5

IIR filters

5.1 Introduction: IIR filters

IIR (Infinite (duration) Impulse Response) filters are the most common linear digital filtering structures. They process the incoming signal $x(n)$ according to the following recursive equation:

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{m=0}^{M} b_m x(n-m) \tag{5.1}$$

which can be expressed in the z-domain as:

$$Y(z) = H(z)X(z) = \frac{\sum_{m=0}^{N} b_m z^{-m}}{\sum_{k=0}^{M} a_k z^{-k}} X(z) \qquad a_0 = 1 \tag{5.2}$$

Note that the FIR-filters that were used in the previous sessions, form a special subset of the possible IIR structures, for which a_1, ..., $a_N = 0$. This means that, while FIR-filters only take into account the $M + 1$ previous values of the input signal $x(n)$ to calculate the output signal $y(n)$, the IIR filters also use the N previous values of the output signal itself in the calculation of $y(n)$. The feed-back component of the IIR filters, caused by the non-zero a_i values, allows for the design of low-order filters with very long (infinite) impulse responses, which can comply to gabarit requirements that would never be possible using a FIR filter with the same number of filter parameters. Unfortunately, this higher computational efficiency also comes

with a number of potential drawbacks that have to be taken into account in the design of the filters:

- Causal IIR filters do not have a linear phase characteristic, which means that different frequency components of the incoming signal may undergo different delays, causing temporal smearing of the signal. For some applications this so called phase distortion not allowed.
- Unlike FIR filters, the stability of IIR filters is not guaranteed, special care must therefore be taken to ensure the filter's stability.
- Small amounts of noise and rounding errors in the filter calculations (e.g. caused by the finite precision of real-life digital calculations) can have a significant influence on the IIR filter's performance, and even on its stability. It is therefore recommended to take these into account in the stability analysis of the filters.

In Matlab's 'Signal Processing Toolbox' (*helpsignal*), a number of standard filter design and analysis functions are available that can be used in the design of both FIR and IIR filters.

5.2 FIR vs. IIR: Implementation of a Simple Echo Effect

Sound waves propagate through the air at a finite speed of about $340m/s$ [1], and can reflect off objects and surfaces. This means that a given sound wave can follow a very complex path through the air, causing a human listener to hear multiple repetitions of the same sound. While in small-scale environments (rooms and small auditoria) this is perceived in the room's reverberation and ambience (this is especially noticeable in a room in which you haven't been before), large-scale environments (in the mountains in the open air, or in large cathedrals) give rise to clearly distinguishable echoes of the sound that gradually decrease in amplitude. In this exercise, we'll implement filters that simulate this echo effect.

5.2.1 Single Echo

In a first approximation of the echo effect, we'll consider a filter for which the output $y(n)$ contains the direct sound from the imput signal $x(n)$, along with a delayed and attennuated version $ax(n-D)$ of the input:

[1] The actual speed of sound is dependent on a number of parameters, such as the temperature

$$y(n) = x(n) + ax(n - D) \tag{5.3}$$

with D the time delay (in samples) of the reverberation signal, and a its corresponding scaling/attenuation factor.
Questions/Tasks:

- What type of filter (FIR or IIR) does the single-echo system correspond to? Implement this filter in Matlab.
- What is the impulse response of this system? [2]

5.2.2 Higher-Order Echoes

Even though it can be used as a very simple approximation of the echo effect, the single-echo system in the previous section does not sound very realistic when compared to natural echoes. To obtain a more life-like result, we will now implement a system with $K > 1$ successive repetitions of the signal. By considering a simple environment in which the soundwaves reflect between two (parallel) rock walls that are at a great distance from each other, we can propose a, echo system with K iterations that occur with uniform time delays iD, and corresponding exponential attenuations a^i (for each repetition, the total attennuation is multiplied by a factor a):

$$y(n) = x(n) + \sum_{i=1}^{K} a^i x(n - iD) \tag{5.4}$$

Questions/Tasks:

- What type of filter (FIR or IIR) does the higer-order echo system correspond to?
- Implement this filter in Matlab with a reconfigurable delay D, attenuation factor a and number of repetitions K. Apply the filter to a real signal ('start.wav')
- What is the impulse response of this system? How long is the impulse response?
- What happens for $a < 1.0$, $a = 1.0$, $a > 1.0$?

[2] Delay of more than $100ms$ correspond to typical echoes, while delays that are shorter than $10ms$ are typical for reverberation in smaller rooms.

5.2.3 Full echo

Based on the implementations in the previous sections, it is clear that a FIR implementation of an echo system easily leads to filters with a very long impulse response, and correspondingly high memory and computational requirements. Therefore, it is interesting to see whether an IIR implementation of the echoes would be possible, in order to lower the required system resources. First, we'll see if we can rewrite the expression of the filter into a recursive IIR form:

$$y(n) = x(n) + \sum_{i=1}^{L-1} a^i x(n - iD) \tag{5.5}$$

$$\Rightarrow \quad h(n) = \delta(n) + \sum_{i=1}^{L-1} a^i \delta(n - iD) \tag{5.6}$$

$$\Rightarrow \quad H(z) = \sum_{i=0}^{L-1} a^i z^{-iD} \tag{5.7}$$

$$\Rightarrow \quad H(z) = \frac{1 - a^L z^{-DL}}{1 - az^{-D}} \tag{5.8}$$

$$\Rightarrow \quad Y(z) = X(z) - a^L z^{-DL} X(z) + az^{-D} Y(z) \tag{5.9}$$

$$\Rightarrow \quad y(n) = x(n) - a^L x(n - DL) + ay(n - D) \tag{5.10}$$

If we now consider L $\to \infty$, then $-a^L \to 0$ for a < 1, causing a significant simplification in the implementation and memory requirements of the filter. (cf. Figure 5.1). Implement this IIR full echo filter in Matlab. Apply the filter to a real signal ('start.wav').

5.3 Notch filters

5.3.1 Operation of a notch filter

A notch filter influences a very small band around a predefined (configurable) center frequency and has no (or a minimal) impact on other frequencies. In

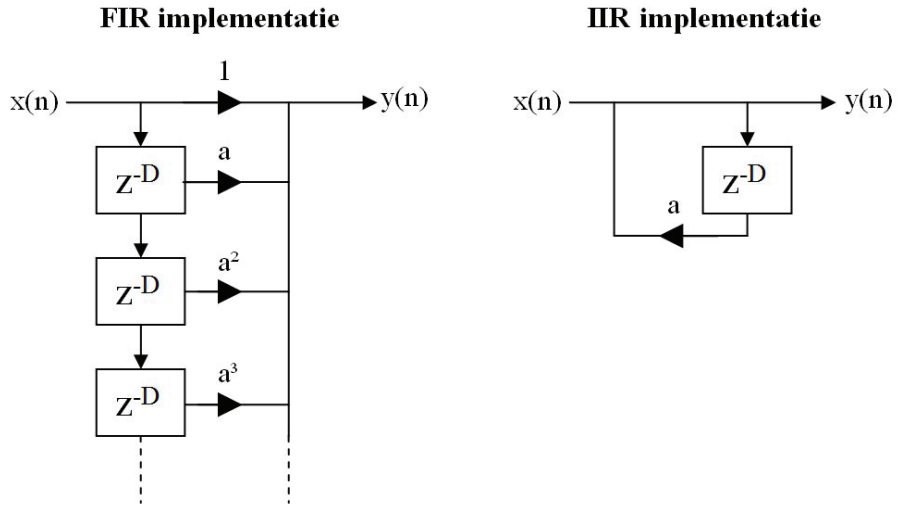

Figure 5.1: *FIR and IIR implementations of an echo.*

this exercice, we'll design a simple digtal notch filter and study its principles. First, let's consider the filter characteristic of a typical notch filter:

$$H(e^{j\omega}) = \frac{(1 - e^{j(-\omega+\omega_0)})(1 - e^{j(-\omega-\omega_0)})}{(1 - ae^{j(-\omega+\omega_0)})(1 - ae^{j(-\omega-\omega_0)})} \tag{5.11}$$

with $0 < a < 1$.

Choose $a = 0.9$ and $\omega_0 = \frac{2\pi}{5}$ and plot the amplitude- en phase characteristics of the filter (e.g. use the matlab function 'freqz').

5.3.2 Application: Remove an Unwanted 50 Hz Component

'strtnoise.wav' contains a 50Hz component and a third harmonic. Listen to the sigal. Implement the notch filter in Matlab and choose ω_0 appropriately, so it will specifically filter out the the signal's 50 Hz component. $H(e^{j\omega})$ gives us information about the steady state condition of the filter. When a signal suddenly appears, a transition period will occur and the filter will go from one steady state to another. For FIR filters this transition period lasts as long as the impulse response. For IIR filters, the impulse response can theoretically last infinitely long, so the transition period could also be infinite. Therefore, the transition period of an IIR filter is typically defined as the moment from which the output remains around a certain wanted steady

state interval. Look at the transient of the filter for a signal that first is 0 and then a 50 Hz sinus. How long does it take for the amplitude to stay smaller than 0.01 times the original amplitude?