[[1]](#footnote-1)

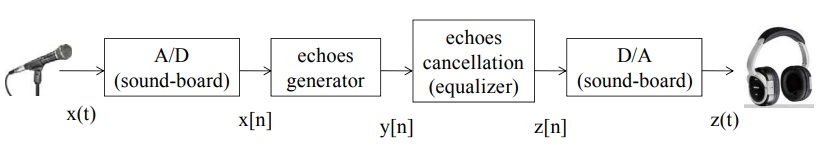
Paper Lab Session 1: Simulation of Acoustic Echoes

Aloma Cuadrado Alvarez, Iker Pirla Casals, EETAC

***Abstract*—Throughout this session we will simulate and characterize various acoustic effects using MATLAB and some basic digital signal processing tools. In short, we will simulate the acoustic reverberation in a room.**

1. INTRODUCTION

In order to carry out the laboratory session we will follow the schema represented below in *Figure 1.*

**

*Figure 1.- General scheme and block diagram*

As can be seen, this process consists of different points that will be analyzed progressively as the proposed questions are answered.

The process starts by converting a signal from analogue to digital, and then simulates the echo effect (the effect of the acoustic reverberation in a room). Later we will reverse this process to eliminate the echo effect, and finally we will convert the digital signal back to an analogue signal.

1. BACKGROUND STUDY

In order to be able to implement these blocks, it is first necessary to know how certain functions work. To do this, with the help of MATLAB we will study the most suitable ones for this exercise. These include the ‘sound’ and ‘soundsc’ functions as well as the ‘audioplayer’ and ‘audiorecorder’ functions.

The first of these functions (“sound”) is responsible for collecting the sound signal and converting it into a matrix that will be sent to the loudspeaker with a sampling frequency of 8192 Hz.

In relation to the second function (“soundsc”) scales the values of audio signal y to fit in the range from –1 to 1, and then sends the data to the speaker at the default sample rate of 8192 Hz. By first scaling the data, “soundsc” plays the audio as loudly as possible without clipping. The mean of the dynamic range of the data is set to zero. These two functions share the same purpose which is basically to play audio.

Regarding the remaining two functions, one is responsible for recording audio data from an input device such as a microphone for processing in MATLAB (‘audiorecorder’), while the other is responsible for playing audio data (‘audioplayer’). The audioplayer object contains properties that enable additional flexibility during playback. For example, you can pause, resume, or define callbacks using the audioplayer object functions.

**Echoes generation**

Firstly, we will perform a simple reverberation, using a signal whose configuration is a simple echo. The input to the function shall be a vector of signal samples (x) at frequency (fs).

The signal output will be created using the following equation:

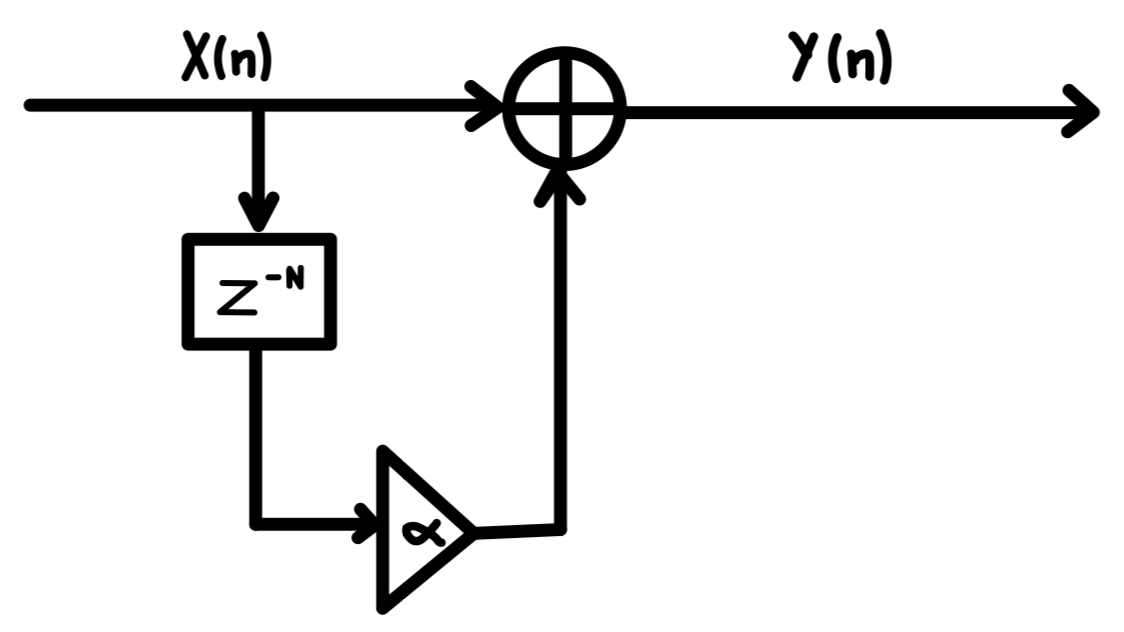
(1)

**=** Amplitude of the single echo

N= Delay of the echo in samples (s)

This signal delay is due to the direct relationship between sampling frequency and seconds.

To visualize what we are going to develop, we have made a programming diagram of this simple echo (figure 2). For this we need to know this relationship:



*Figure 2.- Programming diagram simple echo*

Analyzing the programming diagram we can see that we are dealing with a FIR system as well as a stable system since the poles are less than 1. The corresponding impulse response of the system is :

(2)

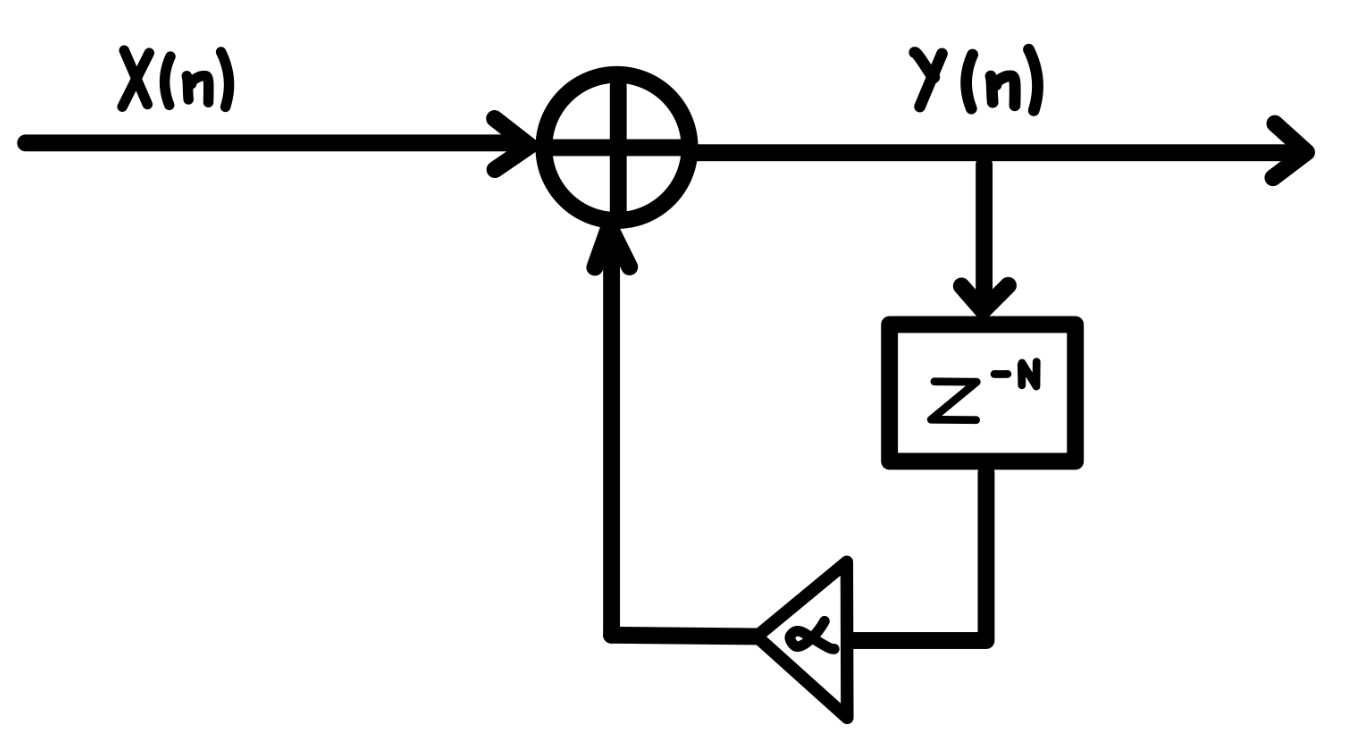
We will continue to analyze complex reverberation.This configuration corresponds to a more complex reverberation producing multiple echoes. In this case, the signal with the echoes is created using the same equation as in the simple reverberation:

(3)

**=** Amplitude of the single echo

N= Delay of the echo in samples (s)

This system will present the following configuration:



Next we will extract the impulse response of the system. First we will perform the Z-transformation and from there we will discuss what kind of system we are dealing with.

(4)

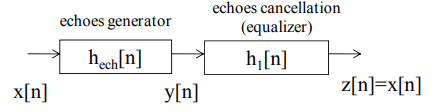
(5)

It can be seen at a glance that for the system to be stable the value of to stay within the circumference of radius 1.

Therefore with this condition we can conclude that it is an IIR system.

**Echoes cancellation (equalizer)**

In this process we simply need to reverse the previous process.



*Figure 4*

To reverse the process we simply need the multiplication of the two to be unitary, therefore:

(6)

So the condition will be satisfied when:

(7.1)

(7.2)

We check that with the (2) and (4) equations we get the result in (6).

Once we know how the simple echo equalizer works we have to make a complex one with multiple echoes, that way we will get a fully equalized sound. This is the equation that the equalizer satisfies:

Its corresponding impulsional response for h1[n] is:

(8)

Finally we check that with the impulsional response (7) and (8) satisfy the condition (6)

= 1

1. [↑](#footnote-ref-1)