Processing of sound signals mini project.

Name: Christian Petersen

Student Nr: 20206244

For the mini project, I decided to make a real-time spectrum analyzer done in Python. Using libraries such as NumPy, Pydub, and Pygames, reading of the data and further processing of the data was possible. The use of Pygames was only to create a screen that could update the current Fast Fourier Transform information quickly enough, so it was synced up with the music being played. It also allowed for buttons and sliders. A diagram of a computer

Description automatically generated

Figure 1 Flow Chart of Spectrum Analyzer.

The flow chart in Figure 1 shows how the overall process of the script goes. All the necessary data from the song is extracted, such as the Sample rate, which defines how detailed the overall data is, or more technically said, it is the number of samples per second that makes up a waveform. In the case of the program, the sample rate of the song was 48k Hz, and with a defined CHUNK size of 2048, the Fast Fourier Transforms (FFT) would contain [ seconds of data. Having a smaller CHUNK size provides better time resolution, but poorer frequency resolution, and larger CHUNK sizes provide worse time resolution but better frequency resolution.

The FFT converts data from the time domain to the frequency domain. One of the main advantages of using the FFT is its computational efficiency. The FFT algorithm reduces the number of calculations needed compared to the standard Discrete Fourier Transform (DFT). While the DFT has a computational complexity of , the FFT reduces this to . This efficiency makes the FFT particularly suitable for real-time applications, where large amounts of data need to be processed quickly. (Christensen, 2019)

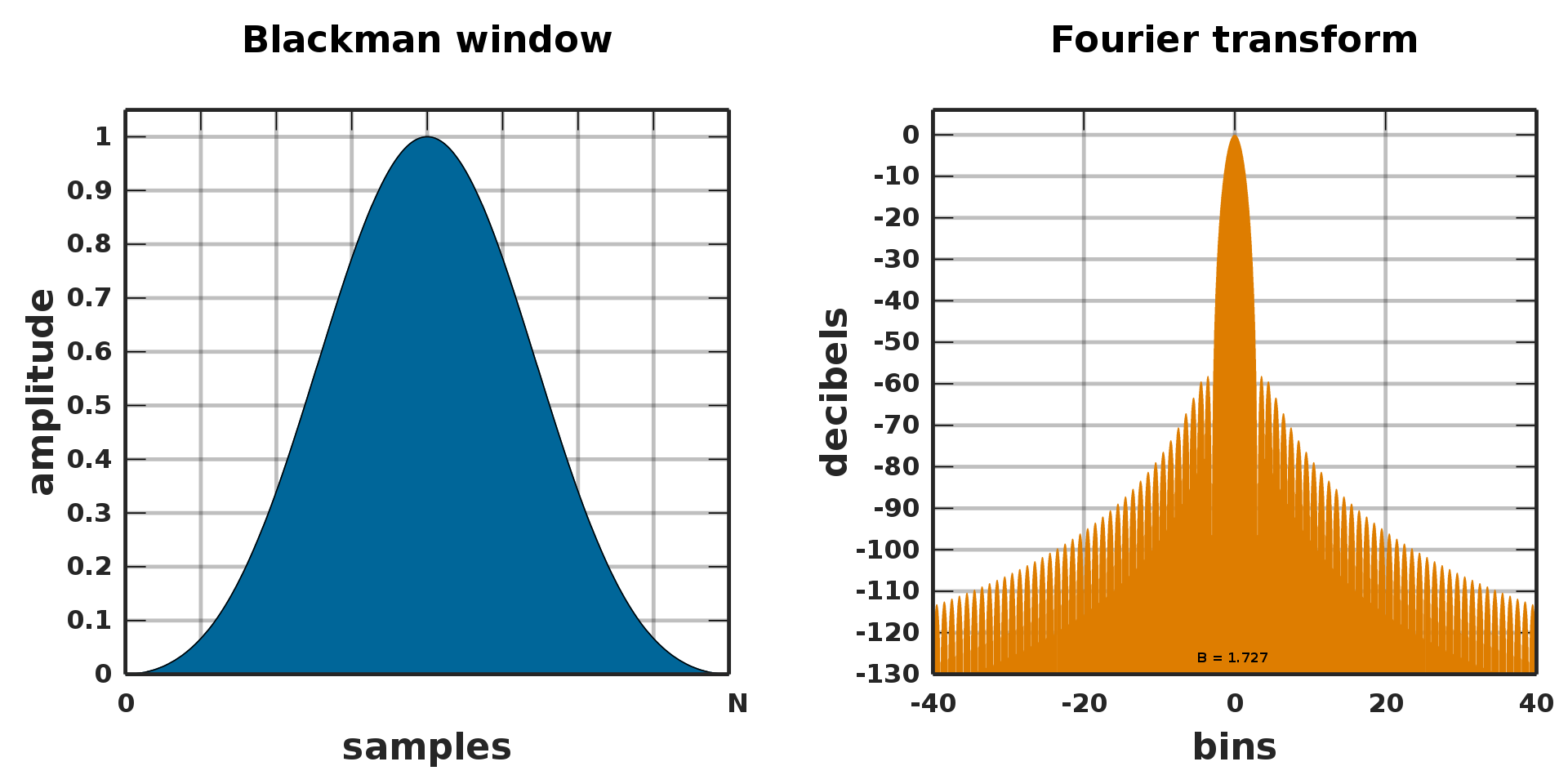
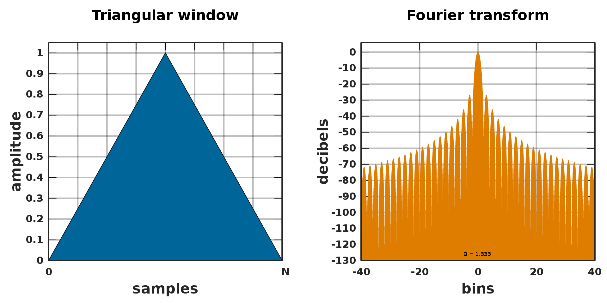
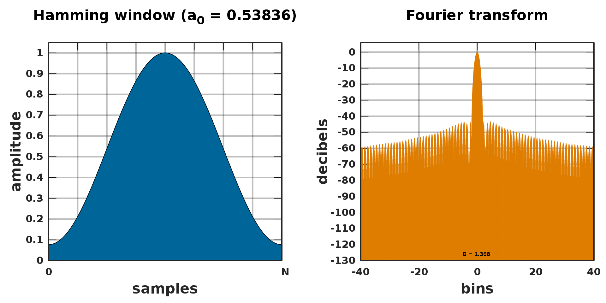
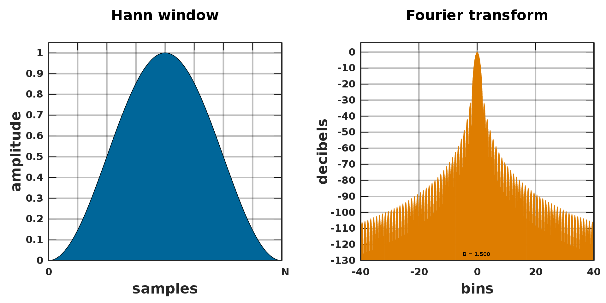
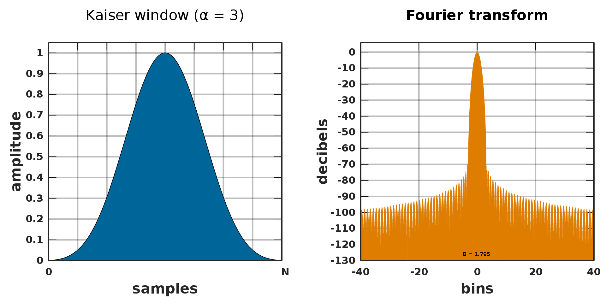


Figure 2 Different windows and their effects on frequency bins outside the focus.

Differences can be determined by the window used when converting the frequencies from the time domain to the frequency domain, as the window can help against spectral leakage. Window functions reduce spectral leakage by tapering the signal at the edges of the CHUNK, thereby minimizing the impact of data outside the selected CHUNK on the FFT results. This helps in limiting how frequencies outside the defined bins affect each other, leading to more accurate frequency analysis. In other words, window functions define how much of the signal's information outside the current CHUNK influences the frequency content within the CHUNK.

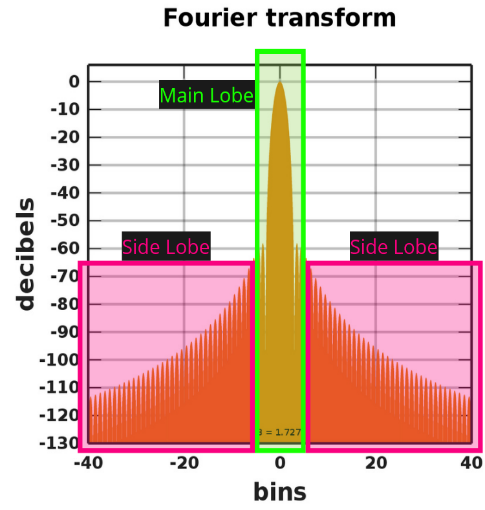


Figure 3 Blackman Window Example

The windows have various trade-offs between main lobe width and side lobe levels. The main lobe refers to the highlighted green section in figure 3 and the side lobes are the areas highlighted in pink. The width of the main lobe determines the frequency resolution, so the narrower the main lobe, the better the window can distinguish between closely spaced frequencies. The side lobes are the unwanted leakage of energy, so the lower the side lobes are, the more accurate the analysis can be. (Understanding FFTs and Windowing, 2024)

For example, the Hann window provides a good balance between frequency resolution and spectral leakage, while the Blackman window offers lower side lobes at the expense of a wider main lobe. The Kaiser window allows adjustable side lobe levels through the parameter β.

In the case of a real time spectrum analyser that visualises the spectral details of a song in real time, a window is necessary to prevent as much spectral leakage as possible, ensuring a clearer and more precise presentation of the spectrum. Real-time processing also requires a lot of computation to minimize latency, making the choice of window and the implementation of the FFT important for this project.

Conclusively, this project effectively demonstrated how to extract and visualize the frequency content of audio signals in real-time. By implementing Fast Fourier Transform (FFT) combined with windowing techniques, the analyzer can accurately convert time-domain signals into their frequency-domain representations. This conversion allowed for real-time visualization of spectral details, synced with the playback of the music.