**IN3031 DSP and Audio Programming Coursework report**

* **Student name, ID and cohort:** Youssef Ayman Abdelmoamen (190054712) - UG
* **Google Drive folder:** <https://colab.research.google.com/drive/1pd59VQDRIQT01VikPfViQa8wOii_oRB7?usp=share_link>

**Introduction**

This report presents the answers to two programming problems: a programmable delay line and a classifier for digit recognition. By modulating the delay period using a sine function at 2Hz and combining the output with the original, an offline delay function was made that added a flanging effect.

In the second task, a 2D-correlation-based digit classifier was developed, and a nearest-neighbor technique was utilized to evaluate multiple versions using a dataset made available on Moodle.

In addition, a thorough description of the frequency response of our programmable delay line was given. Overall, a thorough justification of the solutions as well as any particular points of interest for assessment were given.

**Part 1**

Using a delay line, the accompanying code illustrates how to build a flanger effect. A form of audio effect called flanging involves blending the original signal with a delayed version of it. To produce a sweeping effect, a sine wave or other low-frequency oscillator modulates the delay time.

The NumPy and SoundFile libraries are used to implement the code in Python. Modulate\_delay(), mix\_signals(), and process\_audio() are the main functions within the code. A sine wave is used to alter the delay time by modulate\_delay(). The mix\_signals() function blends the delayed and original signals with feedback gain. The delay line with flanging effect is applied to the input audio signal as part of the process\_audio() function.

The sampling rate and maximum delay time are set by the code after it has loaded an audio file for input. Following the definition of the delay line buffer, a sine wave is used to modulate the delay time. The process\_audio() function is used to process the input signal after which a delay line with flanging effect is applied. In order to compare the input and output signals, the output signal is written to a file after the input signal.

The code uses global variables, which might be problematic in bigger applications, it should be mentioned. Furthermore, the code doesn't deal with errors like a missing input audio file. Future revisions of the code can make improvements in these areas.

Moving on to the image classification task, the code determines the most likely label for each test image by using the correlation between the training and test images. This is accomplished by matching each test image with the training image that has the greatest correlation. This is done by comparing each test image to the collection of training images, using the correlation with each training image without an offset. The system's accuracy is calculated as a percentage of the test labels that were correctly estimated over all test images.

By using negative photos, images that are slightly rotated in the training set, and images that have noise introduced, the code also implements variations of the digit recognition system. In the end of the accuracy of the original system was compared with the accuracies of the other variants. The original system has an accuracy of 90.14% which is a really great score to achieve but the negative variant of the digit recognition system had the highest accuracy of 90.88%.

Future versions of the code could include experimenting with various methods of image enhancement. We included two different forms of picture enhancement in this code: image rotation and noise addition. However, there are numerous other methods, such as cropping, scaling, flipping, and colour transformations, that can be used to enhance images. The accuracy of the digit identification system could be increased even more by experimenting with various picture enhancement approaches.

**Part 2**

By studying the impact of the delay line on the spectral content of the input signal, it is possible to analyze the frequency response of the adjustable delay line with flanging effect. A sinusoidal signal is used to modulate the delay duration, which results in a time-varying phase shift and a comb filter effect in the frequency domain. This produces the flanging effect.

The frequency response of the delay line changes over time when the delay time is modulated by a sinusoid, which causes interference between the delayed and original signals at various frequencies that is both constructive and destructive. Depending on the frequency of the modulation signal, this results in frequency peaks or notches appearing in the spectrum at regular intervals.

The Fourier transform concept can be used to examine the frequency response of the controlled delay line. Using the Fourier transform, a signal can be broken down into its frequency components, allowing us to see the signal's spectral content.

Depending on the value of the delay time control parameter at each sample, the controllable delay line delays the input signal by a variable amount. After then, the original signal and the delayed signal are combined with a feedback gain that modifies the comb filtering effect's strength.

The frequency response of the delay line is a comb filter with notches at regular intervals in the spectrum when the delay period is not modified (i.e., constant delay). The depth of the notches is related to the feedback gain, while the distance between them is inversely proportional to the delay time.

The frequency response of the delay line changes over time when the delay time is modulated by a sinusoid, producing a time-varying comb filter effect. Depending on the frequency of the modulation signal, the frequency notches oscillate up and down, and their depth and spacing change with time.

A spectrogram, which is a representation of the signal's time-frequency spectrum, can be used to see the frequency response of a programmable delay line with flanging effect. We can examine how the frequency response of the delay line changes over time by looking at the spectrogram, which displays the spectral content of the signal as a function of time.

The output signal's spectrogram shows the formation of periodically spaced frequency notches or peaks that match to the comb filtering effect that the delay line introduced. The delay duration, feedback gain, and frequency of the modulation signal all affect the spacing and depth of the notches.

Overall, the frequency response of the controllable delay line with flanging effect can be described as a time-varying comb filter with regularly spaced frequency notches or peaks, depending on the modulation frequency. The notches' position in the spectrum changes over time due to the modulation of the delay time, and their spacing and depth are defined by the feedback gain and delay time parameters. The flanging effect can be utilized to produce a distinctive, whirling sound that is frequently used in music creation and audio effects processing.