Oregon Research Electronics

Application Note

AL100 Acceleration Offset & Temperature

31677 N. Lake Creek Drive, Tangent, OR 97389 USA
+1-541-928-7869 **engineering@orelectronics.net**

July 18, 2016

Application Note

AL100 Acceleration Offset and Temperature

INTRODUCTION: This application note discusses “offsets” in the reported acceleration value, the effect of temperature on the value offset, and methods of dealing with these factors.

Acceleration Offset

Acceleration Offset is the reading that is reported when there should be zero acceleration. There are two sources for this offset:

- Misalignment of the AL100 with respect to the local gravity vector;
- Internally generated offset error

It can be very difficult to tell which of these is at play in any given situation, and it is very likely that both are. For example, it is common to mount the AL100 vertically (that is, the instrument label is “right-side-up”) so that the unit’s X Axis is pointing downward. This places the X Axis parallel to the gravity vector. Then, the projection of the gravity vector onto the Y Axis and the Z Axis should be zero. But, these projections are zero ONLY if the X Axis is exactly aligned with the gravity vector. If the X Axis is off by an angle of θ, then the projected gravitational acceleration on the other two axes can be as large as $1g * \sin(\theta)$. Thus, if the misalignment is only 0.57 degrees, an apparent offset up to 10mg can be generated in the Y Axis or the Z Axis.

The internally generated offset is inherent within the sensing integrated circuit. It varies from axis to axis and from unit to unit. And, it varies with temperature. One AL100 (sn 1050) shows a variation in the offset of about 20mg over a temperature range of about 8C.

Effect of Acceleration Offset

Offset is often somewhat benign. For example, if one is looking for tree sway information, then an FFT will simply take offset as a very low frequency component of the data. The problem comes when you attempt to plot this data. The very low frequency output “bins” may be much higher in amplitude than the data being sought, and cause the graph to be plotted at a scale that hides the other data. Further, during times of rapid temperature change, the number of bins occupied by this very low frequency output can become fairly wide, making it hard to distinguish between the desired data and the data associated with offset.

This effect also makes it difficult to use code-based techniques to detect tree sway frequency and changes in that frequency.

Dealing With Acceleration Offset

There are several ways in which this problem can be approached.

A. Truncate FFT Output Data: This means simply ignoring the lowest output bins within the FFT data.

B. Correct for temperature: ORE spent significant time working with this approach. It was generally possible to reduce the temperature effect by a factor of 10 to 20. It was discovered, however, that the proportionality between temperature and offset change fails in some circumstances. This appears mostly to happen during very rapid temperature changes. It is as if there is a temperature gradient within the sensor IC and the internal temperature sensor (inside the sensor IC) does not represent the temperature of the sensing elements. The precise cause is not known, but it renders this method less useful than it might otherwise be. There is also a problem with the recorded temperature data: it is only recorded with a resolution of 1C since the manufacturer's specified accuracy is 1C. This was an oversight. Multi-sample averaging reduces the consequences of this, but, at normal timestamp intervals, this may impact the acceleration data too greatly.

C. Filter out very low frequency "acceleration": This method filters the raw data before it is plotted or an FFT is run. It uses a digital filter algorithm known as "high pass filter". The filter greatly reduces all low frequency "signal" components including both offset and offset temperature variation. This is the method ORE has chosen to use in tree-sway analysis software now being developed. It greatly reduces temperature variation as well as the basic offset, itself, whether it arises from misalignment with the gravity vector or from inherent internal offset.

Filtering Low Frequency Components

Filtering of data such as raw acceleration is quite simple, but there are several possible pitfalls. ORE uses a first-order IIR (Infinite Impulse Response) or "recursive" digital filter. This seems to provide quite adequate filtering.

The challenges are to pick the most useful filter algorithm and to pick an appropriate filter frequency.

A. Filter Algorithm

An excellent source for information about digital filtering is "The Scientist and Engineer's Guide to Digital Signal Processing" by Steven Smith. It is available as a printed book and for free download (chapter by chapter) from http://www.dspguide.com (1) This section will rely on Chapter 19, "Recursive Filters" (2), from this text.

The single-pole recursive filter has a very simple equation to define it. It can implement low-pass, high-pass, band-pass, and other filter structures. While the high-pass function is needed, ORE has found that the single-pole IIR high-pass filter seems to behave poorly when the filter frequency is much below the sample frequency. Significantly better behavior was found when a low-pass filter was used, then subtracted from the original signal to create a high-pass function.

At this point, some notation needs to be introduced. Let us refer to the nth member of the data to be filtered (the input) as X[n]. Similarly. lets refer to the nth member of the filter output as Y[n]. The previous input value will then be X[n-1]. The equation that describes a general IIR filter is: (3)

$$Y[n] = a0\ X[n] + a1\ X[n-1] + a2\ X[n-2] + a3\ X[n-3] + \ldots + b1\ Y[n-1] + b2\ Y[n-2] + b3\ Y[n-3] + \ldots \qquad \text{EQN-1}$$

The first-order version of this is simply

$$Y[n] = a0\ X[n] + a1\ X[n-1] + b1\ Y[n-1] \qquad \text{EQN-2}$$

For a low-pass result, the coefficients are (4)

$$a0 = 1 - x$$
$$a1 = 0$$
$$b1 = x \qquad \text{EQN-3}$$

And, the high-pass result comes with coefficients (5)

$$a0 = (1 + x) / 2$$
$$a1 = -(1 + x) / 2$$
$$b1 = x \qquad \text{EQN-4}$$

The factor, x, is determined in both cases from the relationship (6)

$$x = \exp(-1/d) \qquad \text{EQN-5}$$

where “d” is the fractional change in the output, on a sample-to-sample basis, after a step input. This is very closely related to “time constant” for simple single-pole RC or RL filters. A more useful expression for our purposes is in terms of the “corner frequency” of the filter; this term will be discussed in more detail in the next sub section. In terms of the corner frequency, the relationship is: (7)

$$x = \exp(-2 \pi Fc) \qquad \text{EQN-6}$$

where “Fc” is the “corner” or “cutoff” frequency of the filter. Note that Fc is really the ratio of the corner frequency to the sample frequency. It cannot be any larger than 0.5 which is set by the Nyquist Criterion. It can be as small as zero (though such a filter would probably have little practical use).

To realize a high-pass filter using a low-pass one, it is useful to recognize these basic relationships:

1. In a low-pass filter, at low frequencies, Y[n] is approximately same as X[n]
2. In a low-pass filter, at high frequencies, Y[n] is approximately zero
3. In a high-pass filter, at low frequencies, Y[n] is approximately zero
4. In a high-pass filter, at high frequencies, Y[n] is approximately same as X[n]

While these relationships do not constitute a proof, they show that if we do

$$Y'[n] = X[n] - Y[n] \qquad \text{EQN-7}$$

where Y[n] is the input, Y[n] is the output of a low-pass filter implementation, and Y’[n] is a high-pass result. ORE implements this in two steps, the first to create the low-pass output, Y[n], and the second step to create the high-pass output, Y’[n], from the input and the low-pass output. This greatly simplifies debugging.

E. Filter Frequency

The choice of the filter frequency is somewhat arbitrary as there are two competing factors.

- The higher the filter frequency is made, the better the removal of temperature-related artifacts. While these artifacts are mostly diurnal, they can have components on the time scale of an hour, depending on how the AL100 is shaded from direct sunlight.

- The higher the filter frequency is made, the greater the negative impact on tree-sway signals, which are likely to have periods in the range of 10’s of seconds (for a very tall rainforest tree) to a few seconds (for a much shorter temperate ornamental).

ORE provides user selectable filter frequencies starting at 0.1Hz (period = 10 seconds) with 2Hz equivalent samples that result from averaging 10Hz samples in blocks of 5. Remember that the corner frequency specified in EQN-6 is really the ratio of the corner frequency to the sample frequency. If the sample rate is 2Hz, then a 0.1Hz filter frequency would be achieved with a value of Fc = 0.1Hz/2Hz = 0.05.

0.01Hz high-pass filter seems to work very well with 2Hz sample rate data for removing temperature artifacts and the offset or “baseline” generally.

Baseline or offset removal also allows the acceleration to be plotted at a much better scale factor. This, in turn, makes sway events much more apparent.

Conclusion

To reduce low frequency “noise” due to temperature-driven changes in sensor offset, whether that offset is inherent or due to misalignment with respect to the gravity vector, filtering of the data is recommended. An Infinite Impulse Response filter algorithm is presented. This algorithm is currently in use in ORE software.

References

(1) “The Scientist and Engineer’s Guide to Digital Signal Processing”, Steven W. Smith, http://www.dspguide.com

(2) ibid, pp 319-332

(3) ibid, p 320, Equation 19-1

(4) ibid, p 323, Equation 19-2

(5) ibid, p 323, Equation 19-3

(6) ibid, p 324, Equation 19-4

(7) ibid, p 324, Equation 19-5