

OBL 4101 – Project on voice coder (vocoder) –

Students: Adelina CUCU & Matej SMID

Teacher: Gaelle LISSORGUES



November 7, 2022

Contents

[1. Introduction 2](#_Toc119543394)

[1.1 Context 2](#_Toc119543395)

[1.2 Abstract 2](#_Toc119543396)

[1.3 Presentation of the main function 2](#_Toc119543397)

[2. Modifying the speed of the voice without changing its pitch 3](#_Toc119543398)

[2.2 Slowing down the signal 4](#_Toc119543399)

[2.3 Increasing the speed of the signal 5](#_Toc119543400)

[2.4 Conclusion 5](#_Toc119543401)

[3. Modifying the pitch without changing the speed 6](#_Toc119543402)

[3.1 Increasing the pitch 6](#_Toc119543403)

[3.2 Decreasing the pitch 7](#_Toc119543404)

[3.3 Conclusion 7](#_Toc119543405)

[4 Robotization of the voice 8](#_Toc119543406)

[4.1 First robotization function 9](#_Toc119543407)

[4.2 Second robotization function 10](#_Toc119543408)

[4.3 Conclusion 12](#_Toc119543409)

[5 Special effects 13](#_Toc119543410)

[5.3 Wahwah effect 13](#_Toc119543411)

[5.4 Saturation effect 13](#_Toc119543412)

[Conclusion 14](#_Toc119543413)

[References 15](#_Toc119543414)

# Introduction

## Context

Audio processing in analogue and digital domains has begun long time ago, having as origins the mechanical recoding devices and old telephony. Even if we can still find traces of these old methods, the Digital Signal Processing audio technologies have evoled a lot since then and they are using algorithmic and mathematical tools, which have a considerable effect on nowadays society. One essential human ability is the ability to perceive and interpret sound. It provides cues for visual perception, situational and geographical awareness, and, most importantly, the capacity for interpersonal communication. Therefore, audio processing has an important role in the engineering field. [1]

## Abstract

This project aims to present different ways of audio processing using Matlab and is based on 4 main parts. In the first part we are going to modify the speed of the voice for different audio without changing the pitch, in the second part of our project we will modify in contrary the pitch of the voice without changing its speed and in the third part, we will apply ring modulation on the voice in order to make it seems like a robot voice. In addition, we have a last part in which we experimented with different filters and audio processing methods on the sounds and displayed and compared the result. [1]

## Presentation of the main function

Our program consists of a variety of function which helped us process the audio signals and obtain the effects that are mentioned above. What plays a main role in this project is the program “VOCODER.m” which integrates all of our functions that we will detail later on. In this program, we can find all the audio files that we tested and that can be changed for testing different sounds effect as well as the representation of our processed signals in Time, Frequency and their spectrograms before and after modifications.

In the first part of this program, we are displaying and listening to the original sound for being able to make a proper comparison after we process it later on. Then we are setting the parameters, calling the function that we want to use and listening to the output sound that is going through a series of tranformation (robotization, modification of the speed, saturation etc.). Beside this, the graphical representation are helping us understand even better what is going on with our audio signal and how it is transformed.

# Modifying the speed of the voice without changing its pitch

The first effect applied to the sound is the speed modification, where we are going to slow down and then increase the speed of the sound without affecting its pitch. For producing this effect we will use the function “PVoc.m” which is calling also the functions “TFCT.m”, “TFCT\_Interp.m”and “TFCT\_Inv.m”.

For testing this effect we used the audio file “PaulHill” from [1] “Audio and Speech Processing with MATLAB” book’s materials, whose representation in time and frequency domain as well as its spectrum, can be seen below:

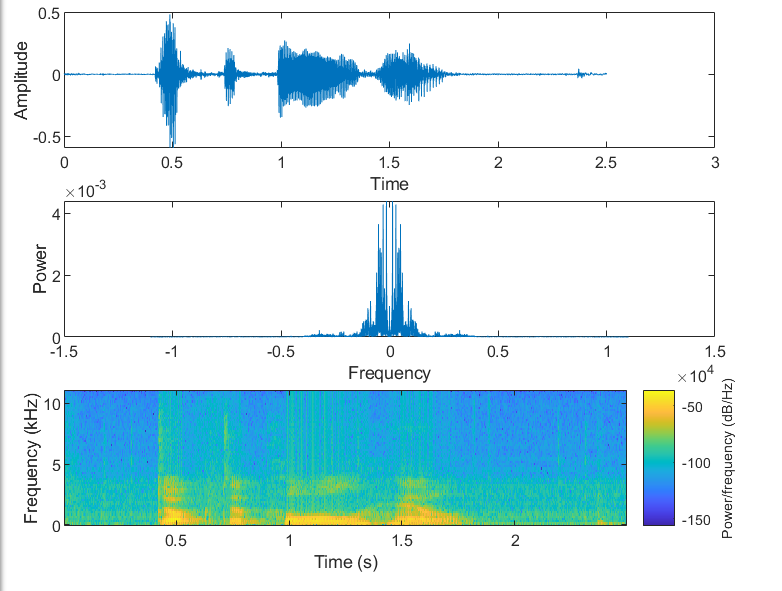


Figure 2.1

In this part, we want to change the speed of speech of the audio signal. The duration of the signal will therefore be modified. To do this, after switching to the frequency domain for each frame, the program must calculate a new time base (in terms of samples) with the chosen speed parameter. This new time base will make it possible to obtain a new number of frames that will be used during interpolation. The interpolation will make it possible to match the FT of each frame obtained to the new number of interpolation windows and thus modify the speed of the signal.

## Slowing down the signal

When slowing down the signal the duration of the signal will increase since the sound will need a longer time. We used an interpolation ratio of 2/3 between the initial time and the final time of the signal for slowing it down and we store it in the variable “rapp”.

If we are comparing the representation in time of the Figure 2.1 and Figure 2.2, we can easily observe that the duration of the shound has increased from 2.5 to 4.

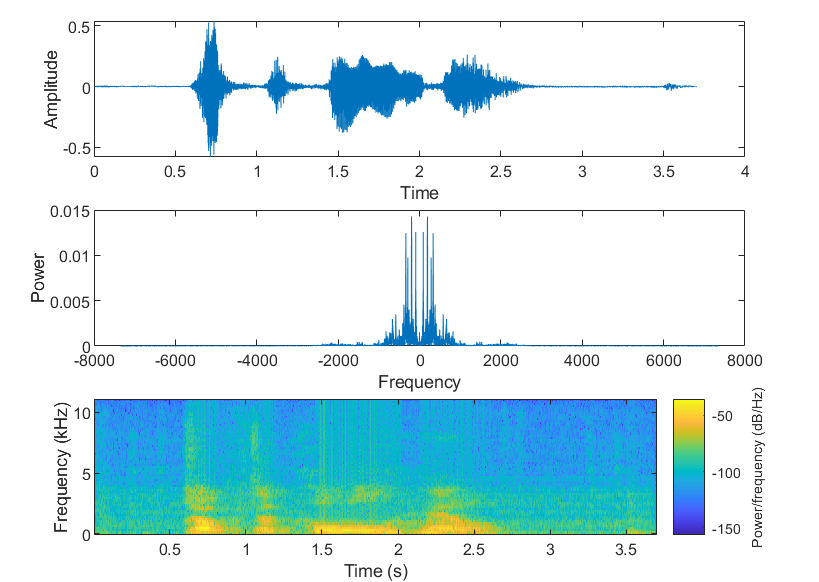


Figure 2.2

## Increasing the speed of the signal

For increasing the speed of the signal we used an interpolation ratio of 3/2 (rapp=3/2).As a result, it can be observed that the duration of the signal has decreased from 2.5 to 1.8, which corresponds to an increase in speed.

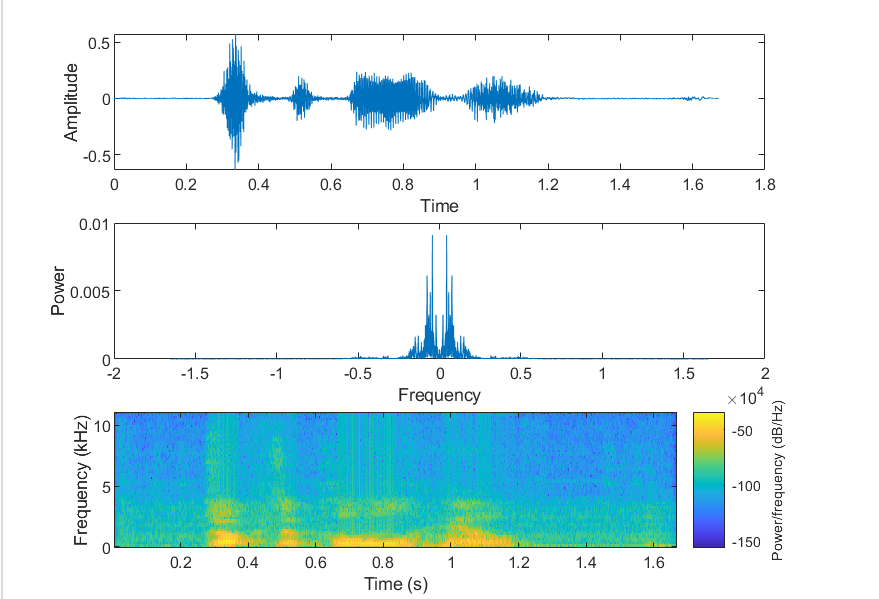


Figure 2.3

## Conclusion

For doing this modifications we used the function PVoc.m and a different interpolation ratio according to each case. We noticed that a ratio of 2/3 will slow down the sound while a ratio of 3/2 will increase the speed. This changes had nothing to do to the way the voice sounds or how the words are pronnouncing, but it simply modified the speed.

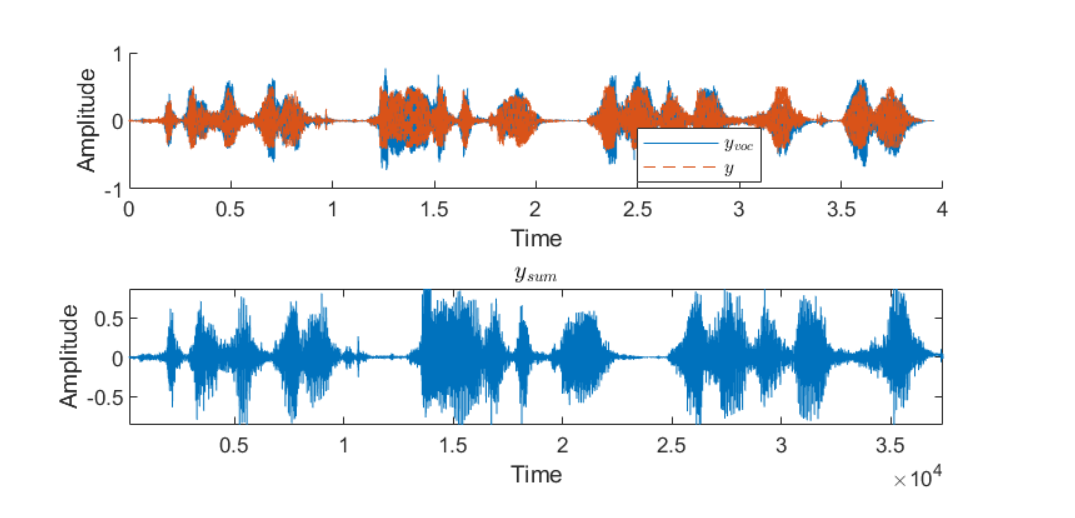
# Modifying the pitch without changing the speed

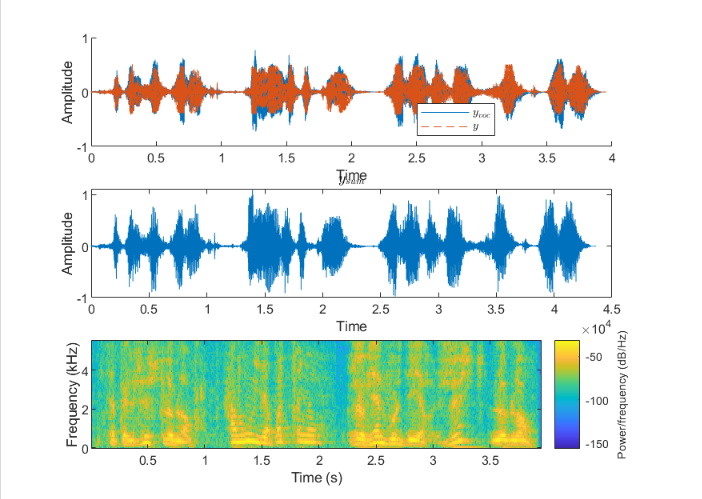
In this part, we want to change the pitch of the voice of our signal. This modification is done on the same principle as the modification of the speed of the voice that we compute previously. Indeed, it has been well observed during listening that the modification of the speed of a voice leads to the modification of the duration of the signal.

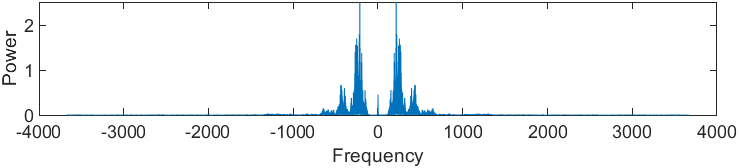
When the speed of an extract is increased, the sound waves are compressed: since the speed of the extract is increased, the duration of this extract is reduced, therefore compressed. This compression phenomenon results in an increase in the frequency of sound waves; indeed, by compressing the extract, the human ear perceives more sound waves over the same unit of time.

If we define our unit of time as the second, we intuitively deduce an increase in frequency, and fin the pitch of the sound, which in our case is a voice. However, here we do not want to change the speed of the extract but only its height. For this, we must re-sample the signal in order to return to the same sampling frequency Fe and thus keep the same speed. We will then have a sample of all the Te. The sum of the modified signal and the original signal is then made. It is possible to add a coefficient to the pitch to increase its influence.

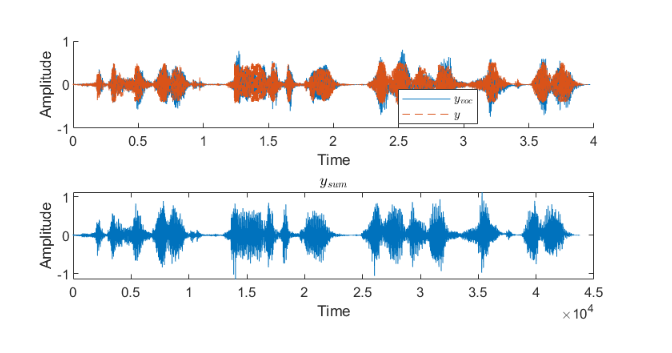
## Increasing the pitch

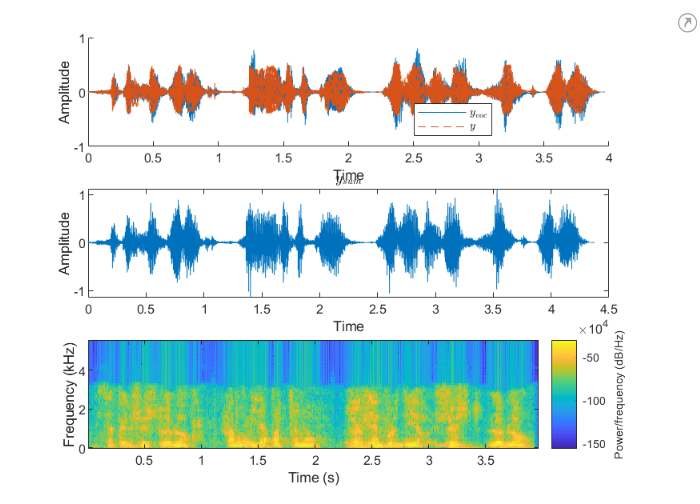


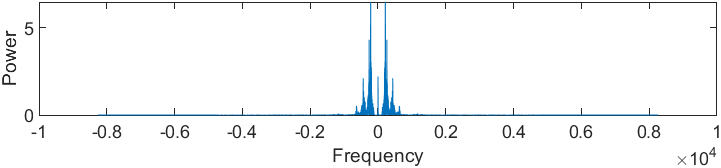




## Decreasing the pitch







## Conclusion

# Robotization of the voice

The robotization of the voice is obtained by using ring modulation. In DSP, modulation means that one signal influences the amplitude of another. RM was first used in radio receivers to carry signals, but was later used as a musical effect, first heard on Harakd Bode's Melochord (Pavlov, 2011). RM became recognized for creating unique metallic alien noises, which were most famously utilized as an effect on the voices of the Daleks in the TV show 'Doctor Who' and are currently employed in a variety of sound and music productions, but it can be used as well for robotization of a voice. [2]

RM is produced by combining two signals, one of which is the modulator carrier signal and the other an input signal. The output signal contains the difference and sum of the frequencies of the two original multiplied signals, which we refer to as the upper (USB) and lower sidebands (LSB).

In the program “Rob.m” we intended to create a function that is taking as input the speech of someone and is transforming its voice in a robot voice with the help of ring modulation. After this we created an additional function “Rob2.m” where the robotization effect is created using 2 frequencies whose effect is weighted relative to each other using the ratio r. These functions are taking speeches and instrument’s shounds and are bringing them a robot effect.

For comparing our audio signal, before and after robotization, we are displaying our original signal “Diner.wav” in Time and Frequency domain as well as its spectrogram and we plot them in the same graph. The result can be seen in the figure below:

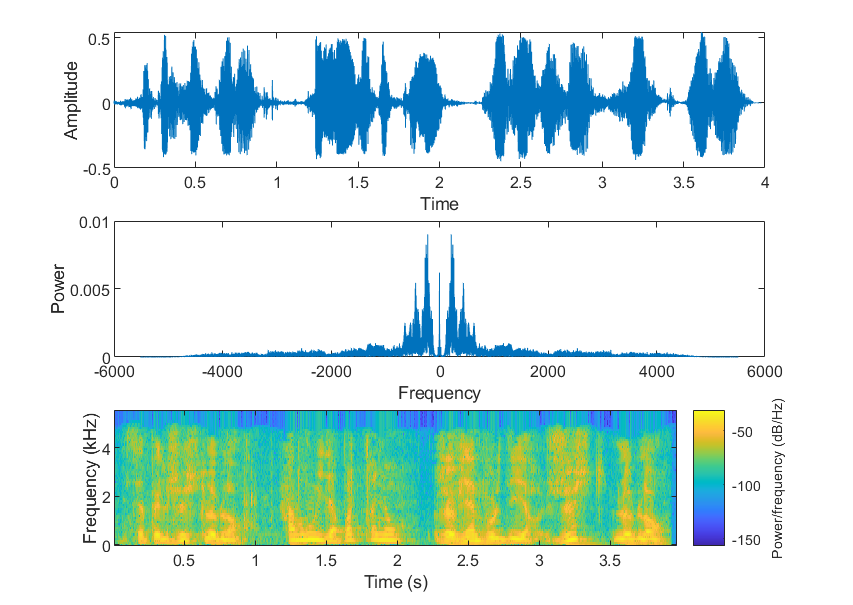


Figure 4.1

We use the 2 functions in our main program “VOCODER.m”.

## First robotization function

For “Rob.m” function we are choosing the carrier frequency and then call our function and listen to the robotization sound by using the function “sound(yrob, Fs)”, where yrob is represented by the following equation:

yrob = real(y .\* exp(-i\*2\*pi\*t\*Fc)');

We are displaying the same information for the audio signal after robotization with our first function “Rob.m” for observing the differences for Fc=1000.

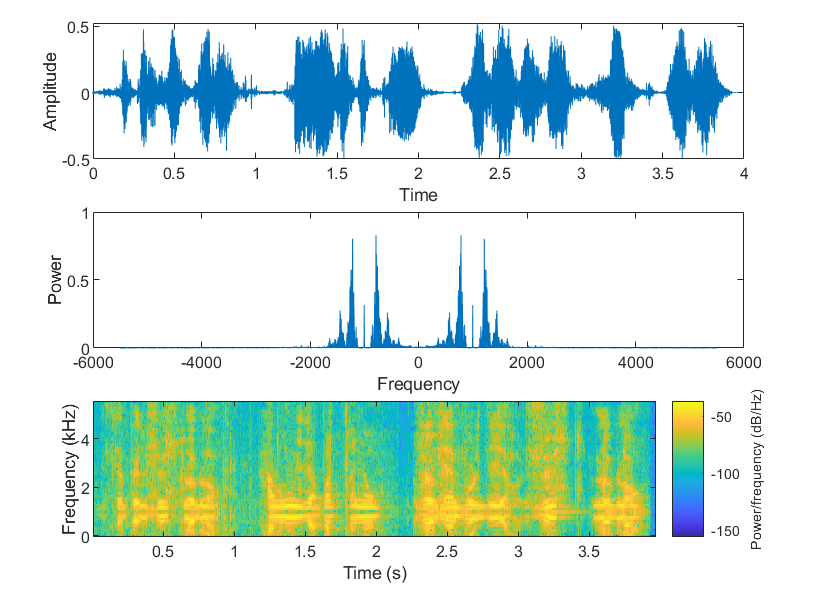


Figure 4.2

We can tell that we successfully obtain a good robotization effect, by observing the output signal of the processed shound in comparison with the original one.

In the figure below can be observed that the processed sound signal, with Fc=1000, for the sound “Diner.wav” appears more extreme than the original sound signal, which confirms that the robotic sound effect has been added to our original signal:

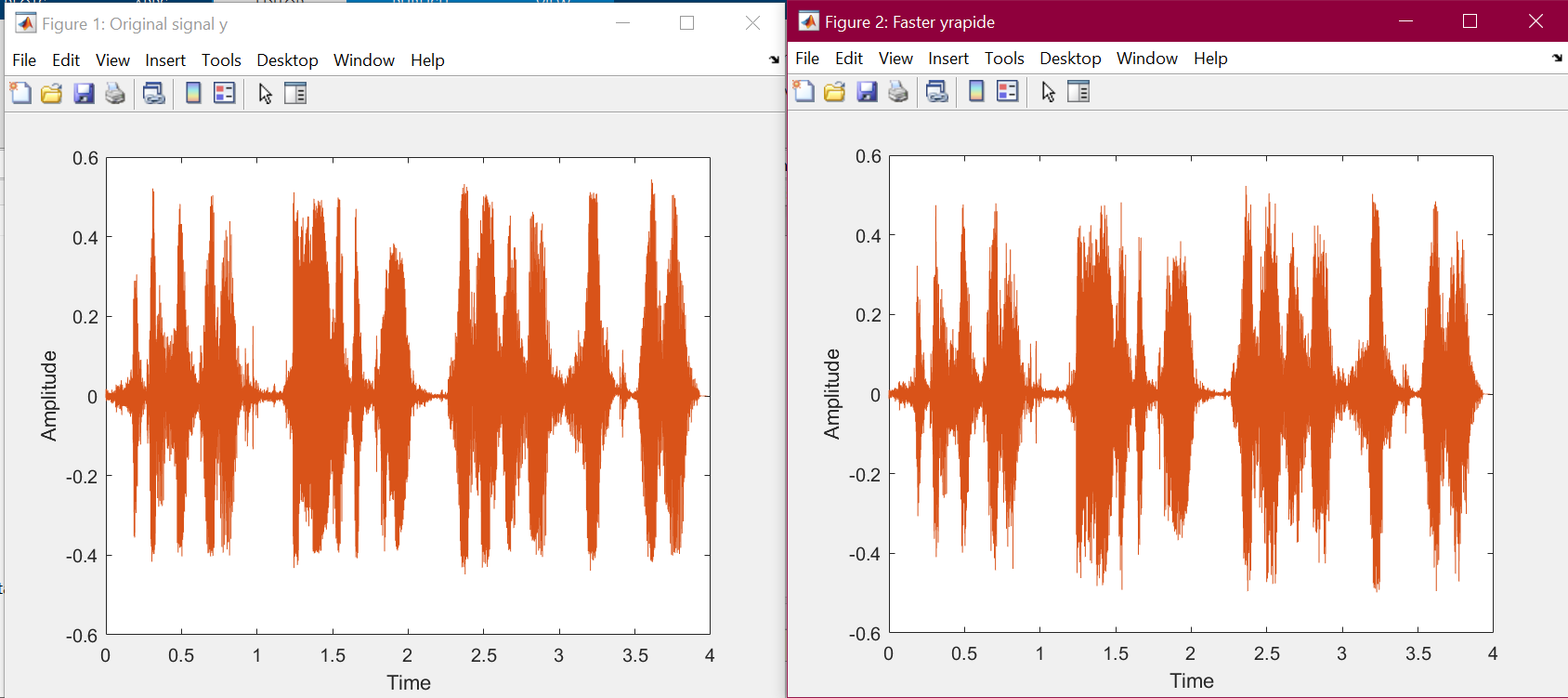


Figure 4.3 Original vs Processed sound for Rob.m

## 4.2 Second robotization function

For the second case, we used the robotization effect and we construct it by using 2 frequecies whose effect is weighted relatively to each other using the ration r. We call this function in our main program “VOCODER.m” by using its parameters and we choose the 2 frequencies prior to this: **y\_rob = Rob2(y, F1, F2, r, Fs);**

In this case, similarly to the first function, yrob is represented by the following expression:

yrob = real(y .\* (r\*exp(-i\*2\*pi\*t\*F1)' + (1-r)\*exp(-i\*2\*pi\*t\*F2)'));

The difference is that this time we are using 2 frequencies, F1 and F2 that are influencing our signal. The graphics obtained by computing “Rob2” function can be seen below:

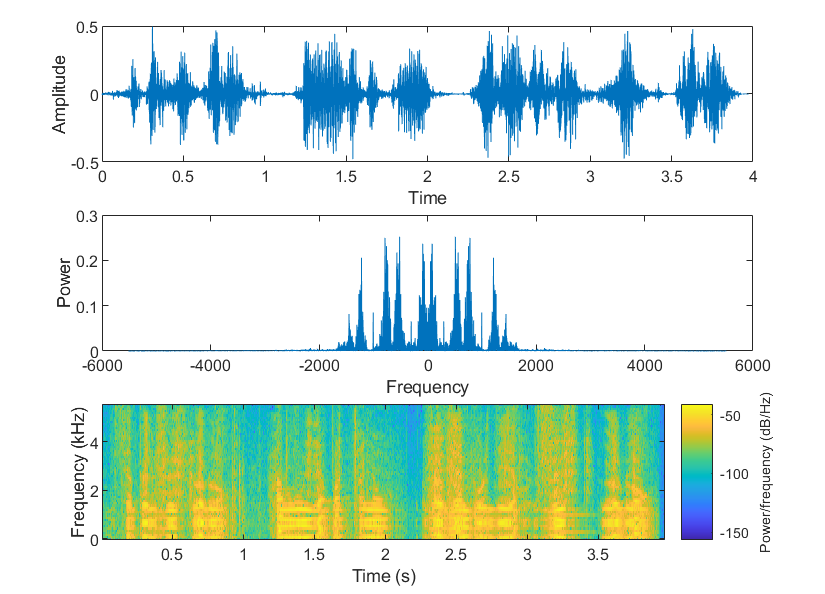


Figure 4.4

As in the previous case, we made a comparison between the original sound and the processed one by using the same audio and F1=1000 and F2=300:

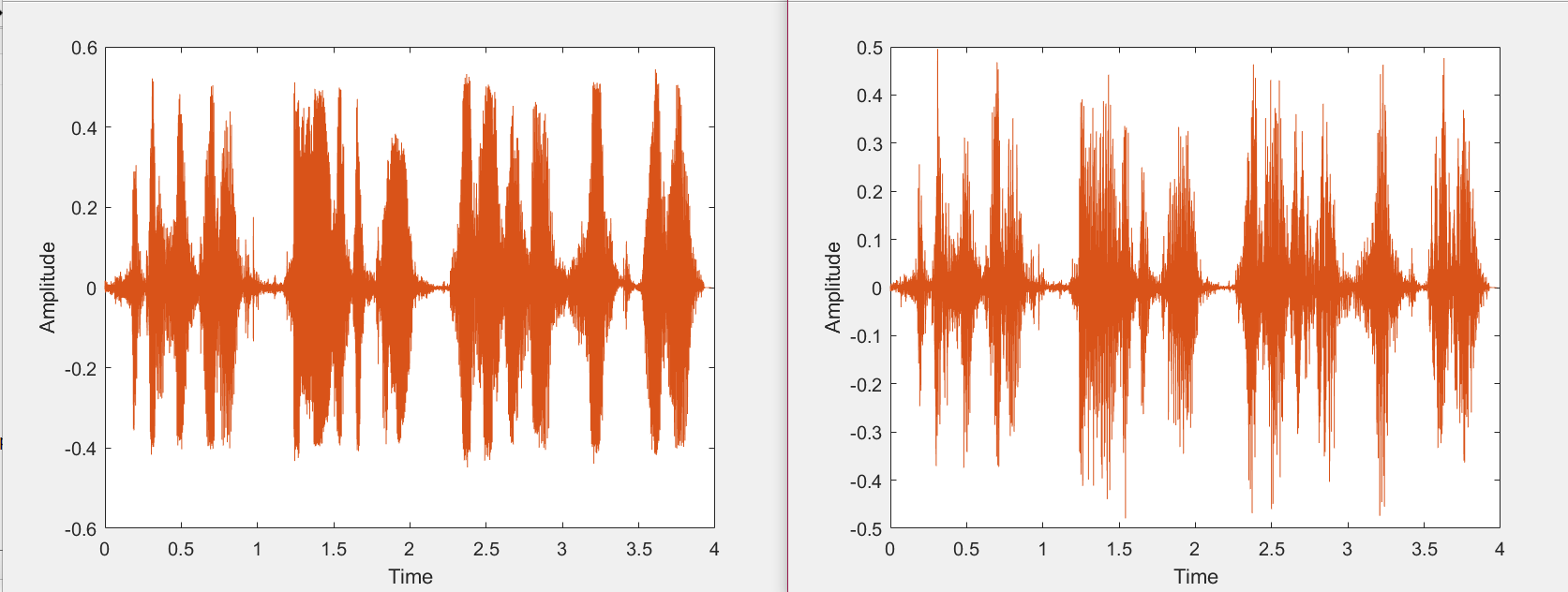


Figure 4.5. Original vs Processed sound for Rob2.m

For both cases, the functions are easy to compute in our main program “VOCODER.m”. First we are choosing our input signal from the list of aoudios we have in our main program, by uncommenting the one that we want to use. Then we are choosing our carrier frequency Fc for the first function or F1 and F2 for the second function. After this we are calling the functions with their parameters: **yrob = Rob(y,Fc,Fs);** for Rob.m and **yrob = Rob2(y, F1, F2, r, Fs);** for Rob2.m and listen to the output signal by using the function **sound().**

Robotic sound effects feature musical qualities that give a sound signal new texture. Vocals that have treated the robotic effect sound like the aliens from science fiction films. When a sound of an instrument, like a violin, is given the robotic sound effect, the violin sounds hollow and tinny, similar to the background music played as aliens appear in a movie scene. [3]

## Conclusion

Both functions perform a good robotic effect, but the “Rob2.m” seems to be more accurate and close to the desired result. The robotic effect becomes more pronounced as the carrier frequency (Fc) increases What is modifying also with the increase of the carrier frequency, is the pitch which rises. But when we tested our function for a carrier lower than 200, the robotic effect was hard barely noticed, and pretty negligible. On the other side when we increased Fc, to over 2000, the effect was too pronounced and it could be described as unpleasant to the ear.

# Special effects

## Wahwah effect

## Saturation effect

We created a saturation function “Saturate.m” and we tested it for 'Extrait.wav'.

For observing the differences between the original and processed sound, we displayed the time, frequency and spectral analysis.

Below we find the original signal representation:

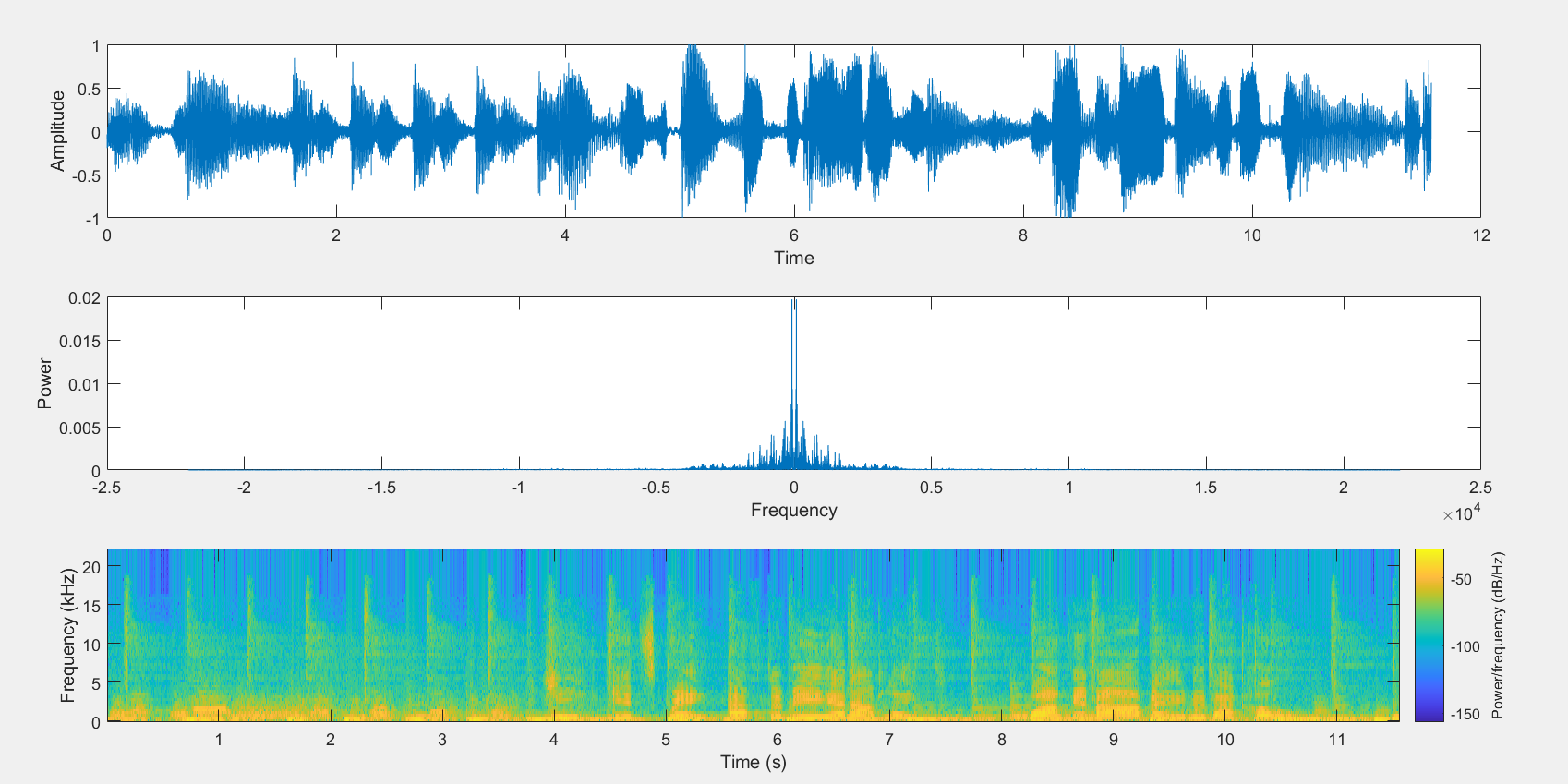
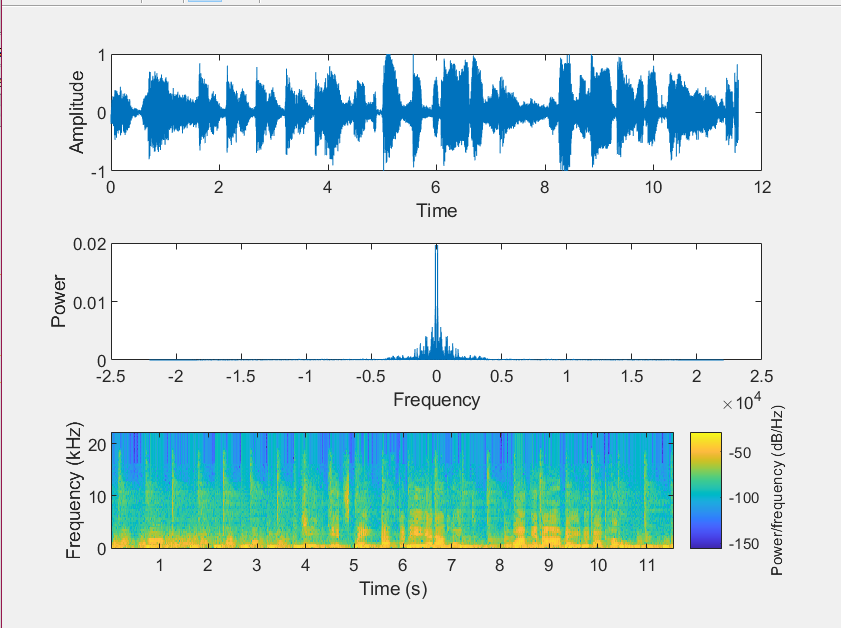


Figure 5.1



# Conclusion

# References

|  |  |
| --- | --- |
| [1] | D. P. Hill, Audio and Speech Processing with MATLAB, 2018. |
| [2] | Unknown, "THE RING MOD: MULTIPLE RING MODULATION EFFECTS," *Digital Audio Systems,* 2017. |
| [3] | C. Yu-Hsien, "ROBOTIC SOUND EFFECT," Vols. Digital Audio Systems, DESC9115, 2012. |