DEEC
DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING
TÉCNICO LISBOA

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

DIGITAL SIGNAL PROCESSING

Report Lab 1

3rd Period 2025/2026

Authors:
João Lopes, ist199971
Marek Pawłowski, ist1119661

Instituto Superior Técnico, University of Lisbon, Portugal

Introduction

This work presents a methodological approach to better understand frequency-domain analysis of continuous time signals by sampling them and using Fourier Transforms. Several subjects were addressed during this process, including but not limited to: aliasing, filtering and signal reconstruction.

R1 - Continuous-Time Chirp Analysis

The goal of Question R1 is to briefly analyze the sampling of a continuous-time chirp signal at a sampling rate equal to the Nyquist Rate of the provided signal. The signal is provided as

$$x_c(t) = \cos\left[2\pi\left(\frac{1}{3}k_1t^3\right)\right], \tag{1.1}$$

being intuitive that the maximum frequency achieved by the signal in the time interval of the requested analysis, $t = [0, 2]$, is given by

$$F_{max} = \left[\frac{d}{dt}\left(\frac{1}{3}k_1t^3\right)\right]\Bigg|_{t=2} = \left[\frac{\Omega(t)}{2\pi}\right]\Bigg|_{t=2} = \left[k_1t^2\right]\big|_{t=2} = k_1(2)^2 = 4k_1 = 4000\text{Hz} = \frac{f_N}{2}, \tag{1.2}$$

where f_N is the Nyquist Rate. Because the sampling frequency used is $F_s = 8000$Hz, it is precisely the Nyquist Rate, meaning there will be no aliasing.

R1.a)

When listening to the signal, one can hear a sound whose pitch increases rapidly over time. This makes sense because the mathematical equation used defines a chirp whose instantaneous frequency increases quadratically with time.

R1.b)

The resulting spectogram of x can be viewed in figure 1.1.

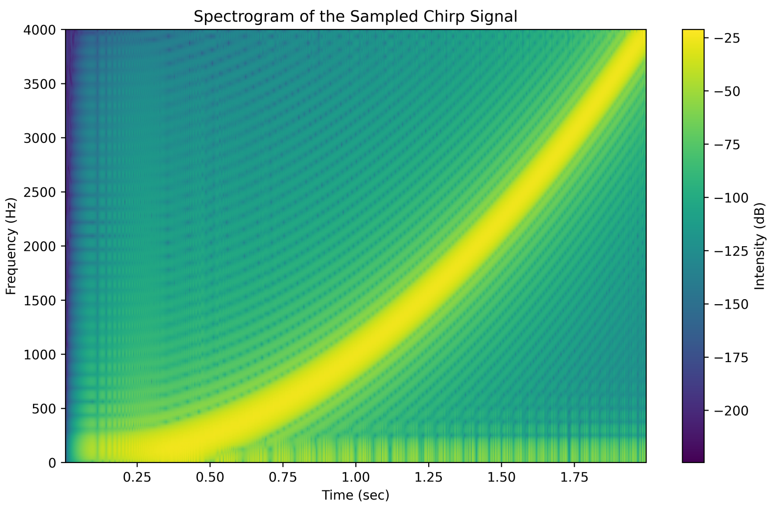

Fig. 1.1. Spectrogram of the sampled continuous-time chirp signal using a Hanning window ($N = 64$). The main yellow curve displays the true instantaneous frequency, while the fainter radiating waves are artifacts of spectral leakage caused by the window's sidelobes.

R1.c)

By looking at the spectrogram, one can visually track a distinct curve that bends upwards toward higher frequencies. This upward visual curve perfectly matches the increasingly high-pitched sweeping sound that is heard when one plays the audio.

R2 - Subsampling and Aliasing

The goal of Question R2 is to understand the effects of aliasing in an interesting way, by sampling in discrete time the previously sampled signal, $x(n)$. By retrieving the signal $y(n) = x(2n)$, we are effectively retrieving the signal values at twice the time step, i.e. half the frequency of the original sampling rate, which leads to a new sampling rate of $F_{s_2} = \frac{8000}{2} = 4000\text{Hz}$. Obviously, $F_{s_2} < f_N$ which leads to aliasing.

R2.a)

The sampling frequency of signal $y(n)$ is 4000Hz. By sampling the original signal at every second index, the time step between consecutive samples is doubled. Consequently, the sampling rate is halved from the original 8000Hz to 4000Hz.

R2.b)

When listening to the new audio file, the pitch increases initially but then starts to decrease, creating a siren-like sweeping effect. In the spectrogram, which is visible in figure 2.1, the frequency curve rises until it hits a frequency of 2000Hz. Once it exceeds this limit, the frequency component folds back down into the lower frequencies. This occurs because we are sampling a signal that has as maximum frequency of 4000Hz (because it has no aliasing), and our sampling frequency is lower than the Nyquist rate of 8000Hz, which leads to aliasing. We may think of it as the frequency components of the replica at -2π and 2π gradually folding back into the baseband as the original frequency increases, creating a mirrored effect in the spectrogram and a corresponding change in the perceived pitch of the sound (the components of the replica at 2π increase in frequency which leads to a decrease in the perceived frequency in the baseband).

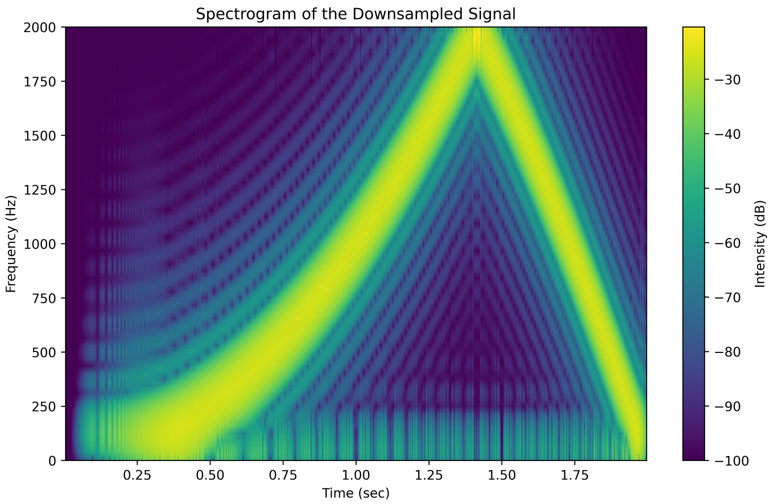

Fig. 2.1. Spectrogram of the downsampled chirp signal ($F_s = 4000$ Hz) using a Hanning window with $N = 32$ for time interval consistency. The main curve rises until it hits the 2000Hz Nyquist limit and then folds back down into the lower frequencies, visually demonstrating the spectral folding caused by aliasing.

R3 - Anti-Aliasing Filtering

The goal of Question R3 is to apply an anti-aliasing order-100 FIR filter to the original sampled signal $x(n)$ with a cutoff frequency of 0.5π. This means that only the frequency components of the original signal $x(n)$ located within the interval $[0, \frac{\pi}{2}]$ in the normalized digital frequency domain are kept. In terms of physical continuous-time frequencies, this corresponds to preserving the components strictly within the $[0, 2000]$Hz interval.

R3.a)

When listening to the filtered and downsampled audio file, the pitch increases initially but then the sound smoothly fades into silence. In the spectrogram visible in figure 3.1, the frequency curve rises until it hits the Nyquist limit of 2000Hz and then abruptly disappears. Unlike the observation in item 2, no frequency components fold back down into the lower frequencies. This occurs because the anti-aliasing filter successfully removes the frequency components above 2000Hz before the signal is sampled. The filter has a cut-off frequency of 0.5 in the discrete-time angular frequency scale, which corresponds exactly to 2000Hz. Consequently, there is no high-frequency content left to cause aliasing, completely preventing the mirrored siren-like effect that was present in the previous signal.

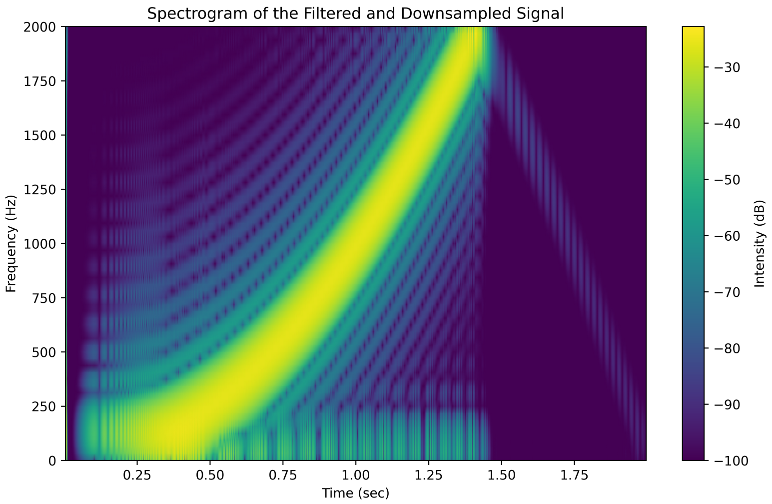

Fig. 3.1. Spectrogram of the filtered and downsampled chirp signal. The application of the order-100 FIR anti-aliasing filter with a cutoff frequency of $\pi/2$ successfully removes frequency components above 2000Hz before downsampling. This prevents the spectral folding (aliasing) that was observed in the previous un-filtered downsampled signal.

Although the laboratory statement states “To avoid distortion, anti-aliasing filters can be used. These filters remove high-frequency components before sampling, thereby preventing spectral folding and information loss,” I believe it is worth keeping in mind that the application of such a filter in this case will result in information loss. However, this loss is exclusive to the high-frequency components that would have caused aliasing, which is preferable to the entire signal being affected by aliasing scrambling.

R4 - Frequency Domain Filtering

The goal of Question R4 is to use the Fourier transform to remove noise from a signal consisting of two sinusoids corrupted by additive Gaussian white noise, and study how filtering in the frequency domain affects the time-domain signal. A comparison will also be made between filtering the signal in the frequency domain directly and filtering in the time domain.

Assumption

No information regarding the time step between each sample within **sum_of_sines.npy** was provided. Since the file has 1000 samples, a default value of 1000Hz for the sampling frequency was assumed.

R4.a)

By simply observing the time-domain plot available in figure 4.1, it is extremely difficult to precisely identify the two frequencies.

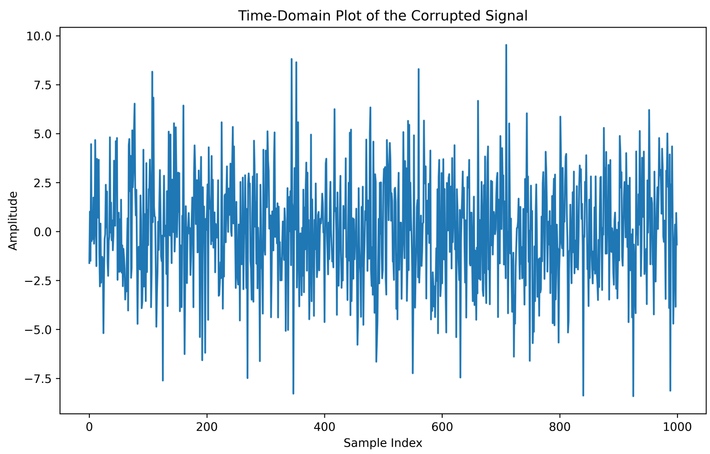

Fig. 4.1. Time-domain plot of the corrupted signal loaded from **sum_of_sines.npy**.

R4.b)

Looking at the magnitude spectrum visible in figure 4.2, two distinct and sharp peaks stand out above a noisy baseline. Based on the extracted data, the two dominant normalized digital frequencies are exactly 0.03 and 0.12 cycles/sample, which correspond to actual frequencies of 30Hz and 120Hz, respectively, assuming a sampling frequency of 1000Hz.

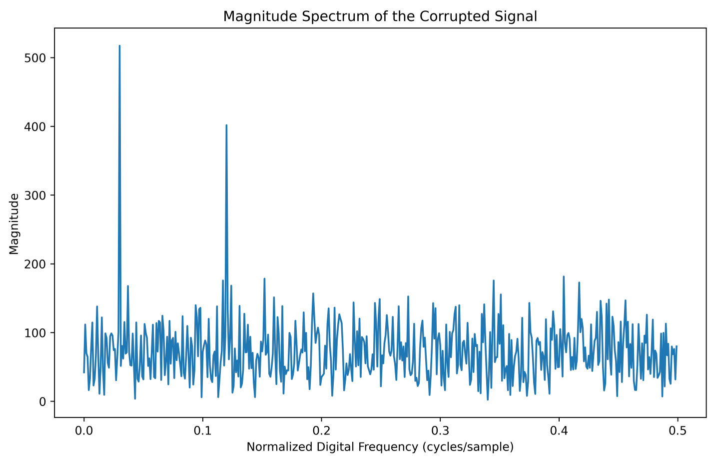

Fig. 4.2. Magnitude spectrum of the corrupted signal computed via the Fast Fourier Transform (FFT).

R4.c)

By comparing the two spectra visible in figure 4.3, it is visually clear that the noise floor has been completely flattened to zero. The only remaining components in the filtered spectrum are the two sharp peaks corresponding to the original sine waves. The frequency-domain thresholding successfully isolated the exact frequencies of interest from the broadband noise.

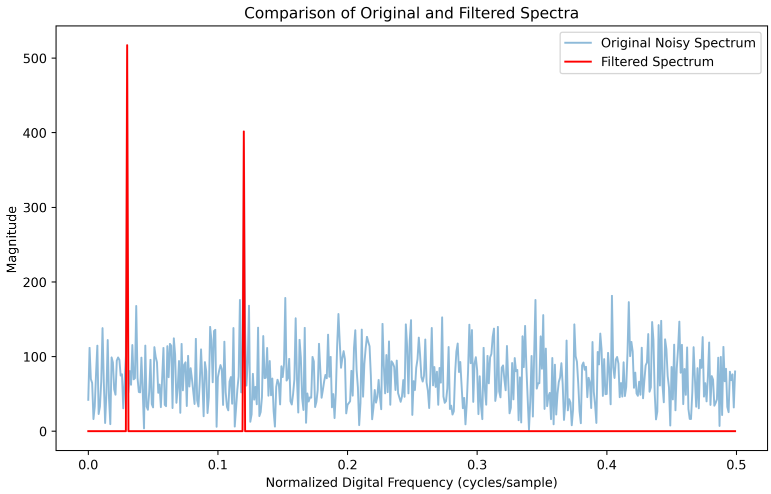

Fig. 4.3. Comparison of the original noisy spectrum and the threshold-filtered spectrum with a threshold set to 181.504, which is 35.1% of the maximum magnitude of the frequencies. By programmatically determining a magnitude threshold to isolate the top two peaks, the broadband Gaussian white noise is completely zeroed out in the frequency domain, leaving only the pure sinusoidal components.

Threshold Calculation For Frequency Domain Filtering

Given that we have the apriori knowledge that there are exactly two dominant frequencies, we can iteratively increase the threshold until only two peaks remain in the positive magnitude spectrum. This leads to the optimal threshold that effectively eliminates the noise while preserving the two sine wave components.

R4.d)

The resulting plot in figure 4.4 shows a significant reduction in the signal's variance. While the original signal was heavily obscured by random fluctuations, the filtered signal clearly displays a clean, periodic waveform. This waveform is the superposition of the two sinusoids identified in the frequency domain. The filtering process successfully removed the additive Gaussian white noise while preserving the underlying structure of the two sine waves.

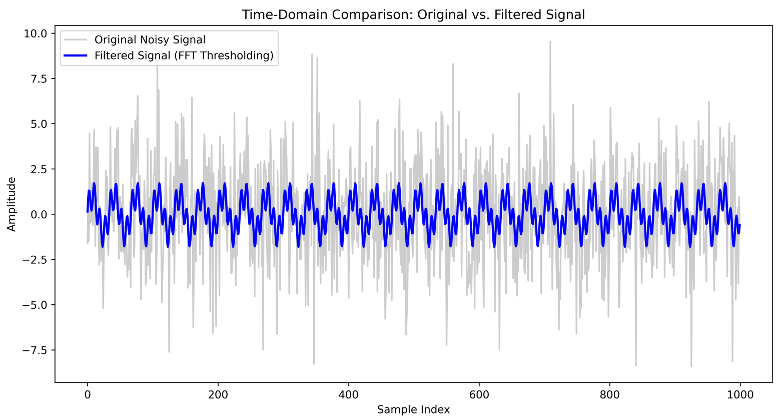

Fig. 4.4. Comparison of the original noisy spectrum and the threshold-filtered spectrum.

R4.e)

It is observed in figure 4.5 that the frequency-domain thresholding provides a much cleaner reconstruction of the original sinusoids. Because the noise is white and spread across all frequencies, zeroing out the frequency bins outside the signal peaks removes the noise entirely without affecting the sinusoids. In contrast, the moving-average filter acts as a low-pass filter in the time domain. While it succeeds in reducing the high-frequency noise, it also attenuates the amplitude of the signal components and introduces a phase lag. Additionally, some noise remains present in the moving-average result because the filter's stopband is not perfectly sharp. Therefore, for signals consisting of pure sinusoids corrupted by noise, filtering directly in the frequency domain is shown to be significantly more effective than a simple running average.

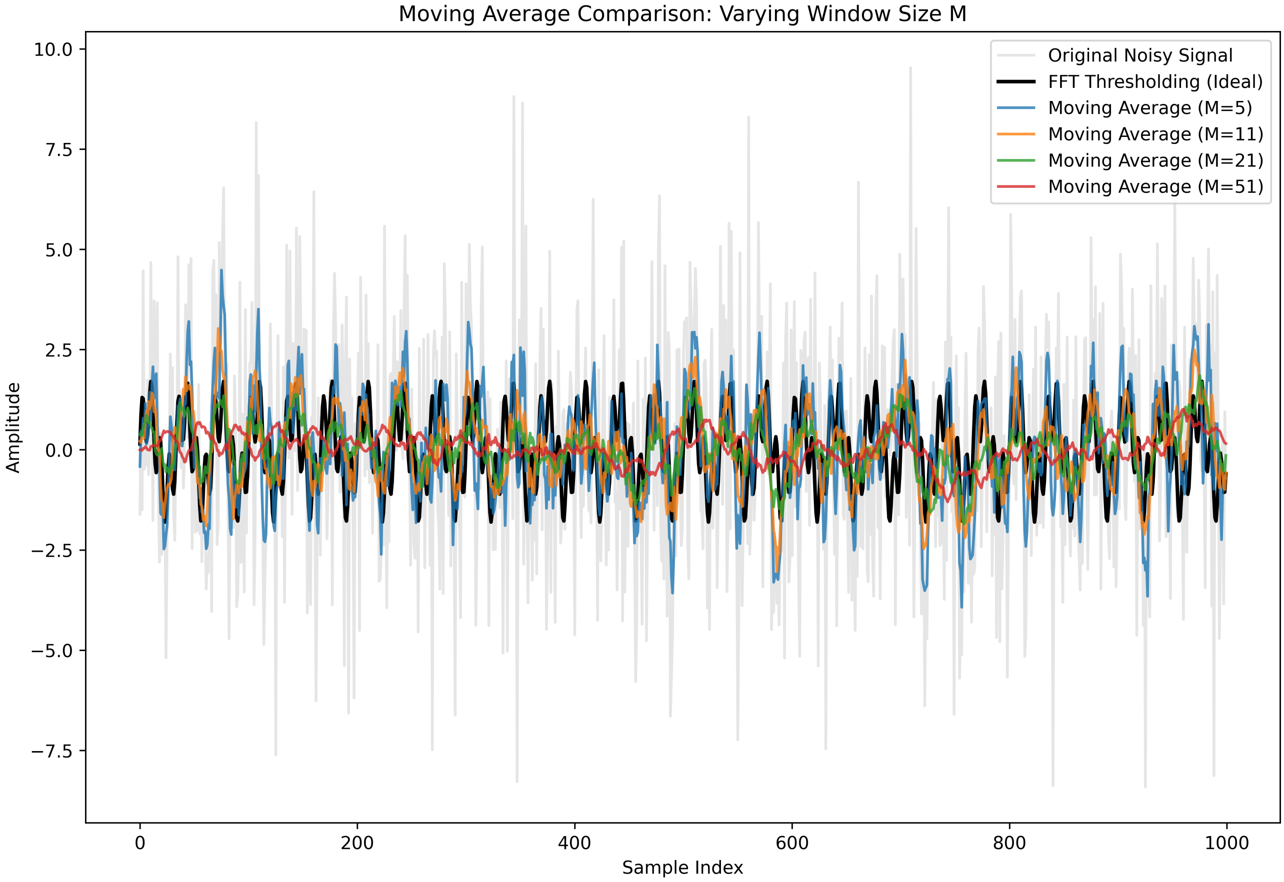

Fig. 4.5. Comparison of filtering techniques in the time domain. The frequency-domain FFT thresholding perfectly isolates the sinusoids. In contrast, the various time-domain moving-average filters reduce the high-frequency noise but simultaneously attenuate the signal's amplitude and fail to completely eliminate the noisy fluctuations.