MATLAB – Using the Fourier Transform

Some concepts and examples how to use MATLAB to compute the Fourier Transform are listed below.

In Matlab the most efficient way to compute the discrete Fourier transform a Fast Fourier Transform:
X = fft(x)

Note that MATLAB expects the length of *x* to be equal to the power of 2.
Either you can prepare your input accordingly (pad with zeros) or let MATLAB do it automatically.

In the following command N is defined to be power of 2.
X = fft(x,N)

The inverse Fourier transform is computed using:
x = ifft(X)

A continuous *x* input must be sampled. The sampling rate should follow the Nyquist theorem – *sampling frequency = 2 maximum frequency*.

MATLAB – Project 0

The objective of this exercise is to gain a visual understanding of how the Fourier transform works and how we can use MATLAB plot the result.

1. Generate in MATLAB a sinusoidal wave with frequency f=1Hz and phase φ=0° and plot it.
2. Calculate the Fourier Transform of the signal, using the MATLAB function *“fft”* and plot it as well. Using the resulting plot, determine the frequency of the signal.
3. In order to make your plot “readable” you have to first normalize the frequencies and to use the MATLAB function *“fftshift”* in order to center the result to the desirable range. Can you explain the numbers on the Y-axis?
4. Now create one more sinusoidal wave with frequency f=10Hz and phase φ=3π/2° and add it to the previous one.
    a. Repeat the steps 2 and 3 using the new signal as input. Can you see the differences on the plot of the frequency domain?
    b. Try the same but create a cosine wave this time instead of a sinusoid one. What do you observe?
5. Create an aperiodic pulse. You can do it using the MATLAB function “rectpuls”.
    a. Apply the Fourier Transform on the pulse and plot the result. What function is the resulting output?