SAE 1.01-02 - BUT INFORMATIQUE - GROUPE 1

Comparative Study of Two Low-Pass Filters for Sound Data Playback: Analysis and Performance

Auteur:
Yahia KHERZA(*A1*)
Olivier MARAVAL(*A1*)

Client:
Gilles PERROT
Référent:
Olivier MARAVAL

December 29, 2023

Abstract

lorem

Contents

Chapter 1

Introduction

The objective is to examine and compare two low-pass filter designs with the end goal of enhancing the clarity of text data transmitted via audio signals. This comparison is a component of a larger project, the DosOok challenge, which involves the development of the two programs DosSend and DosRead for sending and receiving text data through sound.

In this context, a low-pass filter is crucial as it allows the desired signal to pass while attenuating frequencies that are not needed. The necessity for this comparison comes from the fact that we need our data to be kept intact while making the processing as seemless as possible.

For this study, two distinct low-pass filtering techniques have been selected. The criteria for their evaluation are the speed of processing and the accuracy (the filter will get a pass if the initial text isn't altered after the signal has been processed).

This report will detail the theoretical concepts behind each filter, the process of their implementation in Java, and the outcomes of their performance tests. The aim is to determine which filter provides a superior solution for the problem at hand.

Chapter 2

Methodology

We will create our sound containing the data in a text file with the DosSend program. We will then read it back with DosRead and see how fast we can get back the former data.

2.1 DosSend and DosRead usage

2.1.1 Using DosSend to create a wav file

2.1.1.1 Changing the size of the wave visualizer

Before compiling, it is possible to print a specific portion of the signal by changing the second *(start)* and third *(stop)* parameters of this line at the end of the `DosSend.java` file :

```
displaySig(dosSend.dataMod, 1000, 3000, "line", "Signal modulé");
```

2.1.1.2 Compiling the Program

Compile the Java code using a Java compiler

```
javac DosSend.java StdDraw.java
```

2.1.1.3 Run the Program

Run the compiled classes with Java and send it your data written in a .txt file.

```
java DosSend.java < textfile.txt
```

2.1.1.4 Output

The program will convert the input text into an audio signal and save it as `DosOok_message.wav`. It will also print characteristics of the signal to the console and display a graphical representation of the signal.

```
[oli@oli-arch src]$ java DosSend < helloWorld.txt
[ H e l l o   W o r l d   ! ]
[ 1 0 1 0 1 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 1
    1 1 0 1 1 0 0 0 0 0 0 1 0 0 1 1 1 0 1 0 1 0 1 1 1 1 0 1 1 0 0 1 0 0 1 1 1 0 0 0
   1 1 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 ]
Message : Hello World !
Nombre de symboles : 13
Nombre d'échantillons : 49392
Durée : 1.12 s
```

Figure 2.1: Console output for the text "Hello World !"

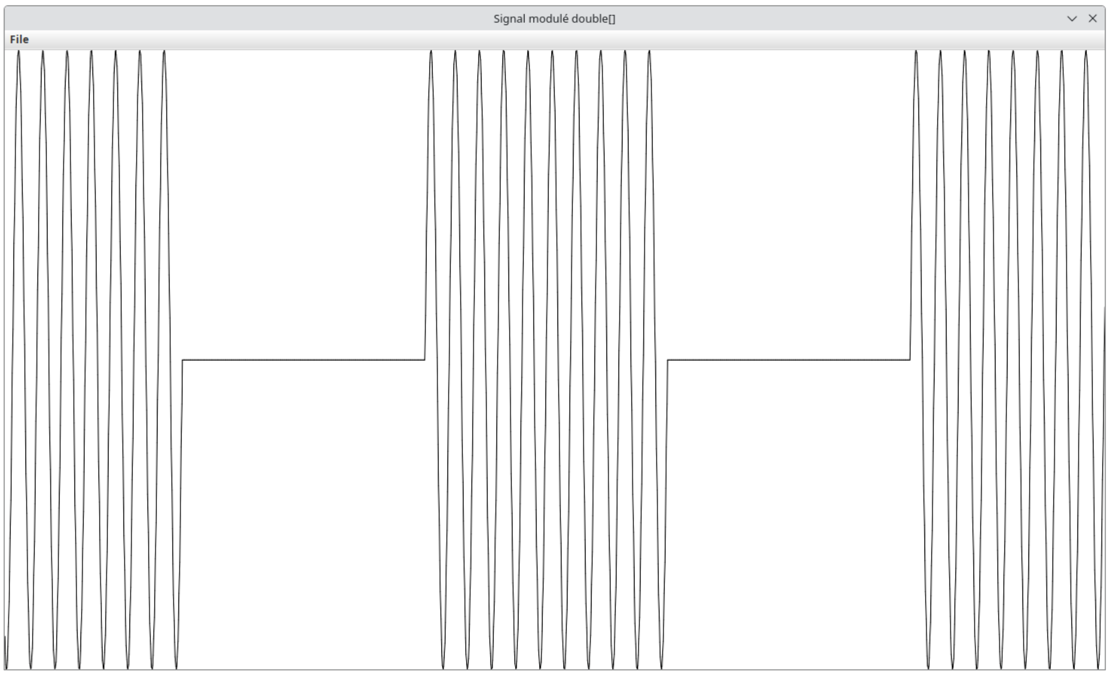

Figure 2.2: Signal output for the text "Hello World !" with start:300 stop:1000 mode:line

2.1.2 DosRead

2.2 Monitoring the speed of the filter

2.3 Checking the filter's accuracy

Chapter 3

Results

Chapter 4

Discussion

Chapter 5

Conclusion