Lab 4: Digital filters

The goal of this lab is to implement, analyse and test some simple echo effects using MATLAB.

1 Lab tasks

Problem 1 (Single echo):

A single echo FIR filter is defined by the difference equation

$$y[n] = x[n] + \alpha \cdot x[n-R]$$

for some real-value, positive constant α and some integer parameter R. Given the system's sampling period T_s, the time delay of the filter is $T_d = R \cdot T_s$.

(a) **Theory:**

(i) Write down the transfer function $H(z)$.

(ii) Calculate and plot (with pen and paper) the square of the amplitude response $\left|H\left(e^{j\omega T_s}\right)\right|^2$ within the frequency interval from 0 to ω_s. Choose $R = 3$ and $\alpha = 1/2$ for this purpose.

(b) **MATLAB implementation:**

(i) Write a MATLAB function implementing a single echo filter that takes the vector x of input values as well as the parameters α and R as input arguments and that returns the output vector y.

(ii) Set $\alpha = 1/2$ and $R = 3$. Test your function using a discrete δ-pulse $\delta[n]$ as input. Use MATLAB to compute the frequency response as the DFT of the impulse response $h[n]$ and plot its magnitude and phase between 0 Hz and f_s. Additionally choose $\alpha = 1$ and explain the observed phase response in this case.

(iii) Apply the echo filter to a sound file (*e.g.*, any .wav file). Choose R so that $T_d \approx 100ms$ as a starting point. Vary α and R and observe the respective effects.

Problem 2 (Multiple echoes):

An N-echo FIR filter is defined by the difference equation

$$\begin{aligned} y[n] &= x[n] + \alpha \cdot x[n-R] + \alpha^2 \cdot x[n-2\cdot R] + \ldots + \alpha^N \cdot x[n-N\cdot R] \\ &= \sum_{k=0}^{N} \alpha^k x[n-k\cdot R]. \end{aligned} \tag{1}$$

(a) **Theory:**

(i) Write down the transfer function $H(z)$ and simplify the expression using the geometric sum. Does $H(z)$ have poles in the complex z plane?

(ii) Use the result of part (i) to write down the difference equation of a recursive filter describing the same N-echo filter.

(b) **MATLAB implementation:**

(i) Implement both the recursive and non-recursive filters for $N = 6$ and verify that they yield the same impulse response.

(ii)
- Use MATLAB to compute the DFT of the impulse response and plot the amplitude and phase responses (Hint: If you do this correctly, you should be able to see why this type of filter is also called *comb filter*).
- Use MATLAB to create arrays representing the coefficients of the recursive filter and plot the pole/zero map using MATLAB's function `zplane` contained in the Signal Processing toolbox. Is the filter stable for $\alpha > 1$?

(iii) Apply both the recursive and non-recursive filters to a sound file. Choose R so that $T_d \approx 100$ms as a starting point. Vary α and R, particularly testing the behaviour for $\alpha > 1$. Explain the different behaviours of the recursive and non-recursive implementations in this case.