FILTERING A NOISY SIGNAL WITH FFT

by Prof. M. Kostic

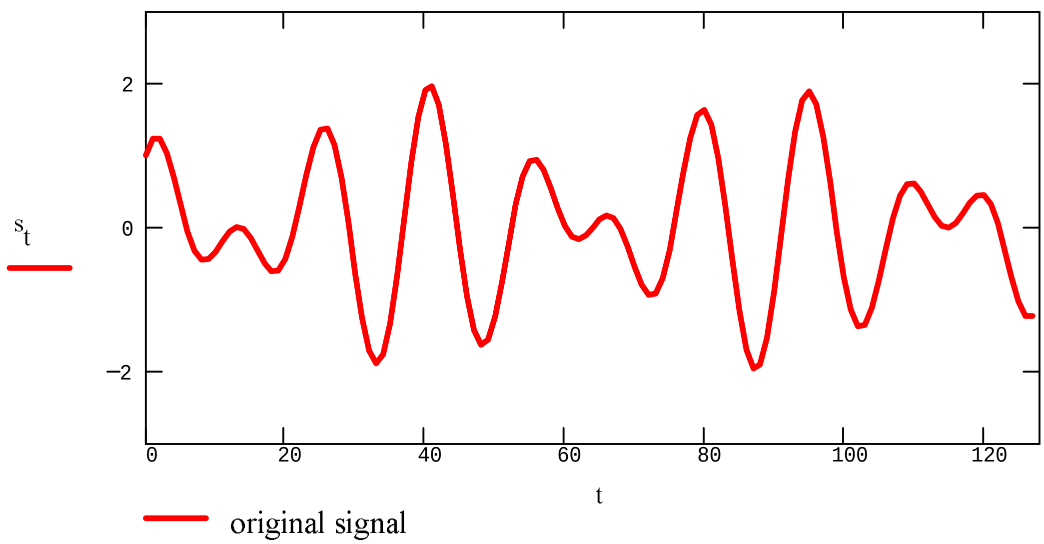
Define an arbitrary signal (s):

$t := 0 .. 127$ ...time range variable

$$s_t := \sin\left(\frac{t}{128}\cdot 14\cdot\pi\right) + \cos\left(\frac{t}{128}\cdot 19\cdot\pi\right)$$

...and plot it

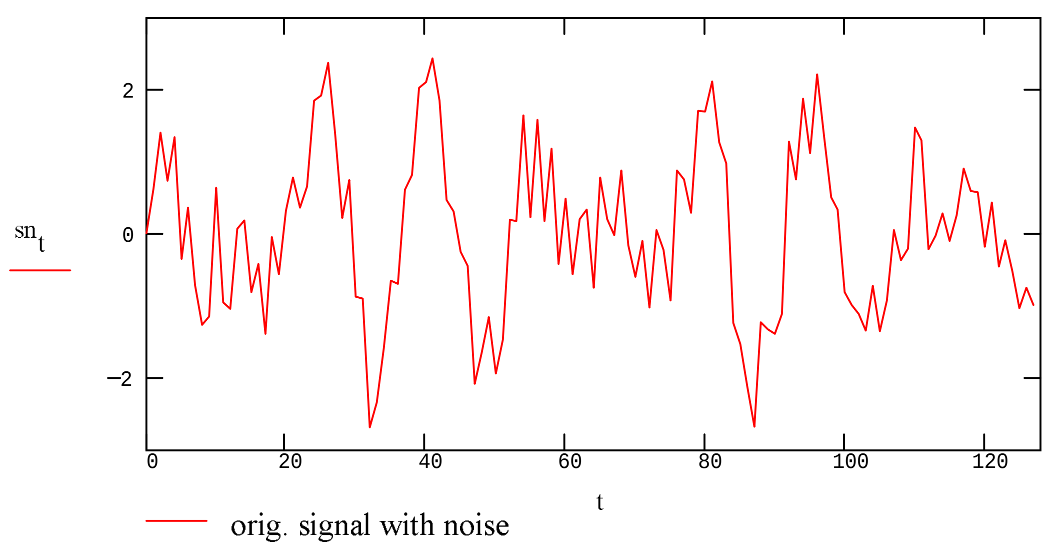
an arbitrary (original)signal



Add some random noise [rnd(2)-1 = random number between -1 and +1] to get signal with noise (sn) and plot it:

$$sn_t := s_t + \mathrm{rnd}(2) - 1$$

original signal with random noise

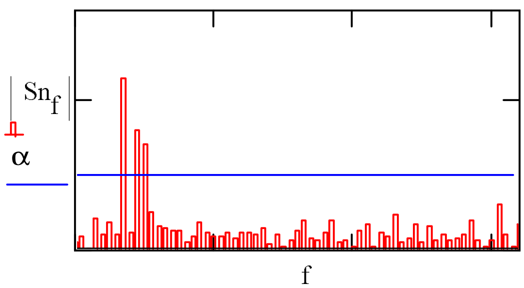


Let's now forget about the original signal (s), but try to get it from noisy one (sn) by filtering the noise using the Fast-Fourier-Transform (FFT or fft) and its inverse (ifft) functions, see below:

$Sn := fft(sn)$ ...noisy signal in frequency domain

$f := 0..64$ ...frequency range counter

$\alpha := 2.5$ ...define threshold for spectral noise rejection.



<-- Signal

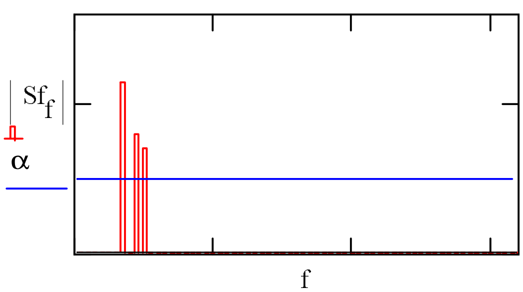
<-- threshold cutoff

<-- Noise

Filter the signal harmonics which amplitudes |Sn| are smaller than threshold a , making them zeros by using the step function F.

$Sf_f := Sn_f \cdot \Phi\left(\left|Sn_f\right| - \alpha\right)$ ...filtered signal in frequency domain

Plot the filtered signal |Sf| in frequency domain and compare it with |Sn|:



<-- Signal

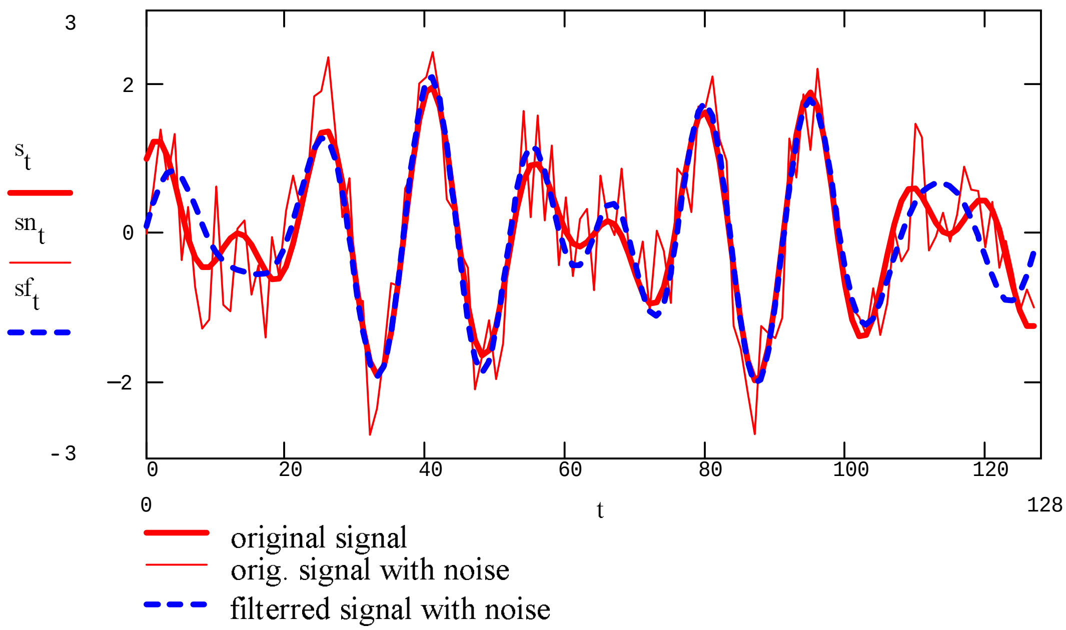
<-- threshold cutoff

<-- Noise

Take the inverse transform of (Sf) to get the filtered signal in real-time domain (sf) and plot it together with original signal (s); they should be "almost the same." Remember nothing is perfect!:

$sf := ifft(Sf)$

Now, plot all three signals together and see the outcome of filtering, i.e. removing the harmonics with smaller magnitudes than the threshold established on the basis of harmonics with major magnitudes.



NOTE:
The low magnitude noise is filtered and the filtered signal for the most part coincides with the original signal. I hope you were exciting!