



## Enhancing Data Acquisition with Intelligent Oversampling

Roger W. Lockhart

Originally Published In SENSORS June, 1997

*IOS pays dividends in so many applications, especially in terms of noise reduction, that it's difficult to think of an application that wouldn't benefit.*

Did you ever wish you could have your cake, and eat it too? In some data acquisition situations, you may be able to do just that. I'm talking about applications where you need only the maximum (max), minimum (min), or average (avg) value of a relatively high frequency waveform. The data acquisition landscape is replete with examples of this nature. Consider the following:

- Min or max value of a 50, 60, or 400 Hz power waveform
- Min, max, or average acceleration (g) from an accelerometer for vibration studies
- Max (systolic), min (diastolic), and average (mean) of a pulsatile blood pressure waveform in life sciences research
- Elimination of noise on near-DC signals (thermocouples, for example)
- There/not there recordings of high frequency waveforms such as audio, vibration, noise, etc.

In each of these examples, the focus of the data acquisition task is not on the waveform itself, but some component of the waveform represented by its max, min, or average value over a unit of time. Here's a more detailed example:

One of our customers wanted to instrument his dynamometer. He had several parameters to measure including oil pressure, rpm, torque, and engine temperature — all low frequency waveforms. The problem was a fifth parameter, vibration, derived from an accelerometer mounted to the engine block. This signal produced high frequency information compared to the other four and forced some unusual conditions on the measurement approach as a result. He had the following options:

1. Sample all channels at a high rate consistent with the frequency response requirements of the accelerometer.
2. Sample the four lower frequency channels at a slow rate and the fifth high frequency channel at a much faster rate.
3. Determine how often you need to report each channel's value (say, five times per second, or 5 Hz) and select that sample rate.

The first option suffers from data bulge — it generates a huge amount of data, the challenge of where to put it, and the question of what to do with all of it afterward. To attach some numbers, assume that the accelerometer needs to be sampled at a 5,000 Hz rate. Accounting for the other four channels requires a throughput rate of 25,000 Hz. Since data needs to be acquired in this application for as long as 8 hours, a staggering 1.4 Gbyte file would be produced for each session. Those not dissuaded from this ap-

proach based on file size alone should further consider the absurdity of sampling engine temperature 5,000 times each second. The second option is the holy grail of data acquisition — selectable sample rates per channel — and is offered by a handful of data acquisition systems. But be prepared to pay dearly for the feature. Over the sample rates we're describing here, system prices range from \$20,000 to well over \$40,000. Option 3 doesn't seem grounded in reality. How can you sample a fast-moving channel at a slow rate and derive any meaningful information? The key is in the application and the sampling approach you use.

From the perspective of the application, our customer wasn't interested in the actual waveform produced by the accelerometer (see Figure 1). He didn't need a continuous vibration signal he could, for example, transform with an FFT to determine all its frequency components and magnitudes. He simply wanted to know the maximum g's produced by the vibrating engine over a unit of time. And in the context of his application, he wanted the value reported five times per second. In other words, he wanted to accumulate acceleration data for 200 ms (1 divided by 5 Hz), then report just the maximum value.

Since a continuous reproduction of the high frequency waveform is not required, we can exploit a little-known technique called over sampling. Many data acquisition products support dual sample rate capability where the data acquisition hardware samples data at a much faster rate than is reported to the software. Also known as "burst sampling", the technique is most often applied to minimize time skew by sampling all enabled channels at a high rate of speed. Higher burst rates yield smaller time skew errors. But for data acquisition products with on-board intelligence, yet another benefit emerges from over sampling: The ability to evaluate and apply all the data that is typically thrown away. This is the acquired channel information orphaned by the software because it needs data much less frequently than the hardware's burst rate makes it available. I'll clarify the concept, called Intelligent Over Sampling (IOS), using our dynamometer example.

The application assumes that a sample rate of 5,000 Hz is adequate to capture the peak g values generated by the accelerometer. This, the rate of the fastest moving signal with respect to time, forms the basis used to calculate the required burst rate of the hardware. Since a total of 5 channels need to be acquired, the burst rate is 25,000 samples per second (5,000 Hz times 5 channels, see Figure 2). Remember that 25,000 Hz represents the rate that our hardware is continuously scanning our 5 enabled channels regardless of how often the application software requests conversions. In the context of our application, the software will be programmed to acquire data at a rate of 5 Hz per channel (see Figure 2). Calculating a throughput number for it yields 25 Hz (5 Hz times 5 channels). At this point it's clear that the hardware generates excess data at a ratio of 1,000:1 (25,000 Hz divided by 25 Hz). What happens to it?

The correct answer for most hardware is "nothing". They simply take the 1,000th point (what we refer to as the "last point", see Figure 2) and report that value to the software — ignoring the other 999 in the process. But data acquisition hardware products with IOS put the excess samples to work. For our accelerometer channel, they can evaluate the 1,000 samples converted each 200 ms and report the maximum value. This approach yields a stream of data values at a 5 Hz rate that precisely describes the peak envelope of the accelerometer waveform — exactly what the application demanded. And we're achieving this at only a 25 Hz effective sample rate which, over the 8-hour test, consumes only 1.4 Mbyte of disk space — three orders of magnitude less than our first option. But there's more to this story. Turn your attention to the other 4 channels of the application. What advantage does IOS offer them? It doesn't make any sense to capture the maximum or minimum values of these near DC signals. But an arithmetic average calculation can yield significant noise reduction. Every 200 ms the 1,000 values acquired from each of the four channels are averaged to a single data value. The result is a waveform where noise

reduces toward zero to cleanly reveal even the most minor fluctuations in magnitude (see Figure 3).

There are some cautionary notes regarding IOS. The effectiveness of IOS degrades as the software's sample rate approaches the burst rate of the hardware. Selecting the average mode when the hardware and software rates are 20,000 Hz and 10,000 Hz respectively yields a virtually useless 2-point average. Another consideration is to ensure that the software's rate is below the highest frequency component of the signal you're measuring. Measurements off a 60 Hz power line, even when the hardware rate is significantly higher than the software rate, will be distorted when the software rate is greater than or equal to 60 Hz. This situation creates a dilemma when the software attempts to report a min, max, or average value before the signal is able to complete one cycle.

IOS pays dividends in so many applications, especially related to noise reduction, that it's difficult to think of a situation that wouldn't benefit. All DATAQ Instruments' hardware products, as well as some high end data acquisition manufacturers, support IOS.

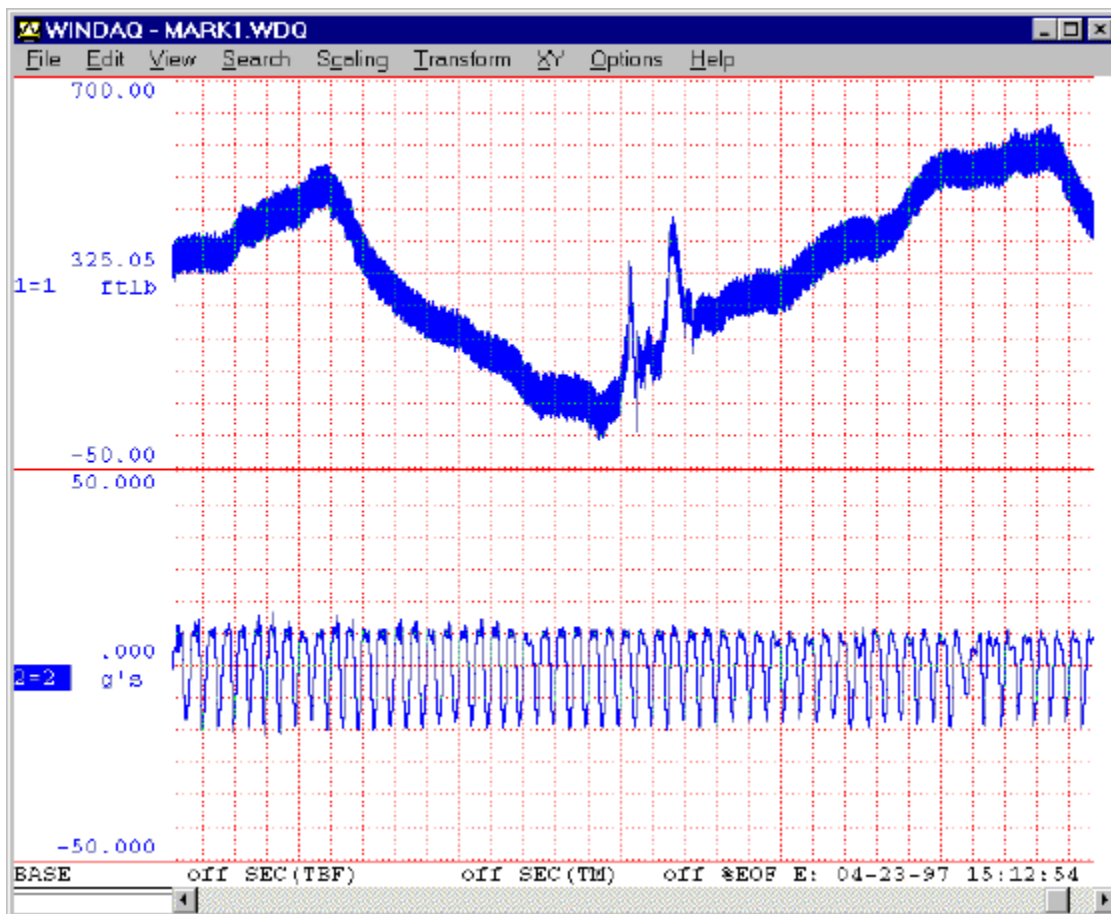


Figure 1. WinDaq data acquisition software screen showing torque (top) and accelerometer waveforms (bottom) sampled at a high rate without IOS.

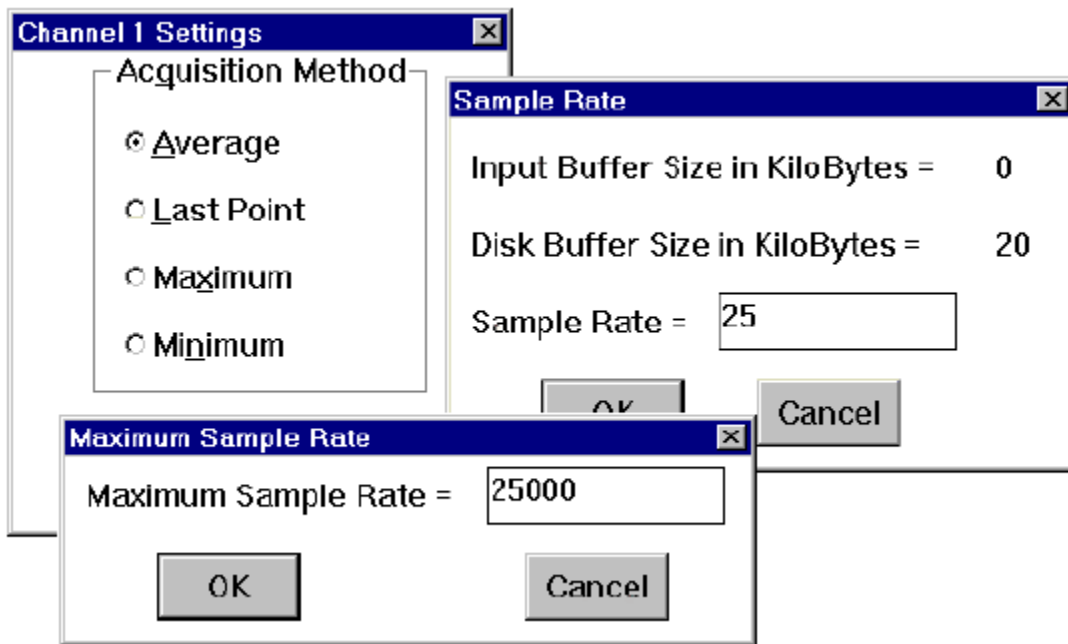


Figure 2. Appropriate intelligent over sampling (IOS) dialog boxes from DATAQ Instruments' WinDaq/Pro data acquisition software package. Clockwise from lower left: Hardware burst rate selection (referred to here as the Maximum sample rate); IOS method (max, min, last point, average) on a per channel basis; Software sample rate selection.

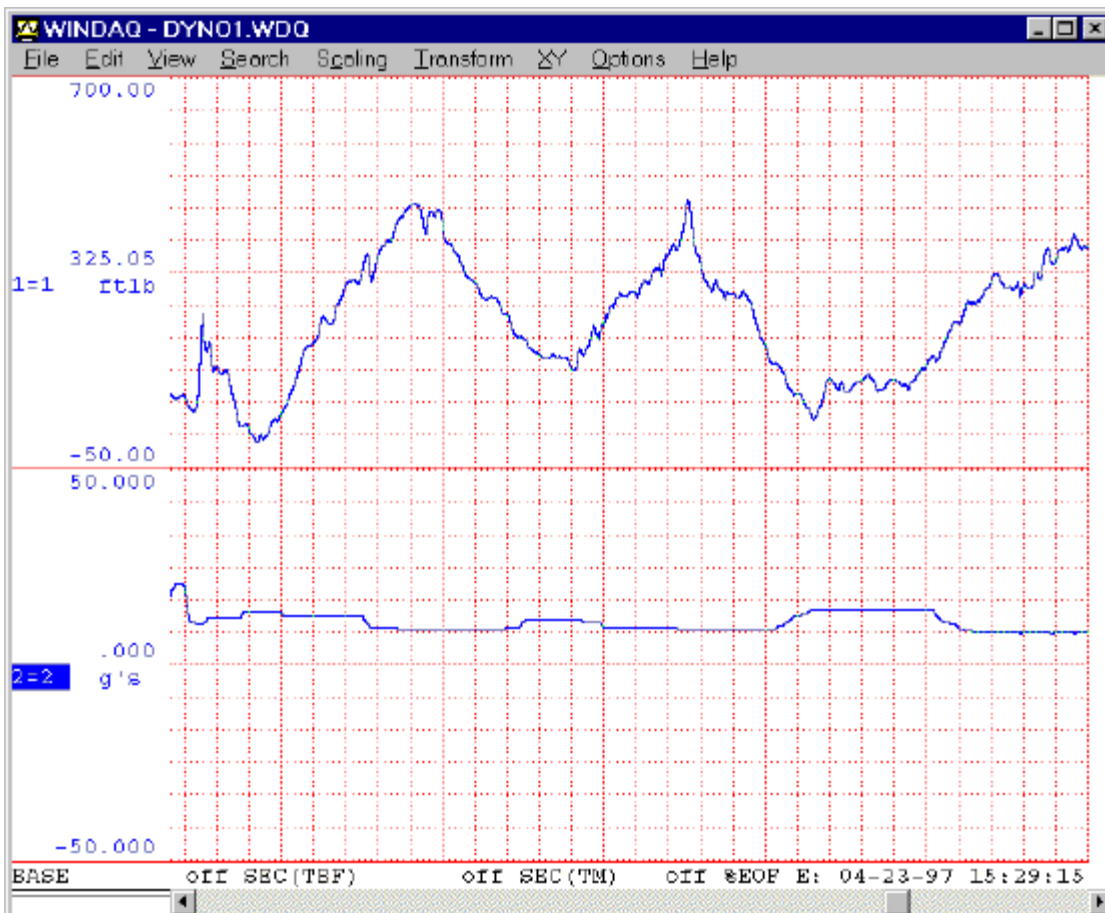


Figure 3. The same waveforms as Figure 1, but sampled at a low rate with average IOS enabled for torque (top), and maximum IOS for acceleration. Note the level of noise reduction for torque, and maximum envelope trending for the accelerometer.