

How to Choose the Right Analysis Tools for Your Measurement System

Overview

Raw data is not always the best way to communicate useful information. Data transformations like removing signal noise, compensation for environmental effects like temperature and humidity, and calibration for equipment error are needed to help turn raw data into useful data. Producing useful data is a primary outcome of engineering applications, so comprehensive signal processing is a fundamental need for any analysis tool used in data acquisition. This white paper outlines five questions to consider when choosing analysis tools for your DAQ system.

Will I need to analyze my data inline, offline, or both?

Most applications require some form of signal processing but a key decision that you need to make is where this processing takes place: inline, offline, or both.

Inline

Inline analysis implies that data is analyzed in the same application where it is acquired. If your application involves monitoring a signal and changing testing variables based on the characteristics of the incoming data, you should perform analysis inline. By measuring and analyzing certain aspects of the signals, you can make the application adapt to certain circumstances and enable the appropriate execution parameters – perhaps saving the data to disk in the case of an alarm scenario or increasing the sampling rate if the incoming values exceed a threshold limit. To perform inline analysis, your application software must have built-in signal analysis functions or an ability to easily integrate external IP.

The caveat to performing signal processing inline is that it takes time to execute these calculations. If your acquisition application has strict timing requirements, then make sure that your signal processing algorithm does not take too long to execute because you could miss some data while it is performing that operation. While you are developing your application, you can benchmark how long it takes to acquire and analyze your data and make sure that you aren't missing any data points. Something else that could help is parallelizing your code so that one section acquires data while another does the signal processing. This helps by using the multiple CPUs available in most machines, but benchmarking this type of application (for example, producer/consumer architecture) also makes sure it meets timing and data gathering requirements.

Point-by-point analysis is a subset of inline analysis where results are calculated after every individual sample rather than on a group of samples. Such analysis is essential when dealing with control processes featuring high-speed, deterministic, single-point data acquisition. The point-by-point approach simplifies the design, implementation, and testing processes because the application flow closely matches the natural flow of the real-world processes that the application is monitoring and controlling.

With streamlined point-by-point analysis, the acquisition and analysis process can move closer to the point of control because the latency between acquisition and decision is minimized. You can further decrease this acquisition latency by deploying your analysis to field-programmable gate arrays (FPGAs), digital signal processing (DSP) chips, embedded controllers, dedicated CPUs, and ASICs.

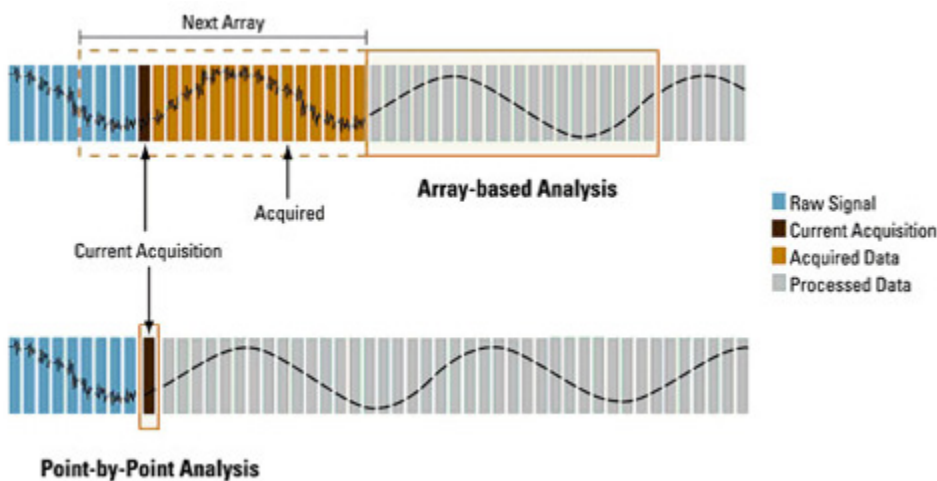


Figure 1.
Array-Based Analysis Versus Point-by-Point Analysis

When you add powerful algorithms and routines to applications, you eliminate the guesswork and create intelligent processes that can analyze results during run time, improving efficiency and iteratively correlating input variables to experiment or process performance.

Offline

Inline analysis is not always the correct methodology when implementing your analysis routines. You may choose to perform offline analysis when you do not need to make decisions as you acquire the data. This involves saving acquired data to disk for unlimited interaction at a later time. Typically, the intent of an offline analysis application is to identify the cause and effect of variables by correlating multiple data sets. Because you perform this analysis after you acquire the data, you are not limited by the timing and memory constraints of data acquisition; such analysis requires only that sufficient computational power is available. This provides several advantages to performing analysis. First, offline analysis offers far greater data interactivity, giving you the ability to truly explore both the raw data and results of the analysis implementation. Histograms, trending, and curve-fitting are all common offline analysis tasks. Furthermore, analysis as a bottleneck for a live acquisition is no longer a concern, considering the amount of time that intense signal processing algorithms can take when operating on large data sets.

There are some applications that need a combination of both inline and offline data analysis. Normally the inline analysis routines are less intensive in these cases, and the offline analysis does most of the heavy lifting of intensive algorithms and data set comparison. An example of less intensive inline analysis would be logic to write data to a file or basic temperature conversion. If your application requires a combination of inline and offline, make sure your analysis tool can adequately support your needs.

Can my analysis tool(s) handle my data (volume, speed)?

A growing concern when choosing data analysis tools is the size and speed of the data it can process. DAQ machines are becoming faster and transducers are becoming cheaper. This means that engineers are collecting more data from more places faster than ever before. If the data analysis tools they use daily can't keep up with these new trends, then engineers have more data than ever, but nothing to effectively analyze it with.

Engineers and scientists are starting to find that their rudimentary data analysis tools do not keep up with their needs. Data analysis tools that were created for financial analysis and not proper data acquisition run into those limits. If you are trying to manipulate or correlate large data sets, then it would be beneficial to use analysis tools that are built for large data sets. Without proper data analysis instruments, you will see that performing any analysis is time-consuming, or you may not be able to analyze at all due to the sheer volume of data.

Talking with vendors about your data size, speed, and analysis requirements helps determine if a data analysis tool is right for you. It is always better to find a tool built specifically for large data sets since this is the trend for data acquisition.

Does my analysis tool(s) offer the functions I need?

If your analysis needs are inline, then examine your application software to ensure that includes built-in or expansion capabilities. If your analysis needs involve offline analysis, then your application software must be able to save data to a format that your offline analysis package can consume.

Most data analysis tool vendors have a well-documented listing of what functions are included with their tool. If you know your specific needs for signal processing, searching the vendor function list works great. If you don't know exactly what you need, look for a tool that has a lot of functions related to your field or application type. Proper data analysis tools have several built-in functions (over 600). While basic and complex math operations are good to have, make sure that there are also functions specific to your area of interest. If your application deals with control, then look for proportional integral derivative (PID) control functions. If your application deals with optical character recognition (OCR), then look for those functions. By having specific functions available (built in or as an add-on), you can be more effective at developing your application because you won't have to spend time creating these functions yourself.

It is also important to remember that your analysis needs often grow over time; consider any analysis requirements you have today, but also verify that your application software leaves room for the inevitable expansion of your analysis needs.

Can I expand my application's analysis options through add-ons?

Investigating whether a data analysis tool has an ecosystem of add-ons is a necessary step. In the early days of software consideration, people picked a tool and were forced to accept only what the vendor put in the product. This meant they were at the mercy of the vendor to add any functionality or features they wanted. As software development progressed, vendors started creating add-ons that people could buy to extend the functionality of their product. This worked well, at the time, because users could purchase the specific functionality they needed; however, they were still limited by what add-ons the vendor offered.

Today, people expect an ecosystem of add-ons to extend their product not only from the vendor, but also third-party affiliates and other customers. This trend is validated by the emergence of "app stores" that extend a product's functionality. When deciding on a data analysis tool, look for one that has a strong ecosystem of add-ons so that you can extend the functionality of your product when you need it. Also look for vibrant communities of users sharing add-ons or IP for the product as these communities often offer added functionality for little to no cost.

Can I integrate my own custom or legacy analysis routines?

Sometimes engineers have proprietary analysis algorithms that simply can't be purchased as add-on software. Additionally, because application requirements change over time, most engineers often invest time and money creating analysis routines or custom IP in older or alternative tools. You should look for a data analysis package that is open to incorporating these external analysis routines. There is no need to reinvent the same functionality in the newer tool when your existing algorithms are validated to work correctly.

Whether you created your analysis routine in another programming language, used a script in an older financial analysis tool, or inherited some configuration file, speak with the vendor to make sure you can incorporate the legacy analysis routine in their data analysis tool. If you can't easily add it, then you will spend a lot of time up front recreating the functionality in the new tool. Modern data analysis tools should be open to using IP created in other environments so that their users can be more efficient.