LABORATORY SESSION AND BACKGROUND STUDY

INTRODUCTION AND OBJECTIVES

The main objective of this lab session is to simulate and characterize several acoustic effects using MATLAB and some basic tools of digital signal processing. More concretely, the objective is to simulate the acoustic reverberation (i.e., the echoes) in a room.

The steps to be followed by the students in the lab are described next:

- Creation of a MATLAB function that simulates a given discrete system described by a time impulse response. This function emulates the effect of a reverberation in a room.
- Creation of a MATLAB function that simulates a discrete system able to eliminate the effects of the echoes in the signal.
- Capture of a voice signal using the sound-board of the PC and MATLAB.
- Processing of the voice signal using the previously programmed functions.
- Playing the voice signal after it has been processed by the programmed systems.

BACKGROUND STUDY

The basic chain to be studied and analyzed in this lab session is the following:

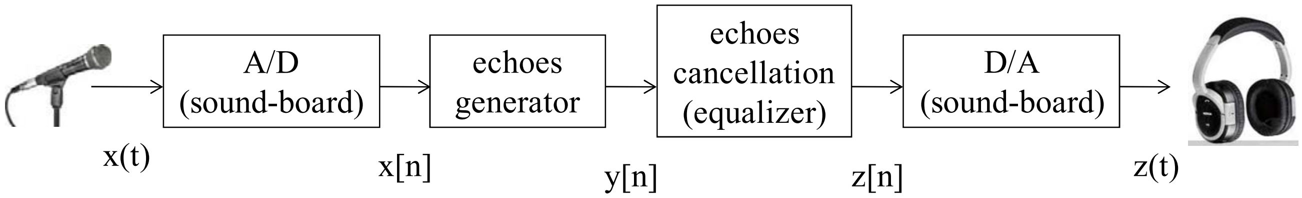

Fig. 1.- General scheme and block diagram for the setup to be used in this lab session.

Description of blocks (all these blocks will be implemented as functions in MATLAB):

- **A/D:** digitalization of the analog signal using the sound-board of the PC. The output from this block will be a vector in MATLAB containing the samples of the signal.
- **Echoes generator:** this block will be implemented as a function. This block will simulate the effect of the acoustic reverberation in a room by creating echoes in the signal. The input signal x[n] will be the "clean signal" and the output signal y[n] will be the "dirty signal", i.e., the signal with the echoes (both the input and output signals will be vectors containing samples). Two different echoes configurations will be implemented: *"simple echo"* and *"multiple echoes"*, as will be explained later.
- **Echoes cancellation:** this block will be implemented as a function. The objective of this block is to remove the echoes from the "dirty signal". Therefore, the input signal y[n] will be the "dirty

signal" and the output signal should be the "clean signal" $z[n]=x[n]$ (both the input and output signals will be vectors containing samples).

- **D/A:** this block produces the analog signal from an input digital signal (i.e., a vector containing samples) using again the sound-board of the PC and the corresponding MATLAB function.

Background study question 1: using the MATLAB help and manual, check the functions "sound", "soundsc", "audioplayer", and "audiorecorder". Which are the input and output parameters for each function? How should they be called to implement the A/D and D/A converters? What is the difference between "sound" and "soundsc"?. Read the help of "audioread" and "audiowrite".

ECHOES GENERATION

In this background study the students are asked to analyze theoretically 2 different echoes configurations that they will have to simulate in MATLAB in the lab. Each of these systems corresponds to a different reverberation of the acoustic signal (remember that 'reverberation' is a synonym of 'echoes').

Simple reverberation - simple echo

This configuration corresponds to a simple echo, i.e., the output signal will be created using the following equation:

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

where α is the amplitude of the single echo and N is the delay of the echo in samples.

Background study question 2: how is N related with the delay of the echo in seconds (Δ) and the sampling frequency in Hz (f_s)?

Background study question 3: plot the programming diagram for this system.

Background study question 4: which is the impulse response for this system $h_{echo}[n]$? Is it a FIR or an IIR system? Is it stable?

Complex reverberation - multiple echoes

This configuration corresponds to a more complex reverberation producing multiple echoes. In this case, the signal with the echoes is created using the following equation:

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

where α is the multiplicative factor for the amplitudes of consecutive echoes and N is the delay between consecutive echoes in samples.

Background study question 5: plot the programming diagram for this system.

Background study question 6: which is the impulse response for this system $h_{echoes}[n]$? Is it a FIR or an IIR system? Is it stable? Under which condition is it a stable system?

ECHOES CANCELLATION (EQUALIZER)

A *zero-forcing equalizer* is a system (with impulse response $h_I[n]$) which is able to counteract the effects of another previous system that introduces echoes in the signal (with impulse response $h_{ech}[n]$), i.e., the output from the second system z[n] should be equal to the input of the first system x[n]. This is plotted in the following figure:

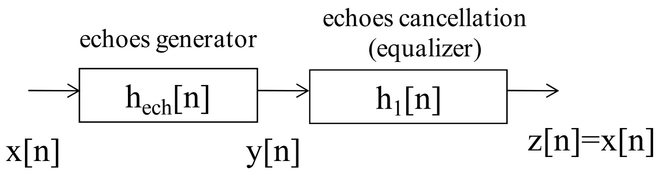

Fig. 2.- Concatenation of the systems corresponding to the echoes generation and equalization.

Background study question 7: which condition should be satisfied by the impulse responses $h_{ech}[n]$ and $h_I[n]$ so that the second system equalizes the first one?

Equalizer for the simple reverberation - simple echo

It can be proved that for the case of a simple echo, the equalizer will be a system satisfying the following equation:

$$z[n] = y[n] - \alpha \cdot z[n-N]$$

Background study question 8: which is the impulse response for the equalizer $h_I[n]$?

Background study question 9: check that the impulse responses found in the “background study questions 4 and 8” satisfy the condition found in the “background study question 7”.

Equalizer for the complex reverberation - multiple echoes

It can be proved that for the case of multiple echoes, the equalizer will be a system satisfying the following equation:

$$z[n] = y[n] - \alpha \cdot y[n-N]$$

Background study question 10: which is the impulse response for the equalizer $h_I[n]$?

Background study question 11: check that the impulse responses found in the “background study questions 6 and 10” satisfy the condition found in the “background study question 7”.

LABORATORY SESSION

In the lab session the students are asked to create a function for each of the blocks corresponding to Fig. 1. The steps to be followed are described next:

1.- Create a function called *analog_to_digital* with the following specifications:

- *Description*: this function has to convert a signal from the analog domain to the digital domain using the sound-board of the PC. A microphone has to be connected.
- *Input parameters*:
  - duration in seconds of the signal to be converted from analog to digital domain (in seconds)
  - sampling frequency f_s (in Hz).
- *Output parameters*:
  - a vector containing the samples of the digital signal.
- *Note*: use the "background study question 1".

2.- Create a function called *digital_to_analog* with the following specifications:

- *Description*: this function has to convert a signal from the digital domain to the analog domain using the sound-board of the PC. Headsets have to be connected.
- *Input parameters*:
  - vector containing the signal samples.
  - sampling frequency f_s (in Hz).
- *Output parameters*: there are no output parameters
- *Note*: use the "background study question 1".

3.- Use the function *analog_to_digital* to save a voice signal with a duration of 10 seconds. Then, use the function *digital_to_analog* to play such saved voice. You can state a sampling frequency of 30 kHz.

4.- Create a function called *simple_echo_generation* with the following specifications:

- *Description*: this function has to generate a signal with a single echo.
- *Input parameters*:
  - vector containing the signal samples of the clean signal (i.e., without echoes).
  - sampling frequency f_s (in Hz).
  - amplitude of the echo α.
  - delay of the echo Δ (in seconds).
- *Output parameters*:
  - vector with the signal samples of the dirty signal (i.e., with the single echoe). The length (number of samples) of this output signal vector has to be the same as the input signal vector.
- *Note*: use the "background study questions 2, 3, and 4".

5.- Use the previous function (step 4) to generate a single echo in the signal obtained in step 3. Choose a proper value for the amplitude and the delay of the echo. Then, listen to the dirty signal with one echo using the function *digital_to_analog*.

6.- Create a function called *multiple_echoes_generation* with the following specifications:

- *Description*: this function has to generate a signal with multiple echoes.

- *Input parameters*:
  - vector containing the signal samples of the clean signal (i.e., without echoes).
  - sampling frequency f_s (in Hz).
  - amplitude of the echoes α.
  - delay of the echoes Δ (in seconds).
- *Output parameters*:
  - vector with the signal samples of the dirty signal (i.e., with multiple echoes). The length (number of samples) of this output signal vector has to be the same as the input signal vector.
- *Note*: use the "background study questions 5 and 6".

7.- Use the previous function (step 6) to generate multiple echoes in the signal obtained in step 3. Choose a proper value for the amplitude and the delay of the echoes. Then, listen to the dirty signal with multiple echoes using the function *digital_to_analog*.

8.- Create a function called *simple_echo_equalization* with the following specifications:

- *Description*: this function has to eliminate the echo in a signal with a single echo.
- *Input parameters*:
  - vector containing the signal samples of the dirty signal (i.e., with one echo).
  - sampling frequency f_s (in Hz).
  - amplitude of the echo α.
  - delay of the echo Δ (in seconds).
- *Output parameters*:
  - vector with the signal samples of the clean signal (i.e., without the single echo). The length (number of samples) of this output signal vector has to be the same as the input signal vector.
- *Note*: use the "background study questions 8 and 9".

9.- Apply the previous function (step 8) to the signal with one echo generated in step 5. Then, listen to the clean signal using the function *digital_to_analog*.

10.- Create a function called *multiple_echoes_equalization* with the following specifications:

- *Description*: this function has to eliminate the echoes in a signal with multiple echoes.
- *Input parameters*:
  - vector containing the signal samples of the dirty signal (i.e., with multiple echoes).
  - sampling frequency f_s (in Hz).
  - amplitude of the echoes α.
  - delay of the echoes Δ (in seconds).
- *Output parameters*:
  - vector with the signal samples of the clean signal (i.e., without the mutiple echoes). The length (number of samples) of this output signal vector has to be the same as the input signal vector.
- *Note*: use the "background study questions 10 and 11".

11.- Apply the previous function (step 10) to the signal with multiple echoes generated in step 7. Then, listen to the clean signal using the function *digital_to_analog*.