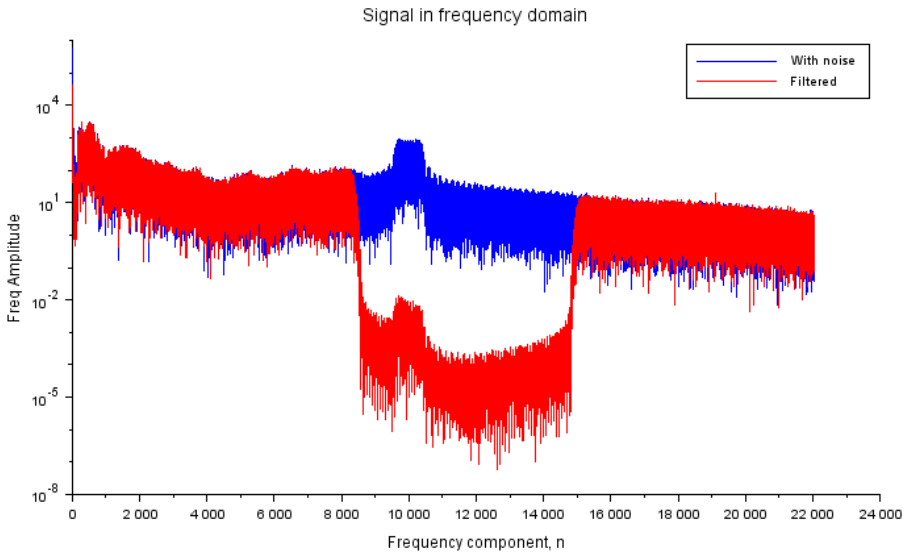


(a) Impulse response (b) Frequency response

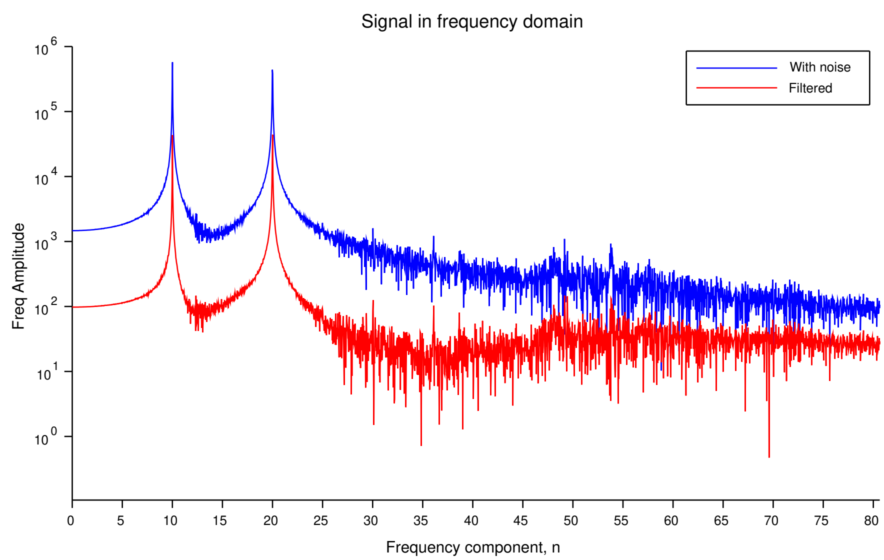
Figure 3: The filter

1.5 Applying the filter

We apply the filter by convolving the audio with it.



(a) Entire spectrogram



(b) Zoomed spectrogram

Figure 4: Comparing the initial audio with noise and result with suppressed bands

1.6 Conclusion

In this task we learned how to make band FIR filters. It shows us that there is much work should be done and creates a slight idea of how real audio processing software works.

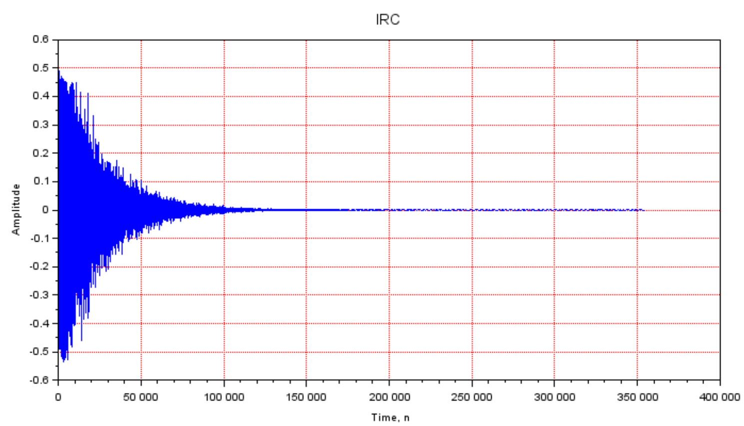
2 Room effect cancellation

2.1 The Task

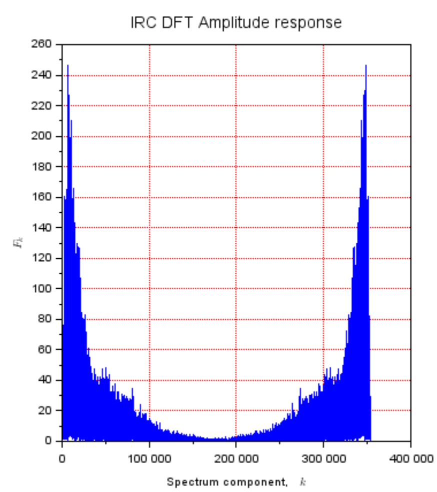
The task was to create the reverse impulse response to cancel the room effect.
Given: Sample track, IRC.

2.2 Original IRC

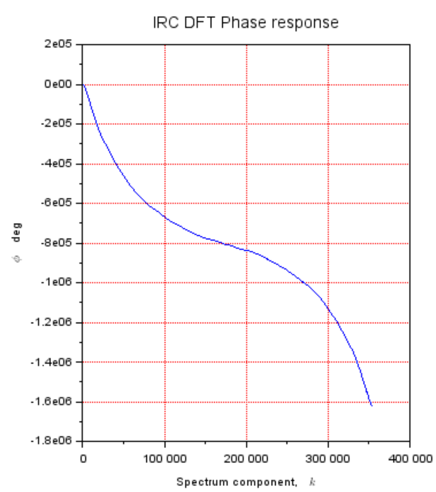
The original IRC does not tell us anything special. It is shown on Figure 5.



(a) Time domain



(b) Amplitude



(c) Phase

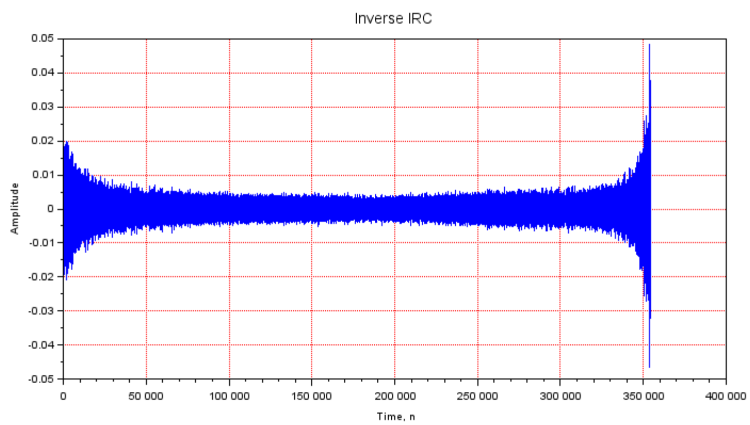
Figure 5: Original IRC

2.3 Inverse IRC

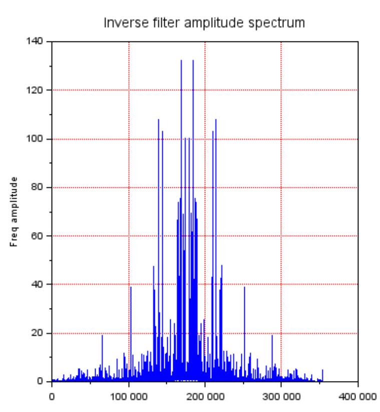
To make an inverse IRC, we have to:

1. Convert IRC to frequency domain (apply FFT)
2. Invert it (apply $1/x$ function to all items)
3. Eliminate possible infinities (it might appear if there were zeros in IRC frequency domain)
4. Convert back to time domain (apply IFFT)

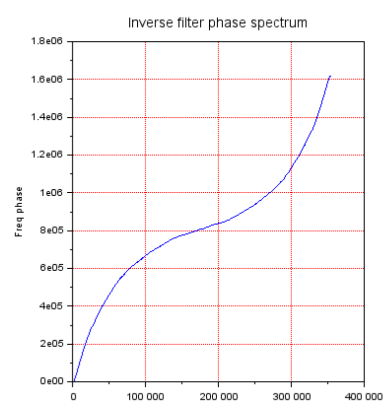
Results are shown on Figure 6.



(a) Time domain



(b) Amplitude



(c) Phase

Figure 6: Inverse IRC

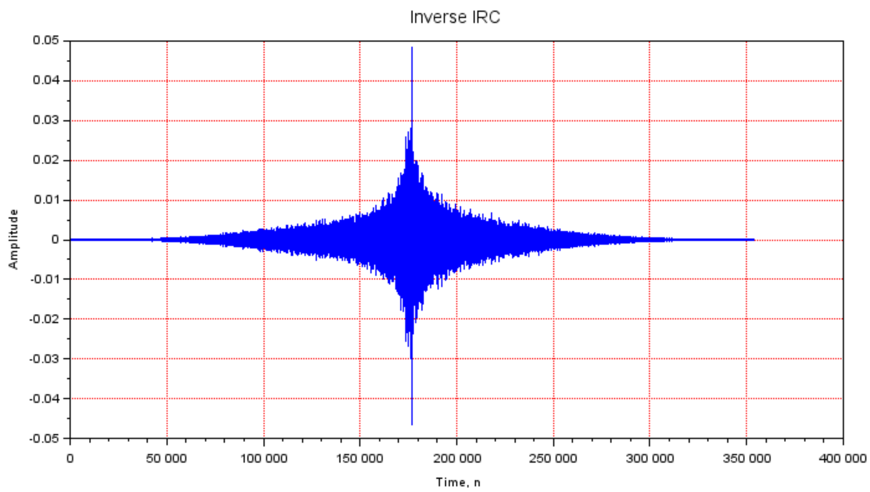


Figure 7: Shifted inverse IRC with window function applied

However, to make this filter applicable, we also have to shift it and apply the Kaiser window function with parameter $\alpha = 6$. I observed that lower window parameter leads to smaller fluctuations on $h * \tilde{h}$ (see Figure 8b), but values smaller than 6 don't make filter to fade.

The resulting filter is shown on Figure 7.

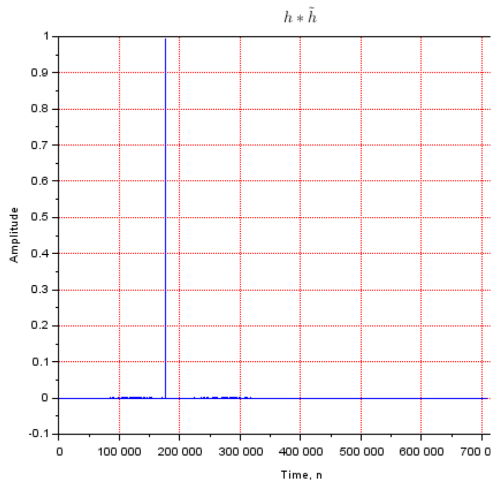
2.4 Convolving with original filter

To make sure that the designed inverse IRC filter reverses the effect of original IRC, I convolved them. The results are shown on Figure 8.

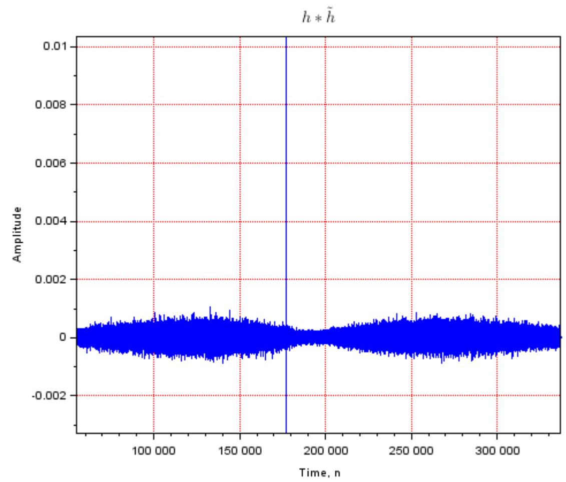
As shown on Figure 8a and Figure 8b, the peak is shifted when comparing with Kronecker delta sequence. Also Figure 8c and Figure 8d shows that amplitude and phase spectrum are not close to zero, thus the inverse IRC is not ideal.

2.5 Conclusion

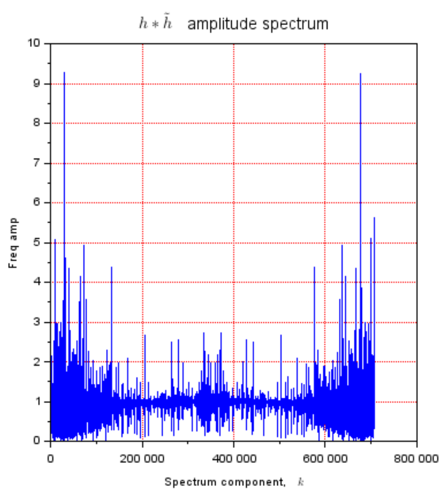
In this task we learned how to suppress the room effects in audio recordings. It shows us that it takes time to experiment with IRCs and tune the filter which can eliminate unwanted echoes and reverberations. And there is still work to be done.



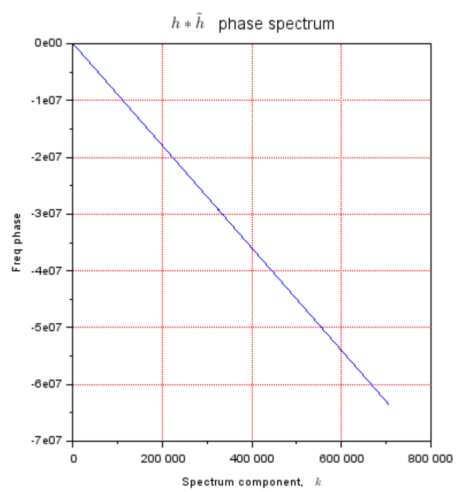
(a) Time domain



(b) Time domain zoomed



(c) Amplitude



(d) Phase

Figure 8: Convolution of original IRC and inverse IRC