Linear Systems and Signal Convolution

Innopolis University, 2020
Digital Signal Processing

Team

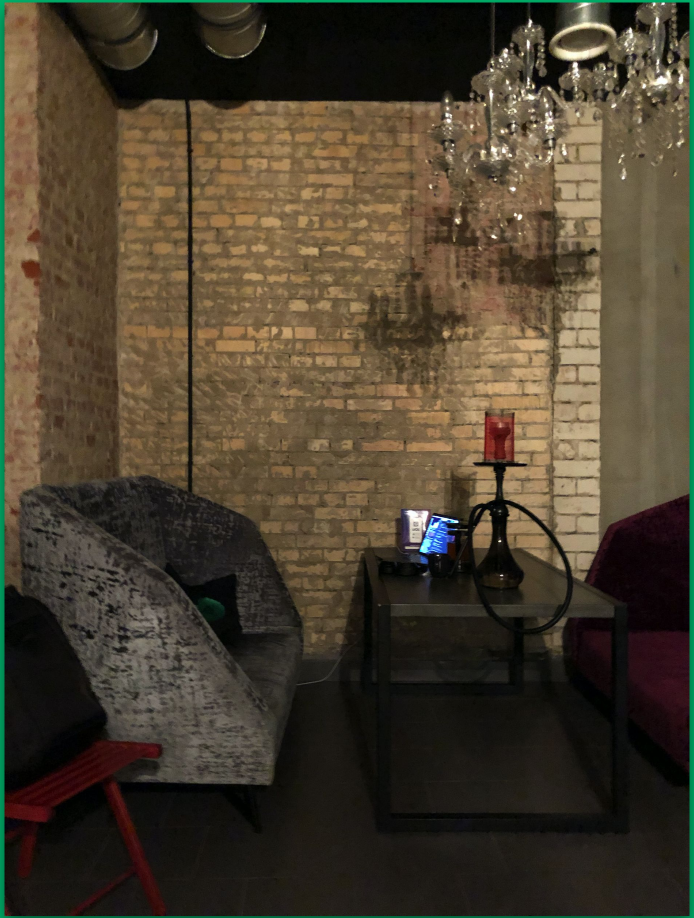
Arina Fedorovskaya

Gleb Petrakov

Task 1

Where we clapped

"Nashe Mesto", Innopolis



Custom convolution algorithm

```
function[convolved_custom] = ConvolveCustom(x, h)
    m = length(x);
    n = length(h);
    N = n+m-1;
    x = [x zeros(1, N-m)];
    h = [h zeros(1, N-n)];
    f1 = fft(x);
    f2 = fft(h);
    f3 = f1.*f2;
    convolved_custom = ifft(f3);
endfunction
```

Using frequency domain multiplication.

Thanks a lot to Prof R.Senthilkumar: https://scilab.in/lab_migration/generate_lab/8/1

Custom convolution times

voice.wav: **0.141054**

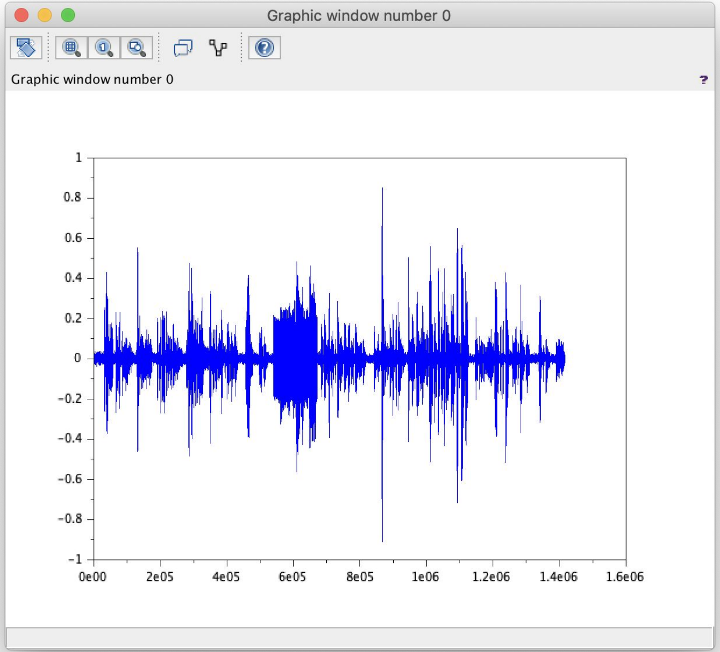
violin.wav: **1.161271**

speech.wav: **0.339162**

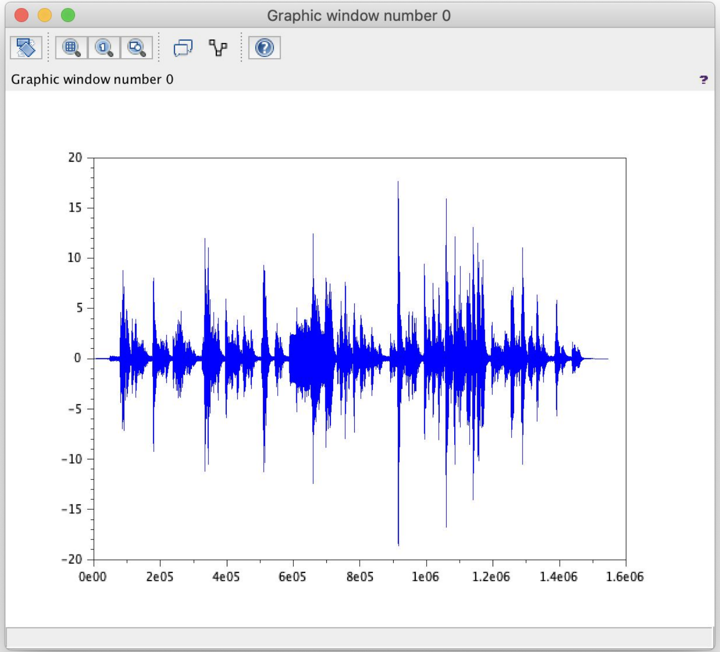
drums.wav: **0.127039**

Source track vs convolved track

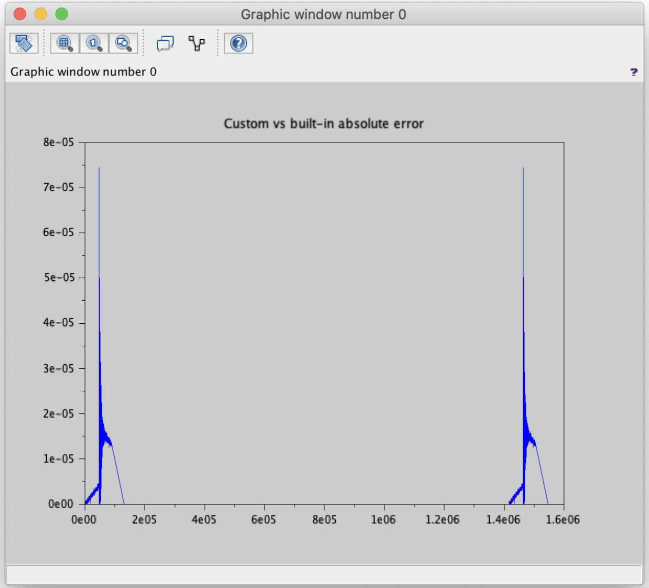
Source track



Convolved track



Default vs our convolution



That's an odd error, we do not really know why, but the error is small

Listen to our mixtape

All records available at:

https://github.com/imajou/IU-S20-DSP-Assignment3

(as well as the code)

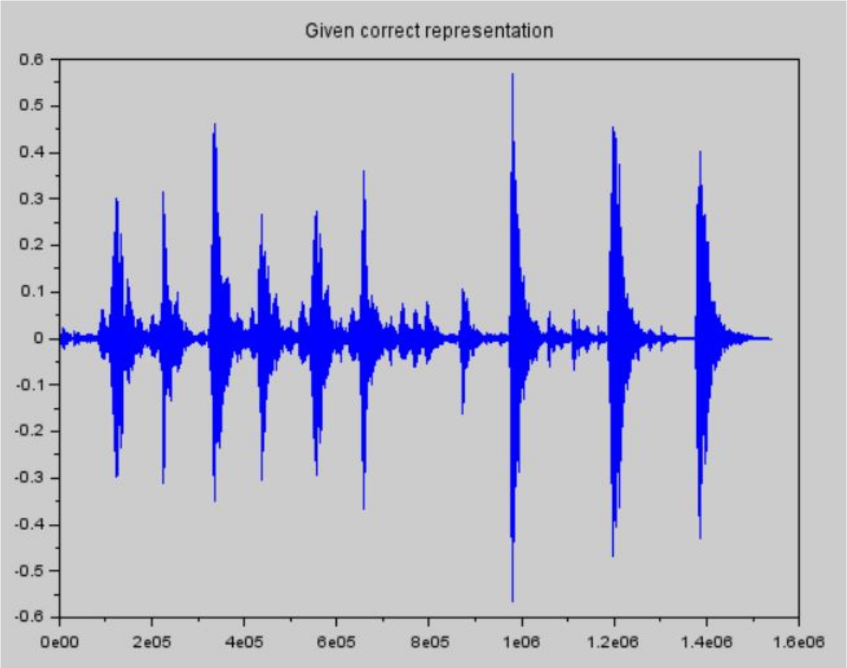
Task 2

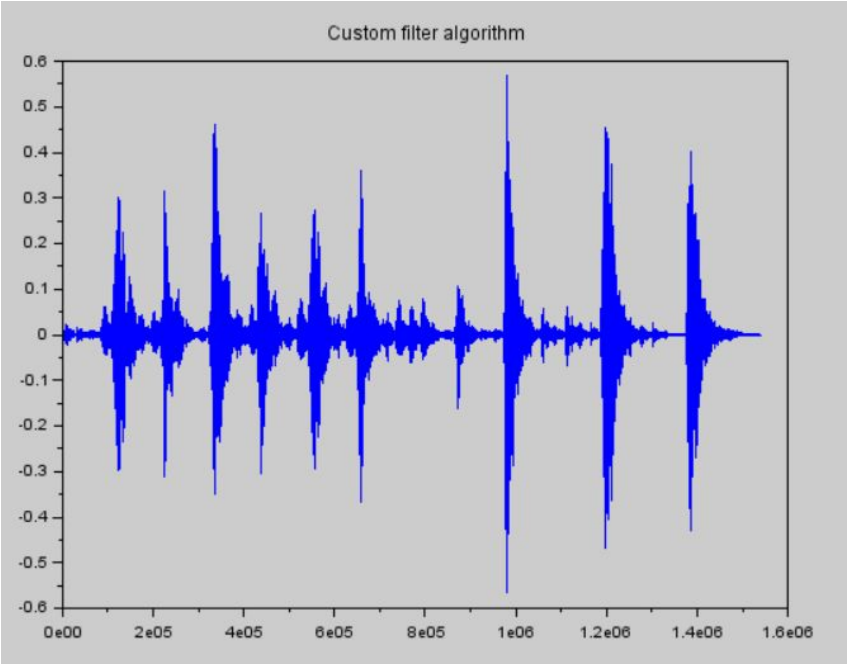
Frequency Filtering with IIR Filters

$$y[k] = \sum_{m=0}^{M} b_m \cdot x_{k-m} + \sum_{n=1}^{N} a_n \cdot y_{k-n}$$

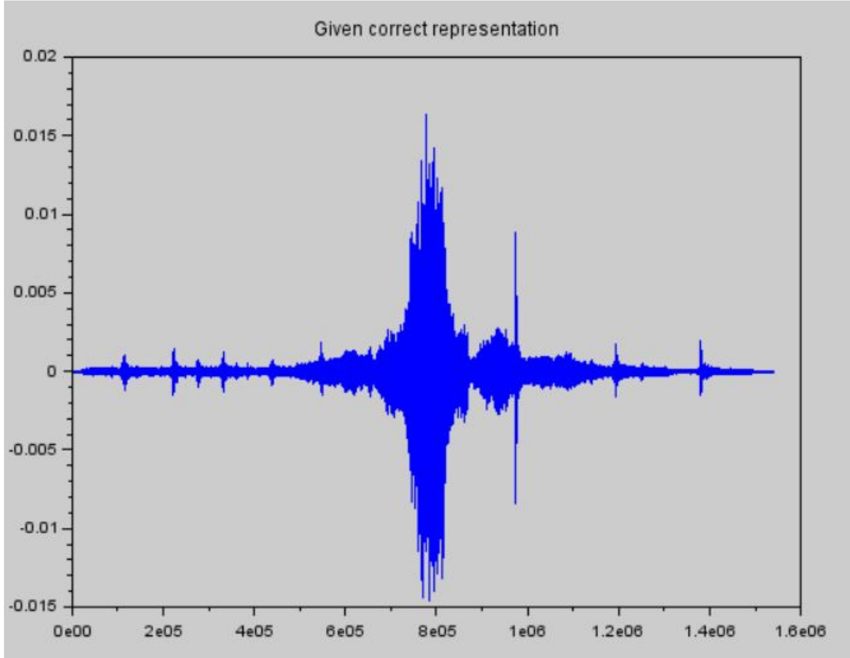
```
// Compute filter function
function [y] = compute(filt)
    m = size(filt.x, "c")
    y = zeros(1, m)
    for n=3:m
        y(n) = filt.b(1)* filt.x(n) + filt.b(2) * filt.x(n-1) + filt.b(3) *
filt.x(n-2) + filt.a(1) * y(n-1) + filt.a(2) * y(n-2)
    end
endfunction
```

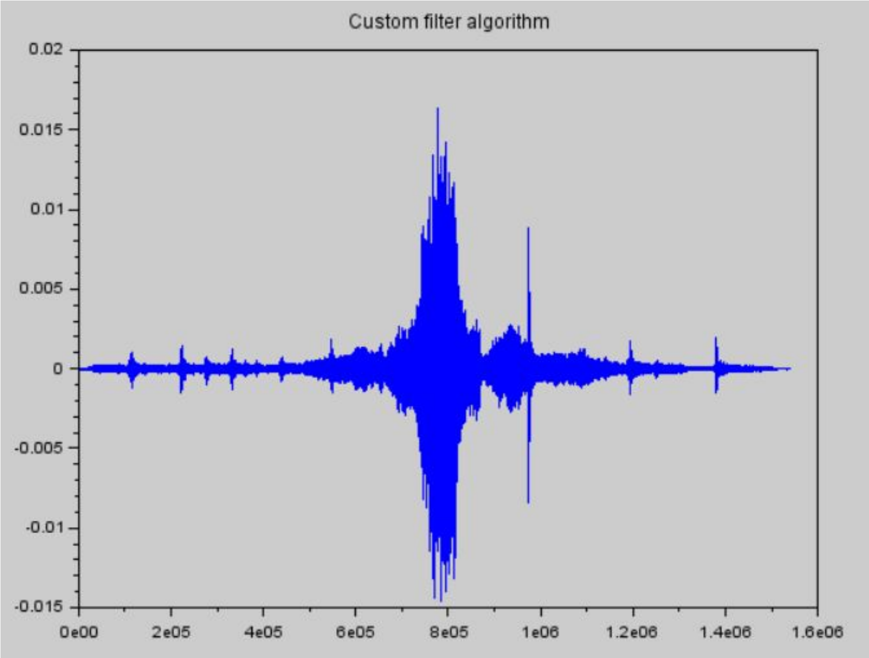
Lowpass filter





Highpass filter





This slide was intentionally left blank