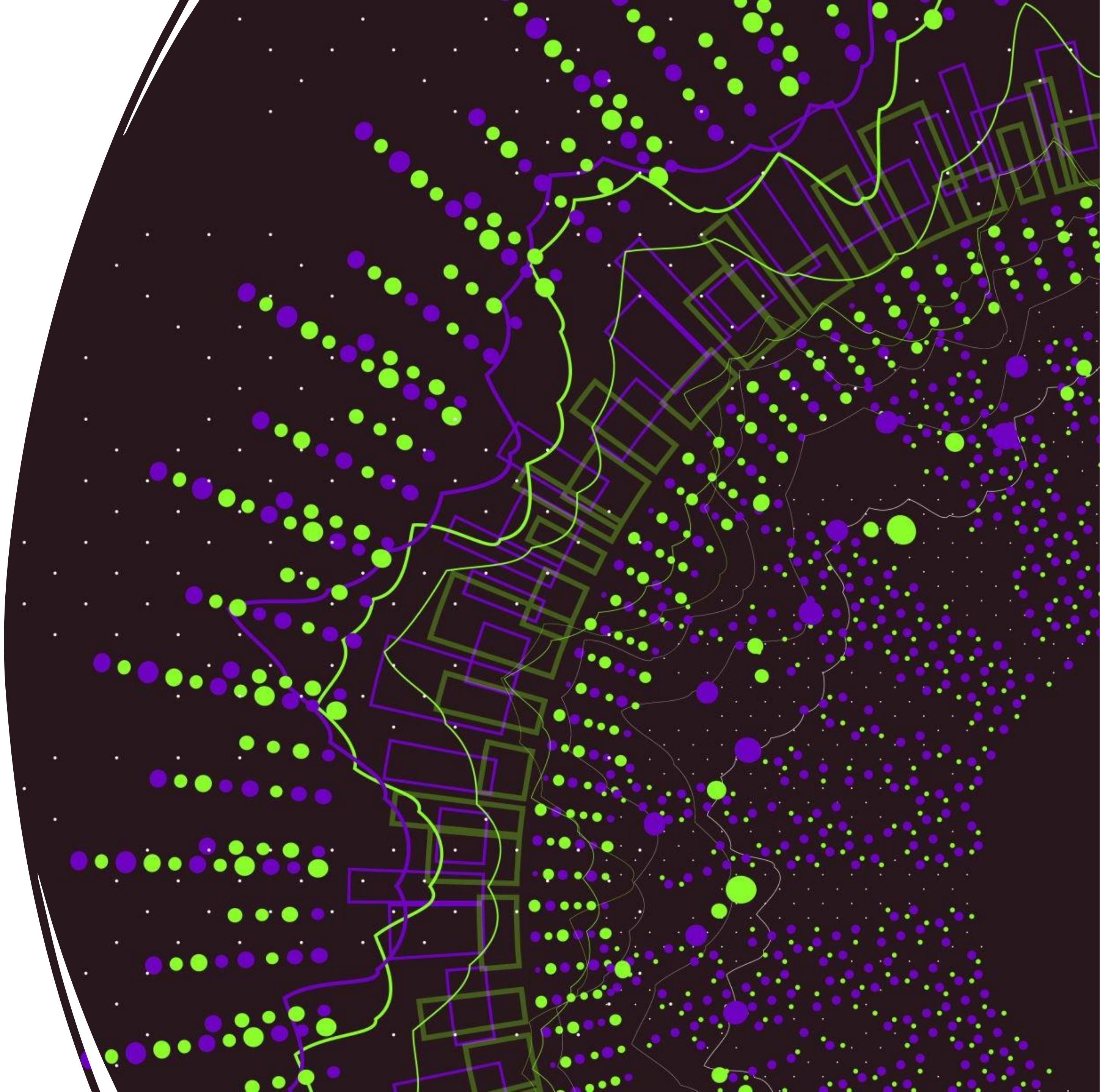
APPLIED SIGNAL PROCESSING LAB

- Paolo Allione s296500
- Camolese Claudio s297378
- Temmuz Sanli s298029



A4.1

Sum of two signals

Consider two signals with random phases;

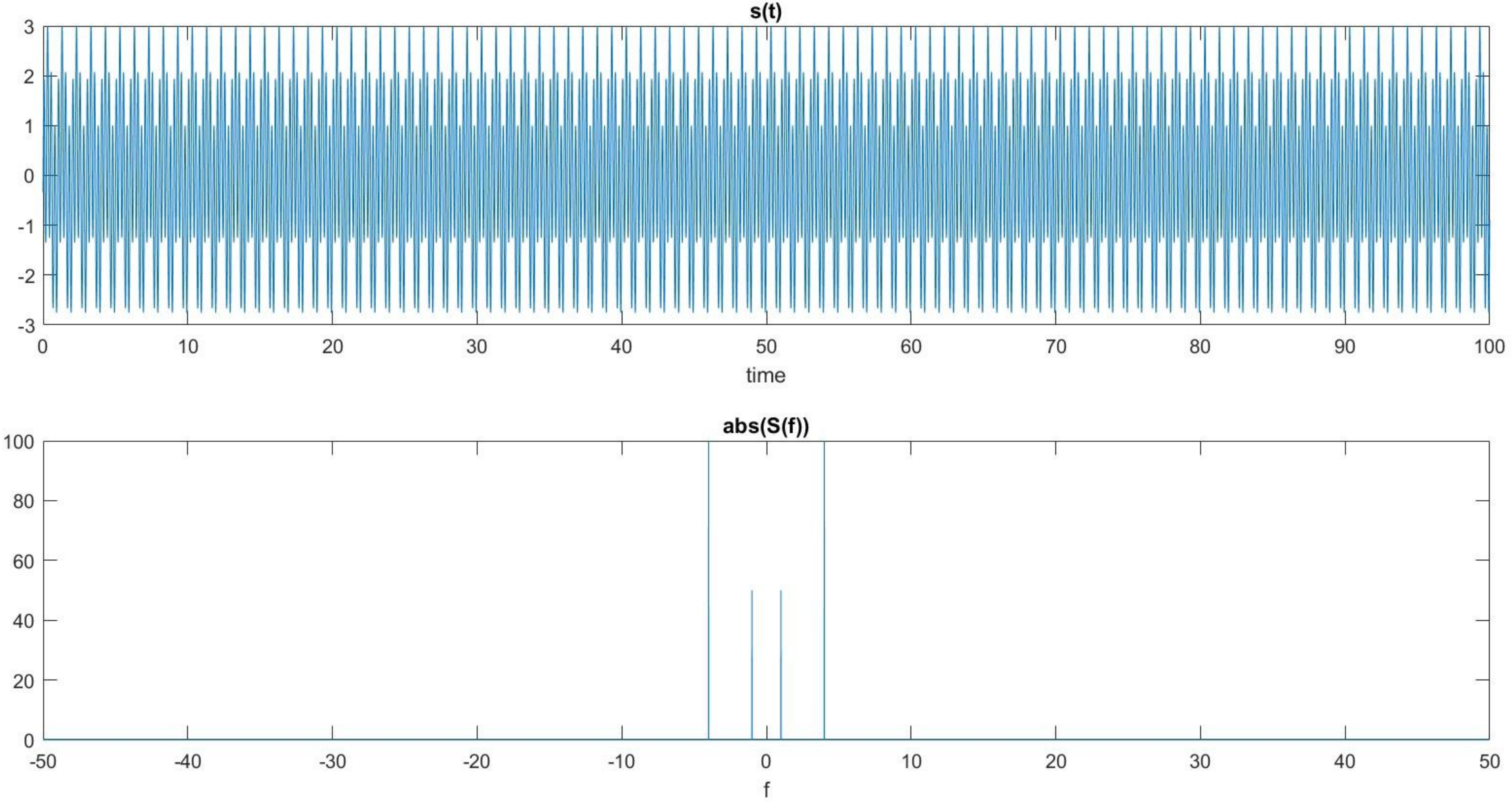
Consider their sum;

Plot s on the time axis, compute the fft and plot abs(S) on the frequency axis.

$$s_1(t) = A_1 \cos\left(2\pi f_1 T + \phi_1\right)$$
$$s_2(t) = A_2 \cos\left(2\pi f_2 T + \phi_2\right)$$

$$s(t) = s_1(t) + s_2(t)$$

Output



s(t)
3
2
1
0
-1
-2
-3
0
10
20
30
40
50
60
70
80
90
100
time
abs(S(f))
100
80
60
40
20
0
-50
-40
-30
-20
-10
0
10
20
30
40
50
f

Signal s’

Now consider a signal s’ where the first part is equal to the first part of s_1 and the second part is equal to the second part of s_2.

$$s'(t) = P_{T/2}(t)s_1(t) + P_{T/2}(t - T/2)s_2(t)$$

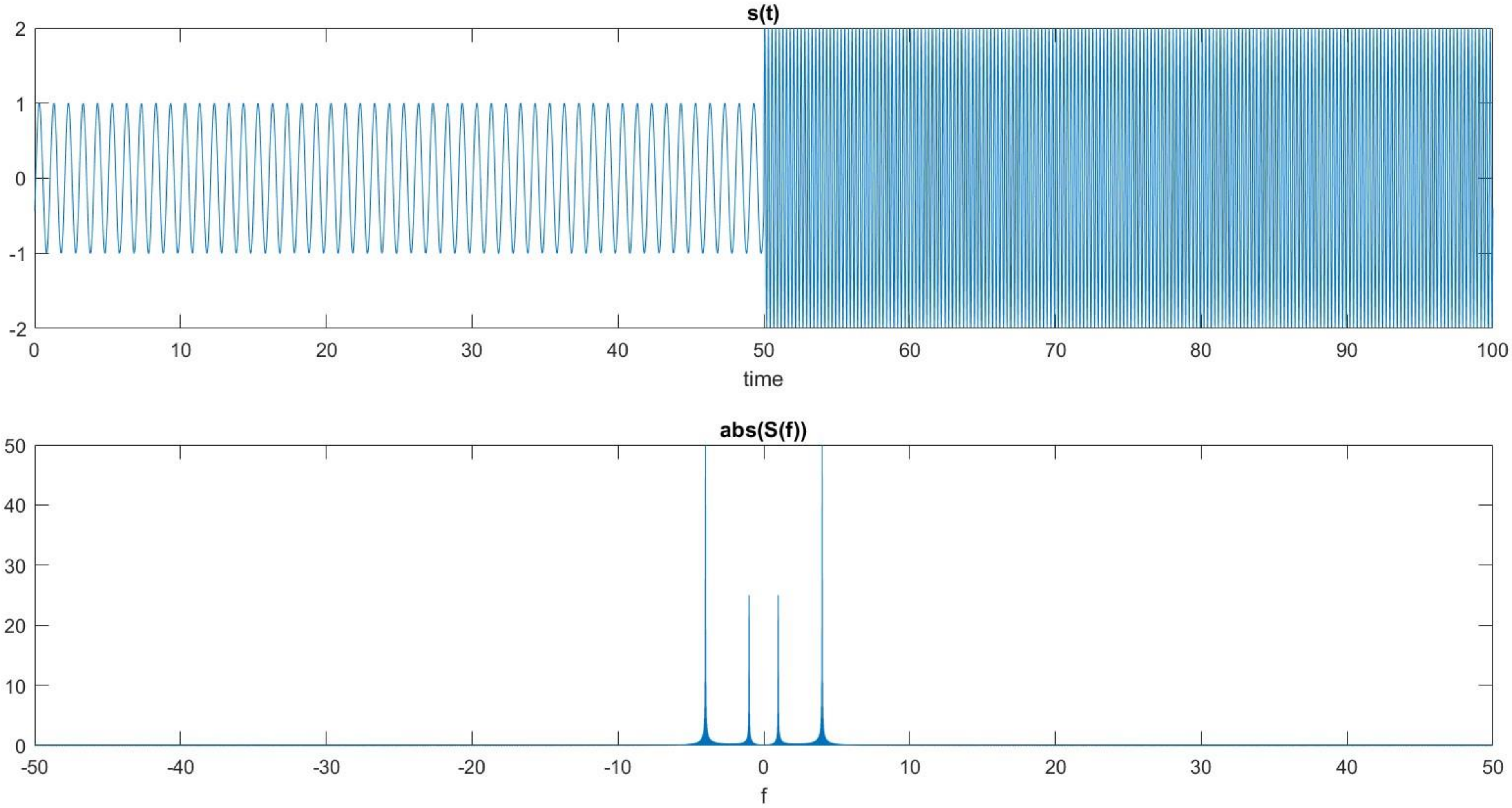
In order to create this signal a filtering function is used as follows:

```
%% s'
P= rectangular(t,T/2);
s_first=P.*s1 + (1-P).*s2;
```

We need to plot s’ on the time axis, compute the fft and plot of abs(S’).
Last part requires to do a comparison of the spectrum profile between s and s’

```
%% Fourier s'
S_first = fft(s_first);
M_first = abs(S_first*Ts);
MM_first = fftshift(M_first);
```

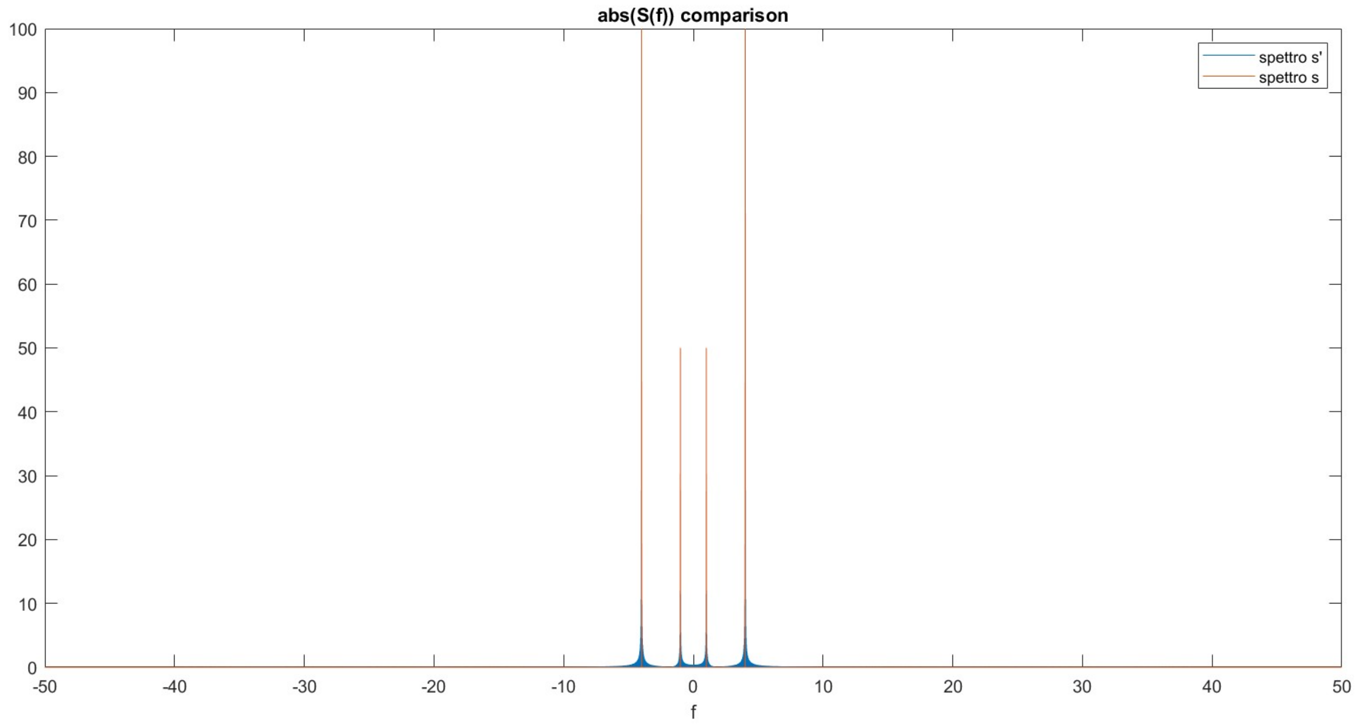
Output



s(t)
2
1
0
-1
-2
0
10
20
30
40
50
60
70
80
90
100
time
abs(S(f))
50
40
30
20
10
0
-50
-40
-30
-20
-10
0
10
20
30
40
50
f

Comparison

As it can be seen, apart from the amplitudes and some spurious components due to the transition, the two fft plots have the same spectral components.



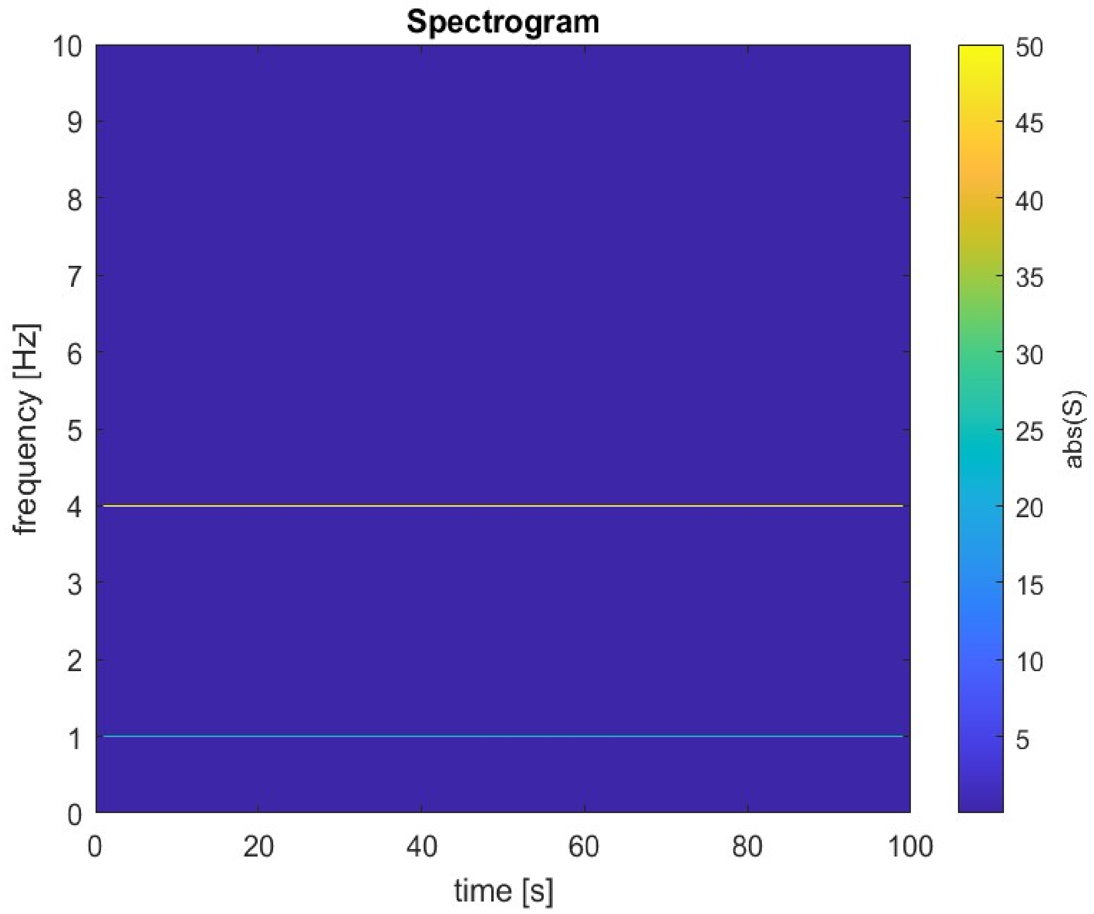
STFT

When a signal is not stationary, the FFT can't provide the true frequency structure of the signal.

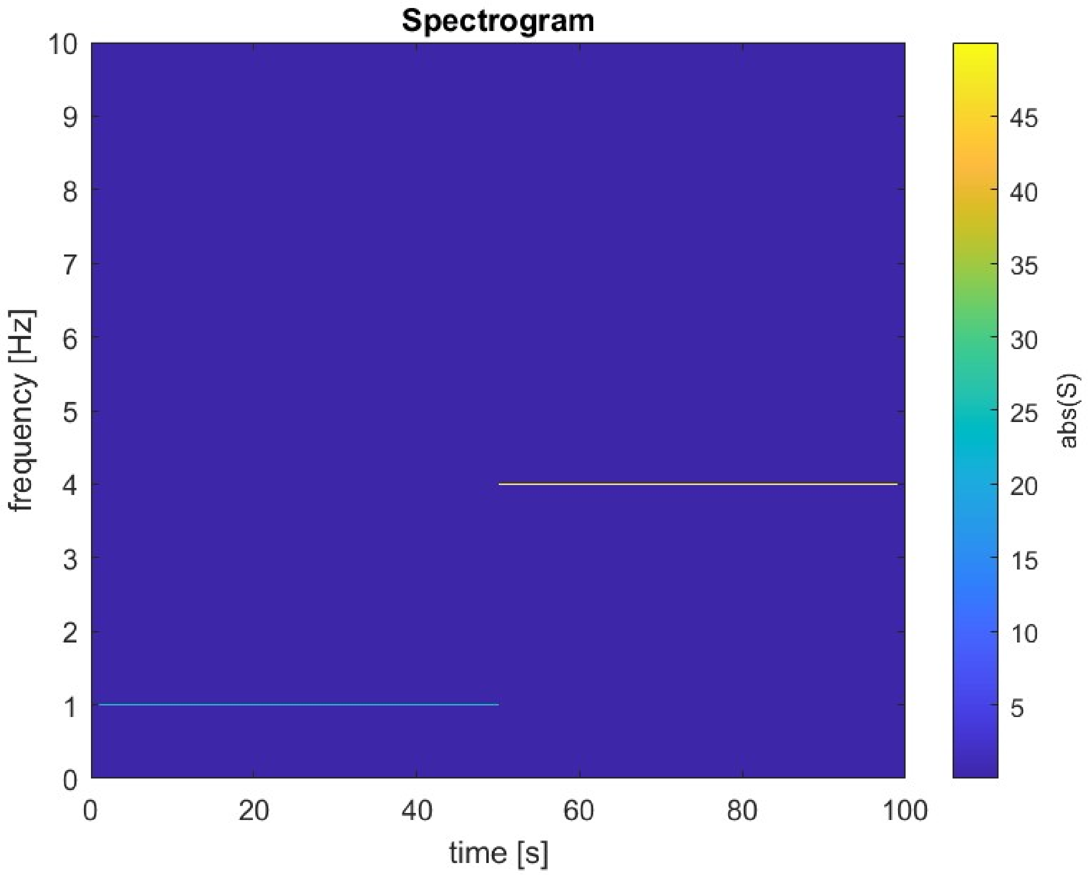
In this case, STFT can be used applying the FFT to a window of $N_{FFT} < N$ samples, then moving forward the window from the left to right until all the N samples have been processed.

The result is a matrix which can be used to plot an image time-frequency where the color represent the magnitude of S.

Output



Spectrogram
frequency [Hz]
time [s]
abs(S)
0
1
2
3
4
5
6
7
8
9
10
0
20
40
60
80
100
5
10
15
20
25
30
35
40
45
50



Spectrogram
frequency [Hz]
time [s]
abs(S)
0
1
2
3
4
5
6
7
8
9
10
0
20
40
60
80
100
5
10
15
20
25
30
35
40
45

Comments

Both images indicate that the signal contains frequency components at 1Hz and 4Hz, which are the initial sinusoidal components we imposed at the beginning. Also the colors, which represent the intensity of these frequencies, are the same.

However, the two images have a significant difference. While the spectrogram of s shows frequencies that are constant over time, the spectrogram of s', due to how the signal is defined, has frequency components that are not constant.

In particular, we can identify two-time windows: in the first one (0-50s), only the 1 Hz frequency component is present, and in the second one (50-100s), only the 4 Hz frequency component is present.

As a result, the signal exhibits non-stationary behavior and is distinguished by the presence of two harmonics appearing at different times.

A4.2

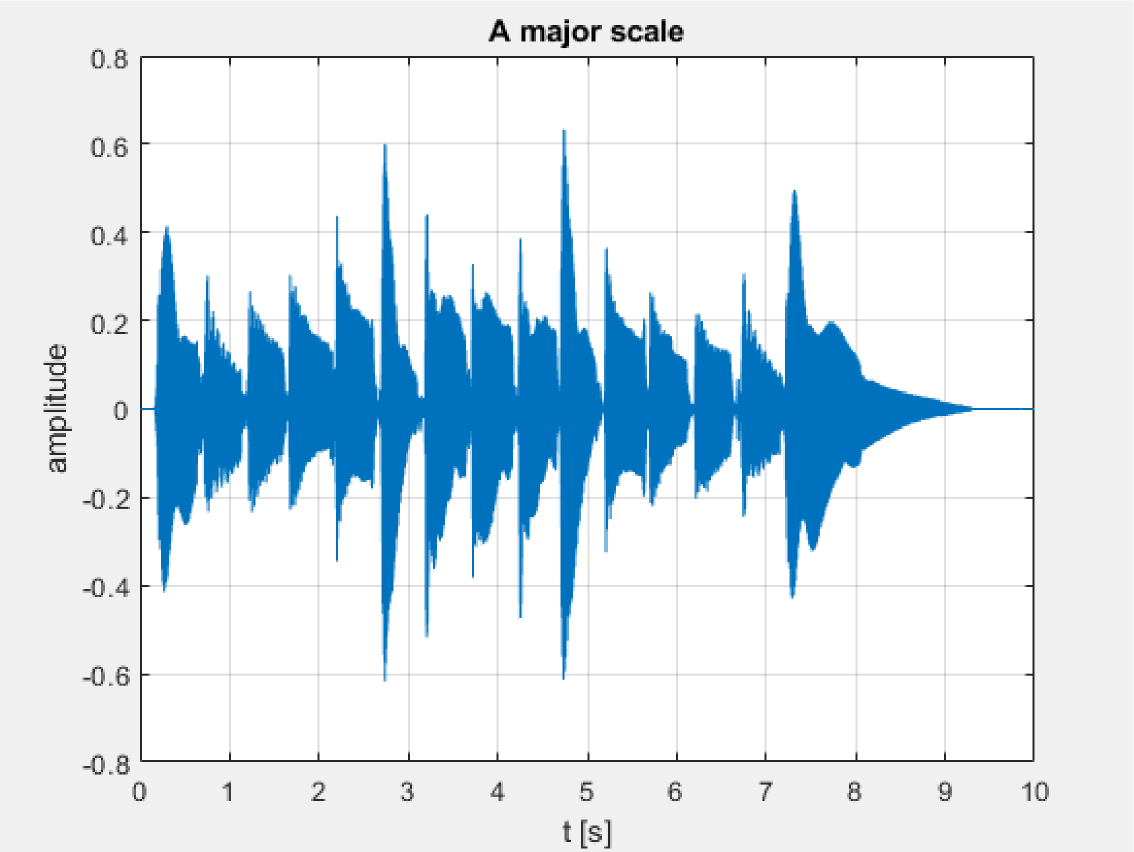
Spectrogram applied to a music audio

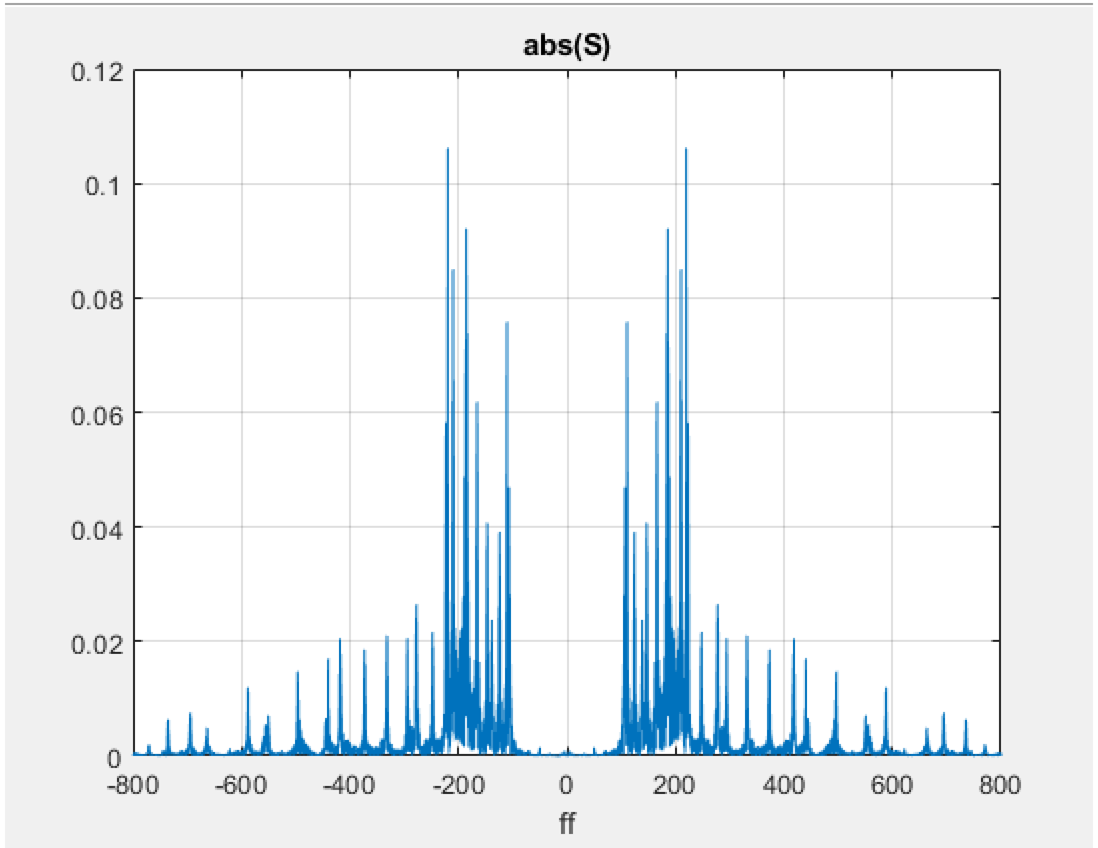
- GOAL : This exercise focuses on the examination and modification of a music signal. Specifically, it involves performing a time-frequency analysis to study the characteristics of the signal both before and after a pitch alteration, which involves raising the pitch by 3 semitones. The goal is to understand how the signal's properties change as a result of this modification.

- The signal analyzed is an A major guitar scale contained in a wav file and it was imported thnaks to the following Matlab program:

```
%% import audio file
[s,fs] = audioread("audio.wav");
```

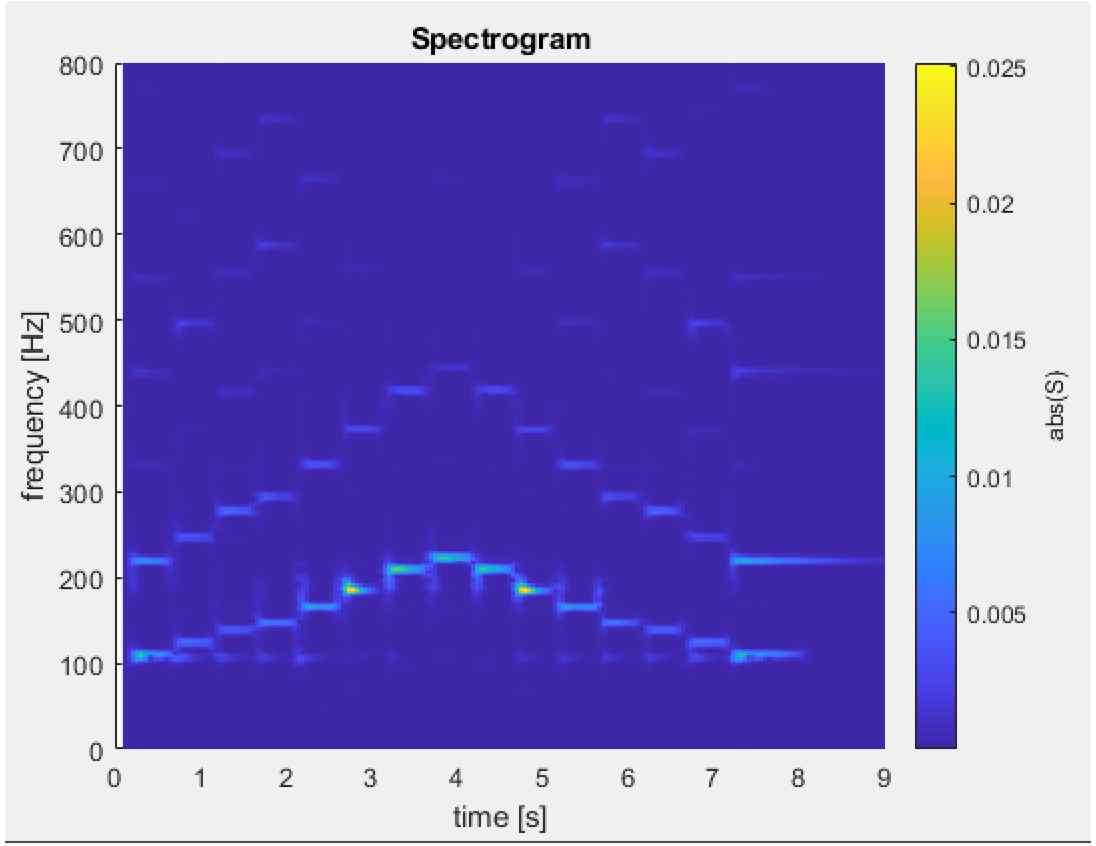
Output : signal in the time domain and its spectrum obtained with the FFT.





SPECTROGRAMS

- The analysis of this graph could be predicted looking at the signal in the time and frequency domain. Indeed, the frequency range between **100 Hz and 200 Hz** is the most pronounced, indicating the primary peaks of the spectrum.
- The audio file, when played, mimics a guitar scale with the pitch ascending and then descending, a dynamic that the spectrogram captures accurately.



Pitch shifted

- The pitch of the audio is shifted by 3 semitones, by using the following Matlab code

s = shiftPitch (s, 3)

As a consequence, we obtein a new spectrogram which has the same scale execution (same shape), but is shifeted along the vertical axis by about 30 Hz (easily visible by looking at the new spectrogram).

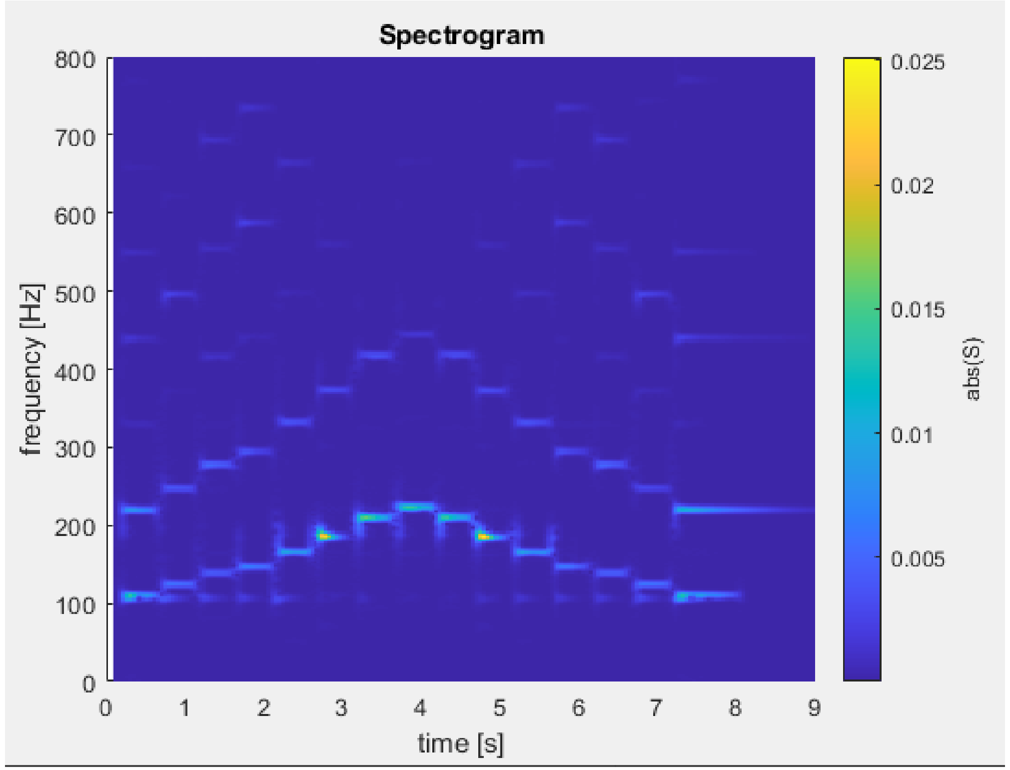
Moreover, playing the signal through:

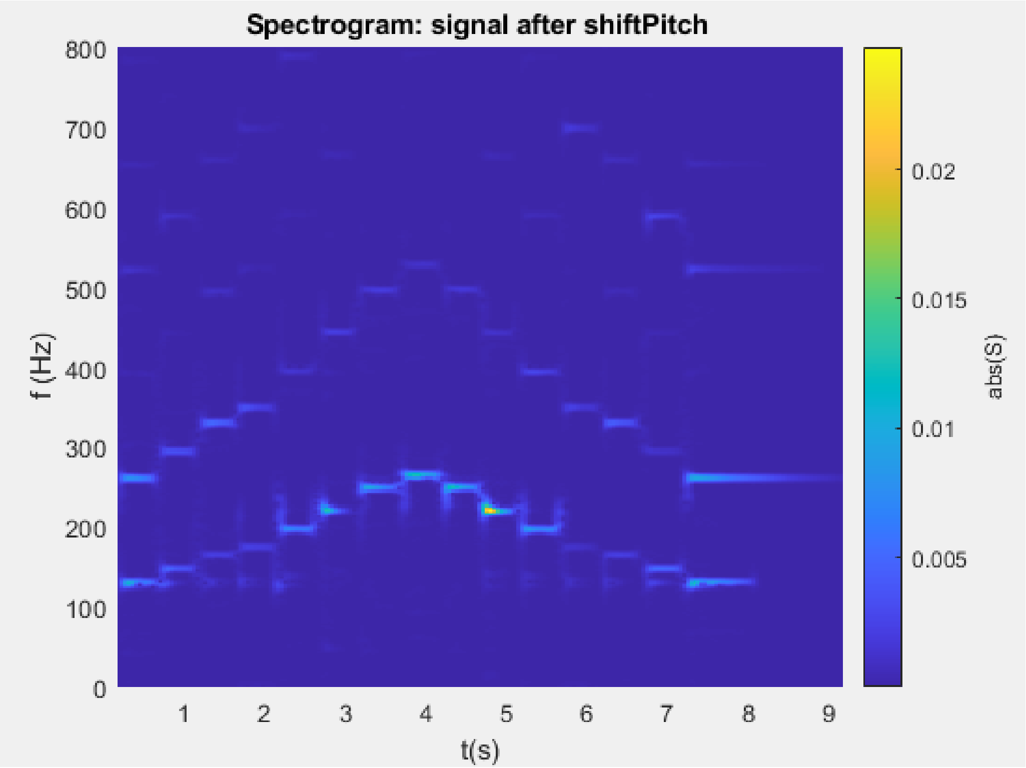
playObj = audioplayer (s, fs)

playblocking (playObj)

It is possibile to hear that the new signal is higher in tone.

Comparison of spectrograms





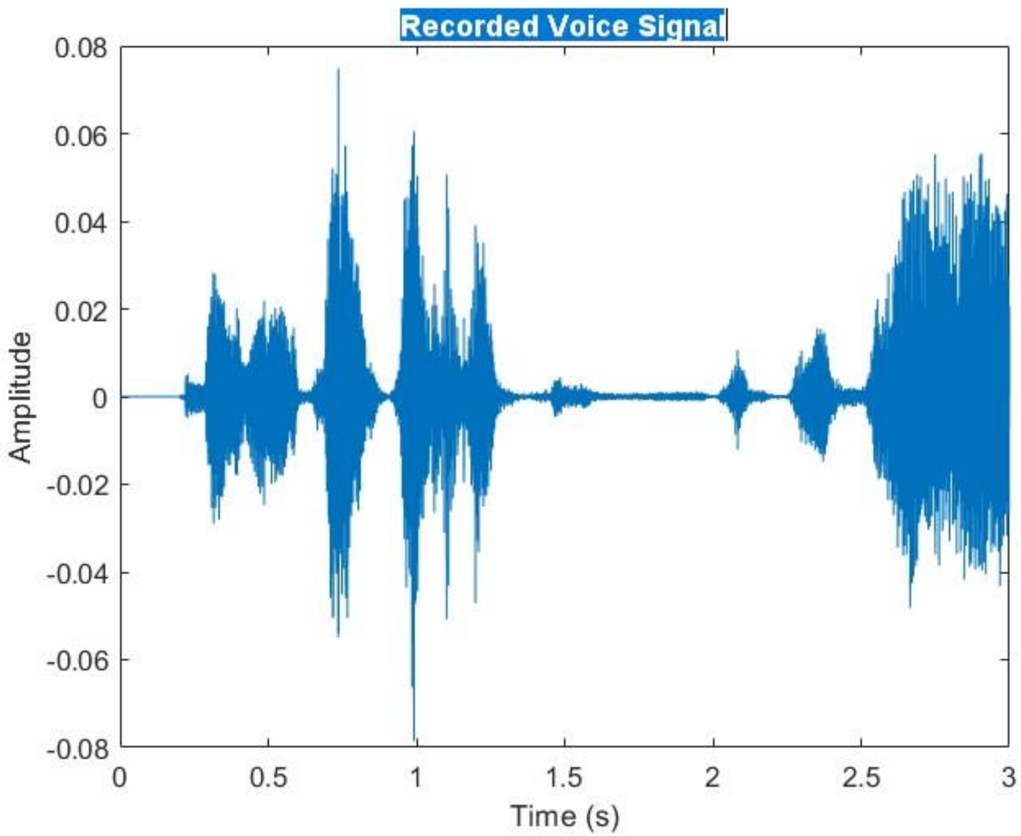
A4.3

Part 3 - Processing of your voice

The goal of this exercise is to analyze and manipulate audio signals by adding echo, shifting pitch, and computing Fourier transforms, then compare their frequency spectra and verify properties through playback.

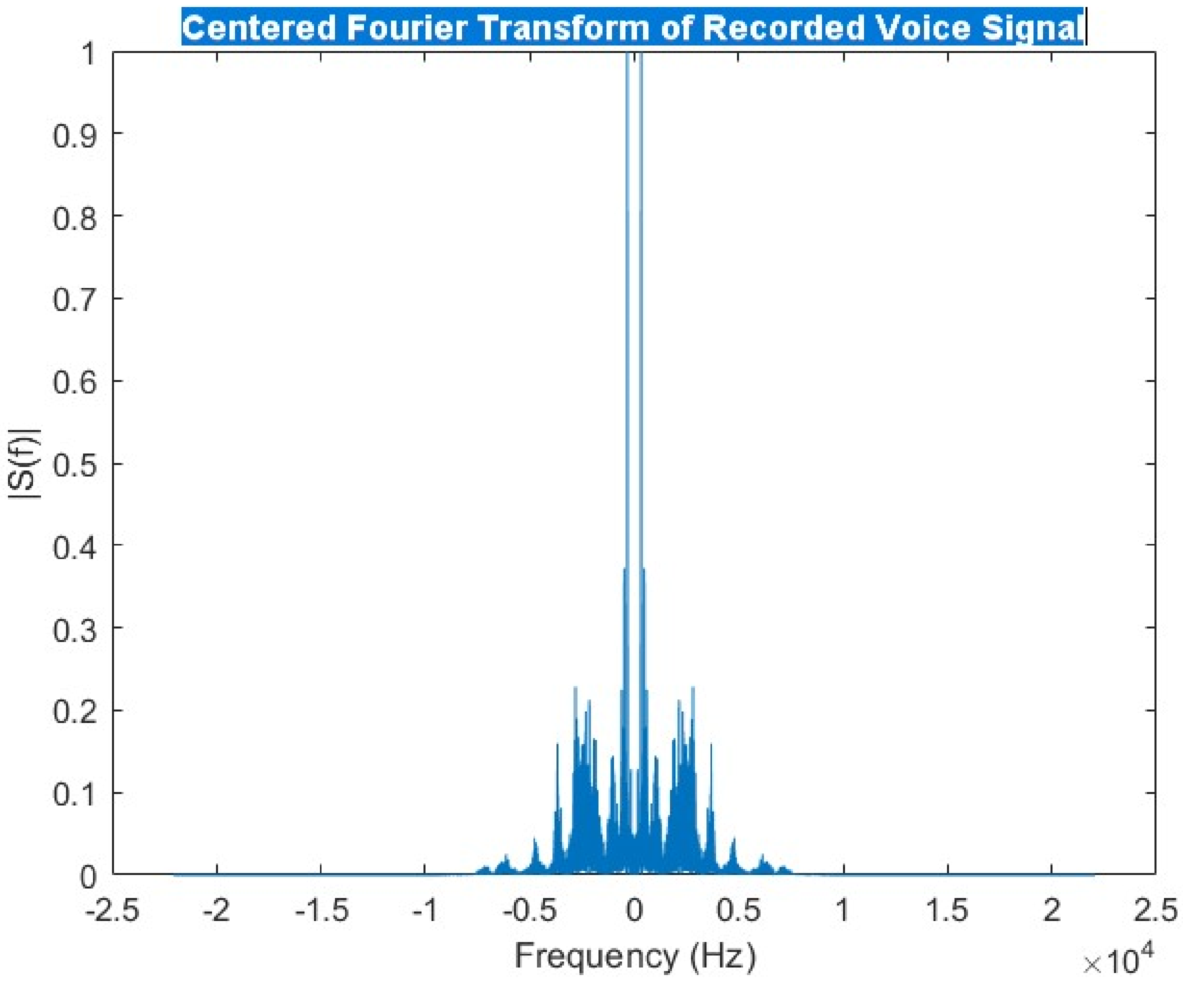
Step 1

- Record a 3-second voice clip using an audio recorder.
- Plot the recorded voice signal's amplitude over time.

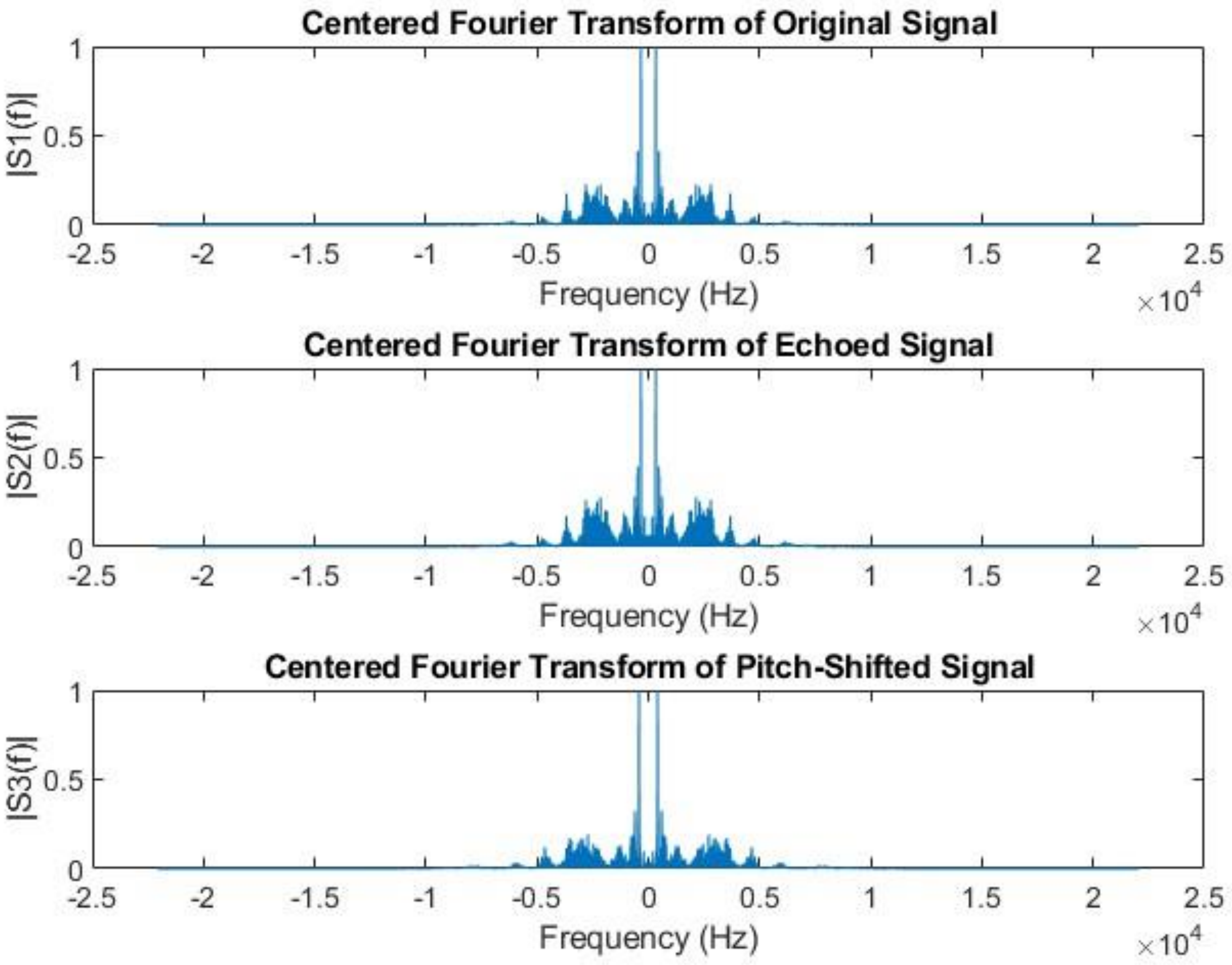


Part 2

- Add delayed and attenuated version to create an echo effect.
- Change the pitch of the voice by shifting its frequencies.
- Extend all signals to the same length with zeros.
- Compute and center the frequency spectrum of each signal.

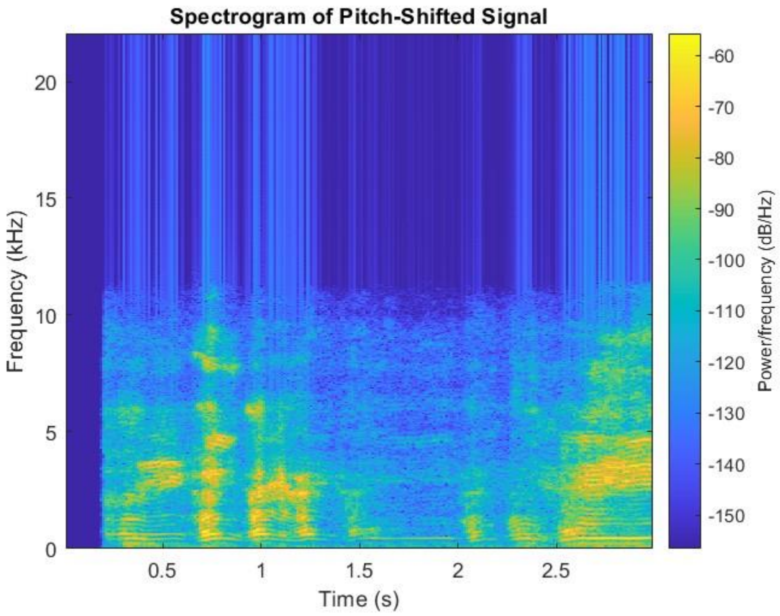


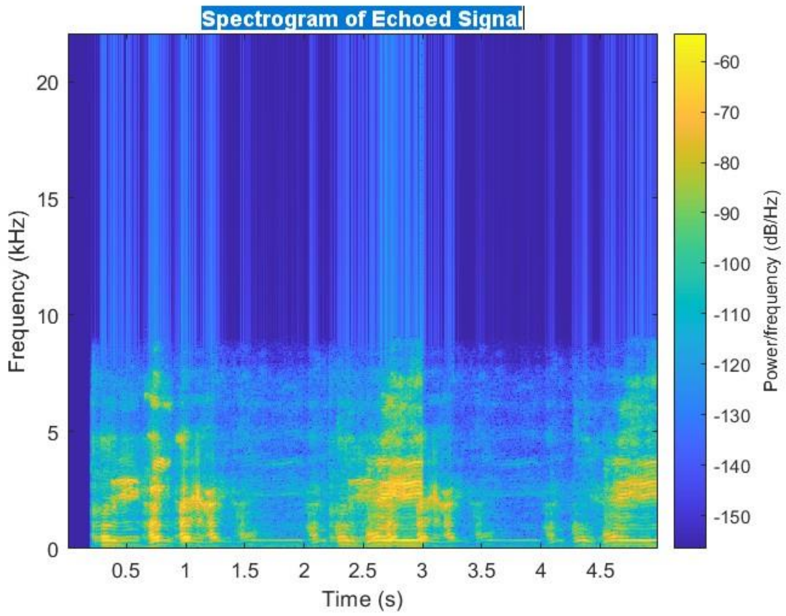
S_2 and S_3

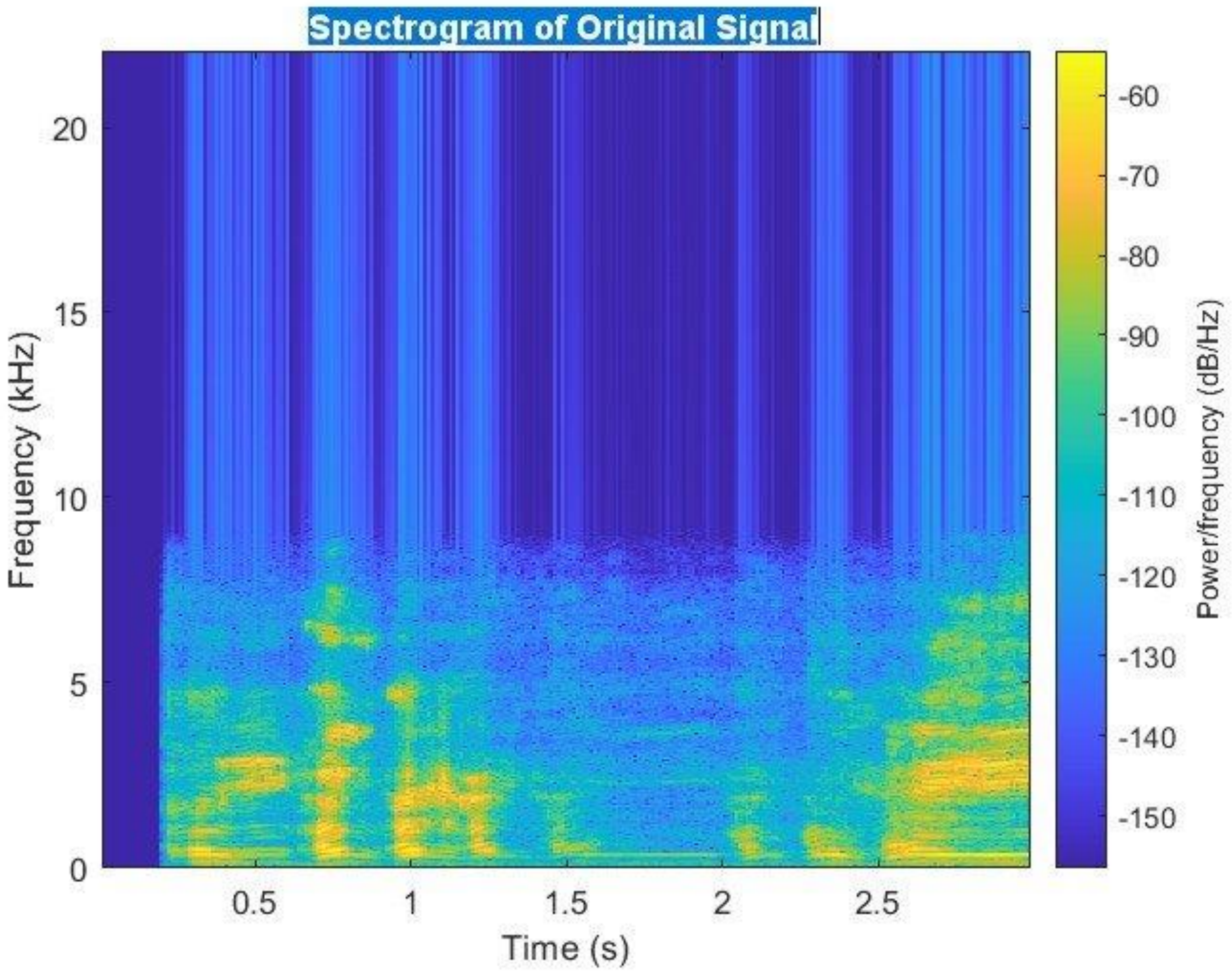

Centered Fourier Transform of Original Signal
|S1(f)|
Frequency (Hz)
×10^4
Centered Fourier Transform of Echoed Signal
|S2(f)|
Frequency (Hz)
×10^4
Centered Fourier Transform of Pitch-Shifted Signal
|S3(f)|
Frequency (Hz)
×10^4

Part 3

- Play original, echoed, and pitch-shifted signals for verification.
- Generate time-frequency plots to visualize signal content (spectogram).







The original signal's spectrogram shows natural speech frequencies over time. The echoed signal's spectrogram reveals repeated patterns due to the delay. The pitch-shifted signal's spectrogram shifts all frequencies uniformly, indicating a pitch change without time alteration