FFT TUTOR: A MATLAB-BASED INSTRUCTIONAL TOOL FOR FFT PARAMETER EXPLORATION

Arvind Singh, Tom De Rybel, and José R. Martí
Dept. of Electrical and Computer Engineering, The University of British Columbia, 2332 Main Mall, Vancouver, British Columbia, Canada, V6T 1Z4 tomr@ece.ubc.ca

1. INTRODUCTION

Although the Fast Fourier Transform (FFT) has been the staple of signal processing [Oppenheim(1998)] for many years, it is still frequently misapplied. In many cases, the confusion stems from misconceptions regarding the relations between time and frequency-domain parameters. Also, spectral leakage due to mismatches between the sample rate and the harmonic contents, and the choice of windowing technique, are a frequent cause of trouble.

In this paper, we will first present a summary of how the various FFT parameters relate and can be chosen in a practical way, followed by a discussion on spectral leakage, windowing and zero-padding. Then, a MatLab-based tool is introduced to help visualize the relevant concepts.

The tool allows the user to graphically evaluate the influence of the analysis parameters on harmonic signals, as well as on a custom dataset, such as a sound recording. This, then, allows the user to experiment with, and optimize, the FFT analysis parameters to enhance the resulting FFT spectrum, as well as visually compare the inverse of the spectrum produced with the original time-domain signal.

2. FFT Parameters at a Glance

In order to use the FFT effectively, the parameters governing the time and frequency domain windows must be well chosen. It are only the samples in these windows which are of concern to the FFT transform and it has no further knowledge of the signal of interest. Errors in the choice of window size and transform parameters are a main cause of poor performance of the FFT.

To select the parameters, it must always be remembered that the FFT assumes that the input signal is periodical. Thus, the signal present in the time window should be precisely an integer number of periods of the signal so that, when the window is repeated periodically, the resulting time sequence should be a faithful reproduction of the periodical input.

Small mis-matches, such as "one sample off" introduce discontinuities and can produce very noticeable degradation of the spectrum. Windowing is then used to improve the end-to-end match.

The relation between the FFT parameters [Marti(2002)] is illustrated in Fig. 1. The rules can now be summarized as:

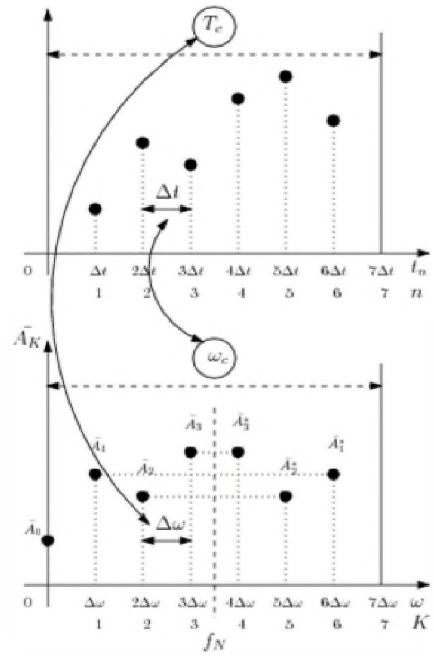

Fig.1: Relations between the time and Frequency-domain windows

$$T_c = N\,\Delta t \qquad \omega_c = \frac{2\pi}{\Delta t}$$

$$\Delta t = \frac{2\pi}{\omega_c} \qquad \Delta\omega = \frac{2\pi}{T_c}$$

$$f_N = \frac{1}{2\Delta t}$$

- The size of the time window T_c determines the lowest frequency that can be represented, and thus the frequency resolution $\Delta\omega$ of the transform.
- The sample rate Δt for a given time window determines the number of points N in the window. This determines the bandwidth ω_c. The Nyquist criterion allows 1/2 of this to avoid aliasing. In practice, 1/10 to 1/5 is used due to filter limitations.
- The frequency-domain components include one DC term A_0, and N complex conjugate frequency components An around the Nyquist frequency f_N.

3. FFT TUTOR

Fig. 2 shows a screen-shot of the MATLAB software tool developed, demonstrating a case where the time window is chosen incorrectly.

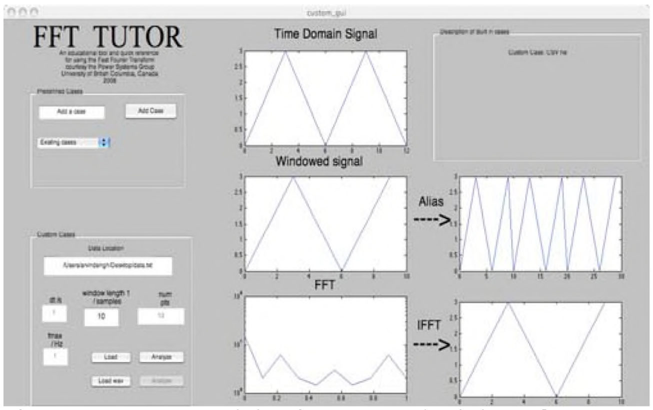

Fig. 2. FFT-TUTOR tool showing a custom loaded case from a CSV file.

The tool features four predefined cases:

- Single Frequency
- Double Frequency
- Zero padding (periodic)
- Zero padding (pulse)

For each results using proper windowing selection is illustrated, as well as results obtained with commonly encountered errors. Each case is furnished with a short explanation in the top right hand corner.

3.1 Single Frequency Case

Fig. 2 illustrates the effect of incorrectly choosing window lengths for periodic signals. An FFT frequency domain window spans from 0 to 2pi radians, the latter non-inclusive. This corresponds to one whole period of the periodical signal in time domain. A common mistake is that the time domain window often contain one point too many because both the 0 rad and 2pi rad values are included in the window. If sampling is done every 90 degrees, for example, then only the 0, 90, 180, and 270 degree values should be included in the window.

3.2 Double Frequency

When multiple known frequencies are present in the signal, proper windowing demands that integer numbers of periods of each frequency are present in the signal. If this condition can not be met, the periods of those frequency components do not match the time window, and spectral leakage will occur. This leakage is visible in the spectrum as additional components spread around the frequency bin that matches the component the closest. Energy that should be purely in the correct, but not available, spectral component thus "leaks" into neighboring bins. One way to think of this is that the various components interpolate the missing required frequency bin.

3.3 Zero padding (periodic)

Zero padding is often used for increasing frequency resolution, since it effectively lengthens the size of window. It is often misapplied, however, in use with periodic signals. In these cases, it results in a different time domain version of the signal as seen by the FFT, as shown for a sine wave in Fig. 3, and corrupts the frequency response by introducing artificial frequency components.

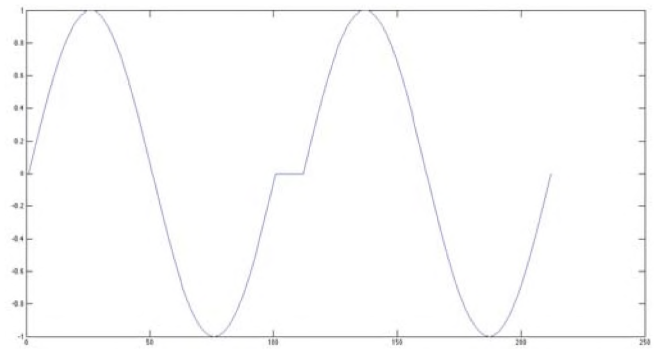

Fig. 3: Time-domain signal due to incorrect zero padding.

3.4 Zero padding (pulse)

Unlike periodic signals, zero padding can be used effectively to increase the frequency resolution for impulses. However this is not true in all cases. In fact, the term zero-padding is somewhat misleading. Padding a transient signal assumes that the padded values are approximately the same as the actual signal if a longer window was taken. Therefore, if a DC offset is present, the signal must be padded to the DC value to avoid corruption of the frequency spectrum.

3.5 Custom cases

In addition to the predefined cases outlined previously, the software tool can read CSV files as well as *.wav files. The user can specify the window length and observe the aliased signal, frequency spectrum and reconstructed signal, as shown in Fig. 2.

REFERENCES

Oppenheim, A. V. and Ronald, W. S. and Buck, J. R. (1998). Discrete-Time Signal Processing, 2nd edition. Prentice Hall.
Martí, J. R. (2002). EECE466 Digital Signal Processing Systems, Course notes.

AUTHOR NOTES

The FFT Tutor program can be obtained free of charge from the authors for private and commercial use.