University of Sheffield

COM3502-4502-6502
Speech Processing

Main Programming Assignment

Leon Singleton

Mingyan Zeng

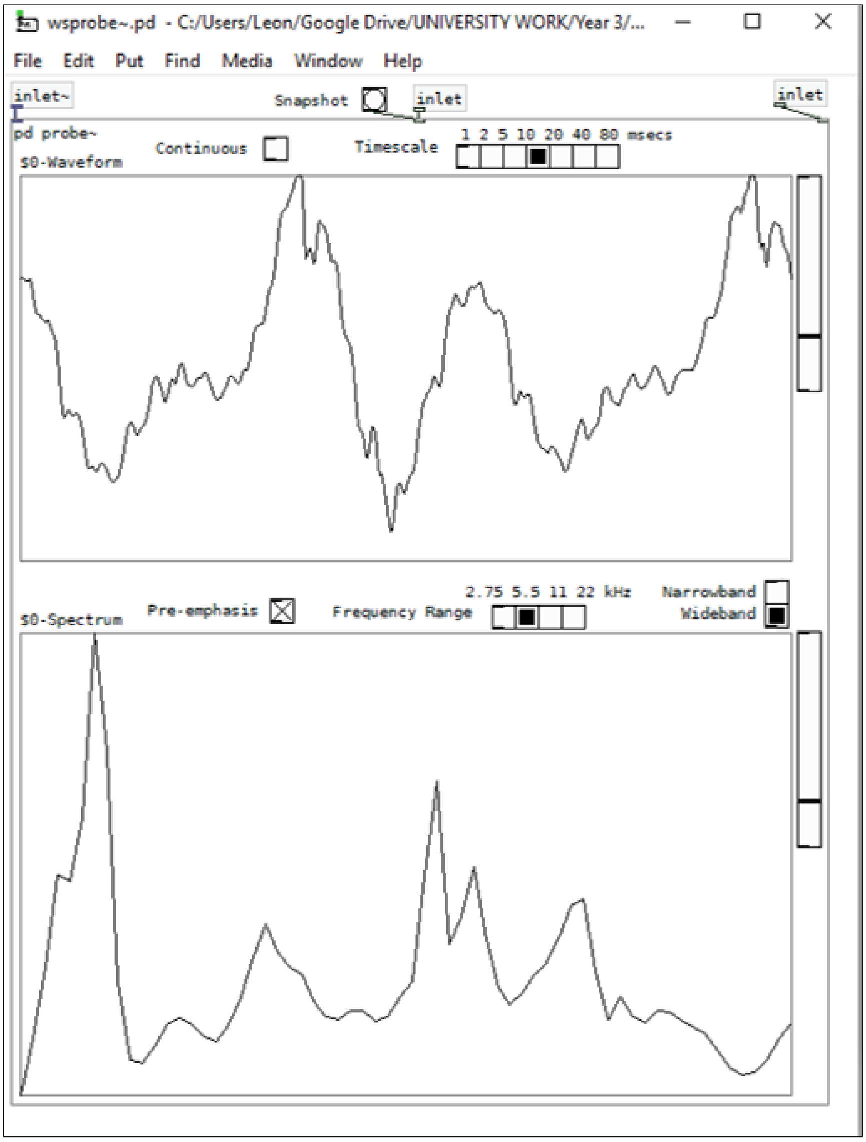
Department of Computer Science

December 7, 2018

QUESTION 1 ***(worth up to 5 marks)***

Provide a screenshot of `[wsprobe~]` for a typical voiced sound, and explain the features in the waveform and spectrum that distinguish it from an unvoiced sound. *Hint: use the 'snapshot' feature in `[wsprobe~]` to obtain a static display.*

A typical voiced sound differs from that of an unvoiced sound due to vibrations of the vocal cords. This causes them to have a more variable oscillatory waveform than that of an unvoiced sound. The wave forms of voiced sounds can be identified by a lower frequency, longer wavelength and a more varying amplitude than a typical unvoiced sound. These features also correlate to that of a voiced sounds spectrum which is more broad than that of an unvoiced sounds spectrum. The screen shot provided below demonstrates the typical characteristics of a voiced sound.



QUESTION 2 ***(worth up to 5 marks)***

Which sounds are most affected when the low-pass cut-off frequency is set to around 500 Hz - vowels or consonants - and why?

When the low-pass cut-off frequency is set to 500Hz the sounds of consonants are affected more than the sounds of vowels. This is because vowel sounds generally have lower frequencies than consonants and smaller frequency ranges. For example the vowel sound 'O' can range between 250-1,000Hz whereas the consonant 'S' typically ranges between 1,500-6,000Hz.

QUESTION 3 *(worth up to 5 marks)*

How is it that the speech is still quite intelligible when the high-pass cut-off frequency is set to 10 kHz?

When the high-pass cut-off frequency is set to 10kHz the entire sound is attenuated but the sound is still intelligible. This is because the high-pass filter object in Pure Data (Hip) does not eliminate all frequencies that are below the high-pass frequency value sent to it. Instead, it is the case that the high-pass filter in Pure Data either eliminates or reduces the frequencies below its cut-off frequency. As such enough of the original signal is still present for the output signal to be intelligible.

QUESTION 4 *(worth up to 5 marks)*

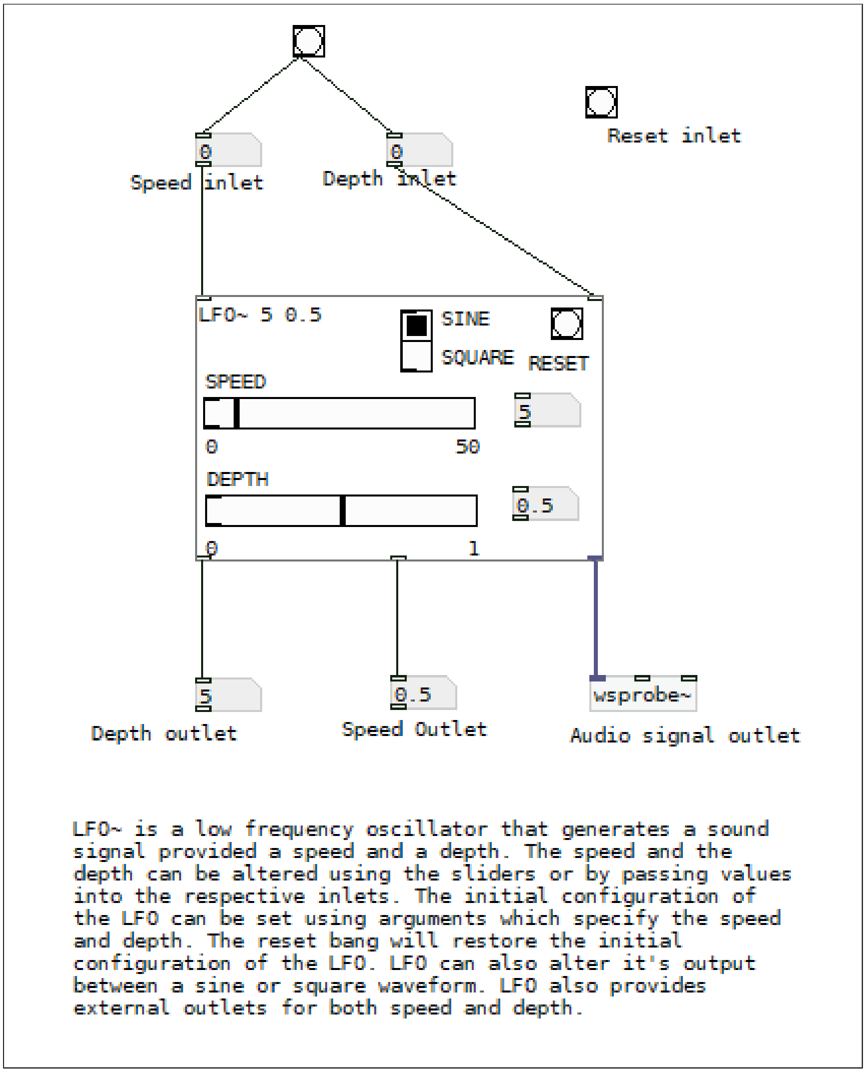
COM3502-4502-6502: The `[GraphicEqualiser~]` object uses an FFT internally; what does FFT stand for and what does an FFT do?
COM4502-6502 ONLY: What is a DFT and how is it different from an FFT?

The abbreviation FFT stands for 'Fast Fourier Transform'. The fast Fourier transform is a mathematical method that takes a signal/function and transforms it from a signal/function of time into it's corresponding frequency components. It does this by factorizing the discrete Fourier transform matrix associated with a signal into a product of sparse factors. This method reduces the complexity associated with discrete Fourier transform and performs computations in a faster (Nlog(N)) time. This is generally why FFT it is the preferred method within Fourier analysis. One common application of fast Fourier transforms is to analyze sound by sampling the spectral energy contained within it at evenly-spaced time intervals.

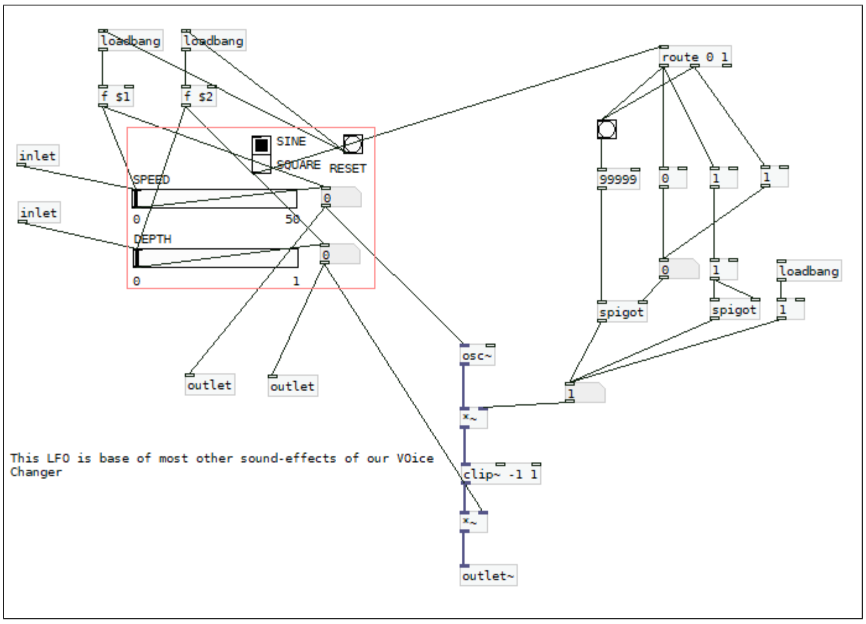
QUESTION 5 *(worth up to 10 marks)*

With `speed = 50` and `depth = 0.5`, what are the minimum and maximum amplitudes of your LFO output, and how do they vary with changes in these two settings? Also, please provide two screenshots: (a) your `[LFO~-help]` object and (b) the internal structure of your `[LFO~]` object.

With a speed setting of 50 and a depth of 0.5 the minimum and maximum amplitudes of the LFO are -0.5 and 0.5 respectively. When the speed setting of the LFO is altered there is no effect on the resulting minimum and maximum amplitude of the output signal. However, when the depth setting of the LFO is altered the minimum and maximum amplitude of the output signal increase as the depth is increased, and vice versa.



LFO~ is a low frequency oscillator that generates a sound signal provided a speed and a depth. The speed and the depth can be altered using the sliders or by passing values into the respective inlets. The initial configuration of the LFO can be set using arguments which specify the speed and depth. The reset bang will restore the initial configuration of the LFO. LFO can also alter it's output between a sine or square waveform. LFO also provides external outlets for both speed and depth.



This LFO is base of most other sound-effects of our VOice Changer

QUESTION 6 ***(worth up to 5 marks)***
In your own words[1], why is this effect known as 'ring modulation'?

Ring modulation of a speech signal can be achieved by multiplying a given speech signal by another waveform, in our case a sine wave produced by the LFO. An effective ring modulation will output sounds that can be characterized by a robotic/metallic nature. This effect is known as ring modulation because when a ring modulator takes wave forms that are not harmonically related and multiplies them, it results in overtones in the output speech signal that are in-harmonic. This in turn results in the speech signal containing the typical ring modulated characteristics that result in a robotic sound quality.

QUESTION 7 ***(worth up to 5 marks)***
Why is SSB commonly used in long-distance radio voice communications?

SSB is commonly used in long-distance radio communications over other methods such as AM or FM because it improves upon the the power consumption required to transmit a signal and decreases the bandwidth of transmitted signals. SSB removes the need for a carrier frequency in the transmitted audio signal since only a single sideband is sent. This results in a 50% reduction in the power required in the audio transmitter. In addition, SSB halves the bandwidth of transmitted audio signals, again due to the fact that only one sideband is transmitted. This in turn improves the signal to noise ratio of the signal, because a lower bandwidth will allow less noise and interference to affect the signal.

QUESTION 8 ***(worth up to 5 marks)***
COM3502-4502-6502: Why can the voice be shifted up in frequency much further than it can be shifted down in frequency before it becomes severely distorted? /emphHint: look at `[wsprobe∼]`.
COM4502-6502 ONLY: Your frequency shifter changes all the frequencies present in an input signal. How might it be possible to change the pitch of a voice *without* altering the formant frequencies?

The reason why voice can be shifted up in frequency much further than it can be shifted down before becoming distorted is due to the natural frequency range of human speech. A frequency range between 1 KHz and 4 KHz is of high importance for intelligibility. This is evident when we use a LP-filter and HP-filter respectively and set them to these frequencies. Approximately 70% of non-tonal language sounds are contained within the range between 0 KHz and 2 KHz with the majority located in the 2 kHz frequency band. As such shifting down causes the majority of sounds within speech to be shifted away from the ideal 1-4 KHz range, resulting in a distorting effect. This distortion effect is not as severe when shifting up because most sounds will still be within the ideal 1-4 KHz frequency range. Furthermore, the distortion

[1] I.e. do not plagiarise from Wikipedia.

effect caused when shifting down is typically more noticeable in male voice over female voice. This is due to the difference in their average fundamental frequencies.

QUESTION 9 *(worth up to 5 marks)*

In a practical system, why is it important to keep the feedback gain less than 1?

In a practical system it is important to keep the feedback gain less than one so as to preserve the stability of the output. When the feedback gain is set as one or higher the output signal is unstable and the gain can be deemed as infinite. This results in the sound of the output becoming completely unintelligible. The reason this happens is because feedback gain with a proportion greater than one causes the frequency range of the signal to become exceptionally high.

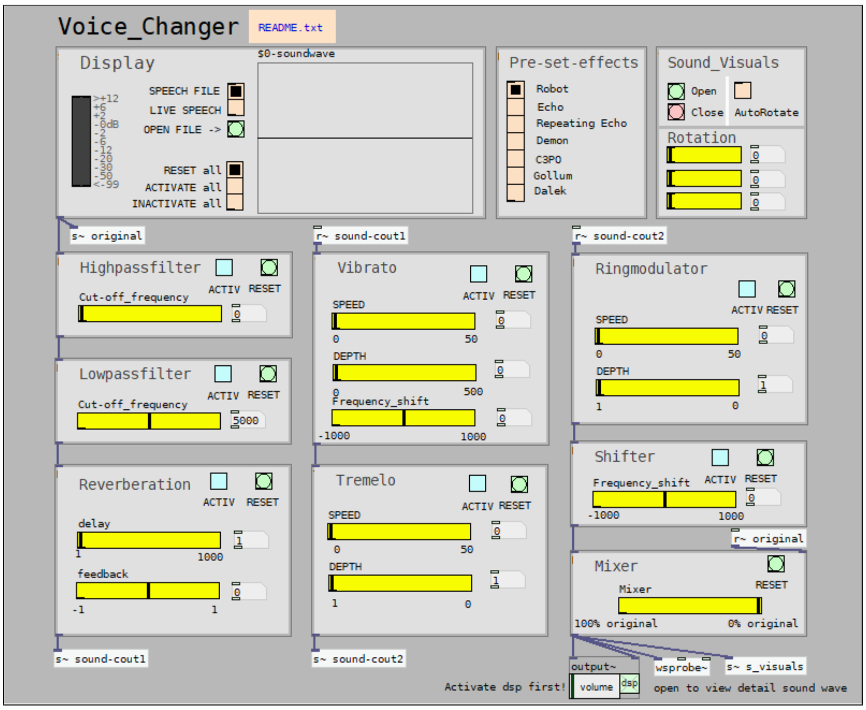
QUESTION 10 *(worth up to 50 marks[2])*

Please provide a short[3] description of the operation of your `[VoiceChanger]` application, together with a screenshot of your final GUI.

The final voice changer utilizes many of the different components that we built in part 1. First of all a user can select to have the input as either live speech or from a speech file. A simple sound wave of the input can be seen at the top of the GUI alongside a decibel reading. The various effects that we have used have been built as abstractions containing sliders that can be altered accordingly using a "graph-on-parent" approach. The slider values can also be set using PD "send-symbols", this allows for easy addition of preset effects, some of which have been included. Each voice abstraction can also have it's values reset using it's corresponding "bang". Furthermore, an abstraction can either be in an activated or inactivated state using the toggle of each abstraction abbreviated to "ACTIV" allowing any combination of effects to be combined together. We have also added functionality that allows a user to reset, activate or inactivate all voice effects. Finally, the output of the sound can be viewed in a multitude of ways. A detailed look at the sound wave and its spectrum can be obtained using "wsprobe " at the bottom of the GUI. A visualization of the sound can also be viewed which has been implemented using the PD Gem library.

[2] 25 for functionality, 15 for design/layout, 5 for `Pd` features, 5 for innovations

[3] no more than 200 words


Voice_Changer
README.txt
Display
$0-soundwave
>+12
+6
+2
-0dB
-2
-6
-12
-20
-30
-50
<-99
SPEECH FILE
LIVE SPEECH
OPEN FILE ->
RESET all
ACTIVATE all
INACTIVATE all
Pre-set-effects
Robot
Echo
Repeating Echo
Demon
C3PO
Gollum
Dalek
Sound_Visuals
Open
Close
AutoRotate
Rotation
s~ original
r~ sound-cout1
r~ sound-cout2
Highpassfilter
ACTIV
RESET
Cut-off_frequency
Lowpassfilter
5000
Reverberation
delay
1
1000
feedback
-1
Vibrato
SPEED
0
50
DEPTH
500
Frequency_shift
-1000
Tremelo
Ringmodulator
Shifter
r~ original
Mixer
100% original
0% original
s~ sound-cout1
s~ sound-cout2
Activate dsp first!
output~
volume
dsp
wsprobe~
s~ s_visuals
open to view detail sound wave