

# Digital Filtering by means of the Fourier Transform Using Mathcad

## Authors:

J.L. Guinon\*, E. Ortega, J. García-Antón, V. Pérez-Herranz

Universidad Politécnica de Valencia/Ingeniería Química y Nuclear, Valencia, Spain

\* E-mail: jlguinon@iqn.upv.es

**Abstract-** *A certain amount of noise is found in all measurements along with the desired signal. Sometimes the signal is so large compared with the noise amplitude that its presence is hardly noticed. However, in other experiments the noise is so large that the signal cannot be found at all. In order to improve the signal-to-noise (S/N) ratio, different digital filtering techniques can be used. Software techniques are programmable; hence they can be adapted to different types of noise. In the filtering process several signal enhancement techniques (boxcar averaging, moving average filter, Savitzky-Golay filter and Fourier Transforms) are used. Of interest is the Fast Fourier Transform, FFT, which enjoys broader applications, including digital filtering, signal differentiation and signal resolution. Among the different symbolic mathematical solvers, Mathcad is one of the most widely used in university and industry. The ease with which Mathcad lets us combine text, equations and graphics in a single document, nearly in the same way as we write them down on paper, makes Mathcad a powerful tool to develop instructional programs. In this paper we report the use of Mathcad to demonstrate the effectiveness of FFT in eliminating sources of noise. Two Mathcad applications are provided, one for environmental noise and another for white noise. In its filtering application, the forward FFT transforms the data array to the frequency domain where, by using some predefined mathematical function, the noise frequencies are truncated. In the frequency domain, the noise frequencies occupy a different region from the signal frequencies. A truncating function is chosen to remove the noise frequencies. The inverse FFT is then used to convert the convoluted frequencies back to the time domain with increased S/N ratio. As soon as the signal, noise and frequency truncation range parameters are modified, Mathcad updates results and redraws graphs, allowing the user to see the effect of filtering on a noisy signal.*

**Index Terms-** *Computer-Based Learning, Fourier Transform, Mathcad, Noise, Signal*

## INTRODUCTION

A certain amount of noise is found in all measurements along with the desired signal. Sometimes the signal is so large compared with the noise amplitude that its presence is hardly noticed. However, in other experiments the noise is so large that the signal cannot be found at all. When the signals are weak, the amount of noise can be so great as to cover the signal completely.

In order to improve the signal-to-noise (S/N) ratio, different digital filtering techniques can be used. Software techniques are programmable; hence they can be adapted to different types of noise. In the filtering process several signal enhancement techniques (boxcar averaging, moving average filter, Savitzky-Golay filter and Fourier Transforms) are used. Of interest is the Fast Fourier Transform, FFT, which enjoys broader applications, including digital filtering, signal differentiation and signal resolution. In its filtering application, the forward FFT transforms the data array to the frequency domain where, by using some predefined mathematical function, the noise frequencies are truncated. In the frequency domain, the noise frequencies occupy a different region from the signal frequencies. A truncating function is chosen to remove the noise frequencies. The inverse FFT is then used to convert the convoluted frequencies back to the time domain with increased S/N ratio. A method of digital filtering using the FFT via Excel has been recently described [1]. However, this method is based on the construction of macros for the forward and inverse Fourier Transform, which is time-consuming and tedious.

Computed-based simulation with computer is a good tool to provide the theoretical foundations for experimentation. Among the different symbolic mathematical solvers, Mathcad is one of the most widely used in university and industry [2-4]. The ease with which Mathcad lets us combine text, equations and graphics in a single document, nearly in the same way as we write them down on paper, makes Mathcad a powerful tool to develop instructional programs. The use of Mathcad in engineering has been discussed in previous papers [5-7]. In this paper we report the use of Mathcad to demonstrate the effectiveness of FFT in eliminating sources of noise. Two Mathcad applications are provided, one for environmental noise and another for white noise. As soon as the signal, noise and frequency truncation range parameters are modified, Mathcad updates results and redraws graphs, allowing the user to see the effect of filtering on a noisy signal.

## RESULTS AND DISCUSSION

### Fourier Transform and Filtering of Environmental Noise

Mathcad application is obtained by means of the following parameters and equations:

$$N := 2^8 \quad N = 256 \quad \text{Total data point} \quad (1)$$

$$t := 0.256 \quad \text{Total time} \quad (2)$$

$$\Delta t := \frac{t}{N} \quad \Delta t = 1 \times 10^{-3} \quad \text{Sampling time} \quad (3)$$

$$i := 0..N - 1 \quad t_i := i \cdot \Delta t \quad \text{Time index and successive times} \quad (4)$$

$$f := 60 \quad \text{Signal frequency} \quad (5)$$

$$y1_i := \cos(2 \cdot \pi \cdot f \cdot t_i) \quad \text{Pure signal in the time domain} \quad (6)$$

$$k := 1 \quad \text{Noise level} \quad (7)$$

$$f1 := 100 \quad f2 := 170 \quad \text{Environmental noise frequencies} \quad (8)$$

$$n_i := k \cdot (\cos(2 \cdot \pi \cdot f1 \cdot t_i) + \cos(2 \cdot \pi \cdot f2 \cdot t_i)) \quad \text{Environmental noise} \quad (9)$$

$$y2_i := \cos(2 \cdot \pi \cdot f \cdot t_i) + n_i \quad \text{Noisy signal in the time domain} \quad (10)$$

$$Y1 := \text{fft}(y1) \quad \text{Fourier transform of the pure signal} \quad (11)$$

$$Y2 := \text{fft}(y2) \quad \text{Fourier transform of the noisy signal} \quad (12)$$

$$k := 0.. \frac{N}{2} \quad f_k := \frac{k}{t_{N-1}} \quad \text{Frequency index and successive frequencies} \quad (13)$$

$$tf1 := 80 \quad tf2 := 400 \quad \text{Frequency truncation range} \quad (14)$$

$$\text{Filter}_k := \text{if} \left[ \left( f_k > tf1 \right) \cdot \left( f_k < tf2 \right), 0, 1 \right] \quad \text{Rectangular filter} \quad (15)$$

$$\text{SignalFilt}_k := Y2_k \cdot \text{Filter}_k \quad \text{Filtered signal in the frequency domain} \quad (16)$$

$$\text{SignalF} := \text{ifft}(\text{SignalFilt}) \quad \text{Filtered signal in the time domain} \quad (17)$$

Mathcad application starts with the sampling parameters, lines (1-4). The application continues with the parameters of the pure signal, lines (5-6), and of the noisy signal, lines (7-10), in the time domain. Lines (11-13) define the parameters of the Fourier transform of the pure signal and noisy signal, and their respective frequencies. The frequency truncation range and the rectangular apodization filter are defined in lines (14-15). By means of the product of the noisy signal by the filter, line (16) defines the filtered signal in the frequency domain. Line (17) allows obtaining the filtered signal in the time domain by means of the inverse Fourier transform.

Figure 1 shows the pure, noisy and filtered signals in the time domain, and Figure 2 shows the noisy and filtered signals in the frequency domain. Curves are offset for visibility. Clearly, the environmental noise amplitude is reduced drastically with little distortion of the signal, because the noise is much localized.

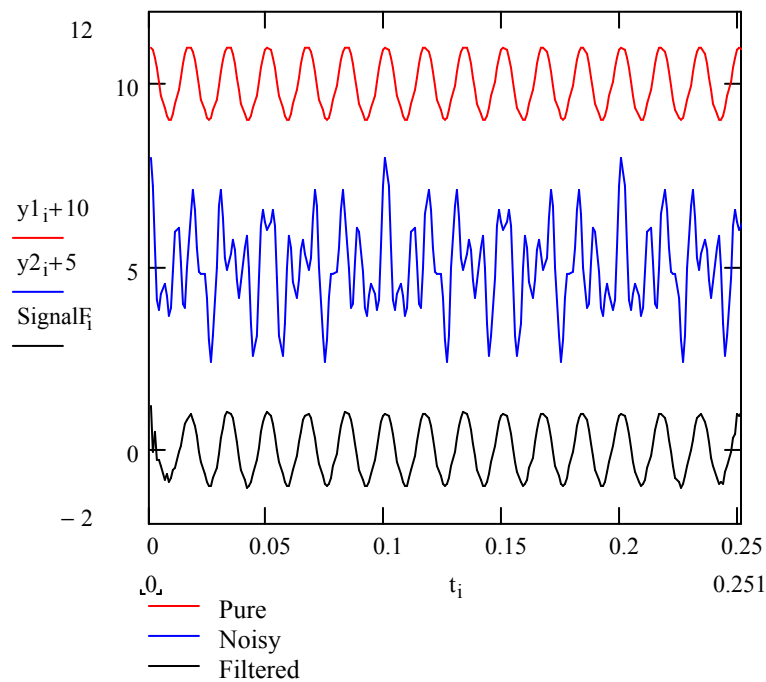


FIGURE 1  
PURE, NOISY (ENVIRONMENTAL NOISE) AND FILTERED SIGNALS IN THE TIME DOMAIN.

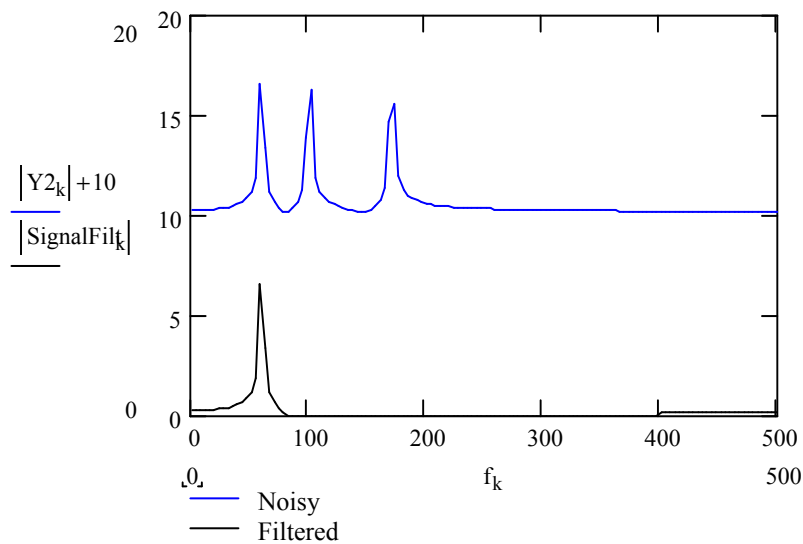


FIGURE 2  
PURE AND FILTERED SIGNALS IN THE FREQUENCY DOMAIN

Finally, lines (18-23) show the procedure to obtain the signal-to-noise ratio (SNR) before and after the filtering process. The use of digital filtering of environmental noise by means FFT is consequently easy to justify because the signal-to-noise ratio is improved by a factor of around 15 relative to the non filtered input signal.

Signal-to-noise ratio (SNR) before filtering:

$$\text{Signal} := \max(y1) - \min(y1) \quad \text{Signal} = 2 \quad (18)$$

$$\text{Noise1} := \text{stdev}(y2 - y1) \quad \text{Noise1} = 1 \quad (19)$$

$$\frac{\text{Signal}}{\text{Noise1}} = 2 \quad \text{SNR before filtering} \quad (20)$$

Signal to noise ratio (SNR) after filtering

$$\text{Signal} := \max(y1) - \min(y1) \quad \text{Signal} = 2 \quad (21)$$

$$\text{Noise2}_i := y1_i - \text{Signal}F_i \quad \text{Noise2} := \text{stdev}(\text{Noise2}) \quad \text{Noise2} = 0.142 \quad (22)$$

$$\frac{\text{Signal}}{\text{Noise2}} = 14.13 \quad \text{SNR after filtering} \quad (23)$$

### Fourier Transform and Filtering of the White Noise

Mathcad application is obtained by means of the following parameters and equations (24-39).

$$N := 2^8 \quad N = 256 \quad \text{Total data point} \quad (24)$$

$$t := 0.256 \quad \text{Total time} \quad (25)$$

$$\Delta t := \frac{t}{N} \quad \Delta t = 1 \times 10^{-3} \quad \text{Sampling time} \quad (26)$$

$$i := 0.. N - 1 \quad t_i := i \cdot \Delta t \quad \text{Time index and successive times} \quad (27)$$

$$f := 60 \quad \text{Signal frequency} \quad (28)$$

$$y1_i := \cos(2 \cdot \pi \cdot f \cdot t_i) \quad \text{Pure signal in the time domain} \quad (29)$$

$$k := 2 \quad \text{Noise level} \quad (30)$$

$$n_i := (\text{rnd}(1) - 0.5) \cdot k \quad \text{Random noise} \quad (31)$$

$$y2_i := y1_i + n_i \quad \text{Noisy signal in the time domain} \quad (32)$$

$$Y1 := \text{fft}(y1) \quad \text{Fourier transform of the pure signal} \quad (33)$$

$$Y2 := \text{fft}(y2) \quad \text{Fourier transform of the noisy signal} \quad (34)$$

$$k := 0.. \frac{N}{2} \quad f_k := \frac{k}{t_{N-1}} \quad \text{Frequency index and successive frequencies} \quad (35)$$

$$tf1 := 65 \quad tf2 := 400 \quad \text{Frequency truncation range} \quad (36)$$

$$\text{Filter}_k := \text{if} \left[ \left( f_k > tf1 \right) \cdot \left( f_k < tf2 \right), 0, 1 \right] \quad \text{Rectangular filter} \quad (37)$$

$$\text{SignalFilt}_k := Y2_k \cdot \text{Filter}_k \quad \text{Filtered signal in the frequency domain} \quad (38)$$

$$\text{SignalF} := \text{iff}(\text{SignalFilt}) \quad \text{Filtered signal in the time domain} \quad (39)$$

Mathcad application starts with the sampling parameters, lines (24-27). The application continues with the parameters of the pure signal, lines (28-29), and of the noisy signal, lines (30-32), in the time domain. Lines (33-35) define the parameters of the Fourier transform of the pure signal and noisy signal, and their respective frequencies. The frequency truncation range and the rectangular apodization filter are defined in lines (36-37). By means of the product of the noisy signal by the filter, line (38) defines the filtered signal in the frequency domain. Line (39) allows obtaining the filtered signal in the time domain by means of the inverse Fourier transform.

Figure 3 shows the pure, noisy and filtered signals in the time domain, and Figure 4 shows the noisy and filtered signals in the frequency domain. For visual clarity, the signals are offset. In this case, the white noise spreads out the whole spectrum and therefore the improvement obtained with truncation and the filtering is only limited.

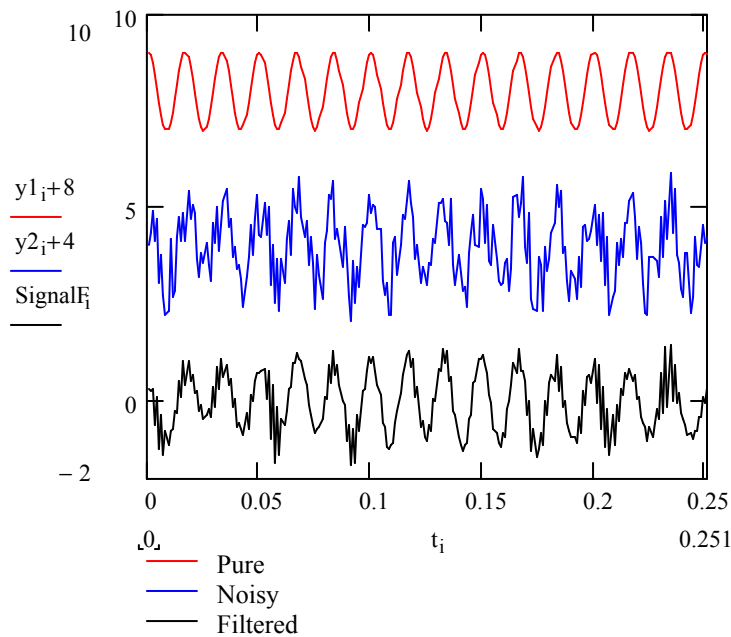


FIGURE 3  
PURE, NOISY (WHITE NOISE) AND FILTERED SIGNALS IN THE TIME DOMAIN

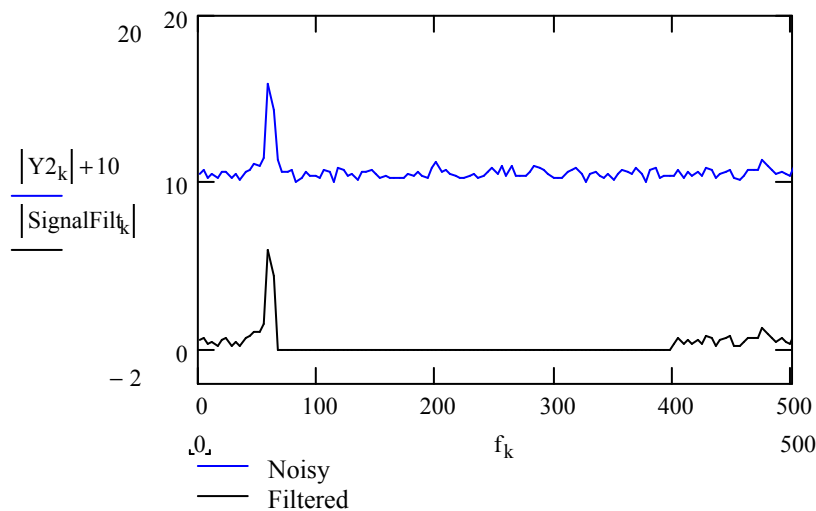


FIGURE 4  
PURE AND FILTERED SIGNALS IN THE FREQUENCY DOMAIN

Lines (40-45) show the procedure to obtain the signal-to-noise ratio (SNR) before and after the filtering process. Contrarily to environmental noise, truncation and filtering hardly increases the signal-to-noise ratio relative to the non filtered input signal.

Signal-to-noise ratio (SNR) before filtering

$$\text{Signal} := \max(y_1) - \min(y_1) \quad \text{Signal} = 2 \quad (40)$$

$$\text{Noise1} := \text{stdev}(n) \quad \text{Noise1} = 0.574 \quad (41)$$

$$\frac{\text{Signal}}{\text{Noise1}} = 3.485 \quad \text{SNR before filtering} \quad (42)$$

Signal-to-noise ratio (SNR) after filtering

$$\text{Signal} := \max(y_1) - \min(y_1) \quad \text{Signal} = 2 \quad (43)$$

$$\text{Noise2}_1 := y_{1_i} - \text{SignalF}_i \quad \text{Noise2} := \text{stdev}(\text{Noise2}) \quad \text{Noise2} = 0.404 \quad (44)$$

$$\frac{\text{Signal}}{\text{Noise2}} = 4.945 \quad \text{SNR after filtering} \quad (45)$$

## CONCLUSIONS

This paper describes the use of the Mathcad software for the implementation and analysis of a fast method for filtering a noisy signal through truncation and Fourier Transform. Two Mathcad applications are provided, one for environmental noise and another for white noise.

## ACKNOWLEDGEMENT

We wish to express our gratitude to the Universidad Politécnica de Valencia, and to Dr Asunción Jaime for her translation assistance.

## REFERENCES

- [1] Overway, K. "FT Digital Filtering. Simulating Fourier Transform Apodization via Excel" *Journal of Chemical Education*, 85, (2008), p. 1151.
- [2] Abbas, A., Al-Bastaki, N. "The Use of Software Tools for ChE Education" *Chemical Engineering Education*, 36, (2002), p 236.
- [3] Shirer, D.L. "Matlab and Mathcad: Updates for popular numeric systems" *Computing in Science & Engineering*, 3(1), (2001), p. 9.
- [4] Pritchard, P.J. "Mathcad. A tool for engineering problem solving" McGraw-Hill: Boston (1998).
- [5] E. Ortega, E., Guiñón, J.L., García-Antón, J and Pérez-Herranz, V. "Nonlinear regression of a second-order kinetics using Mathcad: Ethyl acetate saponification" *Abstracts of Papers 230th ACS National Meeting: Computer 45*. Washington, DC, United States, Aug. 28, (2005)
- [6] Guiñón, J.L., Ortega, E., García-Antón, J and Pérez-Herranz, V. "Moving average and Savitzki-Golay smoothing filters using mathcad", *Abstracts of Papers ICEE 2007*. Coimbra, Portugal, September, (2007).
- [7] Guiñón, J.L., Ortega, E., García-Antón, J and Pérez-Herranz, V. "Lock-in amplifier Response simulation using mathcad", *Abstracts of Papers ICEE 2008*. Pecs, Budapest, Hungary, July, (2008).