Electrical Engineering 3TP4
Lab 2
Signal Convolution

In this lab we use Matlab to compute and study signal convolution. First, some Matlab convolutions are computed and compared with their hand-computed results. An audio file is then read into Matlab and we simulate a telephone network channel where an echo creates a distorted version of the original audio signal. This is done both directly and by forming the impulse response of the echo channel. The echo parameters are then varied to test acceptable echo levels using subjective listening tests. We then form the impulse response of a reverberation channel and test its impact using subjective listening tests.

1 Preparation

1. Make sure that you attend the lectures where is lab is introduced.
2. Signal Convolution Using Matlab: Discrete-time convolution can be easily computed in Matlab using the `conv` function. If `A` and `B` are vectors, then `y = conv(A, B)` returns a vector `y` which is the convolution of `A` and `B` and is of length equal to `length(A) + length(B) - 1`.

2 Experiments

1. Start by doing Question 2.7 from the textbook. In this question you are given three sets of discrete-time signals `x[n]` and `v[n]`. In Part (a) you manually compute the convolutions of `x[n]` and `v[n]`. Then in Part (b) you use the Matlab `conv` function to verify your results. In Part (b), write M-files which do a stem plot of `x[n]`, `v[n]` and `x[n]` * `v[n]` (Use the subplot command to plot them above each other). Include the M-files and your plots in your writeup.
2. Download the audio file, `speech.wav` (in wav format) from the course web site. Make sure that you can play the file on your PC or laptop. If you are using a laptop it is probably better to listen to the sound using a headset or ear-plugs rather than the built-in speakers.
3. Read in the audio file using a Matlab M-file. This can be done as follows.

```
[signal, Fs] = audioread('speech.wav');

L = length(signal);   % Number of samples in the signal.
T = 1/Fs;             % Sampling period in seconds
t = [0:L-1]*T;        % Time vector in seconds
```

After executing these statements, `signal` is a column vector containing the speech samples from the file, `Fs` is the sampling frequency and `L` is the number of samples.

4. Assume that this audio signal is transmitted through a telephone network where distortion in the form of an echo of the signal is created. The person listening hears the original signal plus an echoed version at a reduced amplitude factor, α, i.e., if the original signal is $f_s(t)$, then the received signal heard by the listener is $f_r(t) = f_s(t) + \alpha f_s(t - t_e)$, where t_e is the echo delay in seconds.

   In Matlab, define a variable, `Te`, which is the echo delay *in msec*, and use `signal` to create the received signal, `signalplusecho`. You can do this using a shifted version of `signal`. Then you can create a new wav file as follows.

   It is a good idea to first re-scale your output data to ensure that the magnitude of any of the samples does not exceed 1. Otherwise, clipping of the signal may occur when playing the sound file (Matlab may give you a warning if this happens). To re-scale, do the following to your output sound vector `signalplusecho`.

   ```
   signalplusecho = signalplusecho/max(abs(signalplusecho));
   ```

   Then you can write a new wav file, `speechwithecho.wav`, using the following.

   ```
   audiowrite('speechwithecho.wav', signalplusecho, Fs);
   ```

   You should now be able to listen to the distorted signal by playing the `speechwithecho.wav` audio file.

5. Write a Matlab M-file that creates the echo described above but instead uses convolution. To do that you need to find the impulse response, `IR`, of the system, i.e., the output of the system when an impulse is applied, will be an impulse at time $n = 0$ and a second weighted impulse at a time equal to the desired echo delay. Explain your choice of impulse response and include that in your writeup.

   Now you can use `conv` to take the convolution of `IR` and `signal`. Save the result as a new wav file and verify that things are working properly.

6. Experiment with different values of `Te` when `alpha` is equal to 1. How small does `Te` have to be before the quality of the speech acceptable? Does your answer change when the value of `alpha` is decreased?

7. Design an impulse response vector that will create “reverberation”. This is when the resulting signal contains multiple echos of decreasing intensity. One way of doing this is to create echoes which are exponentially decreasing in amplitude, i.e., you can create echos using

$$\sum_{i=1}^{N_e} \alpha^i fs(t - it_e),$$

   where N_e is the number of echos.

   Repeat Part 6 using your reverberated signal. Include your M-file in your writeup.

Writeup: Submit a writeup for the lab. Each group (2 maximum) is responsible for their own experiments and writeup. Include in your writeup a description of everything that you did including all data, Matlab programs and graphs. Relate the experimental results to theory in as much detail as possible.