[redacted]

Individual Data Stream Coursework Report

Kai Sun Anson Lam [redacted]

https://colab.research.google.com/drive/1vdgGJn16w_q-bU_33-Nb83039t9ApzsT?usp=share_link

1. Introduction

In this coursework, I have completed 2 major tasks, Common DSP: Controllable Delay Line and Data: Digit Recognition. My approaches to both tasks are detailed in the following sections. Please be noted that discussion and description of frequency response of controllable delay line are included in the Common DSP: Controllable Delay Line section.

2. Common DSP: Controllable Delay Line

a) Data

The mono ‘carrier.wav’ file and stereo ‘clavinet.wav’ file in the audio directory are used as inputs of 2 separate implementations.

b) Methodology

Delay Function

The Delay Function is a controllable delay line, with the only input argument being any input file. After creating an input signal, the Delay Function checks its shape length to check if it is mono or stereo, then continue with either ‘if’ or ‘else’ statement. While mono input audio can be directly created and displayed, stereo input signal is transformed to mono input signal before that, followed by plotting the input signal. Created with control frequency, duration, and timepoint, sine wave control audio and signal are also displayed and plotted respectively. After the maximum delay samples are calculated, created output signal is delayed and mixed with input signal. Similar to how input signal and audio were processed, while mono output audio can be directly created and displayed, stereo output signal is transformed to mono output signal before that, followed by plotting the output signal.

Implementation

The Delay Function is called twice separately with mono ‘carrier.wav’ file and stereo ‘clavinet.wav’ file as inputs.

c) Result

Flanging Effect

Please refer to the submitted Notebook for results of the previous run or run all in the Notebook again. The most intuitive way to observe the flanging effect is to listen to and compare audio of both input signal and output signal.

d) Analysis

- Frequency response of controllable delay line depends on control frequency and maximum delay, which affect various frequency components of input signal
- Control frequency set to 0.2 by default (with options for 0.5 and 440 to be uncommented) is low and implies slow variations in delay of input signal

- Slower variations in delay causes lower frequencies in input signal being affected more significantly
- In brief, controllable delay line acts as time-varying filter determining by control signal, especially control frequency

3. Data: Digit Recognition

a) Data

The dataset with a training set and test set is used performed digit recognition. While the training set includes 1934 digits and labels, the test set has 943 digits and labels. A digit can be labelled as any number between 0 and 9. In order to manipulate data, digits and corresponding labels are transformed to lists.

b) Methodology

<u>Digit Recognition Function</u>

The Digit Recognition Function forms the backbone of implementation of digit recognition. In addition to the length, digits, and labels of both training set and test set, the mode is an input argument. The mode, which determines the specific function implementation method, can either be ‘without offset’ or ‘with offset’. While ‘without offset’ compares each test digit and training digit by calculating the sum of the element-wise multiplied digits, ‘with offset’ finds the best matching test and training digit with the two-dimensional correlation function ‘signal.correlate2d’. The remaining process of the Digit Recognition Function is the same for both modes. After being estimated using the training index of maximum correlation, the predicted label of each test digit is added to a prediction list. The accuracy is then computed with the number of correct predictions. 5 random test digits, finally, are shown with associated actual and predicted labels.

<u>Digit Showing Function</u>

The Digit Showing Function is used to initially show 5 random digits with actual labels only, when no prediction has been made yet. This is the same as the final part of the Digit Recognition Function, except showing the predicted label of each random test digit.

<u>Implementation</u>

There are 4 implementation variants with different combinations of training set type and test set type. Their combinations are detailed on the first column of Overview Table in the following Result section. For each implementation variant, a training subset and test subset are created on account of computational feasibility and efficiency. This is followed by creation of their variables to be used as actual parameters when calling the two functions as described above. The Digit Showing Function is called twice for training subset and test subset to show 5 random example digits and labels of them. This is especially useful for special sets, from negative test set to noisy training set. Last but not least, the Digit Recognition Function is also called twice to compare the accuracies and results of ‘without offset’ and ‘with offset’.

c) Result

<u>Overview Table</u>

Implementation Variant (Type of Training Set and Test Set)	Accuracy (Correlation without Offset)	Accuracy (Correlation with Offset)
normal training set and normal test set	0.9	0.95
negative training set and negative test set	0.75	0.75
rotated training set and normal test set	0.5	0.6
noisy training set and normal test set	0.85	0.9

Detail

Please refer to the submitted Notebook for results of the previous run or run all in the Notebook again.

d) Analysis

- In general, 'with offset' cases have higher accuracy compared to 'without' offset' cases, in light of normalised crossed correlation being performed in the former
- For normal training set and normal test set, accuracy is very high in both cases, which is expected when both sets remain unchanged
- For negative training set and negative test set, accuracy is high in both cases but less accurate compared to the previous point due to both inverted sets
- For rotated training set and normal test set, accuracy is medium in both cases and further less accurate compared to the previous point because of different orientation of both sets
- For noisy training set and normal test set, accuracy is very high in both cases and similar to the second point, implying that the noise level of both sets does not reduce the effectiveness of digit recognition much
- In short, accuracy of 4 implementation variants in both cases ranges from very high to medium

4. Conclusion

Solutions in my submitted code are mainly my original work, though there is little reference to labs for specific implementation of in-built functions and plotting.