A Brief Introduction to Noise and Data Filtering

L. Weinman
Schneier/Weinman Consultants, Huntsville, AL 35802, USA
e-mail: Lawrence@Weinman.net

ABSTRACT

This brief article examines some aspects of noise and the effects of filters applied to data.

Keywords: signals, data acquisition, noise, data filter, digital filter, pyrotechnic, experimental design

Introduction

Filtering as used herein, is the removal of noise from the signal under examination

Noise, as defined herein, is any portion of a measured signal that is not desired. The noise may result from fluctuations in the event, from mechanical sources that are not being measured, from inherent electronic sources, from unwanted electromagnetic interactions with the electronics or wiring, or just something that is interfering with what the investigator thinks the signal should look like. (However, as someone famous once said—No data is completely useless, it just isn't quite what you had in mind before you took it.)

Background

The fundamental process of data acquisition, reduction, and analysis is as follows:

- A physical event occurs.
- A transducer changes some physical manifestation of the event into a form that may be recorded.
- A record is made.
- The record is examined and useful information is obtained.

In an ideal world, one would be able to directly record every physical manifestation of any event with perfect fidelity, assign correct values, and obtain all the information that is desired. In the real world, however, things are a bit different.

There Are Two Significantly Different Ways of Filtering: Analog and Digital

Analog filtering removes some frequency(s) portion of the signal by means of passive, active, or both passive and active electronic components. A very common analog filter is the "crossover" used in multi-driver speaker systems. In this application various bands of frequencies are either blocked, or passed, to the speakers, because some speakers are better at reproducing low, mid range, or high frequencies. Since the speakers are transducers (electrical energy to sound energy) and since the filters are (usually) comprised of capacitors, inductors, and resistors, one can readily see how the process would work in reverse.

Imagine that there are two sound-energy to electrical-energy transducers (microphones) with one having good linear response at low frequency and the other, good linear response at high frequency. If a sound, comprising both high and low frequencies excites these transducers, one will produce an electrical signal that is a good representation of the low frequency component of the sound, but with a superimposed poor representation of the high frequency component, and vice versa. The filter, in this case would serve to block the high frequency electrical signal produced by the transducer having good low frequency response, and vice versa.

However, in modern digital data acquisition, the main use for an analog filter is solely to block frequencies that are above the Nyquist Limit. The Nyquist Limit is that data acquisition rate which will allow at least two data points per Hz. [a] (A signal having a maximum frequency component of 50 KHz *must* be digitized at a rate of at least 100,000 samples per second, a 25 KHz signal must be digitized at a rate of at least 50,000 samples per second, and so forth), although a higher digitizing rate is desirable. If this is not done, an effect called aliasing[b] may/will occur. This causes the appearance of spurious data and must be taken into account.

Frequently, either the transducer or the associated electronics will take care of this problem. If a transducer, or its signal conditioner, only has a frequency response of 10 KHz one need never digitize the data from that transducer at (much) over 20,000 data points per second. One must also be aware of frequency response non-linearity in transducers as areas of potential problems. As an example, some transducers may have a stated frequency response of 100 KHz, but a resonance at 200 KHz. In this case, one would have to make sure that no frequency over 100 KHz was digitized.

It is very desirable to not introduce any additional analog filtering before digitizing the signal of interest. While it may be nice to show a "clean" waveform on an oscilloscope, once analog filtering has been performed, it is not possible to recover any information, or frequencies, that might have been present in the signal but have been removed by the analog filtering.

The investigator will frequently discover that upon re-examining data that an "interesting" shape in a curve is noticed. If the data, showing the event of interest, has only been digitally filtered, the data may be re-presented with either no, or different, filtering applied in an effort to emphasize/clarify the "interesting" event. However, if the data had been analog filtered prior to digitizing, no frequency that had been filtered may be recovered.

Conceptually, all filtering may be viewed as a series of steps. These steps, in their most fundamental form, are the following:

- transformation of the time domain data to frequency domain data
- reduction of unwanted frequency data
- transformation of the altered data back to the time domain

The simplest, probably best-known, and most intuitive, low pass filters are the basic "square" smoothing methods. Consider the following example given in the Qbasic[c] computer language:

```
Dim Arr1(1000)        ' create two arrays,
                      capable of holding
                      1000 data
Dim Arr2(1000)        'points each
For I = 1 to 1000
   Arr1(I) = Data(I)  'fill the first array with
                      the raw data
Next I
For I = 2 to 999      ' this is where the
                      smoothing takes
                      place
   Avg = 0
   For J = -1 to 1
      Avg = Avg + Arr1(I+J)
   Next J
   Arr2(I) = Avg/3
Next I
```

This bit of code performs a three-point average smoothing on the data contained in Arr1, and places the smoothed data in Arr2. One might wish to note that the data contained in Arr2(1) and Arr2(1000) has the value of zero. It is obvious that any odd number of data may be smoothed using the above code with minor modifications, and even numbers with only a bit more of a change.

From this simple "square" smoothing, one may extend to a more complex method in which the data points are weighted. For example:

```
Dim Arr1(1000)        ' create two arrays,
                      capable of holding
Dim Arr2(1000)        '1000 data points each
Dim H(-2 to 2)        'this array will hold 5
                      filtering coefficients
H(-2) = 0.1
H(-1) = 0.2
H(0) = 0.4
H(1) = 0.2
H(2) = 0.1            ' note that the sum of
                      the coefficients = 1
For I = 1 to 1000
   Arr1(I) = Data(I)  ' fill the first array with
                      the raw data
Next I
For I = 3 to 998      ' this is where the
                      smoothing takes place
   Avg = 0
```

```
    For J = -2 to 2
        Avg = Avg + Arr1(I+J) * H(J)
    Next J
    Arr2(I) = Avg
Next I
```

In this case, we have applied a "triangular" low pass filter to the original data.

Most importantly, this introduces the concept of using a set of coefficients that are multiplied with the original data to form a new set of data.

$$FilteredArray(I) = \sum_{j=-n/2}^{j=n/2} DataArray(j+I) \times Coefficient(j)$$

Fortunately, and especially because this author is unqualified to properly explain the theoretical background for this area, workers in the field have written readily available programs to generate useful sets of coefficients. These programs are available as commercial software, shareware, and freeware.[d]

By using suitable sets of coefficients, one may obtain low-pass, high-pass, band-pass, band-stop, and differentiated data sets without ever permanently changing the original stored data.

It should be noted that the above method does not incorporate any output terms as an input to a later step, as with recursive filters. Perhaps the simplest recursive filter that the author is aware of[e] is shown in this example:

```
Dim Arr1(1000)  'create an array, capable
                of holding 1000 data
                points
C1 = .7         'a selected value 0 < C1
                < 1
C2 = 1 – C1
For I = 2 to 1000
    Arr1(I) = C1 * Arr(I – 1) + C2 * Arr(I)
Next I
```

Notice, that in this method, any data (after the first) is dependent on earlier data.

A more exact, and flexible, method to accomplish this filtering is to perform a Fourier[f] transform of the time domain data to the frequency domain, remove the unwanted frequencies, and then perform an inverse transformation back to the time domain. There are capabilities to perform these operations in many data analysis programs.[d] However, they are not so quick and easy to incorporate in small programs written to accomplish some task at hand.

As an example, Figure 1 shows the results of a test intended to measure the force exerted by a small pyrotechnic piston actuator. The data was acquired at a rate of 5E-6 seconds per point. Each of the data presentations includes a baseline and an arbitrary upper reference line. The first waveform shows the "raw" data as acquired. The successive waveforms show the effect of increased (lower frequency – low pass) filtering, while the last waveform shows the result of subtracting the 250 Hz filtered data from the raw data. This would represent only the "noise" above 250 Hz in the raw data, and it appears to be caused by two different excitation modes of the test fixture at approximately 3000 and 200 Hz. These two modes were thought to be caused by the vibration of a large mounting plate for the low frequency component, and a much smaller plate mounted at right angle to the large plate for the higher frequency. If desired this could have been determined by mechanically exciting the fixture, or parts of the fixture, using some small impact device to simulate the shock/impulse caused by the pyrotechnic actuator.

The effect of the digital filter before time "zero" is evident in the 500 and 250 Hz filtered waveforms. This could be eliminated/minimized by prefixing additional baseline data to the waveform prior to filtering it.

Inspection of these waveforms will make clear the importance of specifying how data is to be treated along with a set of requirements for the performance of an item. In this case, perhaps, the "time to function" might be specified to be determined from "raw" or "2000 Hz filtered" data, while the "peak effective force" might be required to be determined from a "500 Hz filtered" data set.

While the use of filtering will always have an effect on the magnitude, phase, and absolute time represented by the filtered data, in practice, and with judicious care, these effects will usually be found to be within an acceptable range when making physical measurements. However, one must always keep in mind the possibility that filtered data, under certain cir-

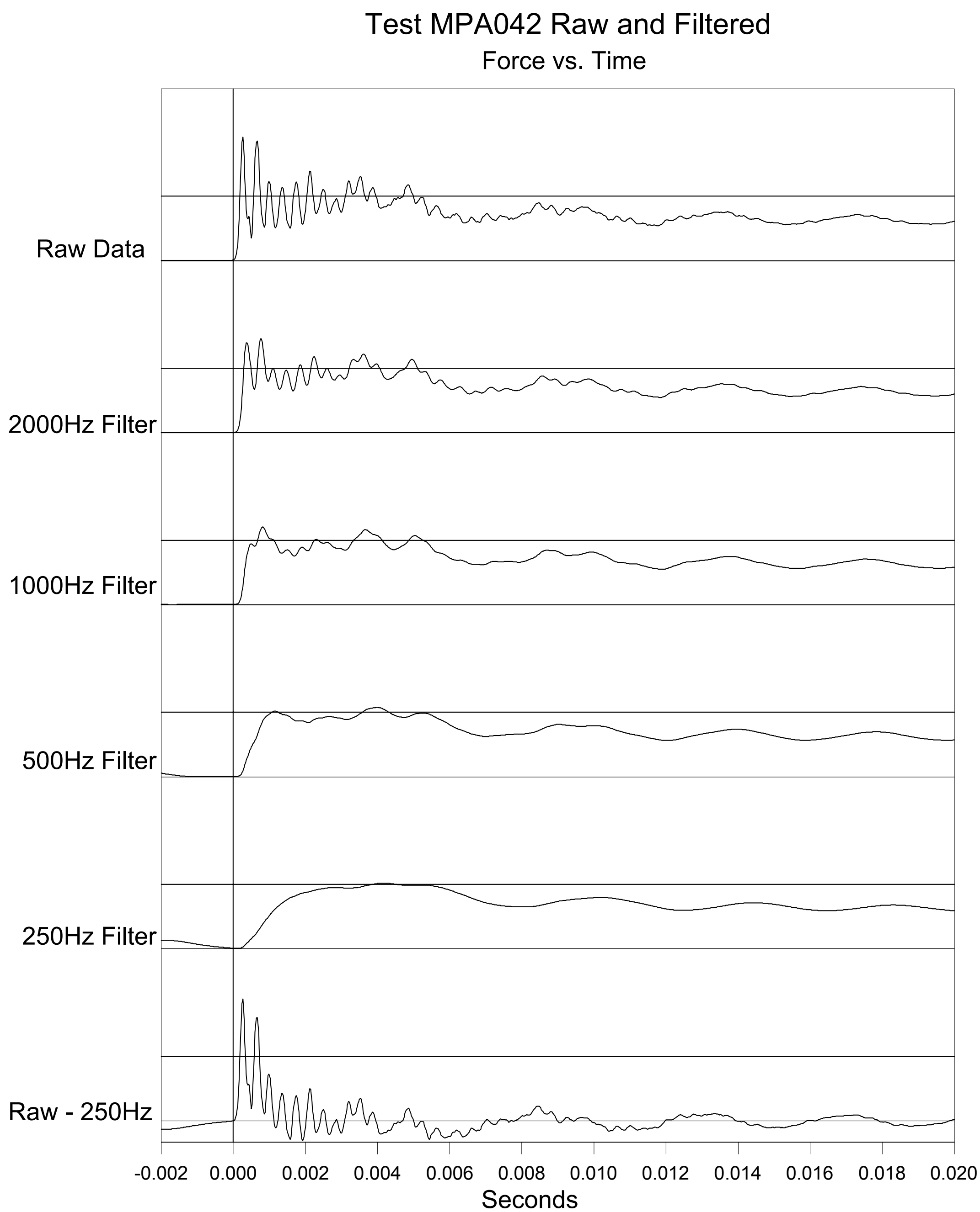

Figure 1. Data Filtering Examples.

cumstances, may not be a sufficiently accurate representation of the actual physical event.

Notes

a. Nyquist Limit: F_s = frequency of sampling, $Y(t)$ has a band limited spectrum

$$F_0 = \frac{\omega_0}{2\pi}\text{, then } F_s > 2F_0$$

b. If sampling is done at a frequency less than the Nyquist Limit, then some high frequency information in the analog signal will be shifted into the lower frequencies giving spurious data.

c. Qbasic can be found in the directory OLDMSDOS on Windows 95 and 98 disks.

d. The author suggests a search of the World Wide Web for finding such software and other available information.

e. Wm. Mattox, personal correspondence.

f. The forward and inverse Fourier transforms are:

$$Y(\omega) = \int_{-\infty}^{\infty} y(t) e^{-j\omega t} dt$$

$$y(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Y(\omega) e^{j\omega t} d\omega$$