Lab 3 - Digital Filters Design, Implementation and Analysis

Task 1. Write your own function to perform FIR filtering. Use the syntax y = myFIR(x,h) where "x" represents a vector containing input signal samples and "h" is a vector containing the impulse response of the filter. Function output, "y" should be a vector containing the filtered signal samples.

Note: Your function should not call Matlab built-in functions such are "filter" or "conv" but perform filtering on sample-by-sample basis using "for" loop for this purpose. This will be a very inefficient way of doing it, but efficiency is not the aim of this task.

Test your function using an FIR filter with an impulse response of {0.2, 0.3, 0.2} and input x = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Use the standard MATLAB "filter()" function to verify the operation of your own function.

Are the outputs of your function and MATLAB "filter()" function identical?

Some questions to get you thinking: What is the transfer function of your filter? H(z)=_____

What is its difference equation of your filter? y(n)=_____

Perform another test of your filtering function by filtering 10 samples long impulse sequence $\delta(n)$.

What is the output of your filter in this case? y = _____

Could you have predicted this result (without doing any calculations)? Why?

Task 2. Assuming a sampling rate of 8 kHz, generate a 1 second long signal - sum of three sinusoids as specified below:

$$x(t) = 5\cos(2\pi(500)t) + 5\cos(2\pi(1200)t + 0.25\pi) + 5\cos(2\pi(1800)t + 0.5\pi)$$

Play and listen to this signal and plot the first 100 samples of it.

Now consider two different FIR filters specified below.

FIR system 1:

h1 = [-0.0012 -0.0025 -0.0045 -0.0068 -0.0073 -0.0030 0.0089 0.0297 0.0583 0.0907 0.1208 0.1422 0.1500 0.1422 0.1208 0.0907 0.0583 0.0297 0.0089 -0.0030 -0.0073 -0.0068 -0.0045 -0.0025 -0.0012]

FIR system 2:

h2 =[0.0004 -0.0017 -0.0064 -0.0076 0.0073 0.0363 0.0458 0.0000 -0.0802 -0.1134 -0.0419 0.0860 0.1500 0.0860 -0.0419 -0.1134 -0.0802 0.0000 0.0458 0.0363 0.0073 -0.0076 -0.0064 -0.0017 0.0004]

Plot the impulse response of each filter using stem() instead of plot().

Filter the generated signal using each of those two filters and plot the input signal spectrum and the output signal spectra for each filter using the frequency resolution of 1 Hz for each plot.

What frequency components/component (frequency locations of the spectral peak(s)) are/is shown in the input and in the output signal spectra for each filter?

Listen to and compare the input signal and output signal from each filter, respectively. Discuss the obtained results.

Task 3. Use Matlab function freqz() to evaluate the frequency response of each filter. Plot the evaluated frequency responses making sure frequency units are in Hz and magnitude is expressed in dBs.

How do those results agree/relate to results you have observed in the previous task?

Task 4. Load the "we.dat" speech signal, sampled at 8000 Hz and filter it using each of the FIR filters from Task 5. Play and listen to two filtered speech signals.

Calculate and plot the input signal spectrum and the output spectra for two filtered signals with a frequency resolution of 4 Hz. Discuss the obtained results.

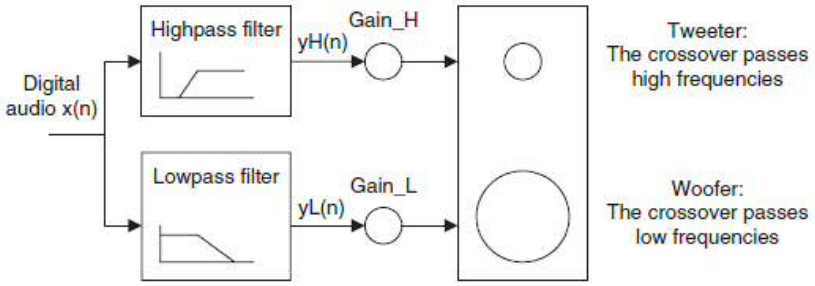
LOUDSPEAKER CROSSOVER DESIGN

In high quality audio systems, there is often a situation where the application requires the entire audible range of frequencies, which is usually beyond the capability of any single speaker driver. A combination of several drivers, such as the woofers and tweeters, each covering different frequency range, to reproduce the full audio frequency range can be employed in these situations.

A typical two-band digital crossover is shown in the figure. There are two speaker drivers. The woofer responds to low frequencies, and the tweeter responds to high frequencies. The incoming digital audio signal is split into two bands by using a low-pass filter and a high-pass filter in parallel. Separated audio signals are then amplified and sent to corresponding speaker drivers.

Hence, the objective is to design the low-pass filter and the high-pass filter so that their combined frequency response is flat, while keeping transition as sharp as possible to prevent audio signal distortion in the transition frequency range.

Although traditional crossover systems are designed using active circuits (analogue systems) or passive circuits, the digital crossover system provides a cost-effective solution with programmable ability, flexibility, and high quality.



Task 5. Design a two-band crossover system according to a specification given below and generate all the necessary plots to illustrate your design. Filter specifications are given below:

Sampling rate:	44,100 Hz
Crossover frequency:	1,000 Hz (cutoff frequency)
Transition band:	600 to 1,400 Hz
Lowpass filter:	passband frequency range from 0 to 600 Hz with a ripple of 0.02 dB and stopband edge at 1,400 Hz with attenuation of 50 dB.
Highpass filter:	passband frequency range from 1.4 to 44.1 kHz with ripple of 0.02 dB and stopband edge at 600 Hz with attenuation of 50 dB.

Hints: Use FIR rather than IIR filters for your design. Use Matlab "designfilt()" function to design two filters.

"help designfilt" should provide sufficient information to use this function properly and complete this task.

INTERFERENCE REDUCTION IN ELECTROCARDIOGRAPHY

Task 6. Load the "ecgbn.dat" signal containing raw electrocardiograph (ECG) data. You have analysed this signal in the previous lab and (hopefully) established that it's contaminated with the interference of 60, 120 and 180 Hz.

Using a zero-placement method, design three second-order notch FIR filters, each with a complex-conjugate pair of zeros in order to remove those three components from the signal. Now, use Matlab to combine three designed filters into one and filter the ECG signal. Plot the original and filtered signals.

Calculate and plot the spectra of both, raw and filtered signals as well as the frequency response of your filter to verify your design.

SIMPLE IIR FILTER ANALYSIS AND DESIGN

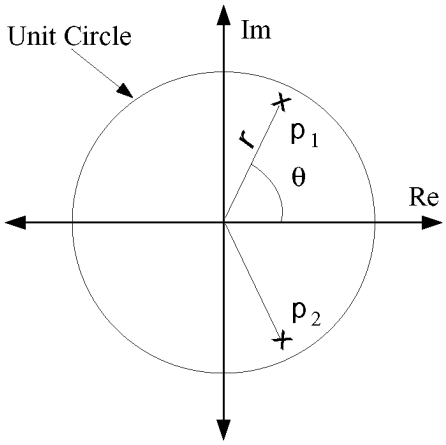
In this section we will consider a simple IIR filter with a complex conjugate pair of poles shown in figure on the right:

$$p_1 = re^{j\theta} \text{ and } p_2 = re^{-j\theta}$$

Here r is the distance from the origin, and θ is the angle of p_1 relative to the positive real axis. The transfer function for this system $H_i(z)$ is given by:

$$H_i(z) = \frac{1-r}{\left(1-re^{j\theta}z^{-1}\right)\left(1-re^{-j\theta}z^{-1}\right)} = \frac{1-r}{1-2r\cos(\theta)z^{-1}+r^2z^{-2}}$$

where $1-r$ is a normalisation constant. A causal IIR filter is stable if and only if its poles are located within the unit circle. This implies that this filter is stable if and only if $|r|<1$.



Task 7. Calculate and plot the magnitude of the filter's frequency response $|H_i(\Omega)|$ on $|\Omega|<\pi$ | for $\theta=\pi/3$ and the following three values of r: $r=0.99$, $r=0.9$, $r=0.8$.

How does the value of r affect the magnitude of the filter frequency response?

Task 8. Calculate and plot the magnitude of the filter's frequency response $|H_i(\Omega)|$ on $|\Omega|<\pi$ | for $r=0.9$ and the following three values of θ: $\theta=\pi/6$, $\theta=\pi/3$, $\theta=\pi/2$.

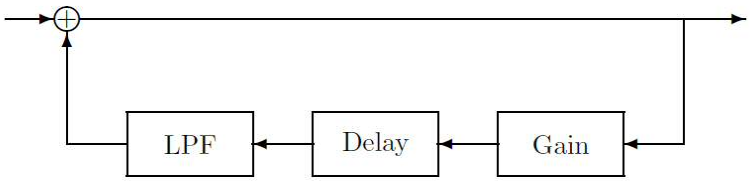
How does the value of θ affect the magnitude of the filter frequency response?

Task 9. Use the conclusions drawn in Tasks 7 and 8 and a material covered in DSP lectures to design a simple IIR filter to process and filter the signal in "pcm.mat" file. Your IIR filter should be designed in order to amplify the narrowband component present in the signal compared to the wideband, background noise. You should first analyse the signal (using "myDFT" or "fft" function), appropriately position the signal poles, obtain the z-transform and the corresponding difference equation for your filter and finally filter the signal and analyse the filter output. You can write your own code to do the filtering, or alternatively, you can use Matlab "filter" function. Listen to the signal before and after filtering, redesign if necessary and draw the conclusions.

Now use Matlab function “filter()” to obtain the impulse response of the IIR filter designed in the previous task and plot it. Use this result to generate 50, 100 and 150 coefficients long FIR filters to approximate designed IIR filter. Obtain the frequency response of each filter, plot them on a single figure, together with the frequency response of the original filter and analyse the obtained results.

PLUCKED STRING SOUND SYNTHETIS USING IIR FILTER

Task 10: A guitar-string sound can be simulated using an IIR discrete-time system shown in figure below.



Three components in the feedback path and the corresponding transfer functions are:

- LPF – two-point moving average filter, $H_{LPF}(z)=\frac{1}{2}\left(1+z^{-1}\right)$,
- Delay – N samples log delay, $H_{Delay}(z)=z^{-N}$,
- Gain - gain of the feedback path, $H_{Gain}(z)=K$, where $K<1$ for stable filter operation.

A short burst of random noise is usually used as an input to this filter. This can be generated in Matlab using: `x = [randn(1,N) zeros(1,L)];` where parameter L controls the length of the synthesized tone.

Design and implement this filter, i.e. construct filter vectors a and b and perform the filtering of the signal x. Assume sampling frequency of 8 kHz and make sure the output signal is 2 seconds long, by adjusting the parameter L when generating the input signal x. Use K = 0.98 or some similar value.

Plot the output signal and play it using “soundsc()”.

Calculate and plot the frequency response of this filter, using “freqz()” and try to relate it to filter output.

Branislav Vuksanovic, March, 2020

branislav.vuksanovic@port.ac.uk