Universitatea POLITEHNICA din București
Facultatea de Inginerie Mecanică și Mecatronică

Sisteme De Achiziţie Si Interfeţe-Instrumentaţie Virtuală

Student: Godari Jani
Grupa: 534B

Signal Processing Program

Signal Processing is a python program that analyzes signals using a well-known scientific module called Scipy. The graphs are created using also a well-known module called Matplotlib and for the graphical user interface PySimpleGUI was used.

Main features of the program:

- IIR Filter [Lowpass & Highpass]
- Image Processing with Lowpass & Highpass
- Correlation

IIR Filter [Lowpass & HighPass]

What is an IIR Filter?

- IIR (Infinite Impulse Response) is a filter that has an infinite impulse response, meaning it dose not become exactly 0 past a certain point but continues indefinitely. This is the opposite of the FIR (Finite Impulse Response) that becomes exactly 0 .

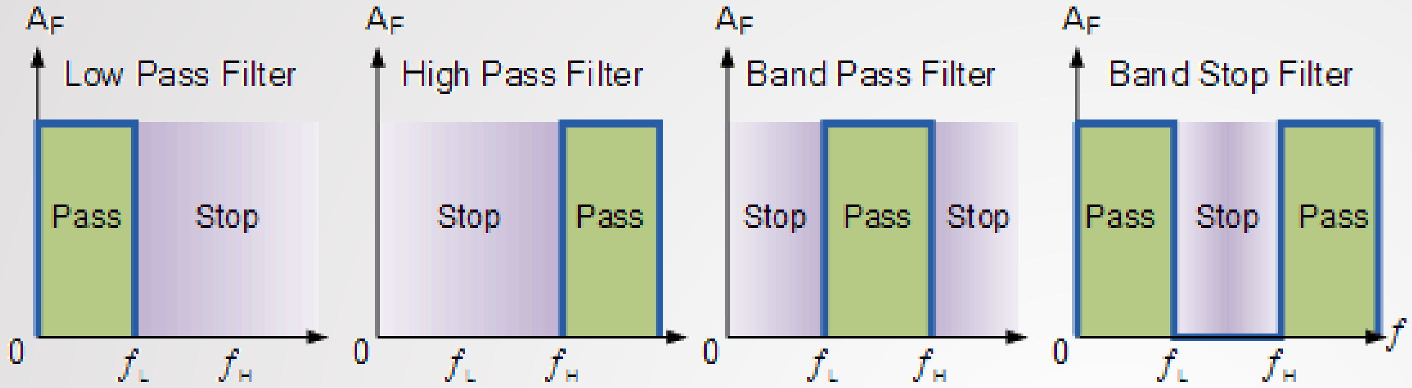
What is Lowpass Filter?

- A Lowpass filter is a filter that is designed to reject all unwanted high frequencies of a signal and only accept/filter the given frequency by the user.
- A Lowpass filter only allows low frequency signal from 0Hz to its cut-off frequency.

What is a Highpass Filter:

- A Highpass filter is a filter that is designed to allow only high frequencies signals from its cut-off frequency.

Ideal filter response curves



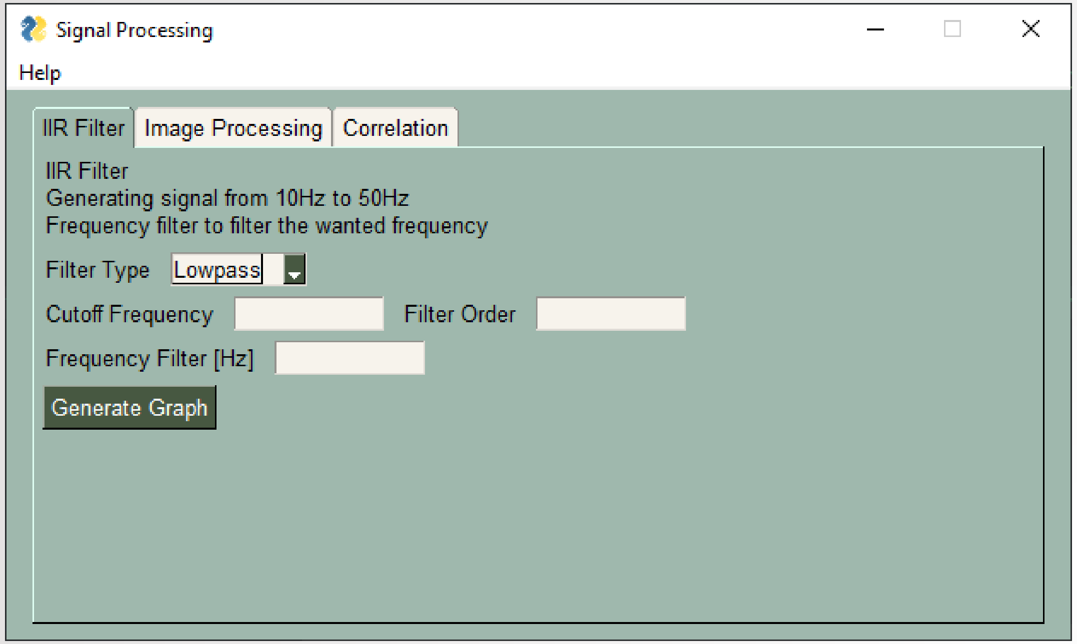
On the picture above the types of filter can be seen how they work

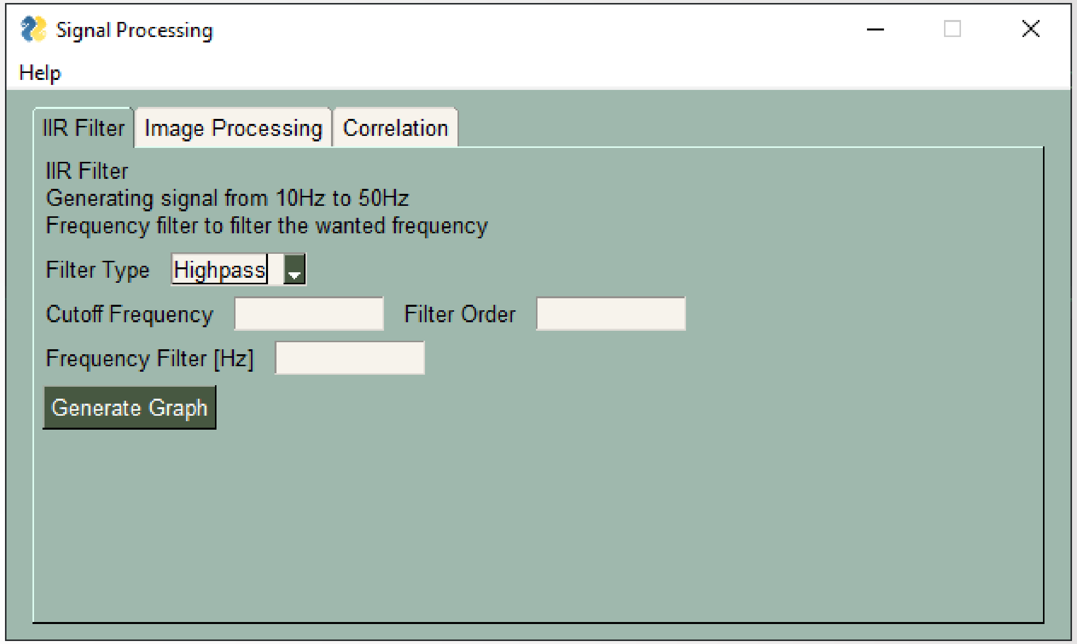
- On a Lowpass filter, the signal passes till it arrives to its cutoff frequency and then stops
- On a Highpass filter, the signal stops until it cutoff frequency and then the signal passes

We also have two other types of filters: Bandpass & Bandstop Filter

- A Bandpass filter needs 2 parameters, Low cut-off & High cut-off frequency. The Bandpass filter allows signals
  that are within a frequency band within 2 points that is setup by the user to pass and block the higher and lower
  frequencies on both sides of the band
- A Bandstop filter is the opposite. It also requires 2 parameters but this time the filter allows most of the frequencies to pass while blocking a some signals into a specific range to be stoped

IIR Filter on the program





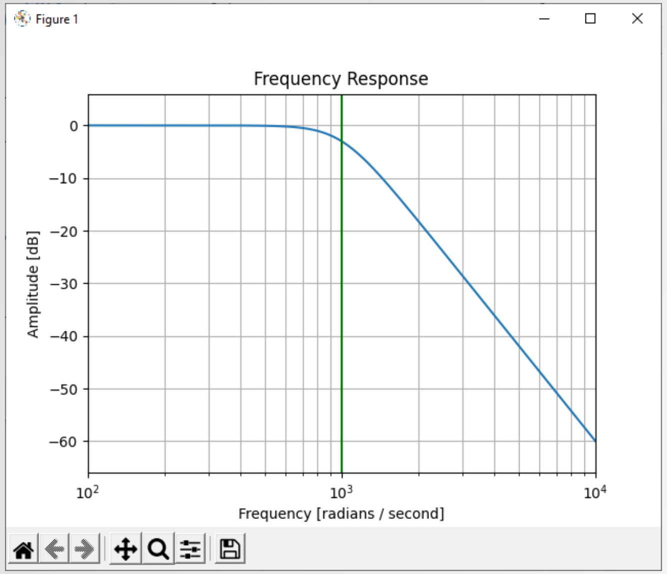
Program supports 2 types of filter: Lowpass & Highpass

Cutoff Frequency: custom based on the user choices. A cutoff frequency is a boundary in which energy flowing through the system begins to reduce and not pass through

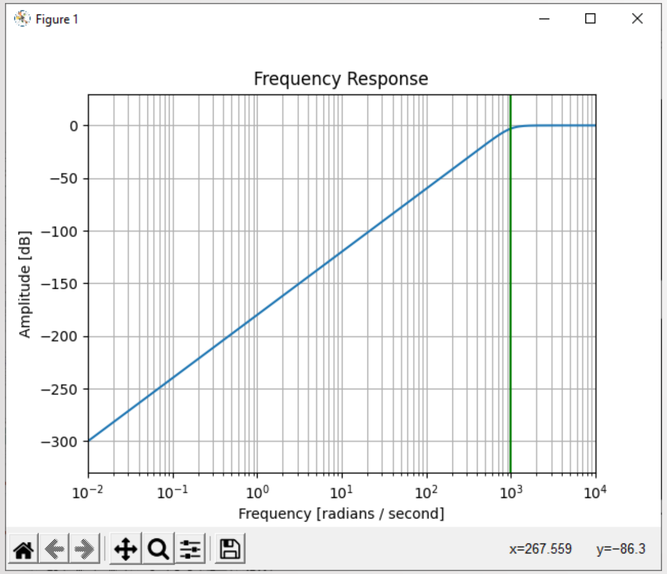
Filter Order: Maximum number of delay elements.

Frequency Filter: The frequency in Hz that we want to filter from the original signal that goes from 10Hz to 50Hz so a correct way to use it is to input values that are inside those frequencies

Cutoff frequency

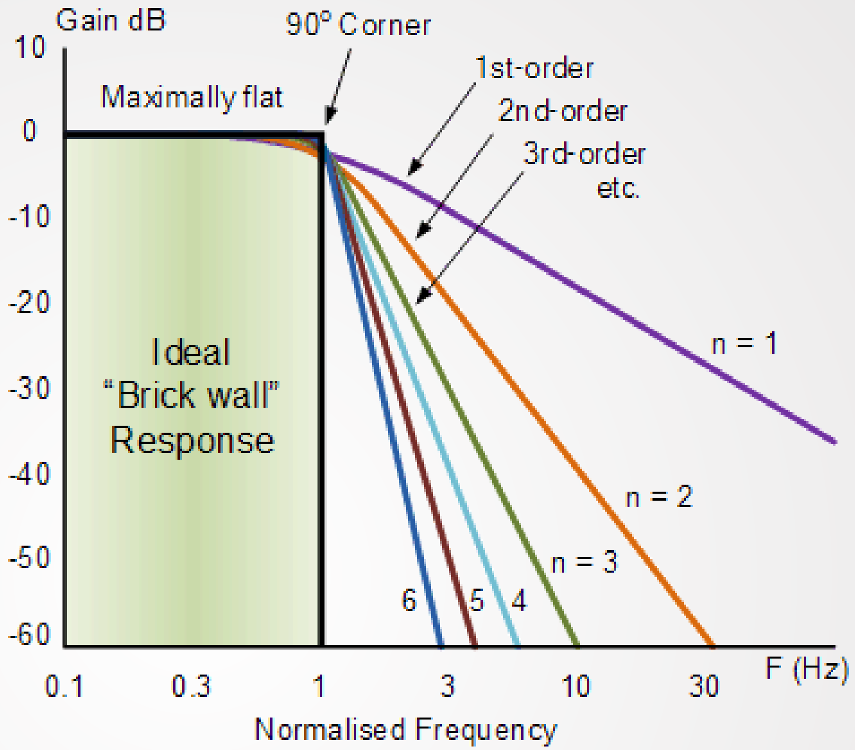


Lowpass cut-off frequency = 1000



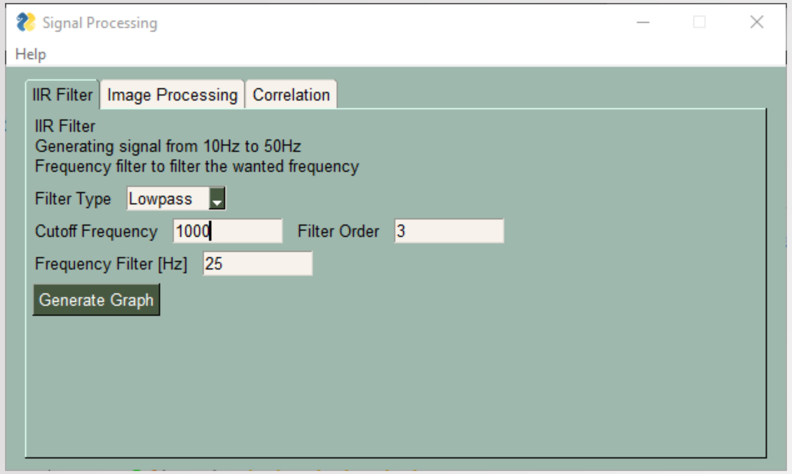
Highpass cut-off frequency = 1000

Filter Order

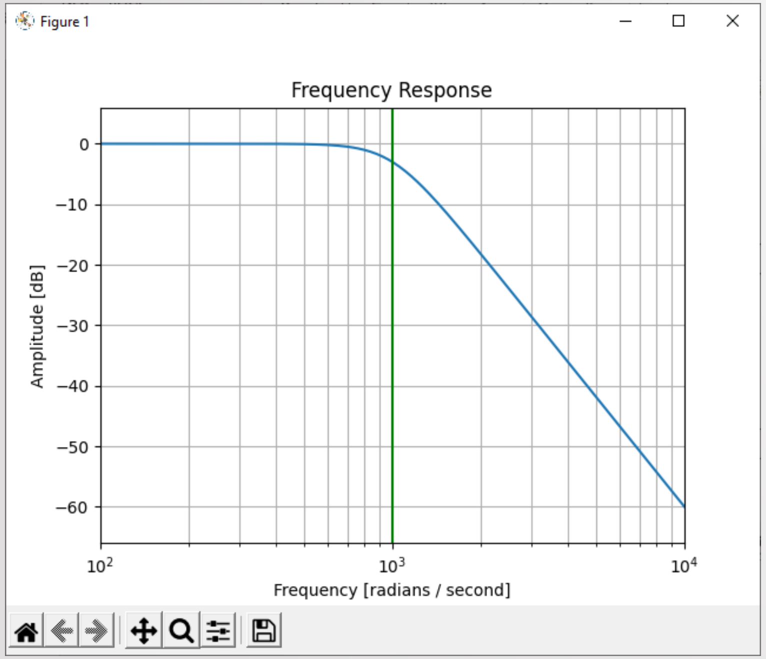


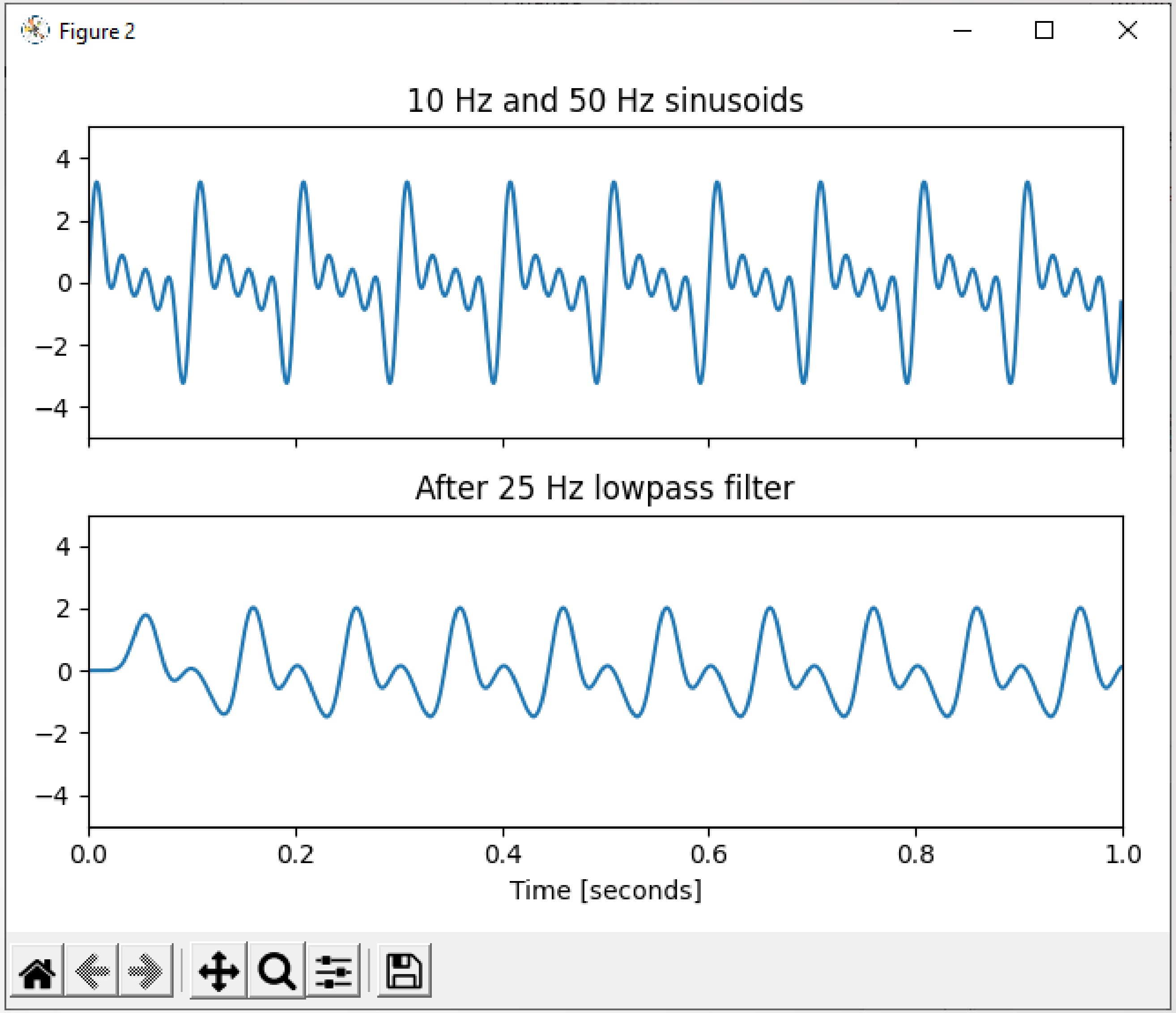
The higher the Butterworth filter order, the higher the number of cascaded stages there are within the filter design, and the closer the filter becomes to the ideal "brick wall" response

In practice however, Butterworth's ideal frequency response is unattainable as it produces excessive passband ripple.

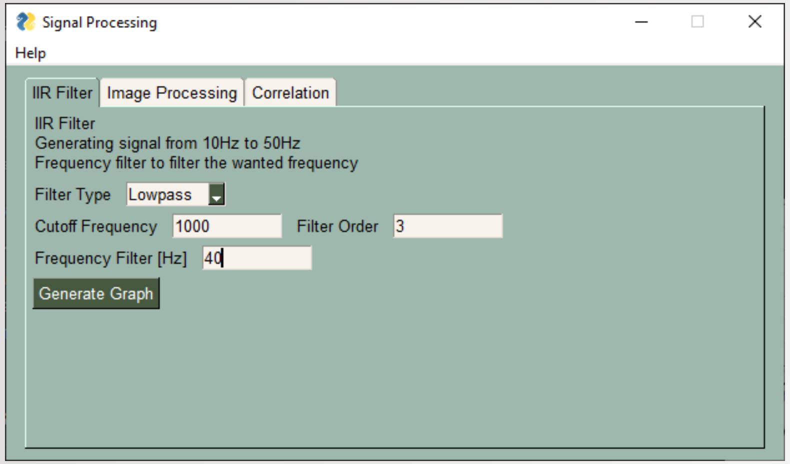


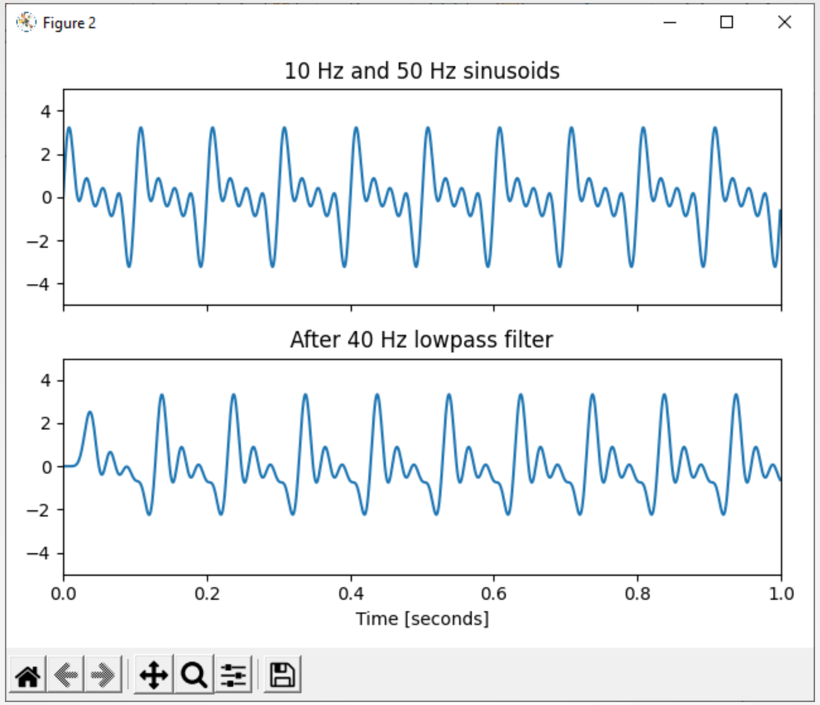
Filter type: Lowpass
Cutoff frequency = 1000
Filter order = 3
Frequency filter = 25



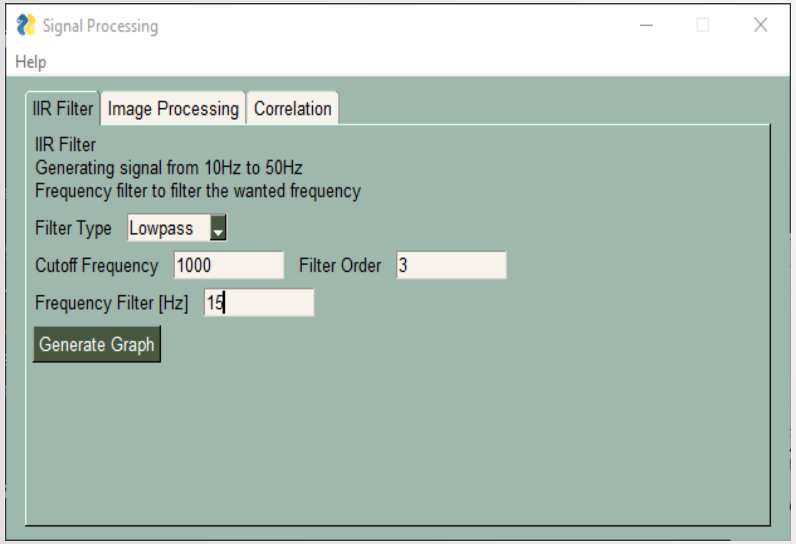


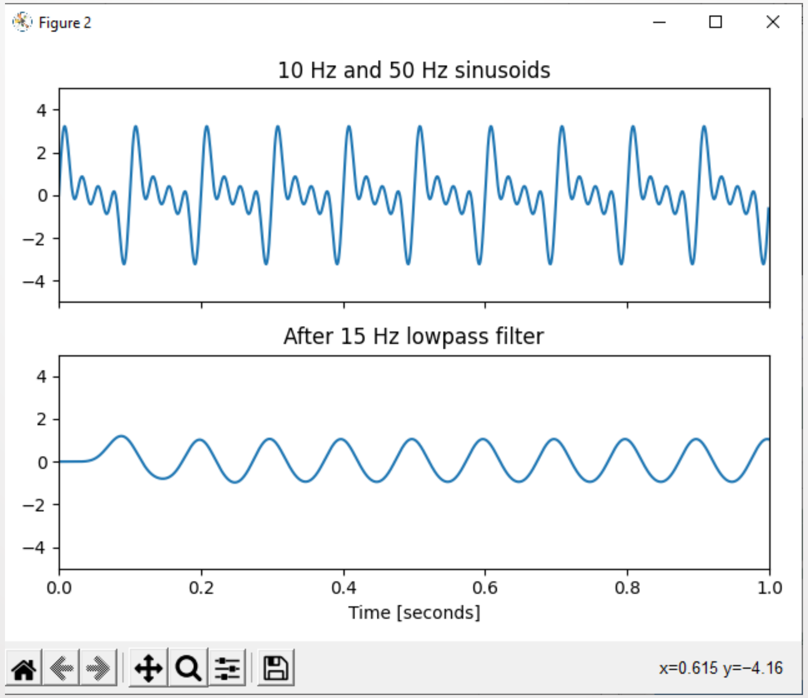
With a filter frequency of 25 we can see that just values that are 25 or lower are passed into the filtered signal





After making the frequency filter 40 we can allow more signals to pass through as seen on the picture above, compared to the previous picture





But if we make the frequency lower, like on the picture above that is 15, a very small range of signals is going to pass. And we can see that executed on the picture above where only approx.. 1 signal is displayed. The reason is that the signal starts from 10Hz, and we have only allowed signal from 15Hz and lower, which is very close to 10Hz

Image Processing

Apart from signal filtering filters can also be used on images. But in this case, we are not applying the filter into a time series which is a signal that varies in time , but we can think about it as a signal that is varying in space. We can think about the image as a square where we can move through 2D space, and we have low frequencies and high frequencies / different values. So, its not a time dependent signal but a space depended signal

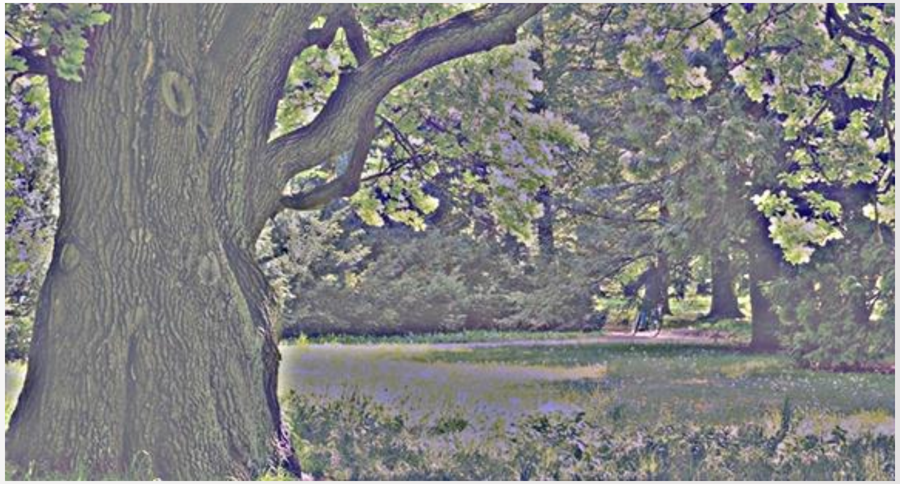
- When we apply a lowpass frequency filter the image becomes blurrier. The reason is that we are finding low frequency/values on the image and making their neighbor pixel values smoother that's why the image becomes blurrier. Low pass filter can also be used to remove noise from images
- When applying a high pass filter on the other end, we are finding the opposite of the lowpass filter. We are looking for high frequencies/values where the change is sudden, and we can detect the edge. This is used to sharpen an image, or to make the edges of an object more clearly by making their neighbor pixels values higher. This method is also used when wanting to enhance an image

Example

Original Image

After applying the low frequency filter

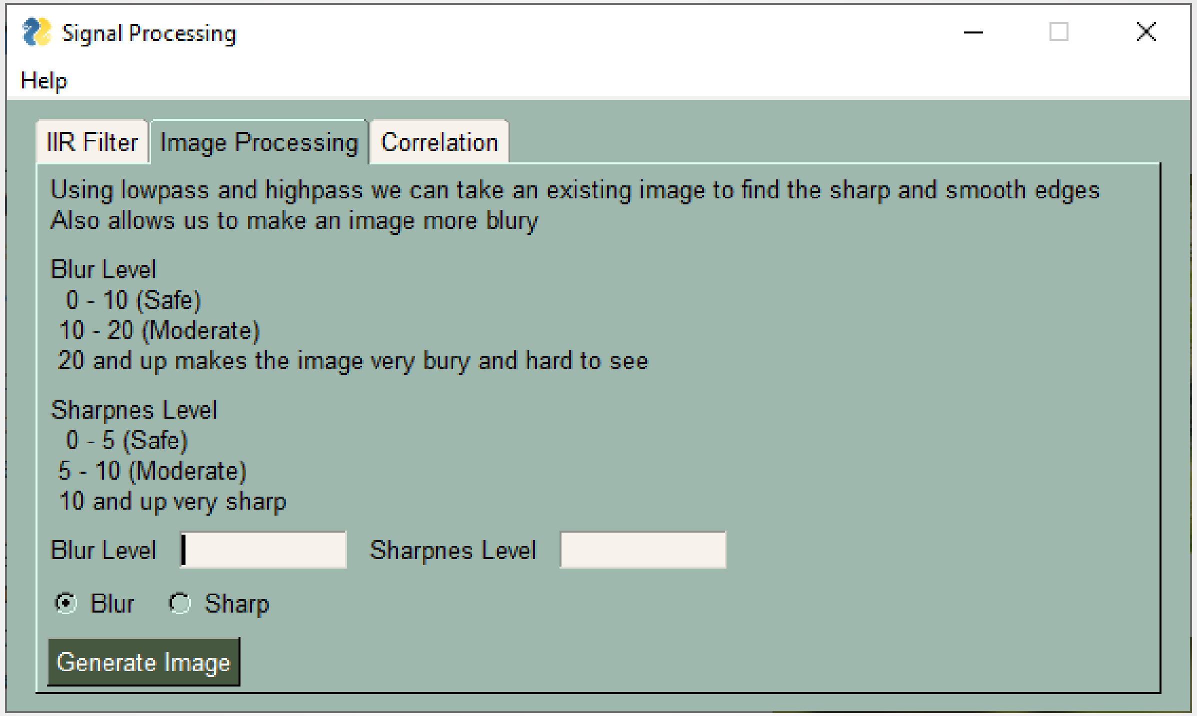
Original Image



After applying the high frequency filter

The image becomes more greyish because to construct the high frequency filter we subtract the low frequency image which contains most of the color as seen on the picture above

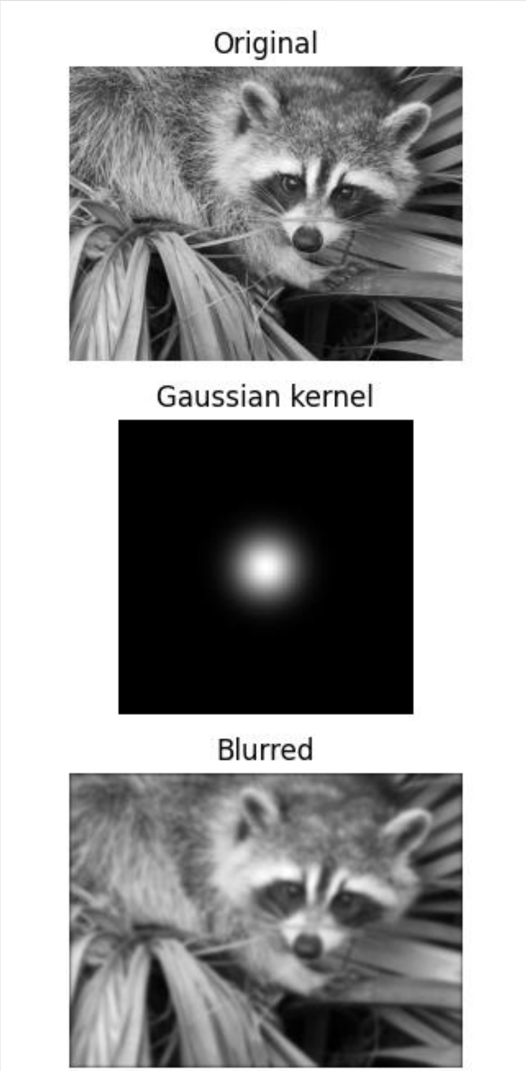
Image Processing on the program



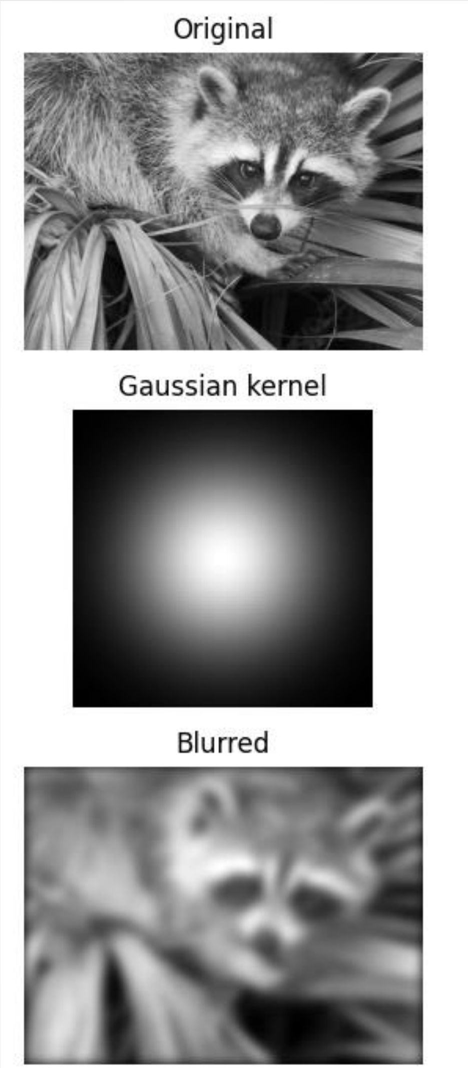
Program supports 2 types of image processing: Blurring & Sharpening (Edge detect)

To make the blurring we use a low pass filter find all low frequencies/values and make their neighbor values smoother, which in turn makes the picture blurrier.

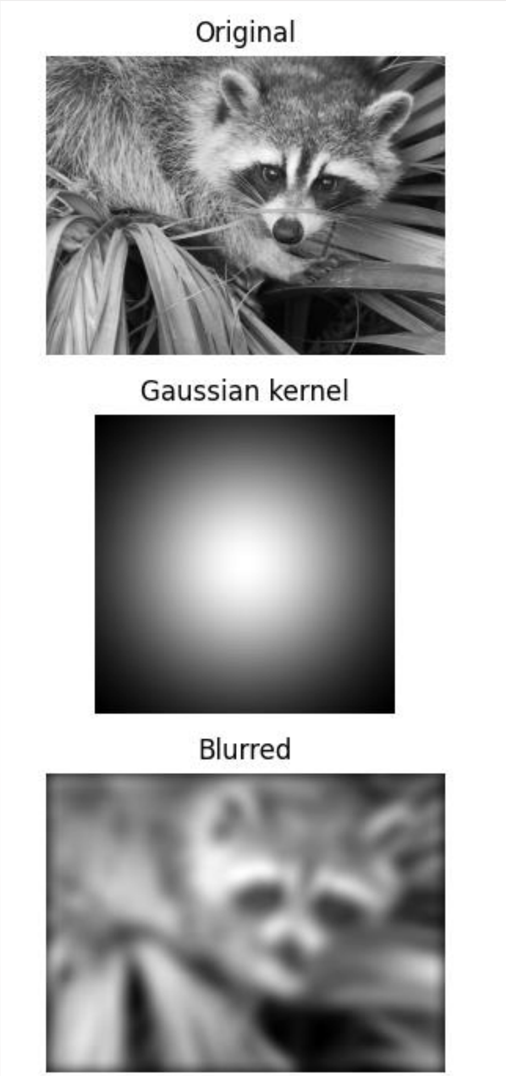
To make the sharpening we use a high pass filter to find all the high frequencies/values. But what we can do is to subtract the low pass filtered image that already contains all the low values, and what remains are the high frequencies/values. And when the filter is applied this also makes their neighbor pixel values higher, creating a sharper image as seen on the images below



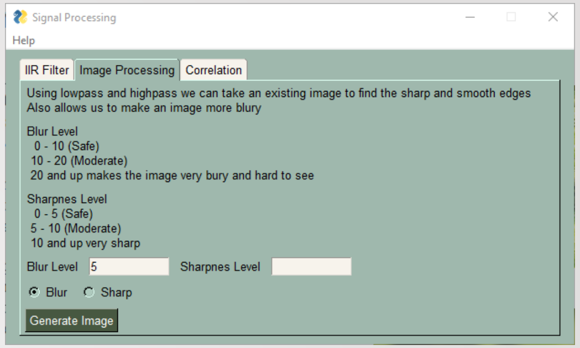
Blur level = 5



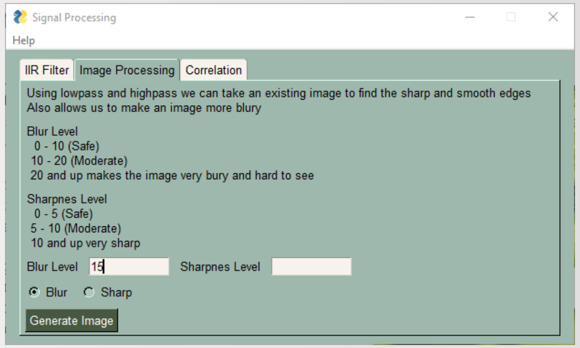
Blur level = 15



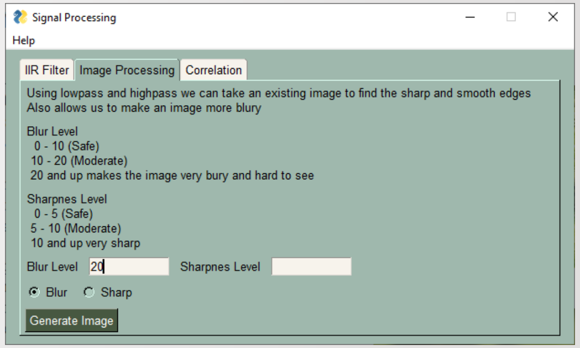
Blur level = 20



Blur level = 5



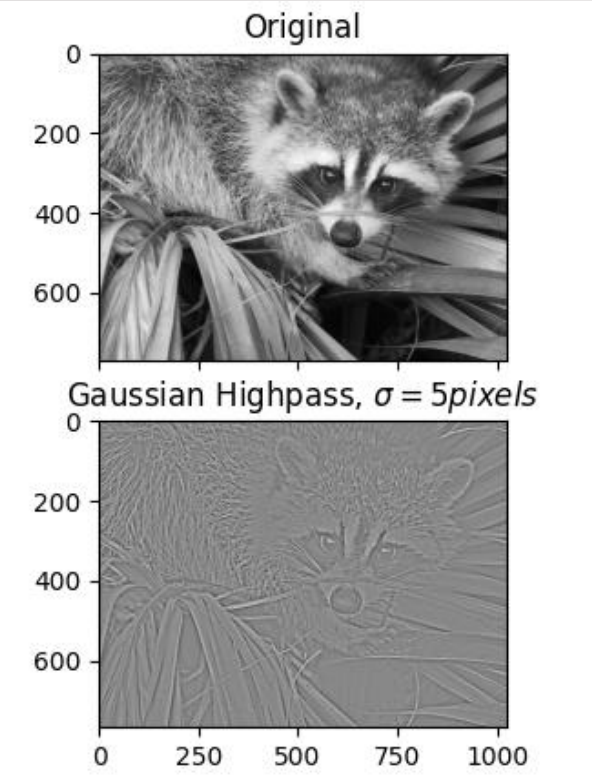
Blur level = 15



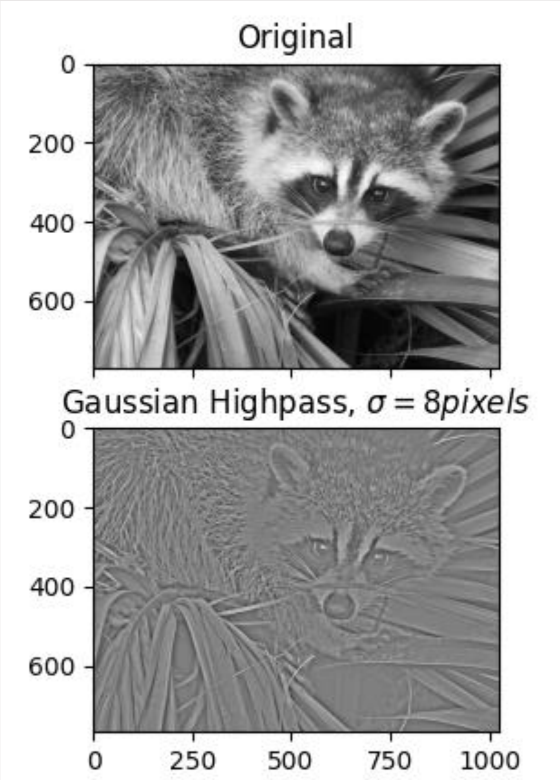
Blur level = 20

Here we have 3 cases with 3 different blur levels. As seen on the images above with the increase of the blur level the image becomes blurrier. The low frequency image is simply a convolution of the original image with a 2D Gaussian filter.

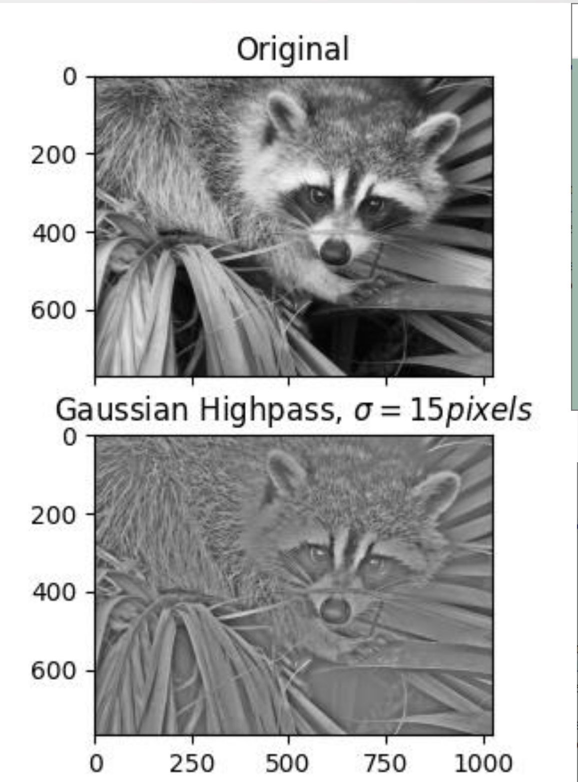
- A 2D Gaussian filter is a filter that uses a kernel that represents the shape of a Gaussian which is a bell-shaped hump.
- On the other hand, a kernel is just a small matrix of numbers that is used in image convolutions.
- And image convolution is a mathematical operation that is very common on image processing, which provides a way of multiplying together 2 arrays of numbers to generate a third one.



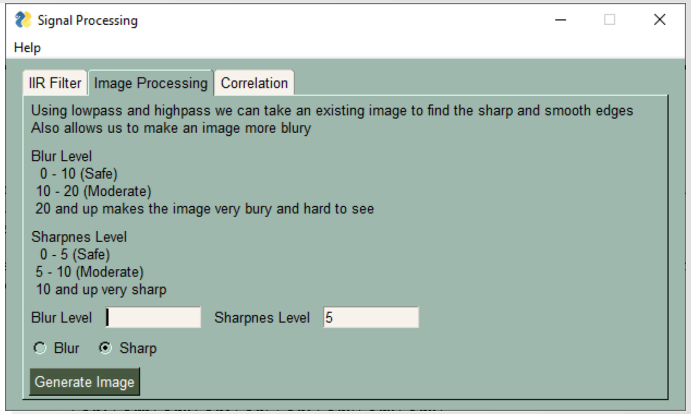
Sharpening level = 5



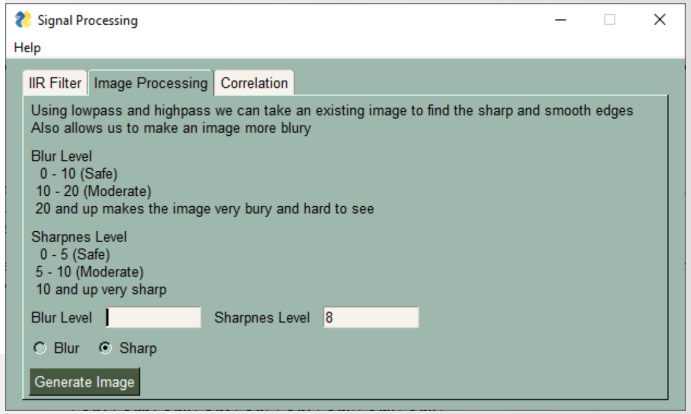
Sharpening level = 8



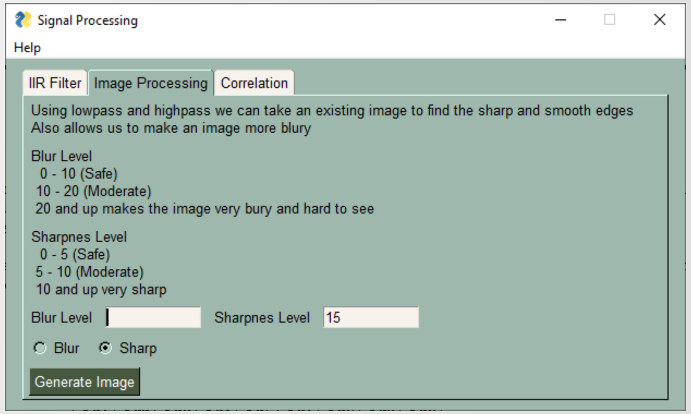
Sharpening level = 15



Sharpening level = 5



Sharpening level = 8



Sharpening level = 15

Here we have 3 cases with 3 different sharp levels. As seen on the images above with the increase of the sharpness level the image becomes sharper.

The high frequency image is obtained by subtracting the low frequency convolution from the original image. The high frequency image is shifted to account for negative values that result from the subtraction.

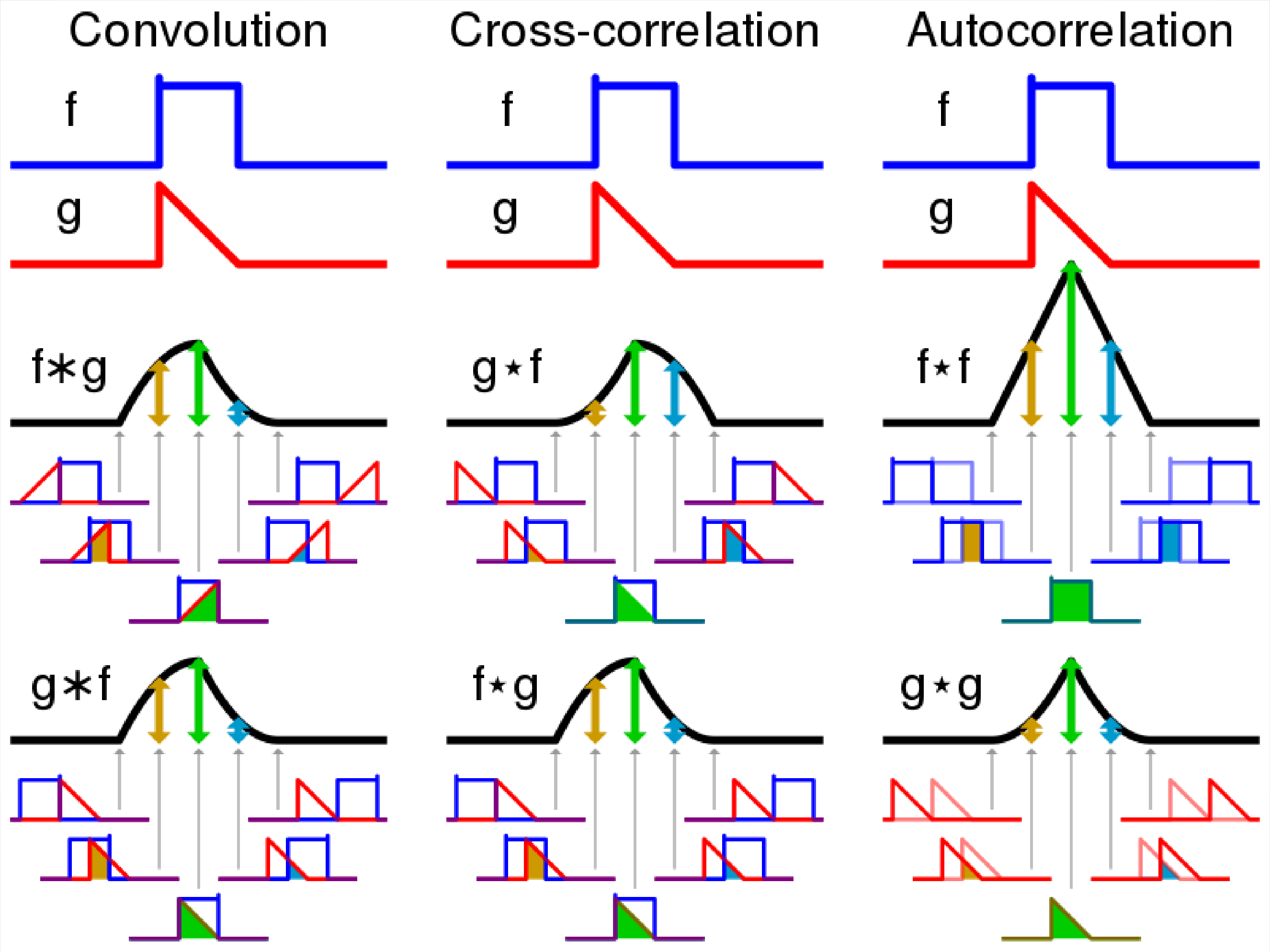
Correlation

Correlation represents a method in which we can verify if a known signal is found into another signal.

Depending on the signals that are used we can have cross-correlation, as we will see on the examples below, or we could have autocorrelation.

Cross-correlation is when we send a signal and receive a signal back, and then we can take those 2 signals and check if the initial signal that we sent is found in the received signal.

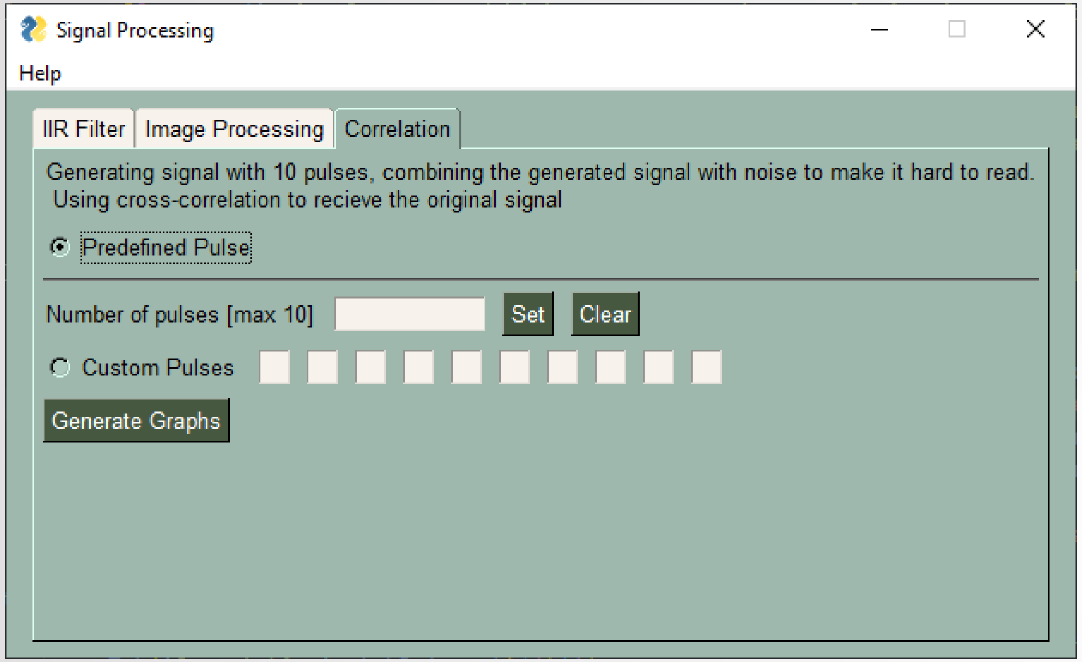
Autocorrelation is when the signal correlation is performed with itself.

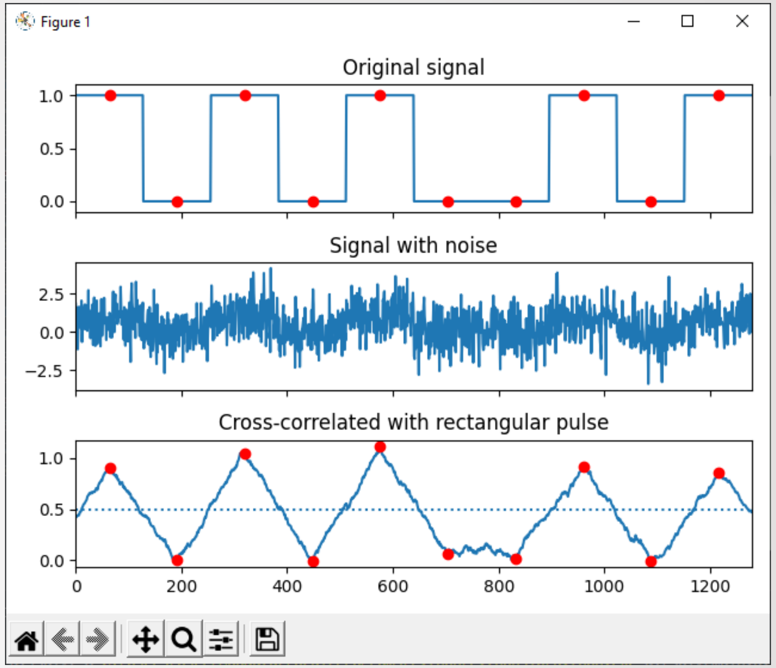


On this image we can see Convolution , Cross-correlation and Autocorrelation.

- Convolution is when we combine 2 signals to produce a third one
- Cross-correlation is when we send a signal, receive a signal back and check if our initial signal is found on the received signal
- Autocorrelation refers to correlation of the signal with a delayed copy of itself.

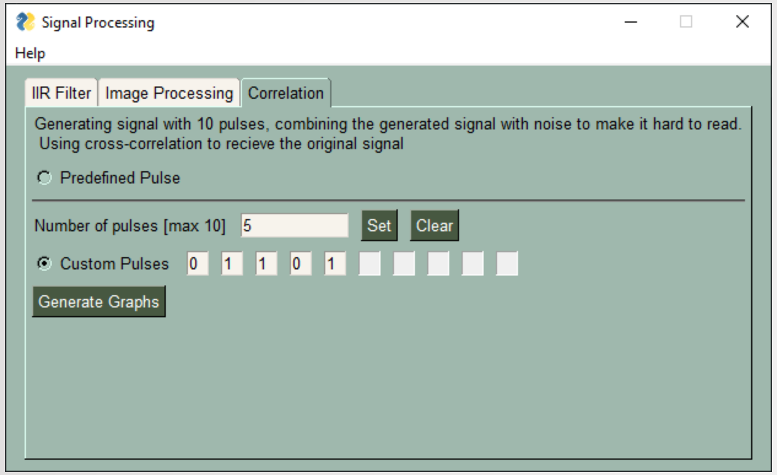
Correlation on the Program

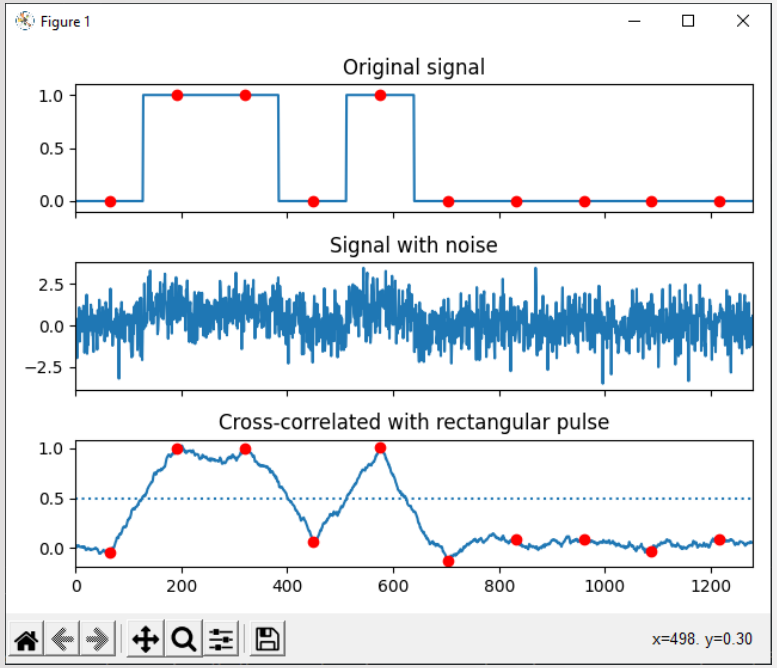




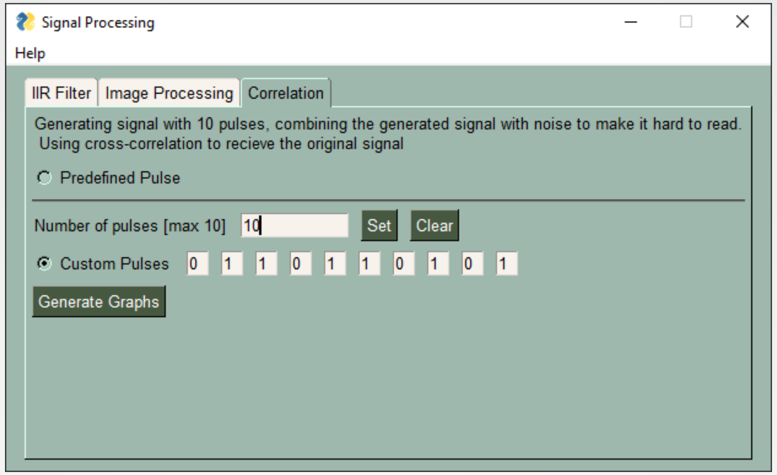
- Predefined pulse: A random pulse that is defined inside the program
- Number of pulses: If we select Custom Pulses, we can set the number of pulses we want to be displayed, the maximum number is 10
- At custom pulses, we can add the values of the pulses from 0 and higher depending on the pulse

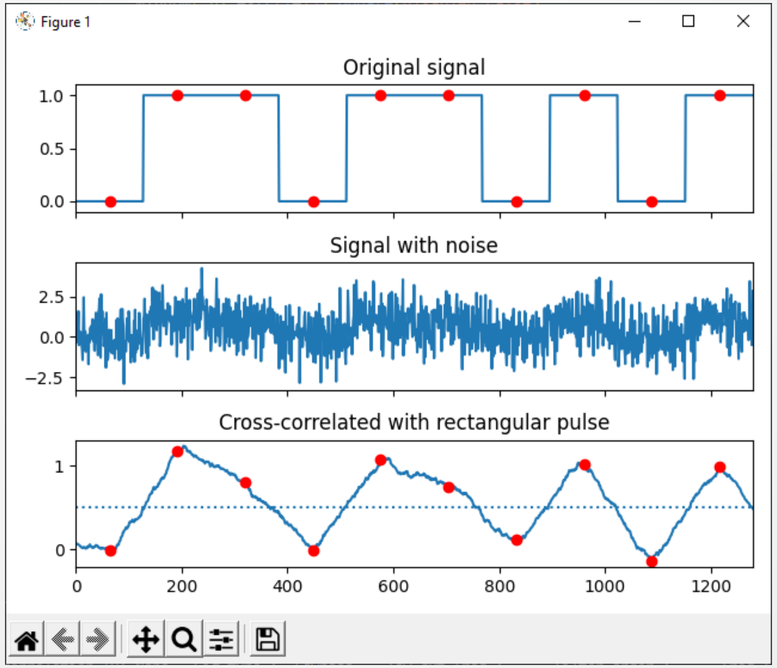
On this image we can se the generated signal pulse that is the signal that is being sent. After we combine that signal with random noise. And we get the second signal that is noisy. After using cross-correlation we can see that we could identify the initial signal on the received signal.



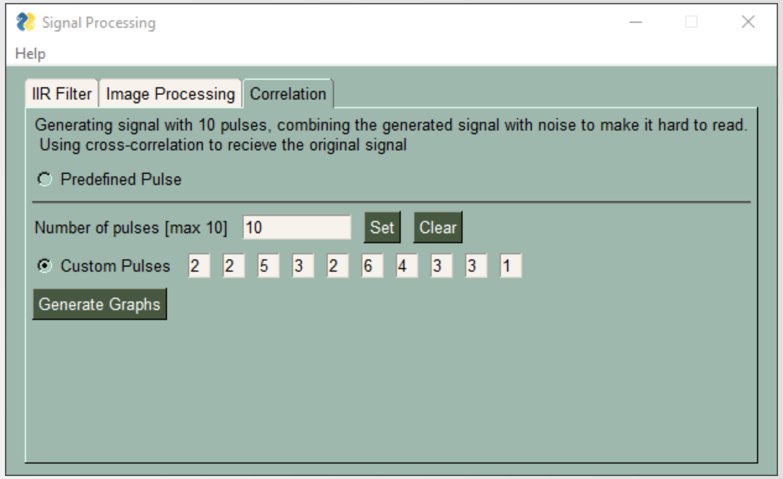


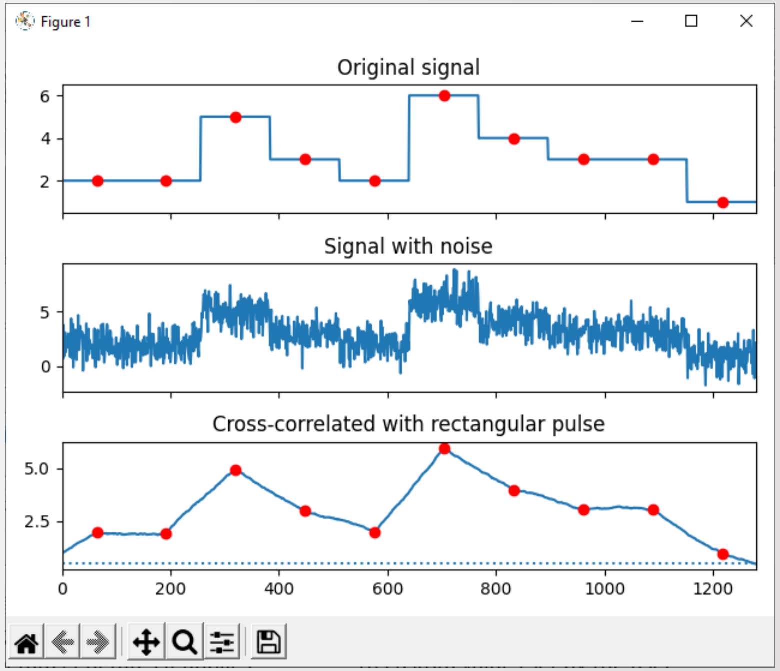
After setting a custom number of pulses to 5, the program generates it applies the noise and then the filter to make the cross correlation



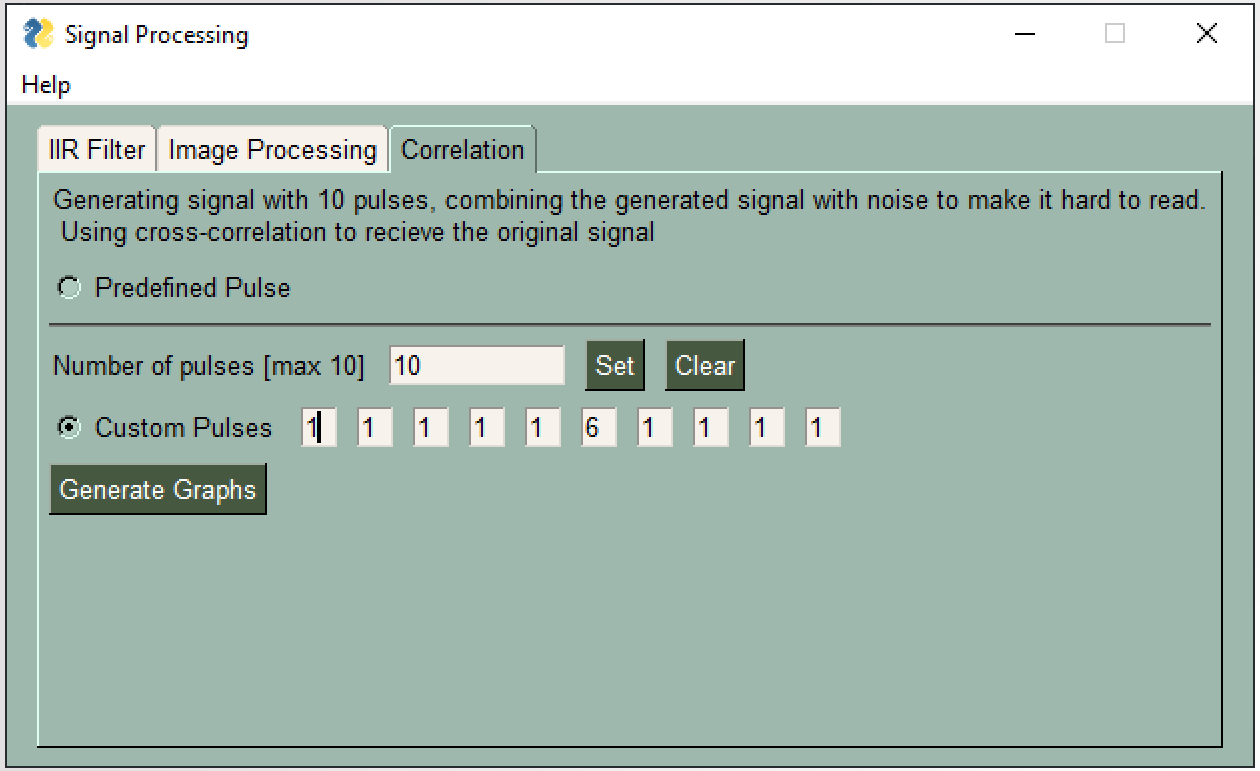


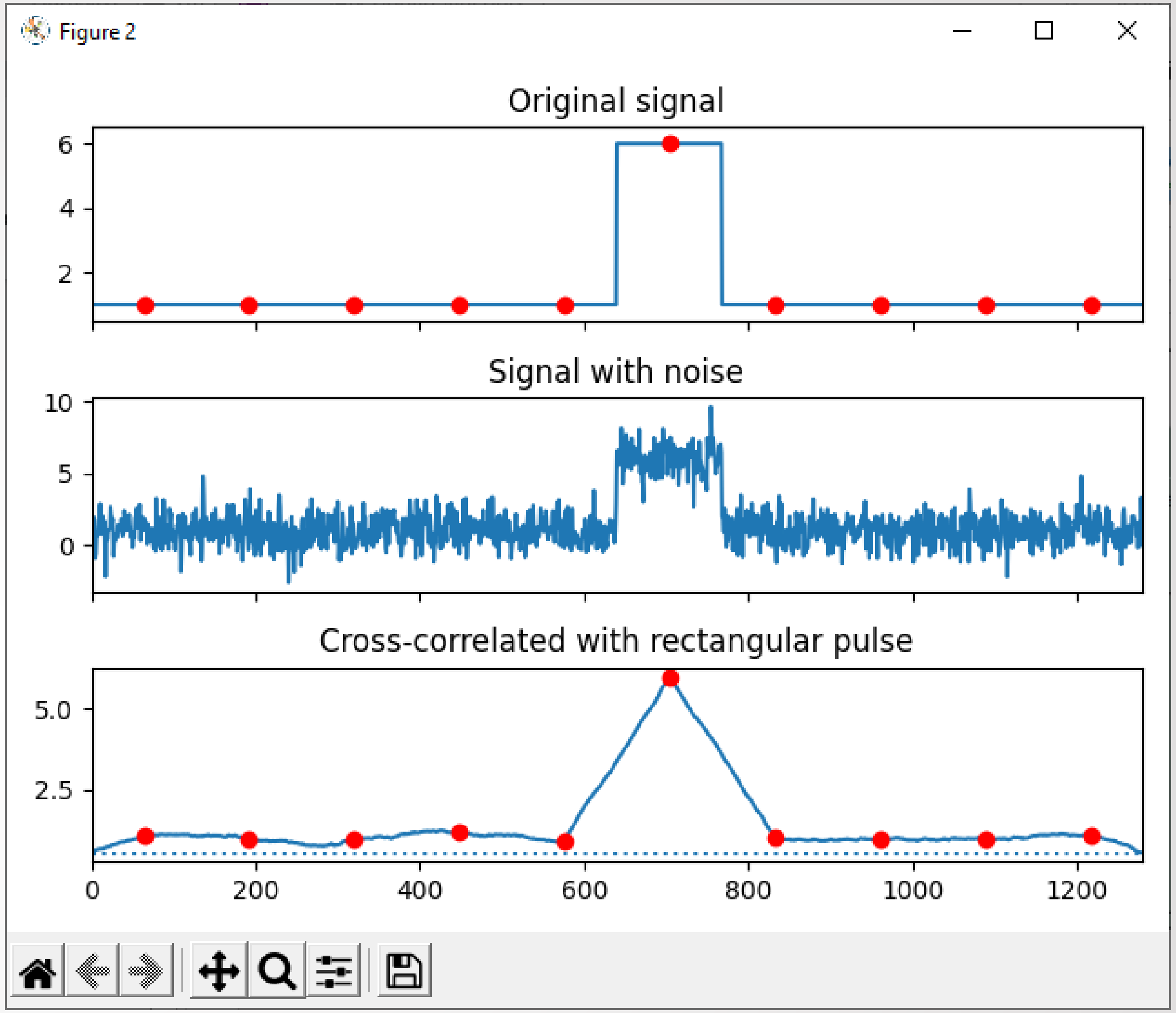
On this image we can see a pulse with 10 custom values set by the user





But we can send other pulses also that are higher than 1 to see them more clearly





Here we can see a pulse that is stronger and actually stands up from the rest on the signal, having higher values

Signal Filtering Use cases

In the 21 century where everything has gone digital, signals are the most important part of this revolution and without understanding them simply would not be possible to have what we have today. Below are a few examples of signal processing and in particular filtering.

- Sound Processing. One of the main use cases for filtering is when we want to process sound and remove all kinds of things, like background noise, wind noise and so on. On todays smartphone when we are on a call our microphone from the cellphone or headphone do active cancellation to reduce the noise coming from the outside environment and just let the voice of the person talking pass through.
- Headphones. We have ANC(Active Noise Cancelling) which uses microphones to listen to the sound signals coming from the environment and using another signal with the opposite frequency to cancel out those noises.
- Image Processing. We can use filtering to manipulate an image and extract valuable data if the case. We can make the image blurrier, or sharper based on the type of filter. We can do edge detection that is used on architectural photography to detect the edges of a building and its form. We can use it to enhance low resolution images to make them clearer and more understandable. Noise reduction on a photo is a big deal on our todays smartphone when we take picture at night, so a low pass filter would be needed to remove all the low frequency/values from a given photo.
- And there are a lot of other fields using filters too like: Voice Recognition, SONAR, RADAR, Seismology etc

Bibliography

SciPy - https://scipy.org/

IIR Filter - Website

SciPy Correlation - Website

SciPy Convolution – Website

Autocorrelation Image - https://en.wikipedia.org/wiki/Autocorrelation

Image Processing Example Images – Website

Ideal Filter Response Curves Image - Website