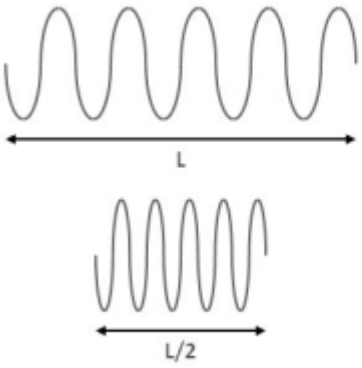
Vocoder Project

Introduction: This project aims to simulate the action of a phase vocoder, a device for performing sound signal processing. It must make it possible to analyze the main spectral components of the voice (or other sound) and produces a synthetic sound from the result of this analysis. By manipulating the parameters (here essentially the signal phase), it is possible to achieve many different effects. We will only see a small part of itin this project: the modification of the speed, the pitch and the robotization of the voice.
This kind of device is widely used today in post-production, at a time when the digitalization of music is omnipresent. We will use Matlab's software here but we will detail all the principles so that this report is accessible to all, and the same for the uninitiated to Matlab.

Principle of the phase vocoder:

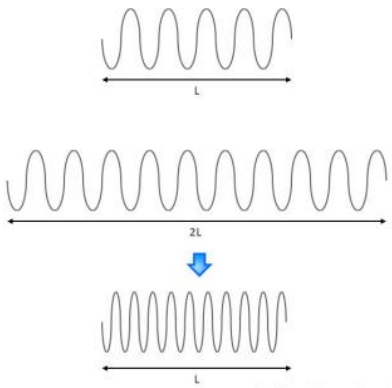
The goal is to change the pitch of an audio signal. The pitch is a fundamental parameter in the production, analysis and perception of speech defined by its fundamental frequency. The modification must therefore be made in the frequency domain. There isa "transposition"

(changing form by moving into another field) in frequency. This has the effect of changing the duration of the signal:



Ref. http://www.guitarpitchshifter.com/algorithm.html#33

However, to have the same modified sound, we must return to the same time base while managingthe changes caused by frequency transposition. We must therefore first interpolate (Intersperse values or intermediate terms (modified signal) in a series of known values or terms (original signal)) to have a signal of duration 2 times longer and then transpose the frequency (and therefore return to the right duration):



Ref. http://www.guitarpitchshifter.com/algorithm.html#33

Then, the passage in the frequency domain can be problematic if it is carried out over the entire duration of the signal: since we will perform its operations on (random) audio signals that are not stationary, the Fourier transform will not give the expected result.
However, if we consider the signal as a sum of several small parts, we can estimate that these parts vary very little over time (very short time) and that they are therefore stationary. We can therefore use the Short-Term Fourier Transform on each stationary part of the signal (which will be called frames) and thus be in the frequency domain.

Finally, after making the desired pitch changes, we will have to go back to the time domain.

To summarize:

- **Passage in the frequency domain for each frame:** the frames are overlap and are weighted by a Hamming window.

- **Transposing frames to modify the signal** : PUT THE ALGO INTERP SCHEMA
- **Retour in the time domain by combining the windowed frames:** we find the audio sound interpolated according to a chosen time vector (interpolation ratio).

A: Changing speed and pitch

We will make the changes on the extract "Dinner.wav", whose analysis (temporal, frequency and spectral) can be seen below:

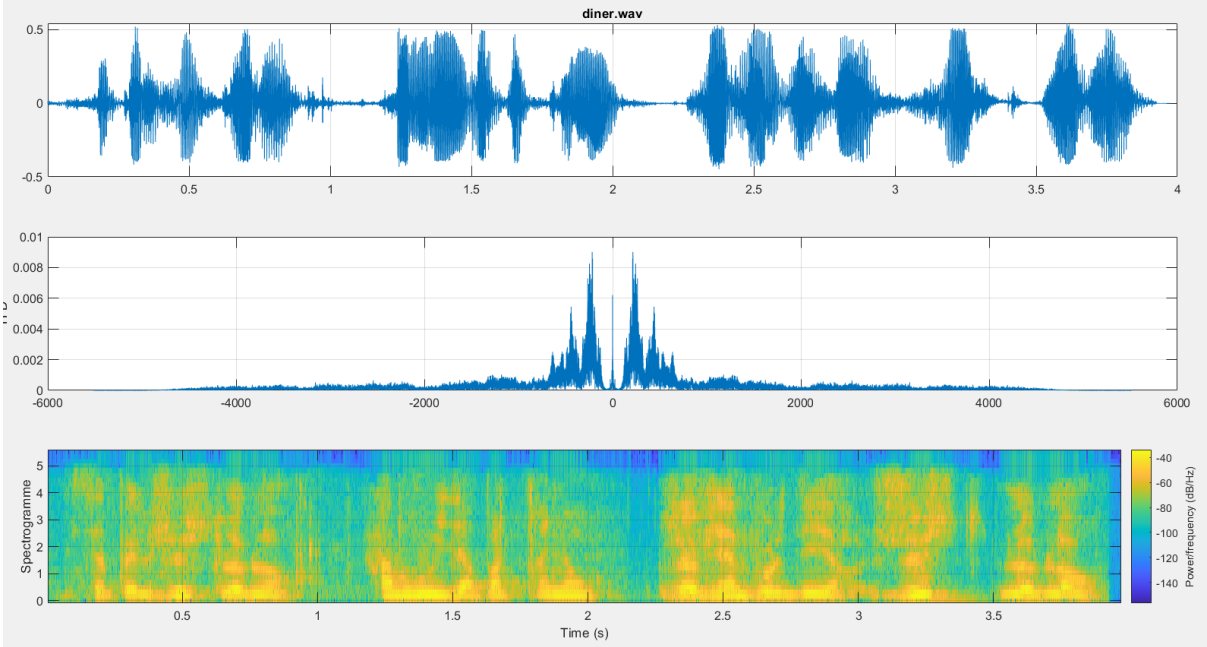
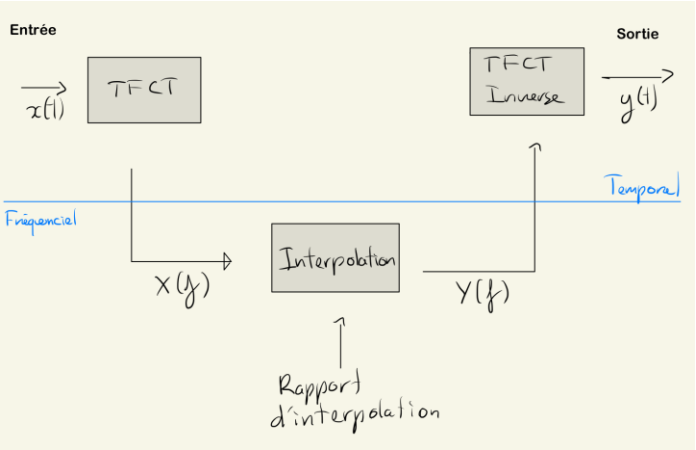


Figure 1: Time, frequency and spectrogram spectrum of the extract

Changing the speed of the voice

In this part, we want to change the speed of speech of the excerpt. The duration of the signal will therefore be modified.
To do this, after switching to the frequency domain for each frame, the programmust 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 TF of each frame obtained to the new number of interpolation windows and thus modify the speed of the signal.
Here is a summary diagram of the program.



For an interpolation ratio of 2/3 (i.e. aslowdown) results are as follows:

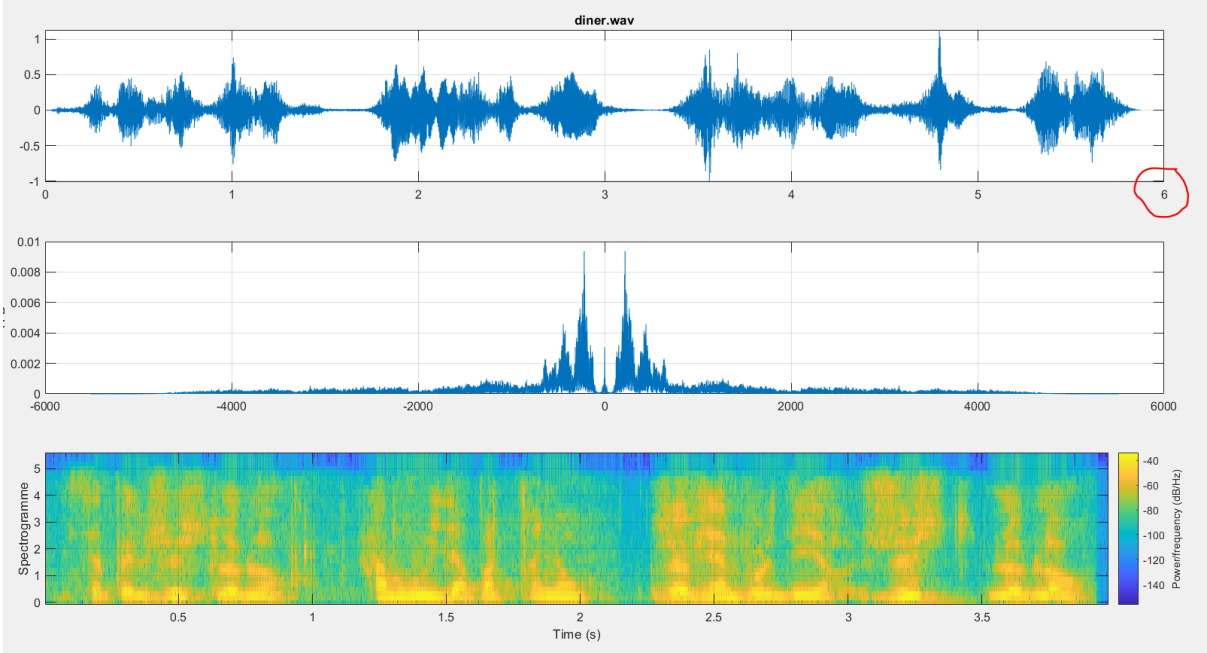


Figure 2: Time, frequency and spectrogram spectrum of the extract after decreasing speed.

It can be seen that the duration of the signal has increased, which corresponds to a decrease in speed.

For an interpolation ratio of 3/2 (i.e. acceleration) results in:

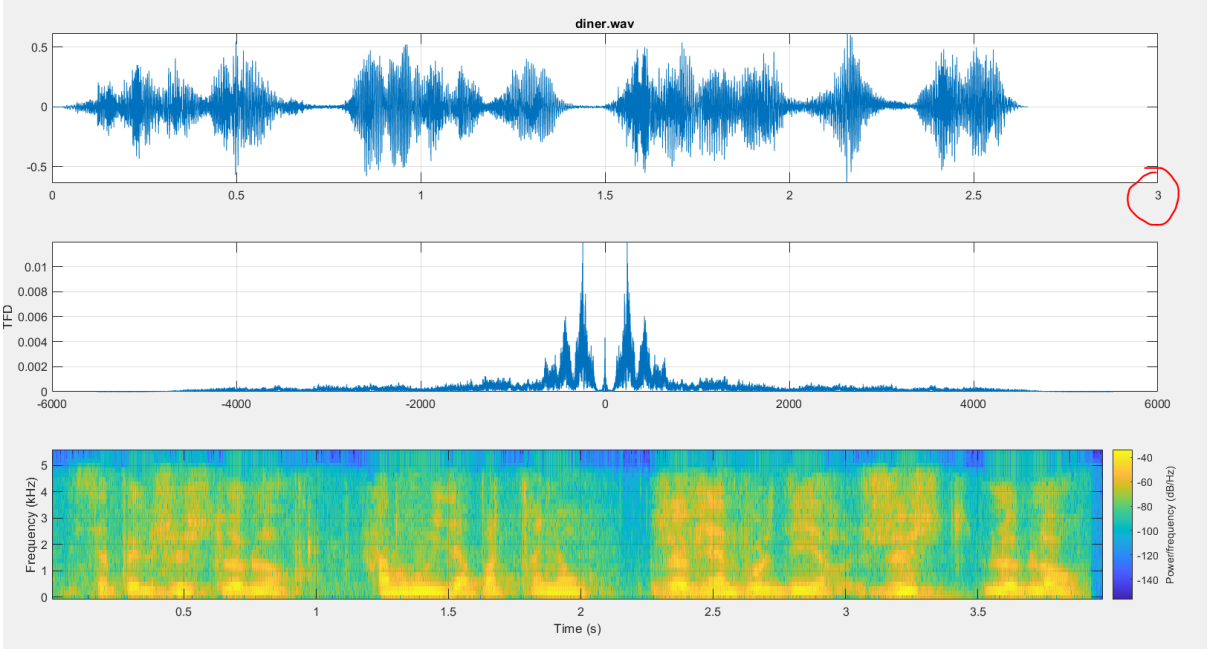
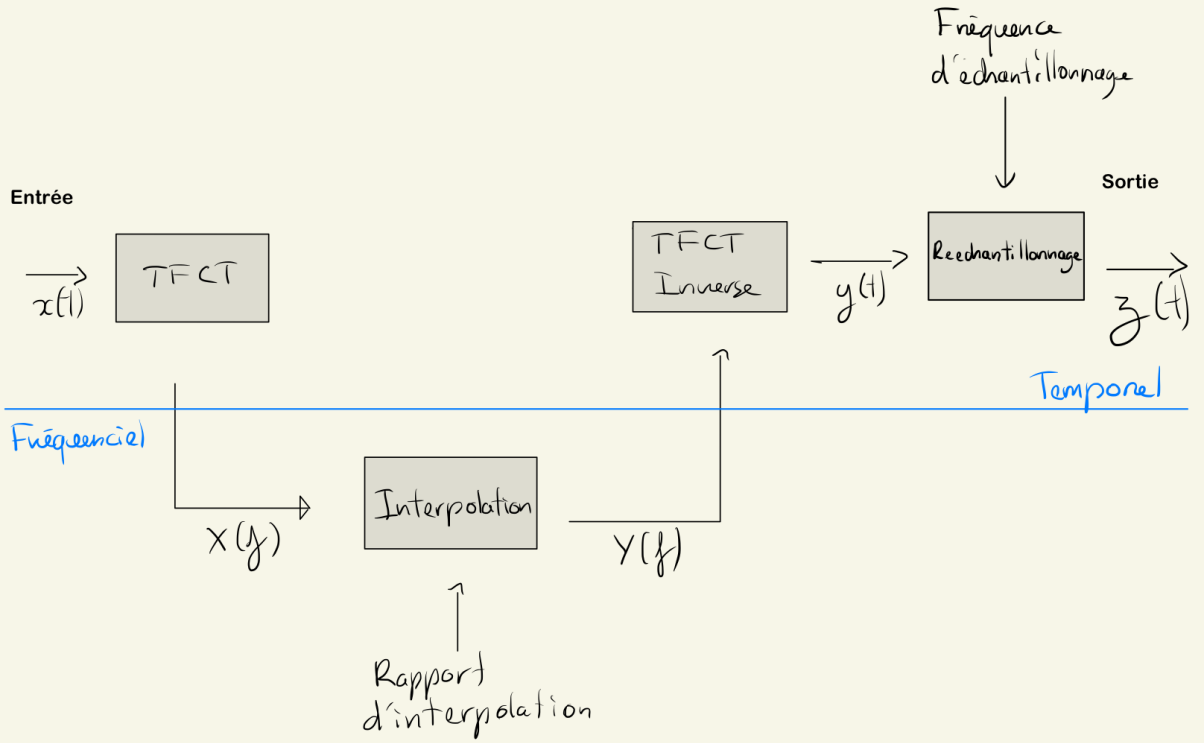


Figure 3: Time, frequency and spectrogram spectrum of the extract after increasing the speed.

It can be observed that the duration of the signal has decreased, which corresponds to an increase in speed.

Changing the pitch

In this part, we want to change the pitch of the voice of the excerpt.
This modification is done on the same principle as the modification of the speed of the voice. 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 isincreased, the duration of this extract is reduced, it is 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 donc from the pitch of the sound, which in our case, a voice.
However, here we do not want to change the speed of the extract but 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 samplelon all the Te. The introduction (above) sums up this principle well.
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.
Here is a summary diagram of the programme.



We can see the results with a coefficient of 1 for the increase in the pitch:

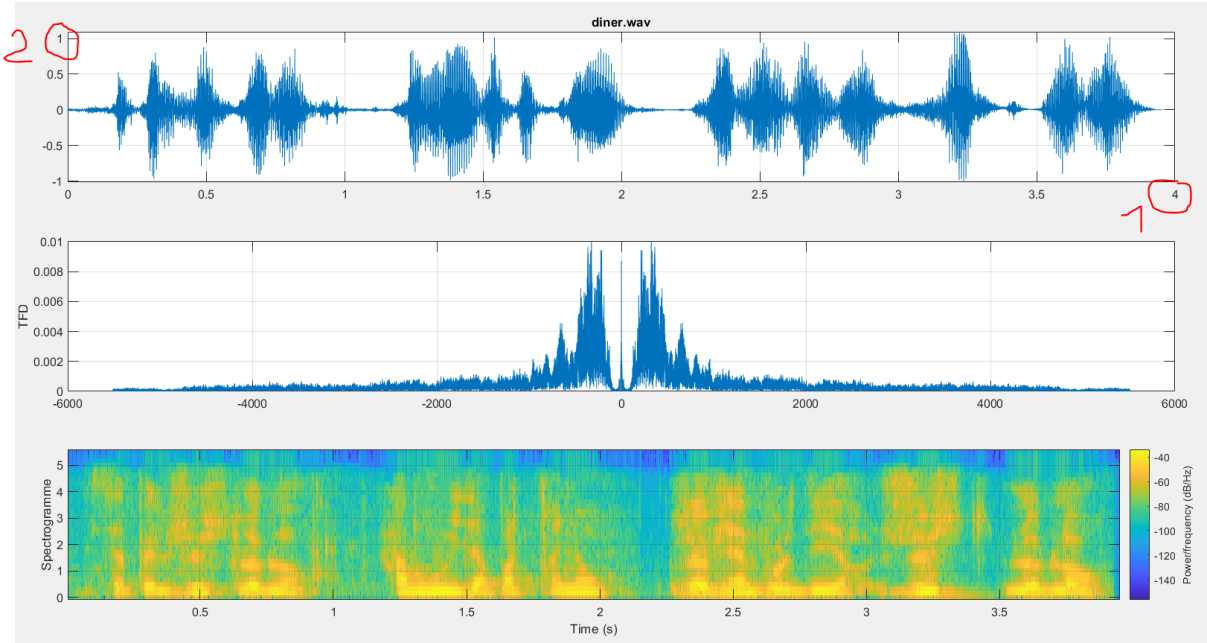


Figure 4: Time, frequency and spectrogram spectrum of the extract after pitch increase (coefficient 1).

We have the same duration as the original signal (1), however the height is modified (1). There is a change in listening.

Now an increase but with a coefficient of 5 for the pitch:

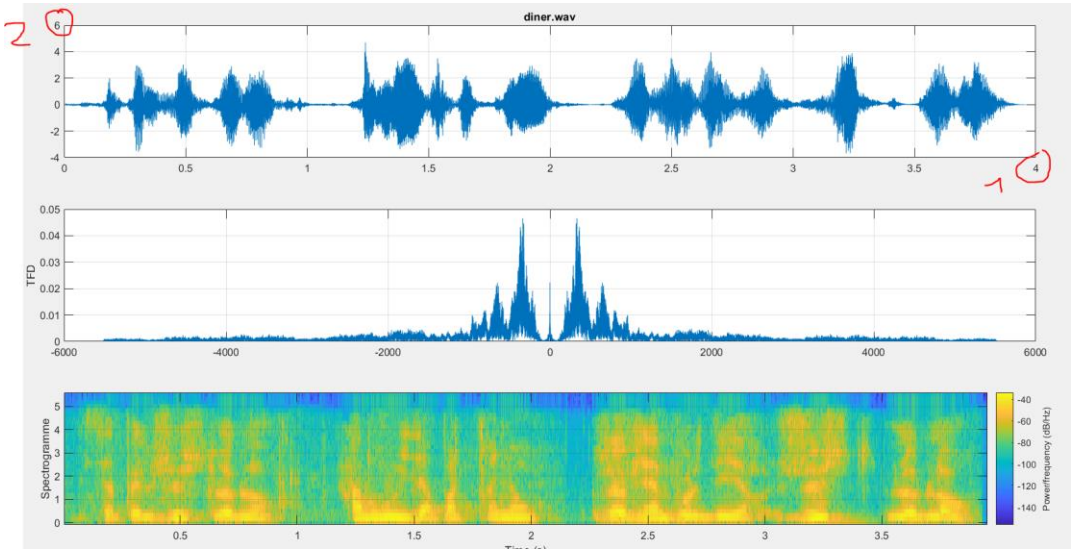


Figure 5 : Time, frequency and spectrogram spectrum of the extract after pitch increase (coefficient 5).

Now we can see that the height is much more important (2). The duration remains unchanged.
Thetemporal scan seems more "compact" (dark blue much more present), so we have an increase in frequency while keeping the general shape of the original signal.
We also see that the spectrum changes: the maximum amplitude has been multiplied by 5. The low frequencies are less present and this is felt when listening, we have the impression that the sound is higher.

We will now simulate a decrease in the pitch with a coefficient of 5:

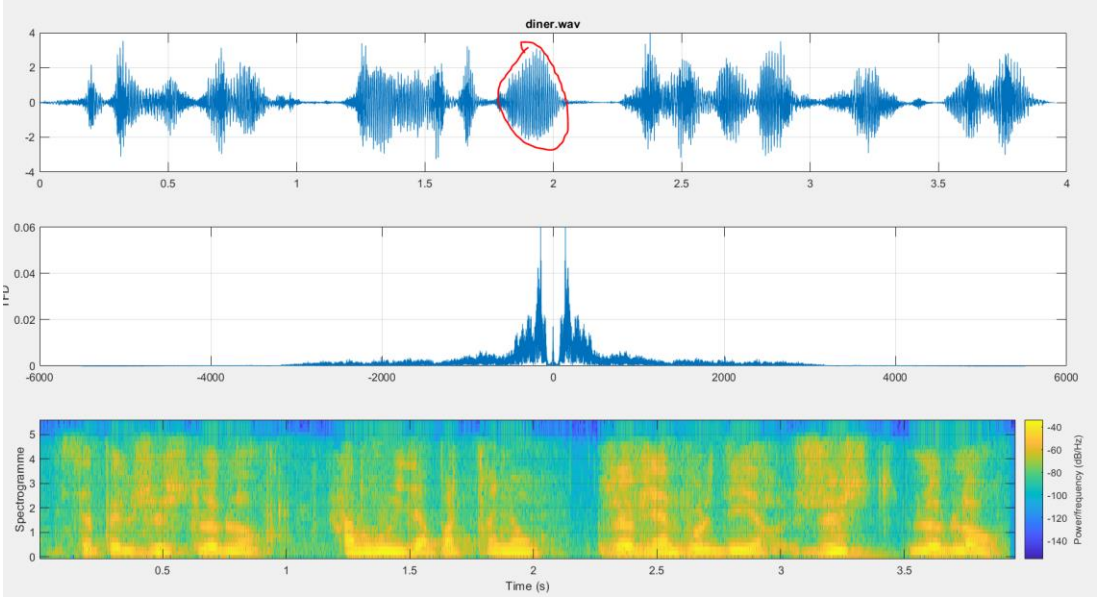


Figure 6: Time, frequency and spectrogram

Spectrum of the extract after pitch decrease (coefficient 5)

We notice a "lightening" of the time signal, it is less provided in high frequencies. The spectrogram shows a "weakening" of high frequencies which is felt on listening, one has the impression that the voice is more serious.

The pitch therefore corresponds to the frequency composition of the sound. Its modification makes it possible to make a sound lower or higher while keeping the same sound.

B: Robotization of the voice: " rob.m "

In this part, we want to get a robot voice from a voice snippet.
The modification is done in the time domain, so the principle is different from the other 2.
To achieve this, the functionn rob.m will have to modulate the audio signal by a complex exponential at a frequency fc and recover its real part.
This frequency fc will be representative of the degree of robotization of the voice.

To achieve this, we must first generate a time vectort from the sampling rate Fs and the size of the audio signal (number of samples).

Then we must multiply each element of the audio signal by the complex exponential, so we must use the operator "x.".
To endir, if we use the vector t directly, we let MATLAB do a vector multiplication (very long calculation). It is therefore necessary to use the transpose of t or t'.

Finally, the real () function allows you to select only the real part.

We havenow shown the results obtained for different values of Fc.

For Fc = 200 Hz:

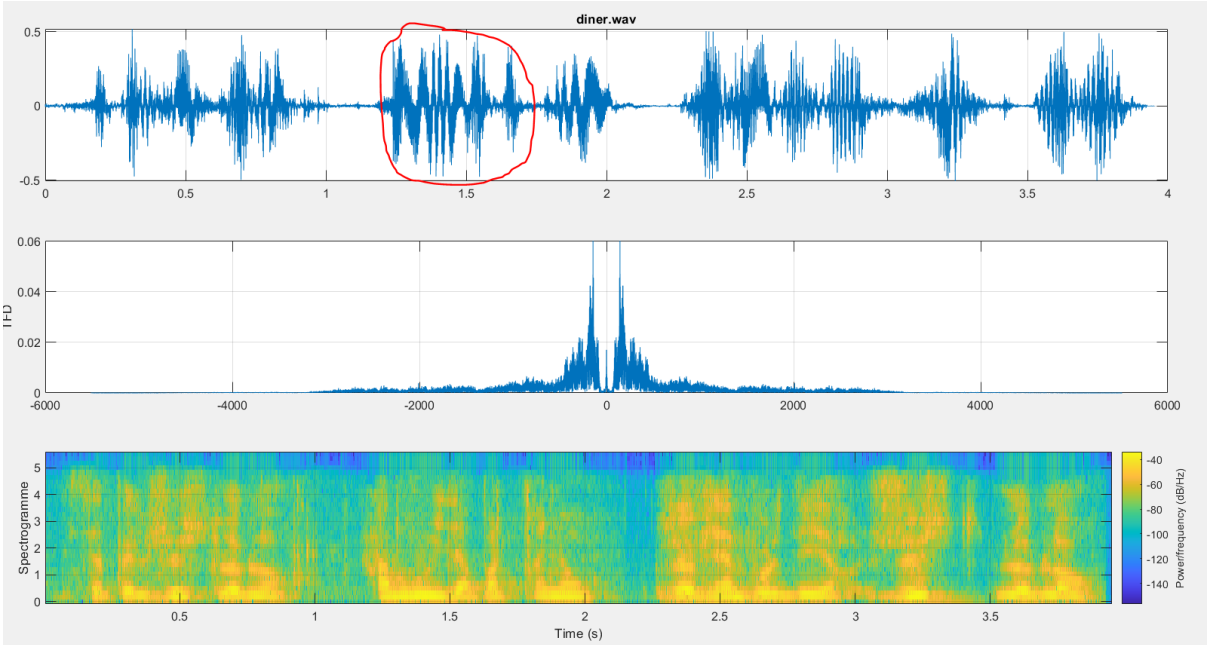
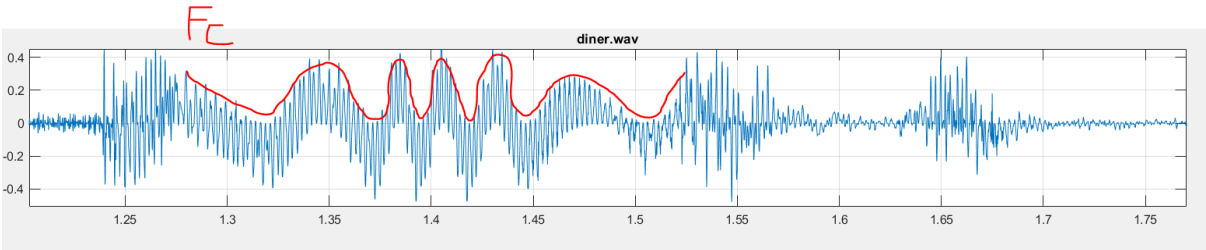
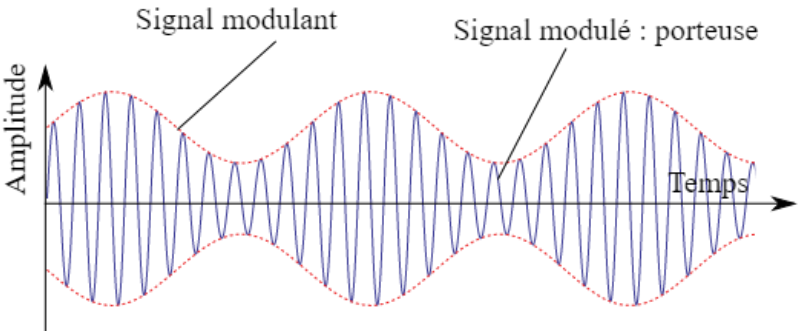


Figure 7: Time, frequency and spectrogram spectrum of the extract after robotization (frequency 500Hz).

For Fc = 200 Hz, we can see that the time sig nal retains the different "packets" (which correspond to the lyrics) but that these packets, although they retain their places, are modulated by the frequency Fc:

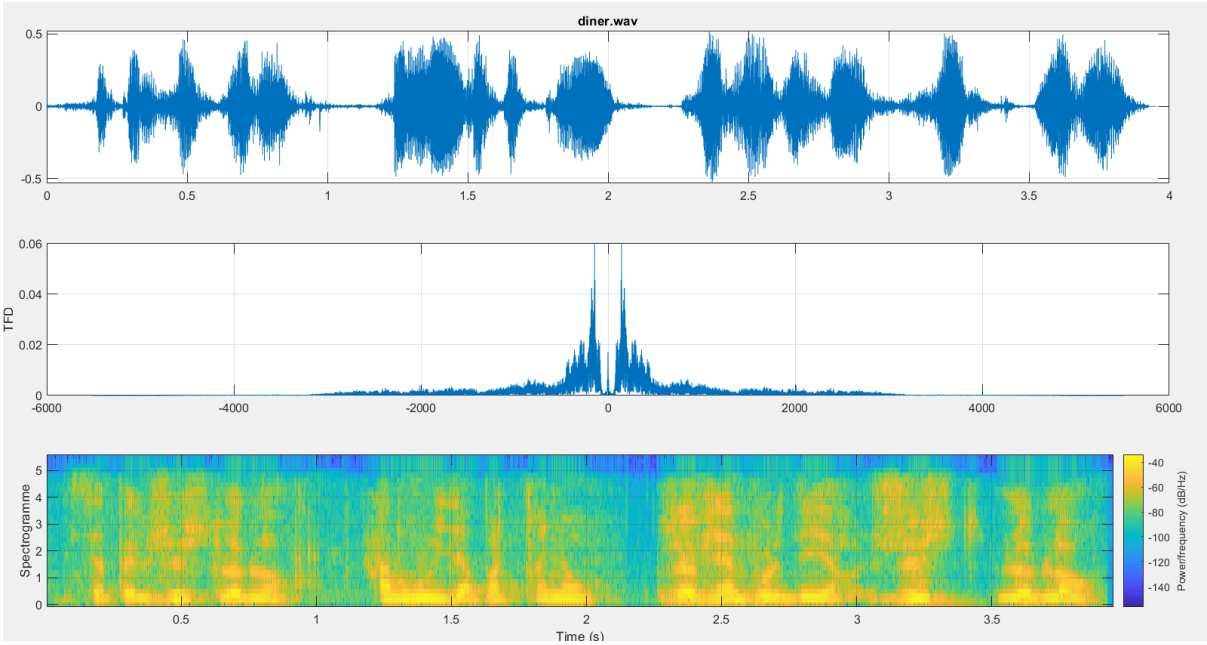


The diagram below summarizes the basic principle:

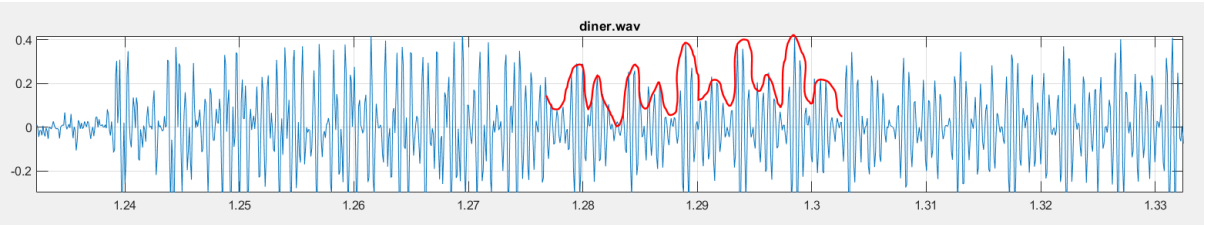


With exponential Fc the modulating signal and speech the modulated signal.
It can be seen that the modulating signal must have a lower frequency compared to the frequency of the modulated signal, otherwise there will bedistortion of the modulated signal (therefore of speech).
The fact of modulating the signal leads to frequency variations and therefore a "tremor" of the voice: this is felt when listening: we have the impression that the voice is robotized.

Now with an Fc = 2000 Hz:



We see that we can not see the modulation observed previously without zoom. We find the packets of lyrics but they are distorted because the frequency Fc is much too high:



So we lose the original signal completely and it is felt when listening, the lyrics are unintelligible.

C: Bonus effects:

All the effects that we will describe and the explanations that we will provide have been described in the document available on the internet at this address:
https://users.cs.cf.ac.uk/Dave.Marshall/CM0268/PDF/10_CM0268_Audio_FX.pdf

We just transcribed what we understood to include it in our project. Not all of the effects detailed below are implementedin our project.

We will now implement other effects to our project. These effects can be classified by the way they are generated:

-**By Filtering:** Low-pass, High-pass etc ... which make it possible to suppress/attenuate sounds of the spectrum and therefore of the original signal (e.g. suppression of the double bass)
Equalizers (timbre corrector): which make it possible to amplify or reduce certain frequency bands
-**Variable filter in time:** Wah-wah and phaser effects...
-By playing on the **delay:** Vibrato, Flanger, Chorus and Echo ...
-With **modulators (amplitude and phase):** Ring modulation, Tremolo and Vibrato wah-wah..
-With **non-linear treatments**: Compression, Distortion, Limiters, Exciters, Enhancers ...
-Spatial effects: Panning, Reverb, surround...

We will not detail all these effects but 1 per category mentioned above which we will try to implement matlab.
It can be noted thatsome effects can be obtained with several different techniques (e.g. wah-wah), more or less effective depending on the audio signal.

For digital filtering effects, TP2 summarizes the possibilities well with the example of the suppression of thedouble bass and the enhancement of the voice.

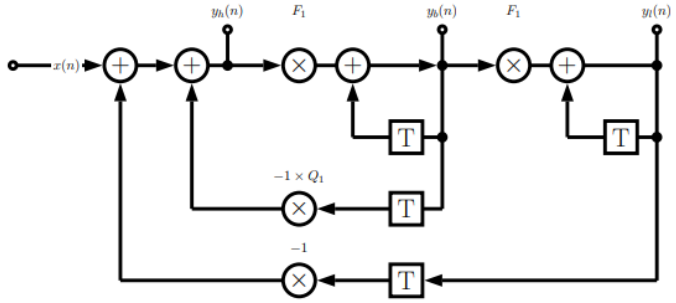
Wah-wah effect:

For effects using a time-varying filter, we will detail the Wah-wah effect:

The Wah-Wah effect is mainly used by guitarists: it creates anexpressive character by producing a sound that seems to be pronounced by an "oua" voice.

It is achieved through a bandpass filter with a central frequency (resonant) variable in time and a small bandwidth. It is a mix between the filtered signal and theoriginal signal.
Its realization can be summarized in 3 steps:
-A triangular wave is created to modulate the central frequency of the
Bandpass filter.

-We implement the state variable filter which works according to the following principle:

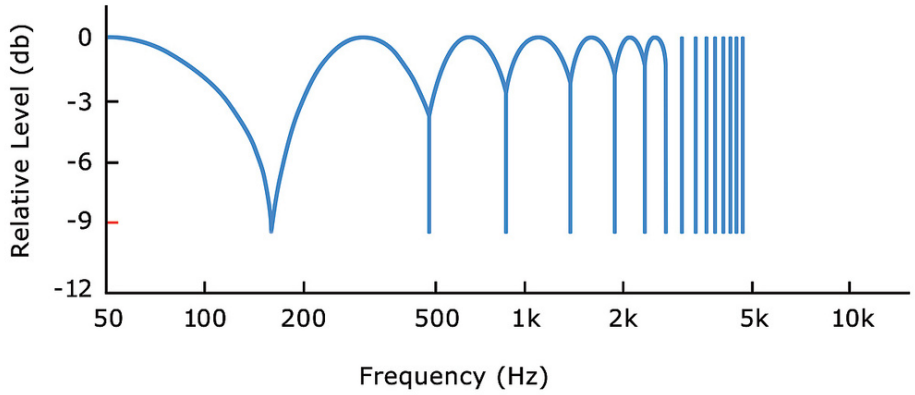


We find theinput signal, the filtered signal with a low pass (yl), a band pass filter (yb) and a high-pass filter (yh).

-As long as the center frequency is in the state variable filter loop, the operation is repeated.

Delay-based effects:

These effects are based on the use of FIR and IIR "comb" filters, or the combination of the 2 depending on the desired effect, different comb filters are used:



These filters cause a delay of the signal, then the delayed signal and the original signal are added. The different effects are obtained by varying the delay as shown in the table below:

Effect	Delay Range (ms)	Modulation
Resonator	$0 \ldots 20$	None
Flanger	$0 \ldots 15$	Sinusoidal (≈ 1 Hz)
Chorus	$10 \ldots 25$	Random
Slapback	$25 \ldots 50$	None
Echo	> 50	None

We will detail the vibrato effect, which allows you to vary the height of the signal at a much faster speed than manually.
To achieve this, we will have to vary the delay periodically between 5 and 10 Ms.

Modulation-based effects:

Voice robotization isan effect based on phase modulation.
There are others based on amplitude modulation such as the tremolo effect (mainly guitar): it gives the impression that the notes "shake, go back and forth".

To do this, the o riginal signal is modulated by a sinusoidal m (n) signal at a frequency of less than 20Hz (low frequency oscillator):

$$y(n) = (1 + \alpha m(n)).x(n)$$

Alpha is the modulation ratio with alpha = 1 = > maximum effect and alpha = 0 = > modulation cancelled.

1+αm(n) = `(1+ alpha*sin(2*pi*index*(Fc/Fs)))`

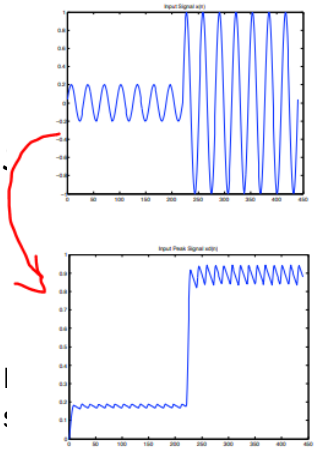
with alpha and Fc to fix (here we took Fc = 5 and alpha = 0.5) and index the number of samples of the signal.

It works well.

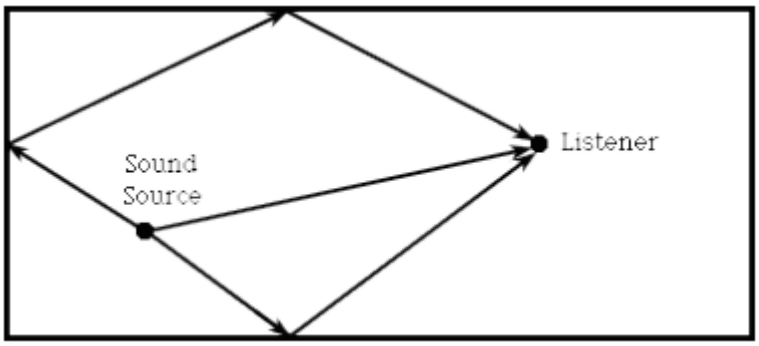
Effects based on non-linear treatments:

Nonlinear processing results in the creation of harmonic/non-harmonic frequencies in the original signal.
There are 3 types of non-linear processing: dynamic processing (compressors/limiters), intentional harmonic processing (distortions...) and exciters/amplifiers.

In this project, we will simply make a limiter: it avoids amplitude peaks while keeping the dynamics of the signal.
It must react quickly to measure the peak and reduce it if it exceeds the desired value.
An illustration can be seen below:

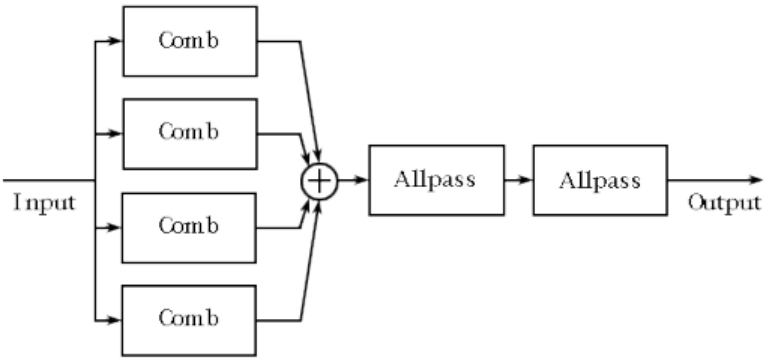


e effect of reverberation: the result of the reflection of the original
om as shown inthis example:



As can be guessed, a sound reverberated by a wall will have a delay and attenuation compared to the original sound. In addition, it can reverberate several times, so we have a series of sound that is increasingly attenuated and retardé that allows to create this effect.

In this project, we will simulate a Schroeder reverberation that uses IIR filter types, comb filters and boilerplate filters that can be summarized by this diagram:



This is the version we haveincluded in our project.

D: Conclusion

We can say that this project has allowed us to understand and apply certain principles of sound processing (and therefore audio signal). We were able to play on different characteristics of a sound:

- Its intensity/volume: depends on the amplitude of the signal, son unit of measurement is the decibel (dB).
- Its pitch/frequency: defined by the vibrations of the object (including vocal cords) creating the sound, is expressed in Hertz (the higher a sound is, the higher its frequency is and vice versa).
- Its timbre/color: the nshadow and the intensity of the harmonics that compose it and allows to recognize the person who speaks or the instrument that is played.

We understood the importance of phase and amplitude continuity, and the actions put in place to ensure that thiscontinuity is respected.
We were able to observe the effects of these treatments on the temporal representation, spectrum and spectrogram of the signal.

We could have taken the study of modified signals a step further by adding (for example) correlations/auto-correlation and improved the project by providing a graphical interface. However, we took a long time to understand the basics of the treatments carried out in this project (which is still the purpose of the course) and we feel that we have done everything we were capable of to complete this project.

Thank you for reading.

Martin Schetter & Vincent Lisette