“Voice Modification”

ECE 160 Multimedia Systems

Antonio Javier Samaniego Jurado
samaniegojurado@umail.ucsb.edu

***Abstract*— I propose an audio processing algorithm that modifies how a human voice recording sounds, without changing the pitch or speed of the speech, so that the identity of the person speaking is obscured, but the speech can still be understood. I present the methodology, results and discussion in this report.**

I. Introduction

Voice modification is an useful way to obscure the identity of the person speaking for applications such as crank calls or espionage activities[1]. Even though that can be achieved many ways, in this implementation I show how applying amplitude modulation, a “robot” effect, high-pass filtering and an echo/reverb, can be a very effective one, carried out by manipulating samples from the audio file. I have used Matlab© for my implementation.

II. Methodology

A. Overall Description

Input: .wav voice audio file
Output: modified .wav voice audio file

Prepare data for modification
audio_samples, Fs ← audioread(file)

modulated ← amplitude modulation
delayed_robot ← robot effect (modulated)
filtered ← high_pass (delayed_robot)
final ← echo/reverb (filtered)

Frequency domain analysis

plot of results in time/frequency domain

Alg. 1. Overall Algorithm

B. Reading the Audio File

First, the algorithm reads a .wav audio file using Matlab built-in function ‘audioread()’, saving the audio samples in variable ‘audio_samples’, a new 1D ignal (array) that is going to be modified to eventually get the final result, as well as its corresponding sampling frequency value in Fs.

C. Amplitude Modulation

By creating a sine function, the algorithm starts its modification performing amplitude modulation, just like it is done in applications such as a telecommunication transmission, where we have a carrier, a modulating wave and the modulated result (as shown in Fig. 1), being the audio samples signal the equivalent to the carrier, the sine to the modulating wave and the modified audio to the modulated result.

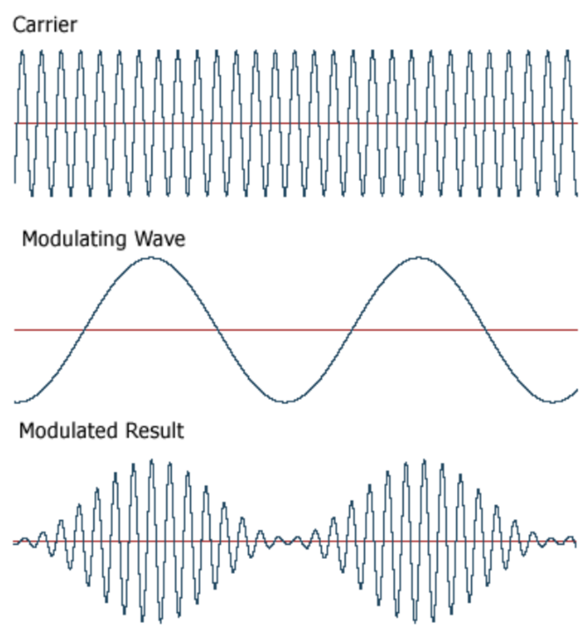


Fig. 1. Amplitude Modulation

D. Robot Effect, by Applying a Fast Delay

The “Robot Effect” can be achieved by applying a fast delay to the signal. In my implementation, I have created a certain number of delayed versions of the original with different delay values (from a minimum delay to a maximum delay), proceeding as shown in Fig. 2, using short/fast delay values.

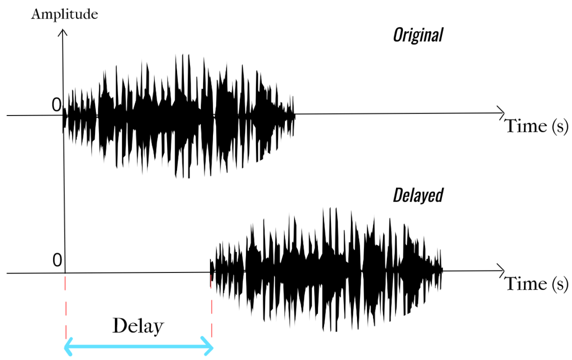


Fig. 2. How a certain Delay is applied to the original audio signal

Next, the algorithm adds up all delayed signals, saving the result in 'sum_delayed', which already sounds like a robot.

How it specifically works can be seen in Alg. 2.

Input: audio_samples
Output: sum of delayed signals

Set delay parameters strategically
Set min_delay
Set max_delay
Set num_delayed_signals

create array with delay values, from min_delay to max_delay
Prepare delayed_signals array

for all delayed_signals
    delayed_signals((all positions-max_delay)+delay) ← audio_samples(:)

sum_delayed ← sum(delayed_signals)

Alg. 2. 'Robot Effect', by Applying a fast Delay

Note that the delay parameters are already strategically set to achieve such effect, based on a regular, standard human voice speed.

E. *High-Pass Filtering*

After getting the sum of all delayed signals, achieving the robot effect, such result is filtered by a Butterworth high-pass filter, saving the filtered signal in 'filtered'. This gets rid of the low frequencies, making it harder to find out who is speaking.

The pipeline that describes the process is shown in Fig. 3.

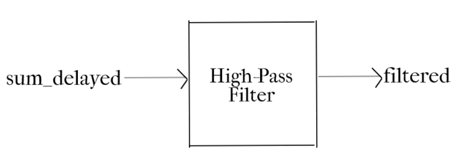


Fig. 3. High-Pass Filtering

F. *Echo/Reverb Effect*

This step of the algorithm takes the previously filtered signal and adds a little (but still noticeable) echo/reverb effect.

This is achieved by applying a very specific filter with a gain (k) and delay (D) parameters. Such parameters allow to design a filter that result in a sum of several delayed versions of the input signal, that have different offsets.

It should be pointed out that, in order to get a great echo effect, parameters k and D are already set strategically. As it is a filtering operation, the pipeline that describes the process is equivalent to the one shown in Fig 3., replacing the high-pass filter with the one previously indicated.

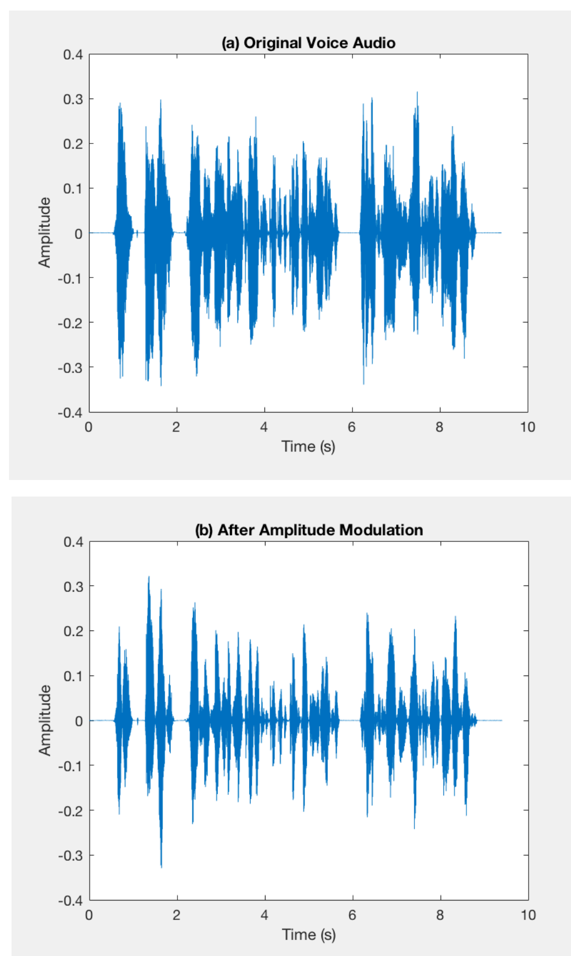
After applying this effect, the final result is achieved and saved in 'final'.

III. Results and Discussion

Next, I present and discuss the results, performing an analysis in both time and frequency domain, as well as a subjective evaluation of the resulting modified audio.

A. *Time Domain*

After plotting the resulting signals for every step of the algorithm, results in the time domain are shown in Fig. 4.



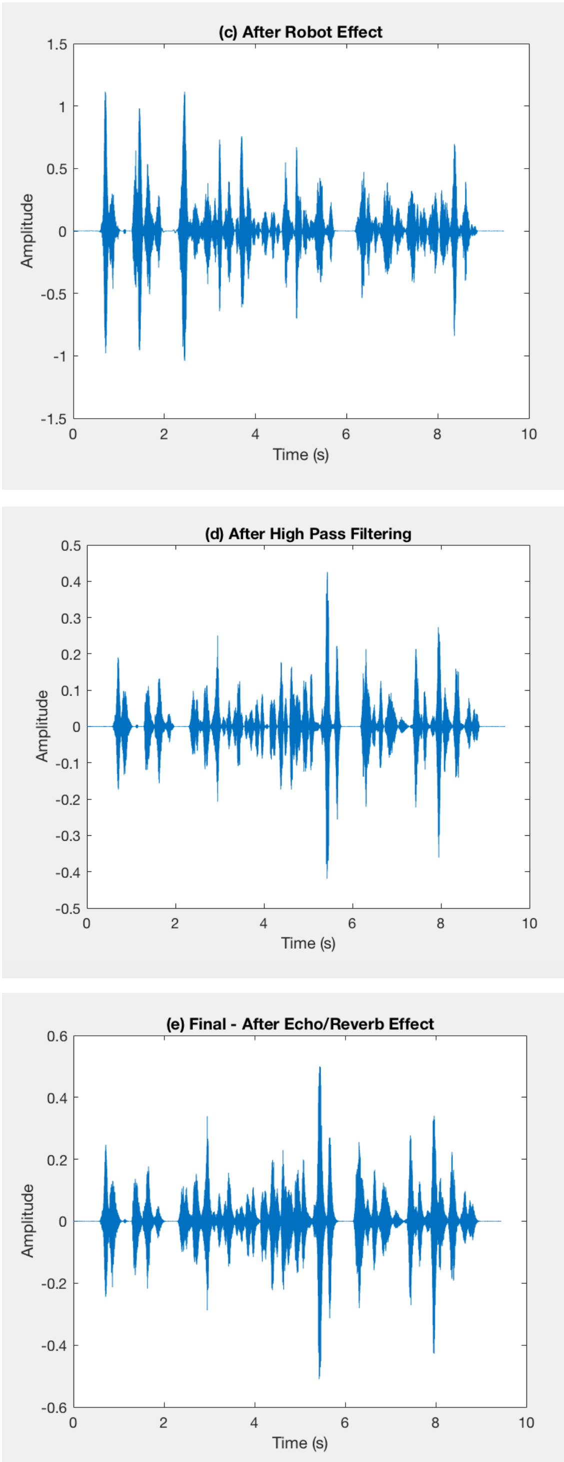


Fig. 4. Plot of resulting signals after each step of the algorithm - Time Domain.

Compared to the original signal (Fig.4 (a)), we can see changes in amplitude values after the first step (amplitude modulation - Fig.4 (b)). The resulting signal is sort of reshaped (see Fig.1) based on the modulating wave used for this particular case, which can be seen in Fig. 5. Apart from that, the most representative results within the time domain can be seen in Fig.4 (c) and Fig.4 (e), as in both we can notice new samples as a result of applying the delay.

More representative results for the high-pass filtering operation will be shown in the frequency domain.

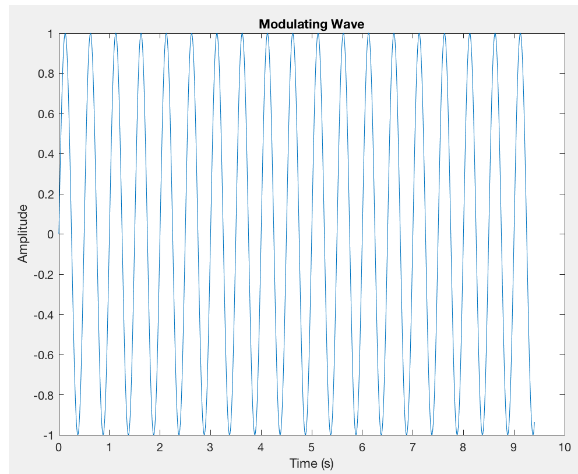
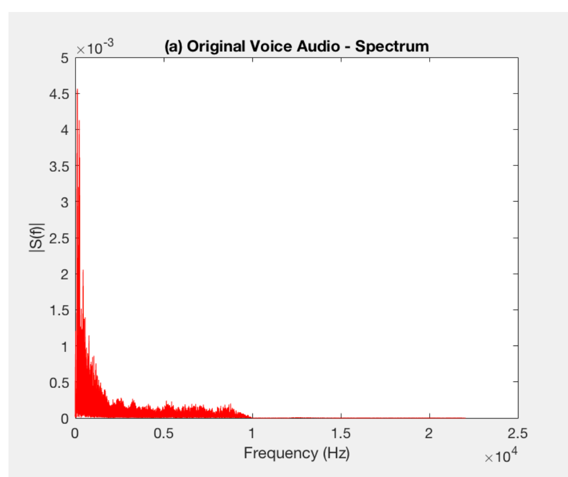
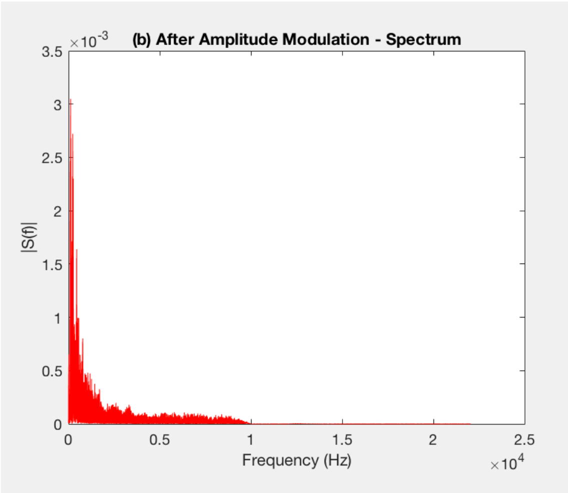


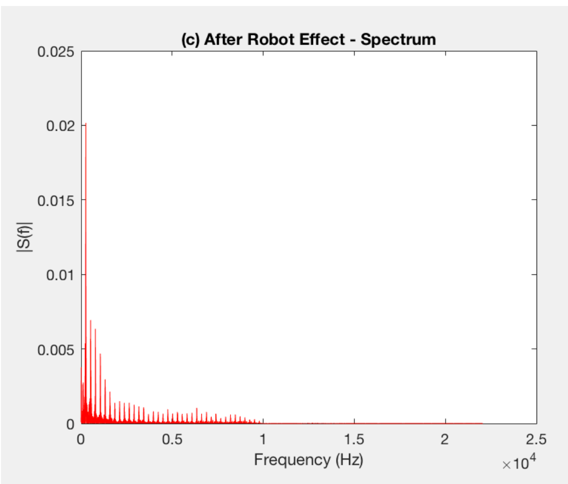
Fig. 5. Modulating wave used in the amplitude modulation step.

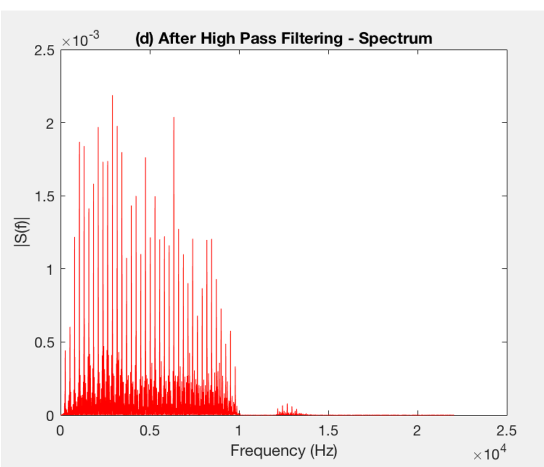
B. *Frequency Domain*

Proceeding in a similar way as in the time domain, results in the frequency domain are shown in Fig. 6.









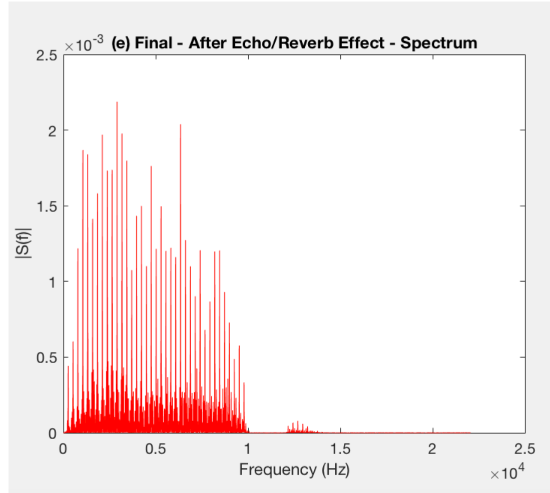


Fig. 6. Plot of resulting signals after each step of the algorithm - Frequency Domain.

The first representative result within the frequency domain plots can be seen in Fig. 6 (c), where after applying the robot effect using a fast delay we can see how certain frequencies in the spectrum lose energy.

Next, after filtering the resulting signal with a high-pass Butterworth filter, we can see how higher frequencies gain some energy, resulting in a more distributed spectrum, as seen in Fig. 6 (d).

The final result has a similar but not as noticeable effect as the robot effect, showing how the final, modified audio spectrum looks like.

C. *Subjective Evaluation*

The first impression of the modified audio is that it sounds like a robotic, metallic version of the original voice.

The filtering operation is also noticeable as it cuts off low frequencies of the voice. This is why the algorithm gives an even more effective result with a male voice recording.

Finally, the last step adds an echo/reverb to the speech that helps it be more understandable and nicer to hear.

IV. Conclusion

Voice modification can result in very effective way to obscure the identity of a person speaking, which is useful for applications such as crank calls or espionage activities. This implementation achieves such result by giving a robotic, metallic and reverberated version of the original audio, with its corresponding differences in both time and frequency domains.

References

1. ECE 160 Voice Modification Challenge Description.