**Introduction**

In this laboratory exercise, we explored the characteristics and practical applications of various types of filters, particularly focusing on their frequency responses and their effects on different input signals. The main aim of the tasks was to understand how filtering can isolate, suppress, or shape signals depending on frequency content. Through a series of experiments and analysis, we applied and evaluated the effects of **band-pass**, **low-pass**, and **band-stop** filters on both deterministic and noisy signals. The behavior of the filters was analyzed using frequency response plots and power spectral density estimates, with special focus on understanding the bandwidth, cutoff frequency, and imperfections in filter performance.

**🔹 Task A: Determine the Characteristics of a Band-Pass Filter**

In this task, we evaluated the frequency response of a **band-pass filter** by analyzing the ratio of output amplitude to input amplitude (**Aout/Ain**) across a range of input frequencies. The data was plotted to visually observe the filter's behavior and attenuation outside its passband.

A graph with blue dots and red dots

AI-generated content may be incorrect.

From the measured data, we can conclude the following:

* The band-pass filter begins to pass the signal from around **20 Hz**, reaching maximum transmission (Aout/Ain ≈ 1) at around **30–34 Hz**.
* The output starts decreasing gradually after **50 Hz**, and the filter still allows a non-negligible amount of signal to pass even up to **100 Hz**, though attenuated.
* At frequencies beyond **100 Hz**, the Aout/Ain ratio becomes close to **zero**, indicating that the filter strongly attenuates those components.

**🔸 Filter Settings:**

* **Low cutoff frequency**: ~20 Hz
* **High cutoff frequency**: ~50 Hz
* **Passband**: ~20 Hz to 50 Hz
* **Center frequency**: ~30–34 Hz (where Aout/Ain ≈ 1)

**🔸 Imperfections in Filter Response**

Although the ideal band-pass filter should allow frequencies only between 20 Hz and 50 Hz and sharply attenuate frequencies outside this range, our plot shows a **gradual roll-off**:

* The output doesn’t drop immediately to zero after 50 Hz.
* Instead, significant energy still passes through even at **70–90 Hz**, and only **beyond 100 Hz** does it approach zero.

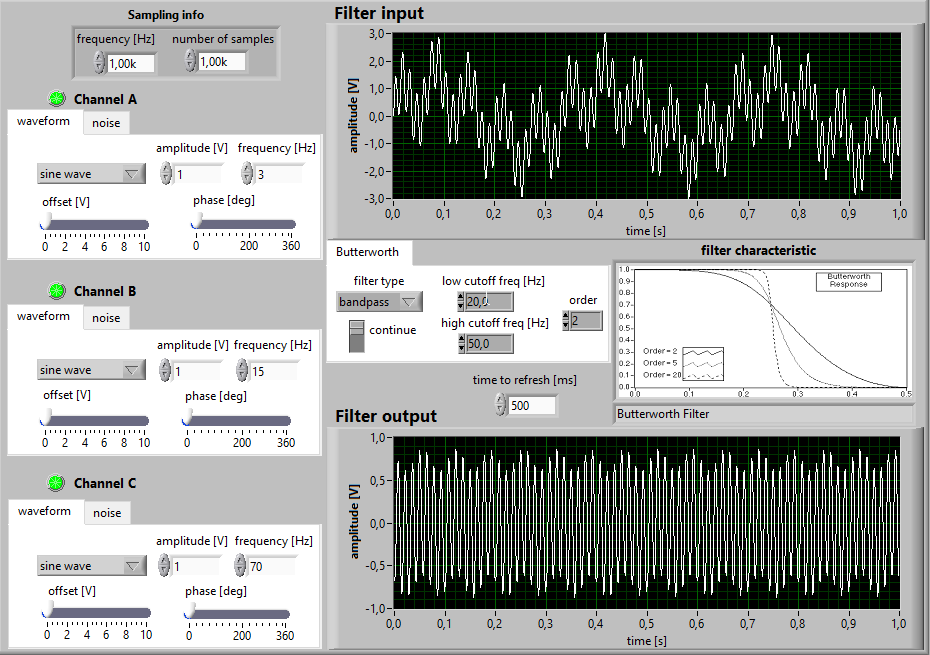
This **"imperfection"** is common in real-world filters and is caused by the filter's transition band and design limitations. Ideally, the cutoff would be sharp (vertical drop), but practical filters always show some slope in their attenuation.

**✅ Measure of Imperfection:**

* The **attenuation beyond the high-pass cutoff** (50 Hz) is not ideal, as it takes ~50 Hz (up to 100 Hz) for the filter to suppress the signal to near zero.
* This can affect applications requiring very sharp filtering.

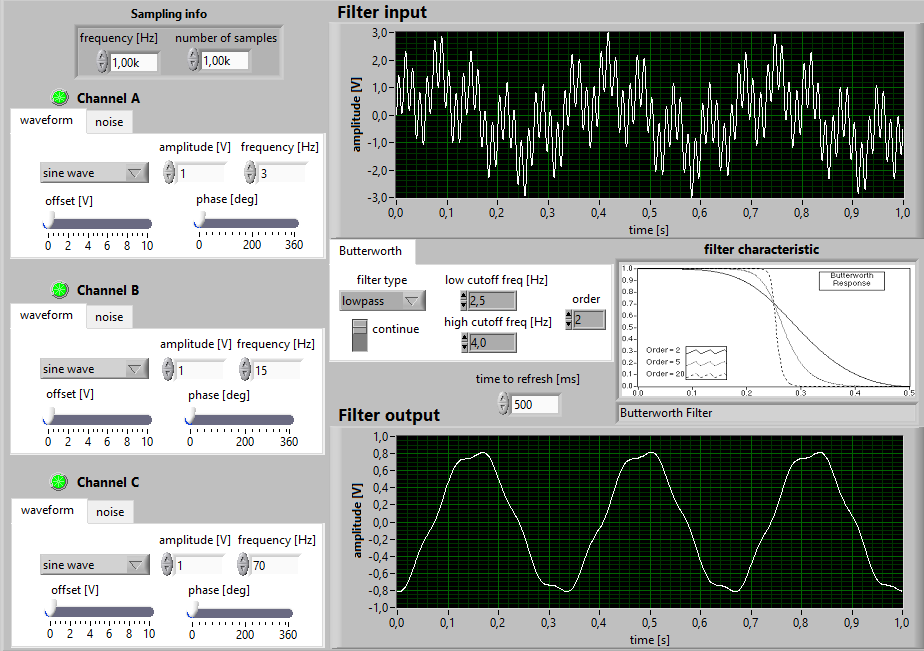
**B) Analyse the influence of filtering (its type and settings) on a polyharmonic signal composed of 3 sine waves**

In this part, I used three sine waves combined together to create a polyharmonic signal.

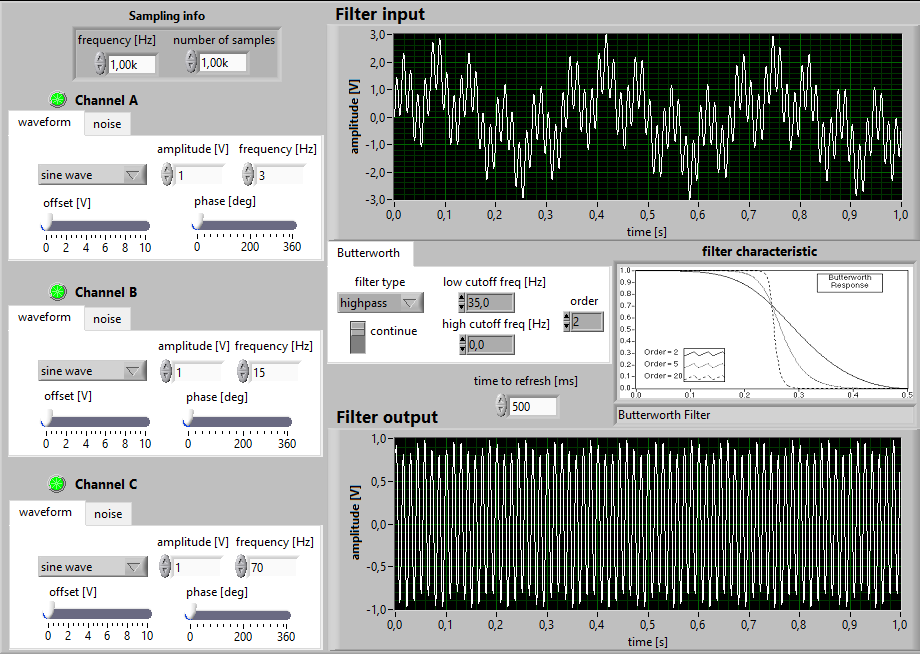
* these are my setting for this

First, I wanted to keep **only the first (lowest) frequency**, so I used a **low-pass filter**. This allowed the low-frequency sine wave to pass while removing the middle and high components.

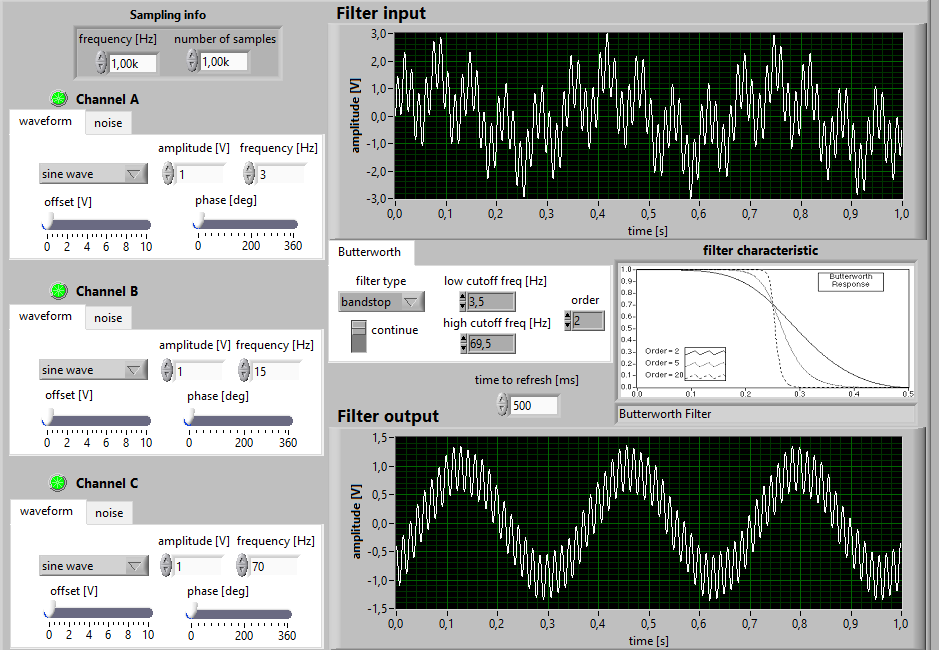
✅ **Result**: Only the lowest sine wave was visible in the output.



After that, I wanted to keep **just the highest frequency**, so I used a **high-pass filter**.

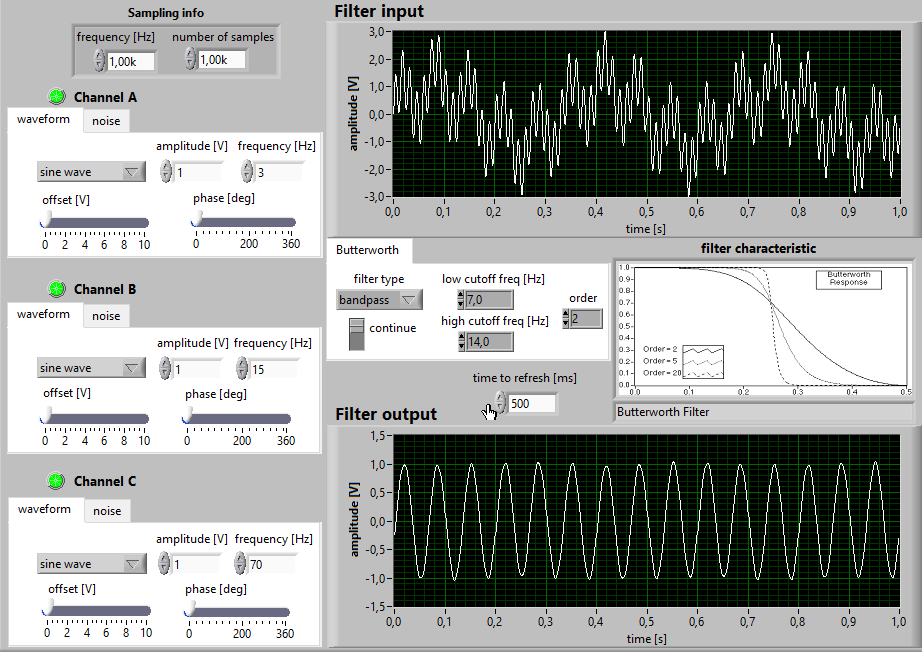
✅ **Result**: Only the high-frequency sine component remained.

Then, I wanted to **keep both the lowest and the highest frequencies**, and **remove the middle** one. For this, I used a **band-stop filter**.

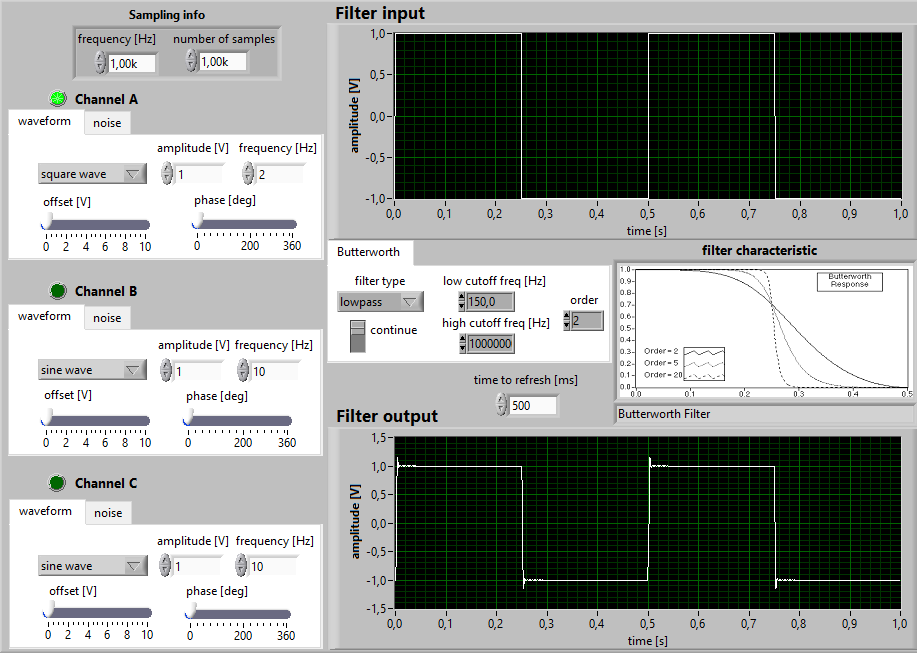
✅ **Result**: The middle frequency was removed and only the lowest and highest frequencies were present.

Finally, I wanted to **keep only the middle frequency**, so I used a **band-pass filter**.

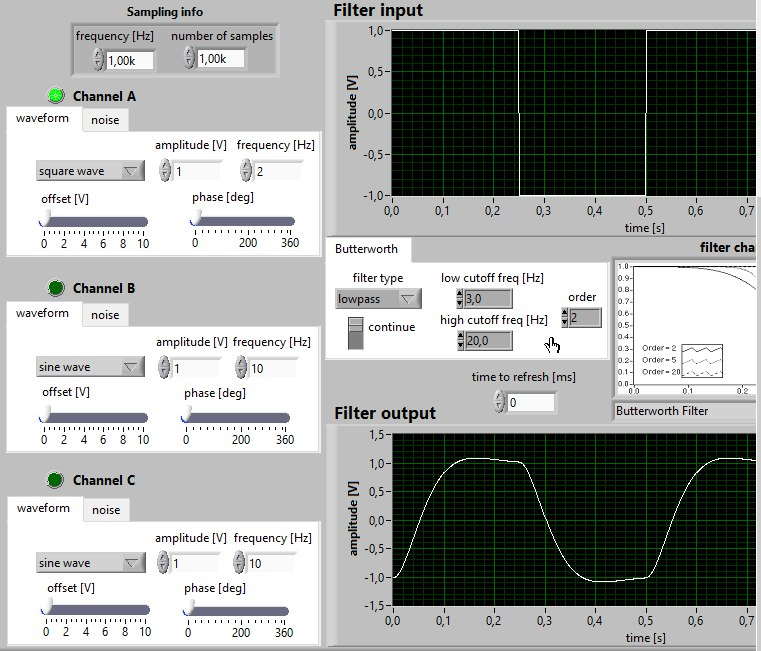
✅ **Result**: The low and high frequencies were filtered out, and the middle sine wave was preserved



**C) Analyse the influence of the setting of a low-pass filter on square wave filtering**

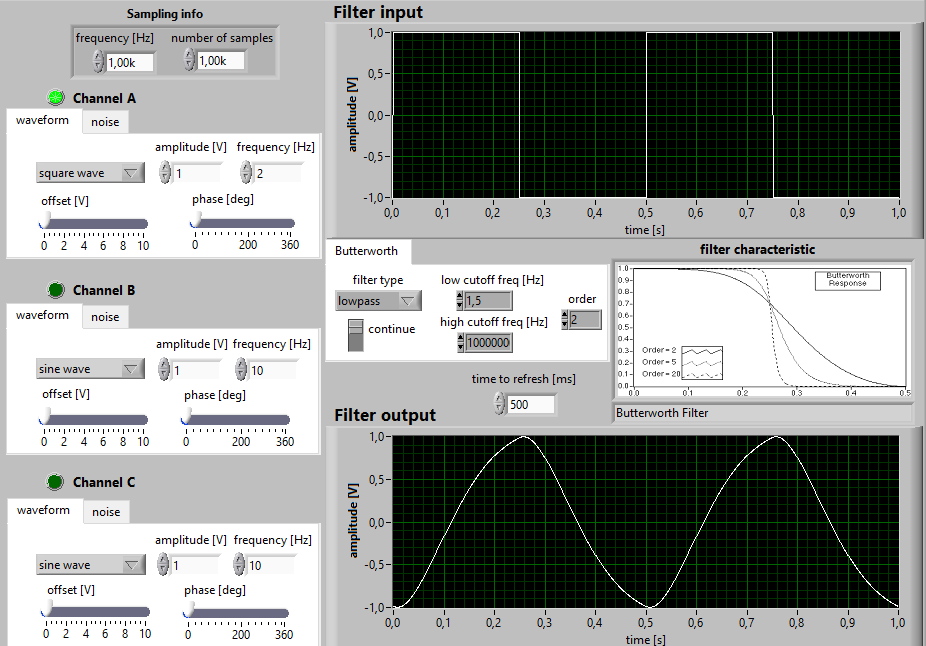
In this part, I applied a **low-pass filter** to a **square wave** signal and observed the changes:

* When I reduced the low-pass cutoff frequency, the square wave started to look more **rounded**, closer to a **sine wave**.



When I lowered the cutoff even more (close to 0 Hz), the output almost became a **pure sine wave**.

✅ **Conclusion**: The square wave contains many high-frequency harmonics. By applying a low-pass filter, those high frequencies are removed, and we can clearly see that the original square wave is actually built from multiple sine waves



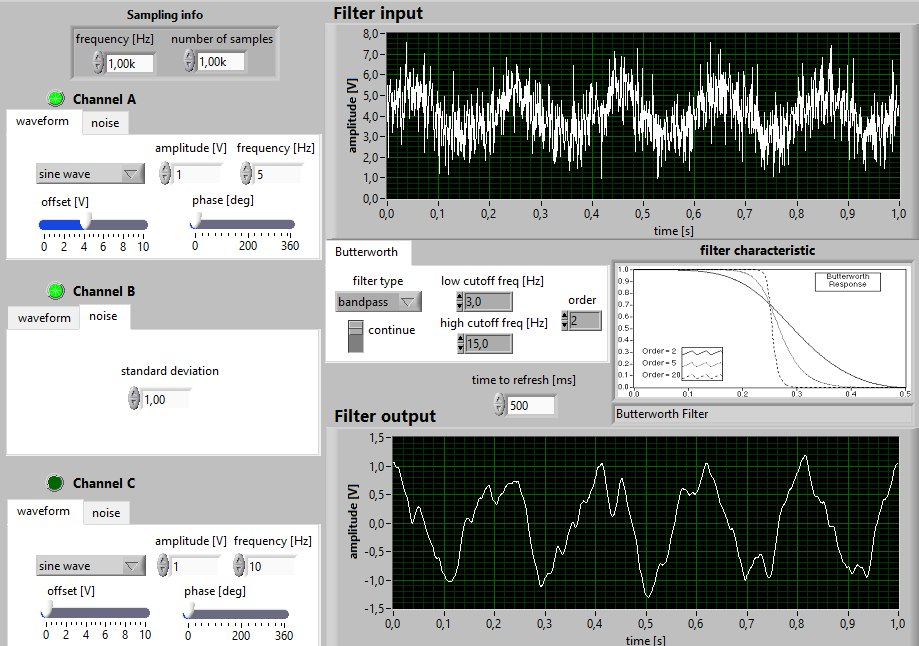
**D) Apply the band-pass filter to noise and mean removal**

In this task, I applied a **band-pass filter** to a **noisy signal** (with mean offset) to remove both noise and the mean value:

A screenshot of a computer

AI-generated content may be incorrect.

First, I reduced the **high cutoff frequency** to **15 Hz**, and this gave a much better result. The signal started looking like a **clean sine wave**.

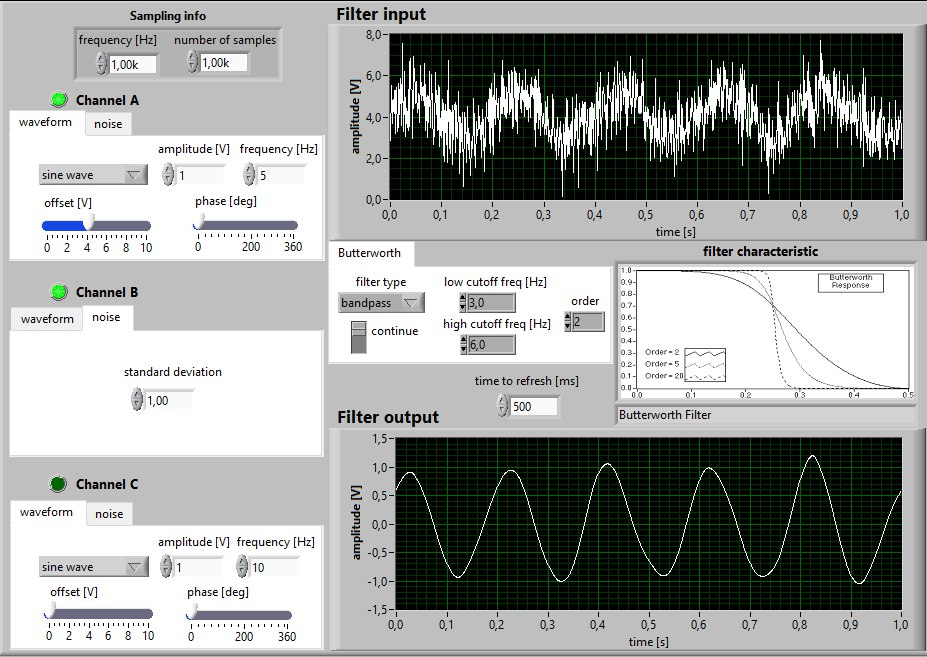


Then, I changed the **low cutoff frequency** to **4 Hz**, but it reduced the amplitude too much.

A screenshot of a computer

AI-generated content may be incorrect.

After adjusting more, I found that a **low cutoff frequency of 3 Hz** gave the best result.



✅ **Final Result**: The signal was smoothed, the **amplitude stayed close to 1**, and the **mean value was effectively removed**.

MOHAMMED ALMASHHOR SEMIH GUNDUZ