CHAPTER 7

DATA ACQUISITION AND DISPLAY SYSTEMS

Philip C. Milliman
Weyerhaeuser Company
Federal Way, Washington

1 INTRODUCTION **209**

2 DATA ACQUISITION **210**

3 PROCESS DATA ACQUISITION **211**
3.1 Sampling Interval 211
3.2 Accuracy and Precision of Data 213
3.3 Time-Based versus Event-Driven Collection 214

4 DATA CONDITIONING **214**
4.1 Simple Linear Fit 215
4.2 Nonlinear Relationships 215
4.3 Filtering 216
4.4 Compression Techniques 218
4.5 More on Sampling and Compression 220

5 DATA STORAGE **222**
5.1 In-Memory Storage 222
5.2 File Storage 224
5.3 Database Storage 225
5.4 Using Third-Party Data Acquisition Systems 226

6 DATA DISPLAY AND REPORTING **226**
6.1 Current-Value Inspection 226
6.2 Display of Individual Data Points 227
6.3 Display of Historical Data 228

7 DATA ANALYSIS **230**
7.1 Distributed Systems 230
7.2 System Error Analysis 231

8 DATA COMMUNICATIONS **231**
8.1 Serial Communications 231
8.2 Parallel Communications 232
8.3 Networks 232
8.4 OSI Standard 232
8.5 OPC Standard 233
8.6 Benefits of Standard Communications 234

9 OTHER DATA ACQUISITION AND DISPLAY TOPICS **234**
9.1 Data Chain 234
9.2 Web Programs and Interfaces 235
9.3 Configuration versus Implementation 235
9.4 Store and Forward 237
9.5 Additional Communications Topics 237

10 SUMMARY **238**

REFERENCES **238**

MAGAZINES THAT CARRY RELEVANT INFORMATION **238**

1 INTRODUCTION

The industry has changed significantly since this chapter was first written in the months before 1990. The personal computer has become part and parcel of everyday life. Control systems have become increasingly based on standard systems and interfaces; sensors themselves are often based on just smaller versions of the same operating system as large manufacturing systems. This has tended to change the focus from the technology of data acquisition to the software and systems to support data acquisition.

The trend has been away from requiring the engineer to understand the science of how sensors work and the lowest levels of data acquisition and more toward the engineer understanding the collection, coordination, storage, access, and manipulation of data. With that in

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

mind, this chapter has been updated to focus more on the latter and less on the former. Other chapters in this book cover details of the electronics, transducers, sampling, and calibration.

To control any process or understand what occurs during the life cycle of a process, the system (a human or machine) must have information about what is occurring. In the simplest of control loops, the measured variable must be converted to usable units, comparison in some form to a target occurs, and a response is determined. At the plant level, improvement of plant operation relies upon understanding the relationships between processes within the plant (not only current, but historical), which in turn requires collecting data throughout the plant, characterizing the relationship of the data with other data, storing the data in such a way as to be retrievable in a useful, timely way, and manipulating the data for presentation and hopefully providing an aid to understanding the relationships between processes. In today's competitive environment, focusing on local control and ignoring the interaction between processes, both internal to the plant and external, can be disastrous. If one is not focused on improvement, one can bet the competitor is. Larger corporations, especially, can bring analytical tools to bear to improve local processes, plantwide processes, and their relationships to external influences, such as the supply chain. On the other hand, today's computing tools bring very powerful data acquisition and analysis capability within the reach of the average technician with a little bit of motivation.

Data acquisition and display systems have changed dramatically. Twenty years ago, terms referring to specialized systems such as SCADA (supervisory control and data acquisition) and data loggers were common terms. Now, with the proliferation and broadening role of computer systems and their intrusion into every aspect of manufacturing, many of the features that used to be in specialized instruments and systems are now part of the everyday tools available to anyone with a computer. This chapter attempts to cover aspects of data acquisition and manipulation that may help the engineer better understand issues and give a foundation for using and even constructing tools. The organization is as follows:

- The initial sections cover the nature of data and the acquisition and conversion of data to usable units and includes some discussion of useful display techniques. The discussion attempts to identify issues of which the engineer should be aware and give guidelines on how to manage data.
- The latter sections cover the coordination, storage, access, and manipulation of data. A discussion of pros and cons of different strategies should help the reader understand the trade-offs in system selection and construction. It is difficult to do this without describing specific technologies and brands, but the author has endeavored to level the discussion in such a way that changes in technology will not change the value of the discussion. Time will tell if the approach is effective.

2 DATA ACQUISITION

Data acquisition includes the following: (1) acquiring raw data from the process being measured and (2) converting data to usable units. Included in this section are also some topics of data display closely related to the nature of the data being acquired. Other aspects of data display will be covered in later sections.

In process industries, much of the data are analog in nature, such as pressure, temperature, and flow rate. The values acquired are sampled representations of process data that have a scale and a range, with various issues around effective range and whether values over a range are linear or more complex. When acquired in a data acquisition system there are a number of issues that must be addressed related to how data is sampled; how it is

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

represented in the computer as a digital value but still able to be manipulated as an analog number; and how a continuously changing value can be stored without exceeding the capacity of storage or computation of the acquisition system. Discrete manufacturing still has a number of analog data sources, but a larger proportion involves discrete data, such as motor stops, starts, and pulses. These have their own issues of acquisition and storage and are often related to attributes of the process. The next section deals primarily with process data, additionally covering some discrete data and issues around data collection, representation, and storage.

As businesses begin to broaden the scope of optimization and understand their global processes, the context of the data in terms of product, plant conditions, market conditions, and other environmental aspects has increasingly added discrete data to the set of data to be obtained. The interaction of the process with factors such as which crew is managing the process, which customer's needs are highest priority, legal controls such as environmental limits impacting allowable process rates, operator decisions, which product is being manufactured, grade achieved, and a large number of other factors become important when a company is competing with other companies that have already achieved excellent local control of processes. These data involve less understanding how to deal with continuous data and more with the coordination of data within and between processes. These can be termed manufacturing attributes to emphasize their importance in providing an environment around process data. Later sections deal with manufacturing attribute data and issues around their collection and coordination with process data.

3 PROCESS DATA ACQUISITION

Most modern data acquisition is via digital systems that may have a lower level analog collection mechanism but is now so removed from the engineer that the engineer is only concerned with the digital portion of the system. The ability to use digital microprocessors as building blocks for data collection, the prevalence of computer tools, and the creation of widely available operating systems that operate on small footprints have virtually eliminated the need for analog instrumentation. While the data may be analog in nature, the technology has been developed to such a degree that the engineer decreasingly needs to pay attention to the analog aspects of the data.

A digital-to-analog (D/A) and analog-to-digital (A/D) converter performs the actual processing required to bring analog information from or to the process. While the resulting signal may be digital, it is a representation of a continuous number that has characteristics that, if not understood, can result in erroneous conclusions from data, including missing data, misinterpreting trends, or improperly weighting certain values.

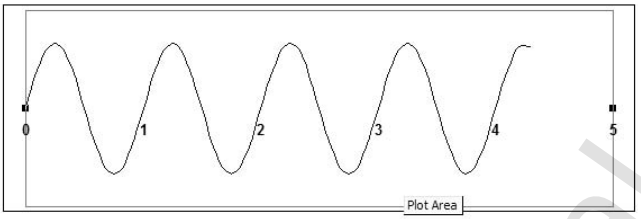
The engineer should be aware of several features of analog data to ensure that the data are used properly. An understanding of sampling interval, scaling, and linearization will facilitate the use of data once collected.

3.1 Sampling Interval

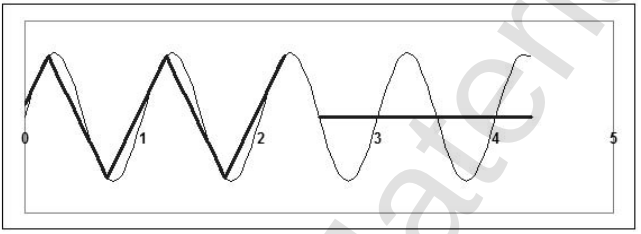
One of the important steps with any data collection process includes the proper choice of sampling interval. As an example of the impact of selection of sampling interval, or frequency, a sine wave with a period of 1-s (Fig. 1*a*) is measured with several sampling intervals. 0.5-s (Fig. 1*b*) and 0.1-s (Fig. 1*c*) intervals both provide different impressions of what is actually happening. The 1-s sampling rate being in phase with the sine wave yields the impression that we are measuring a nonvarying level. The 0.5-s sampling rate yields several

Copyrighted Material

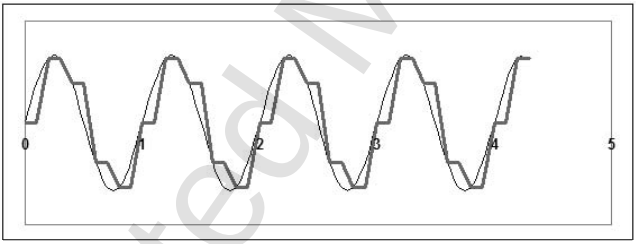
Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com



(*a*)



(*b*)



(*c*)

Figure 1 Sampling: (*a*) sine wave form; (*b*) aliasing of data; (*c*) sampled every tenth second.

different results depending on what phase shift is encountered. This is known as aliasing (see Ref. 1, pp. 122–125). If a 0.1-s period is used, we finally begin to obtain a realistic idea of what the waveform truly looks like.

The sampling interval has a different impact when collecting manufacturing attribute data. With manufacturing attribute data, every change in value has importance. They provide a context to the process that assists with the tieback to business interactions. The values are often coded. The sampling must be close enough to the time of occurrence to allow determination of state in relation to other events. The sampling interval must be fast enough to capture any change in state. Sampling at slower than the change rate of the data will mean lost events—potentially critical in a situation where the count of items processed is important. Sampling at a rate slightly faster than the maximum change rate of the manufacturing attribute data assures that no change will be missed. Another important consideration is to know at what time an event occurred. For instance, if a value changes only infrequently but other related manufacturing attribute values are changing at a faster rate, then the scan rate has to be fast enough to match the fastest change rate of all the related manufacturing attribute data. This is sometimes called the master scan rate, meaning that the frequency of

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

scanning must be fast enough to capture faster events and determine the state of other variables relative to those events. Similarly, if there are related analog data, the scan rate may have to be fast enough to even characterize the curve of the analog data (remember Fig. 1).

The capacity of the target system must also be taken into account. Storing large volumes of data is becoming more feasible as systems increase in speed and power, but the retrieval and organization of those data may become a time-consuming, overly complex task with too much data or poorly organized data. Consequently, even though storage itself is less of an issue, other factors impact how much data are retained and how organized for later retrieval. Later sections examine approaches for organizing and retrieving data.

3.2 Accuracy and Precision of Data

Accuracy and precision are dependent on the sampling interval as well as the resolution of the system (Ref. 2; Ref. 3, pp. 78–80; and Chapter 1 in this volume). When dealing with the A/D conversion process, the step size or number of bits used is critical when determining the system precision and accuracy (Ref. 1, pp. 78–81). Figure 2 illustrates the difference between accuracy and precision of data. Table 1 illustrates the effect the number of bits has on the precision.

This also interacts with range, which will be discussed later, since having an accurate number over a small percent of the desired range would not allow the ability to fully characterize the process. For example, highly accurate readings with 1% moisture resolution over a range from 10 to 20% moisture content would be inadequate if one were attempting to measure moisture over a 5–40% range. When selecting transducers, it is necessary that they have both the accuracy and range needed for the process being observed. When selecting converters, one should be aware of the settling time (governs how often readings can be obtained), resolution of the converter (affects range and detail of measurements), and accuracy of sensor.

Chapter 1 includes some characteristics of transducers, including calibration, and the sampling of data.

One should be aware that an event that has been stabilized in a data collection system may be offset in time, resulting in a potential discrepancy between events or values when values are compared from different sources or from multiplexed data. It should be verified

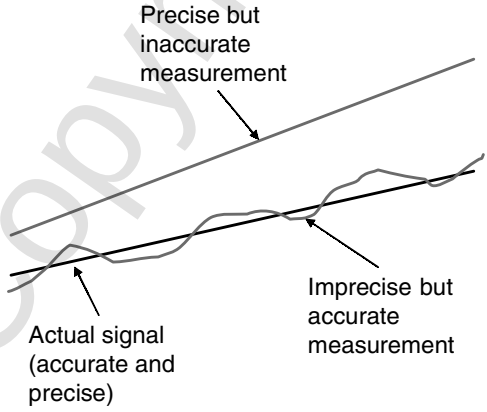


Figure 2 Difference between accuracy and precision.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

Table 1 Relationship between Number of Bits and Precision

Number of Bits	Steps	Resolution on 5-V Measurement	Percent Resolution
8	256	0.01950	0.3900
10	1024	0.00488	0.0980
12	4096	0.00122	0.0240
16	65,536	0.00008	0.0015

that the transducer is collecting the data fast enough to allow one to have relevant times of collection in the data acquisition system. Also one should ensure that the potential relationship between events from different sources and their intended use is understood when considering the speed and accuracy of transducers.

3.3 Time-Based versus Event-Driven Collection

There are two major approaches when collecting data with a general-purpose data acquisition system. In one approach data are collected on a regular frequency based on time, such as once per second. This is easy to institute and it is relatively easy to analyze the data and their relationships after the fact. This approach tends to require more data storage and can make it difficult to identify events or the interactions with manufacturing attribute data. The other approach is event-based acquisition. An event is identified, such as when a package is dropped onto a platform, the time of that event is recorded, and the values of related variables are collected for that time. The sampling rate of the transducers to acquire the other variables may be important, as their values may become irrelevant if too long a time interval has passed after the event has occurred when the related variables are sampled. Batch processes, such as mixing a tankful of chemicals, often have some data collected only at the start and end of the process. Other data may be recorded at fixed time intervals during the batch process. Depending on the needs, the data during the actual reaction may be of great or of little use. The time between sampled events may be several minutes, hours, or even days in length, but the time of the event may be critical, as well as detailed data at the time of the event, resulting in a common tactic of using high-speed scanning to detect the occurrence of an infrequent event. Approaches for combining and analyzing data will be covered in a later section.

4 DATA CONDITIONING

Often the data obtained from a process are not in the form or units desired. This section describes several methods of transforming data to produce proper units, reduce storage quantity, and reduce noise.

There are many reasons why process measurements might need to be transformed in order to be useful. Usually the signals obtained will be values whose units (e.g., voltage, current) are other than the desired units (e.g., temperature, pressure). For example, the measurement from a pressure transducer may be in the range 4–20 mA. To use this as a pressure measurement in pounds per square inch (PSI), one would need to convert it using some equation. As environmental conditions change, the performance characteristics of many sen-

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

sors change. A parametric model (equation) can be used to convert between types of units or to correct for changes in the parameters of the model. The parameters for this equation may be derived through a process known as calibration (Chapter 1 covers much of the process of calibration and sampling). This involves determining the parameters of some equation by placing the sensor in known environmental conditions (such as freezing or boiling water) and recording the voltage or other measurable quantity it produces. Some (normally simple) calculations will then produce the parameters desired. (See the following discussion of simple linear fit for the procedure for a simple, two-parameter equation.) The complexity of the model increases when the measured value is not directly proportional to the desired units (nonlinear). Additionally, as sensors get dirty or age, the parameters might need adjusting. There are a variety of control techniques to assist with compensating for changes in the environment around the sensor, including adaptive control.

4.1 Simple Linear Fit

The simplest formula for converting a measured value to the desired units is a simple linear equation. The form of the equation is $y = ax + b$, where x is the measured value, y represents the value in the units desired, and a and b are parameters to adjust the slope and offset, respectively. The procedure for finding a and b is as follows:

1. Create a known state for the sensor in the low range. An example would be to put a temperature sensor in ice water.
2. Determine the value obtained from the sensor.
3. Create a known state for the sensor in the high range. An example would be to immerse the sensor in boiling water.
4. Determine the value obtained from the sensor
5. Calculate the values of a and b from these values using the equations

$$a = \frac{\text{actual high value} - \text{actual low value}}{\text{measured high value} - \text{measured low value}} \quad (1)$$

$$b = \text{actual low value} - (a \times \text{measured low value}) \quad (2)$$

Figure 3 demonstrates example relationships between measured values and engineering units.

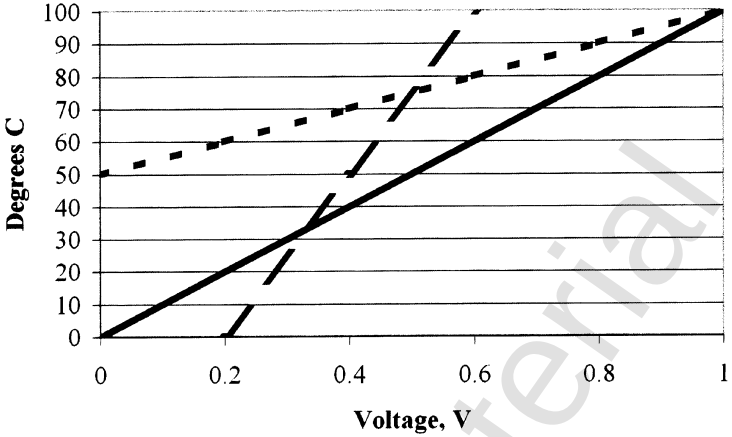
4.2 Nonlinear Relationships

Often, there is not a simple linear relationship between the engineering units and the measured units (Fig. 4). Instead, for a constantly rising pressure or temperature, the measured value would form some curve. If possible, we use a portion of the sensor's range where it is linear, and we can use Eqs. (1) and (2). When this is not possible, we have to characterize the sensor by a different equation, which could be a polynomial, a transcendental, or a combination of a series of functions.

One can imagine several sensors which are linear in different ranges to be used in conjunction to create a larger range of operational data. This variety of formulas should make one point clear: Without an understanding of the basic model of the sensor, one cannot know what type of conversion to use. Many sensors have known differences in output depending on the range of sensed data. Be aware of the effect environmental conditions have on the sensor readings. If the characteristics of a sensor are unknown, then the sensor must be measured under a variety of conditions to determine the basic relationship between the

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com



Legend						
Line	Voltage for 0°C	Voltage for 100°C	Equation for *a*	Equation for *b*	Value for *b*	Equation for °C
_ _ _ _	0	1	$\frac{100-0}{1-0}$	0 – 100	0	$C=100V$
__ __ _	–1	1	$\frac{100-0}{1-(-1)}$	0 – 50(–1)	50	$C=50V+50$
_____	.2	.6	$\frac{100-0}{.6-.2}$	0 – 250(.2)	-50	$C=250V-50$

Figure 3 Relationship between measured values and engineering units.

measured values and engineering units. Some knowledge of the theory of the sensor's mechanism will help to give an idea of which model to use. The development and evaluation of a model is beyond the scope of this chapter, but other chapters in this volume provide assistance.

4.3 Filtering

Even after data are converted to the appropriate units, the data may have characteristics that inhibit understanding the important relationships for which one is looking. For instance, the data may have occasional fluctuations caused by factors other than the process or the process may have short-term perturbations, which are not really an indication of the major process factors.

Filtering is a technique that allows one to retain the essence of the data while minimizing the effects of fluctuations. The data may then appear to be ''smoothed.'' In fact, the terms ''filtering'' and ''smoothing'' are often interchanged. Filtering can occur when the data are still in an analog state (Ref. 1, p. 54) or can occur after the data are converted to digital data (digital filtering). Measurement variability comes from a variety of sources. The process itself may undergo fluctuations that result in variation in measurement but that are only temporary and should be ignored. For instance, if the level of an open tank of water were to be measured but waves cause fluctuations in the height of a float, then the exact value at any given time would not be an accurate reflection of the level of the tank. The sensor itself may have fluctuations due to variability in its method for acquiring data. For instance, the

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

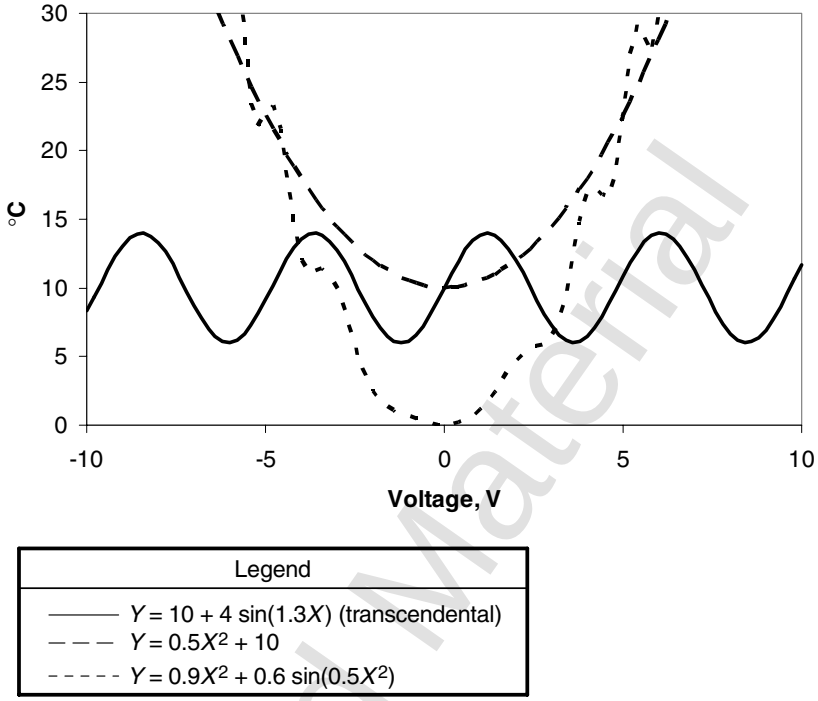


Figure 4 More complex data relationships.

proximity of a 60-Hz line may induce a 60-Hz sinusoidal variation in the signal (measured value). Examples of filtering approaches are as follows and are also given in other chapters in this volume (Ref. 12, p. 538):

(a) Repeated sample average: Take a number N of samples at once and average them:

$$\frac{1}{N}\sum \text{Value}(i)$$

(b) Finite-length average (moving): Take the average of the last N measurements, averaging them to obtain a current calculated value.

(c) Digital filters:

$$y = (1 - \alpha)y_{i-1} + \alpha x_{i-1}$$

The simple average is useful when repeated samples are taken at approximately the same point in time. The more samples, the more random noise is removed. Chapter 1 addresses some of the issues with sampling and the concept of population distribution. The formula for an average is shown in (a) above.

However, if the noise appeared for all the samples (as when all the samples are taken at just the time that a wave ripples through a tank), then this average would still have the noise value. A moving average can be taken over time [see (b) above] with the same formula, but each value would be from the same sensor, only displaced in time. A disadvantage of

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

this approach is that one has to keep a list of previous values at least as long as the time span one wishes to average.

A simpler approach is the first-order digital filter, where a portion of the current sample is combined with a portion of previous samples. This composite value, since it contains more than one measured value, will tend to discard transitory information and retain information that has existed over more than one scan. The formula for a first-order digital filter is described in (c) above, where α is a factor selected by the user. The more one wants the data filtered, the smaller the choice of α; the less one wants filtered, the larger the choice of α. Alpha must be between 0 and 1 inclusive.

The moving average or digital filter can tend to make the appearance of important events to be later than the event occurred in the real world. This can be mitigated somewhat for moving averages by including data centered on the point of time of interest in the moving-average calculation. These filters can be cascaded, that is, the output of a filter can be used as the input to another filter.

A danger with any filter is that valuable information might be lost. This is related to the concept of compression, which is covered in the next section. When data are not continuous, with peaks or exceptions being important elements to record, simple filters such as moving-average or digital filters are not adequate. Some laboratory instruments such as a gas chromatograph may have profiles that correspond to certain types of data (a peak may correspond to the existence of an element). The data acquisition system can be trained to look for these profiles through a preexisting set of instructions or the human operator could indicate which profiles correspond to an element and the data acquisition system would build a set of rules. An example is to record the average of a sample of a set of data but also record the minimum and maximum. In situations where moisture or other physical attributes are measured, this is a common practice. Voice recognition systems often operate on a similar set of procedures. The operator speaks some words on request into the computer and it builds an internal profile of how the operator speaks to use later on new words. One area where pattern matching is used is in error-correcting serial data transmission. When serial data are being transmitted, a common practice to reduce errors is to insert known patterns into the data stream before and after the data. What if there is noise on the line? One can then look for a start-of-message pattern and an end-of-message pattern. Any data coming over the line that are not bracketed by these characters would be ignored or flagged as extraneous transmission.

4.4 Compression Techniques

For high-speed or long-duration data collection sessions there may be massive amounts of data collected. It is a difficult decision to determine how much detail to retain. The trade-offs are not just in space but also in the time required to store and later retrieve the data. Sampling techniques, also covered in other chapters, provide a way of retaining much of the important features of the data while eliminating the less important noise or redundant data. As an example, 1000 points of data collected each second for 1 year in a database could easily approach 0.5 terabyte when index files and other overhead are taken into account. Even if one has a large disk farm, the time required to get to a specific data element can be prohibitive. Often, systems are indexed by time of collection of the data point. This speeds up retrieval when a specific time frame is desired but is notoriously slow when relationships between data are explored or when events are searched for based on value and not on time. Approaches to reducing data volume include the following:

- Reduce the volume of data stored by various compression tools, such as discovering repeating data, storing one copy, and then indicating how many times the data are

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

repeated. There are many techniques for compressing data, covered elsewhere. Zip files are a common instance of compression to make the data consume less storage and to take less time in transmission from one computer to another. Compression tends to increase the storage and retrieval time slightly. Increasingly, file systems associated with common operating systems include compression as a standard option or feature of mass storage. These systems are quite good at compressing repeated data but are less effective when data vary but have a mathematical relationship, such as a straight line between two points.

- Normalize the data. When the developer knows relationships between data, redundancy can be avoiding by normalizing the data—following some basic principles to organize the data in such a way that redundancy is avoided.[5] C. J. Date describes the levels of normalization of data in a relational database.[5] For instance, if a person has several addresses, then one could store the person's name once, store each address, and store the links from the person to the address. While very similar to compression, it relies on the developer identifying and taking advantage of the relationships between data to eliminate redundancy and reduce space. This creates significant effort in planning for acquisition and storage of data. It pays off in reduced storage and significantly improved retrieval and analysis times.
- Eliminate nonessential data. If one is not interested in the shape of a sinusoidal signal, for instance, but only interested in how many cycles occurred during a given time frame, then sampling techniques can be used to characterize the data without having to store significant data.

The engineer or researcher has to make assumptions about how the data will be used and factor those into the acquisition and storage system. A project attempting to discover the relationships between waveforms would require high-frequency sampling and probably time-based storage. A project attempting to record the number of times a boiler went over a certain temperature level might have a high-speed scanning capability but only store those values that were above the temperature limit. An inventory tracking system may have triggers that cause scanning only when some event occurs.

Often, a batch or pallet of product may contain a large number of items. The items can be sampled for some process attribute. The customer may want to know summary statistics about the pallet, but the storage of all the data may not be feasible. In this case, statistical results can normally be derived from summary data. Average, standard deviation, total, correlation, maximum, and minimum are easily calculated from summary, accumulated data*:

Averages. Keep a running sum of data and count of readings:

$$\text{Average} = \text{Sum Of Values}/\text{Count Of Values}$$

Totals. Keep a running sum of data.

Standard Deviation

$$\sqrt{\frac{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}{n(n-1)}}$$

*Standard deviation and correlation from Ref. 6, pp. 473, 477.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

Correlation

$$r = \frac{n \sum x_i y_i - (\sum x_i)(\sum y_i)}{\sqrt{[n\sum x_i^2 - (\sum x_i)^2][n\sum y_i^2 - (\sum y_i)^2]}} \tag{3}$$

Range. Save largest and smallest values.

Median. Find the middle value of a distribution, which requires keeping all values.

The median (the true center of the data) requires the raw data to be calculated. A compromise for depicting the distribution of data without having to store the full details is to store a distribution of the data. For instance, the range of possible important data can be broken into a series of totals, reflecting the count of items that fit into the particular total. A histogram representing the distribution of data can be created from the totals without requiring the full set of original data. In addition, the median can be approximated using this technique. The distribution can also be used to supply data for statistics based on distribution of data, such as the Taguchi loss function (Ref. 7, pp. 397–400).

4.5 More on Sampling and Compression

Rather than just sample the data, why not save all the changed values of the data, discarding values which are the same or within some limits of the previous reading? This really applies best to continuous processes. Quite significant space reduction can be maintained in processes that are slowly changing and have only occasional large upsets. Variations of this technique can provide additional improvements. For instance, rather than just checking to see if the current value is the same as or within some limits from the previous reading, see if it is on the same line or curve as the previous value. This can result in a great reduction of storage requirements at the loss of a slight amount of accuracy in reconstruction. The more flexible the compression technique, the more work must be done to reconstruct the data later for examination. For instance, if the user of the data acquisition system wants to retrieve a data point within data that has been reduced to a line segment, the user or the system must determine which line segment is wanted using the time stamp for the beginning and ending of the line segment interval and then recalculate the point from the equation. This is referred to as a boxcar algorithm. Values that are close to the line segment can be treated as on the line segment if one can afford to lose some accuracy.[8] The formula for a boxcar has to take into account the length of the interval (maximum), the height of the box (how much noise is allowed), and how peak or exception values are treated. For instance, Table 2 presents a set of data with several types of compression applied for data sampled at a constant interval.

In Table 2, the simple repeating-value compression will not lose data but will result in little or no compression if the data are changing value frequently, including having any noise. The boxcar compression technique results in much higher compression for slowly changing data with only the loss of fine detail data (depending on the height of the window). For data that are nonlinear or changing frequently, the boxcar compression method results in little compression. Process information systems often use the boxcar compression method. If data are slow moving with occasional bursts of activity, the boxcar and repeating-value methods can result in dramatic reductions in space required. If data changes tend to be linear, then the boxcar algorithm tends to be superior to the repeated-value approach. For an extreme example, see Table 3.

The raw data would have resulted in 631 data points being stored. The boxcar method would result in 5 data points being stored, less than 1% of the storage required. In the

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

Table 2 Examples of Compression Techniques

Time Stamp and Raw Value		Simple Repeating Value			Boxcar Compression		
Time	Value	Time	Value	Count	Start Time	Start Value	Slope
12:00	1	12:00	1.0	3	12:00	1	0
12:01	1	12:03	2.0	1	12:03	2	1
12:02	1	12:04	3.0	1	12:08	7	−2
12:03	2	12:05	4.0	1	12:11	1	4
12:04	3	12:06	5.0	1	12:12	5	0
12:05	4	12:07	6.0	1	12:16	5	−4
12:06	5	12:08	7.0	1			
12:07	6	12:09	5.0	1			
12:08	7	12:10	3.0	1			
12:09	5	12:11	1.0	1			
12:10	3	12:12	5.0	5			
12:11	1	12:17	1.0	1			
12:12	5						
12:13	5						
12:14	5						
12:15	5						
12:16	5						
12:17	1						

example above, the repeated-value method would have resulted in almost exactly the same compression as the boxcar. However, if there had been a 0.1% slope in data throughout the period, the boxcar would remain the same but the repeated-value method would have resulted in no compression. A deadband (the height of the boxcar) around the repeated value (meaning two values within some small deviation from each other would be counted as the same value) would result in very high compression in the above example. A long ramp-up of the value during that time frame would have further differentiated the two compression methods.

Table 3 Data Compression: Raw Data versus Boxcar

Raw Data[a]		Boxcar Method		
Start Time	Start Value	Start Time	Start Value	Slope
12:00	1	12:00	1	0
12:01	1	15:00	1	1
. . .		15:01	2	3
14:59	1	15:02	5	−4
15:00	1	15:03	1	0
15:01	2			
15:02	5			
15:03	1			
15:04	1			
. . .				
18:59	1			
19:00	1			

[a] Gaps represent no change in data.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

There are many techniques for compression of data. As mentioned above, many rely on assumptions about the underlying nature of the data, such as being continuous data. Where data are directly related to events, more traditional compression techniques which look for repeating patterns in the data may be used. These are typically performed by operating systems and database systems and can therefore be taken advantage of with little or no work on the part of the engineer.

5 DATA STORAGE

In whatever ways data are sampled, collected, filtered, smoothed, and/or compressed, at some point the data must be stored on some media if long-term data are to be analyzed (covered later in this chapter). There are several approaches to data storage that will be discussed in brief here.

5.1 In-Memory Storage

There are normally limitations on how much data can be stored, particularly when low-frequency events have high-frequency data surrounding them that are of interest. For example, if scientists are monitoring Mt. St. Helens for seismic data, it would be prohibitive to capture millisecond data for years while waiting for an eruption. It would be of interest to capture data at high density just before, during, and after each eruption but not in the quiet times in the intervening years. Collecting the millisecond data on many sensors would overflow the storage capability of most systems. There are techniques to store subsets of the data that allow high-density data from constrained time intervals to be stored.

An approach for collecting and later reporting high-density data around an event of interest is to collect the data continuously using the triggered snapshot method. High-speed data are temporarily retained for a fixed time interval or memory capacity, with the start and end time of the data moving forward with time. Older data are discarded as the time range moves past it. This moving window is useful for creating trends and summary data for that interval. The user can be shown dynamic displays that update over time and reflect characteristics of the moving window of the process. Periodically, a set of the data can be extracted to mass storage, especially triggered by some event of interest. An event is recognized by some means (automatic or user generated) that the engineer has preconfigured to cause the transfer of the current instance of the moving window to permanent storage.

The relationship between the trigger and the moving window can be configured several ways, as depicted in Fig. 5. The handling of the data involves moving values through a data array, adding more recent values at the end and pushing the rest toward the beginning—a queue.

The triggered snapshot is particularly useful when knowledge about the sequence of events just before the event of interest can help discover problems. As an example in manufacturing, in sawmills there are often very high speed sequences of events, such as where a board may come out of one conveyor and is transferred to another conveyor and some event such as the board leaving the conveyor occurs. High-speed video can be always in progress, and the detection of the board leaving the system can be used to trigger the transfer to permanent storage of the video. Events of concern can be safety issues and the triggered snapshot method can be used to help eliminate potential life-threatening situations. The triggered snapshot method is particularly useful for discovering the causes of unusual events.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

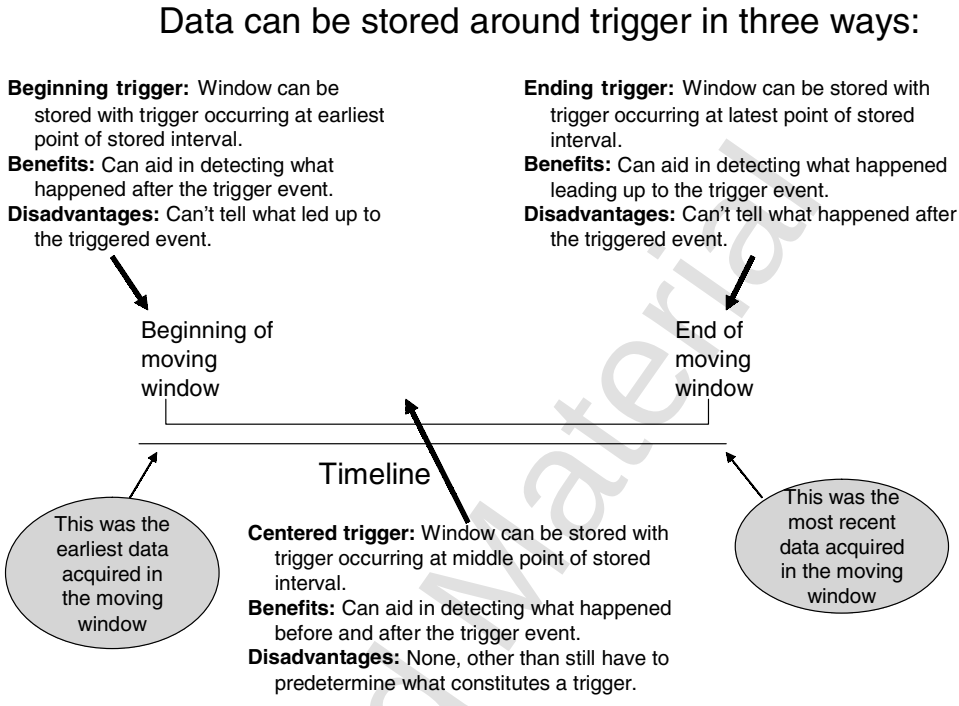


Figure 5 Relationship between trigger and moving window.

This has the advantage of allowing monitoring and analysis of high-speed events and still capturing some data to enable determining some data relationships. This is most useful if the engineer has some idea of what events may yield valuable relationships. It is much less useful when events, triggers, or relationships are unknown or unexpected. Sampling data at slower intervals may serve to allow accidental capture and identification of useful relationships, but the work required to find those relationships is much higher and of questionable probability of success.

As an example, in the sawmill many variables are changing state at high speed. For diagnostic purposes, it is valuable to see a high-speed snapshot of states of photo eyes compared to saw drops, gate openings, and grade decisions. However, the volume of data is normally too great for storage and later analysis. There are some events that are of more importance than others, such as when a gate is opening early or a saw is failing to drop consistently. These can often be recognized and the data captured in the window of time before and after the event can be stored, allowing later analysis of what led to the event and what happened shortly afterward. Some characteristic data can be summarized for each time window, stored, and used later for analysis, such as the number of photo eye changes, number of saw drops, and number of gate openings. More complex relationships can be tallied to aid in diagnostics, such as number of gate openings for grade 2. The more complex the relationship, the more difficult the programming task to ensure capturing the incidence to storage. A typical pattern is to collect process variables that may be of interest, often from a programmable logic controller. As a given problem begins to be identified, additional logic can be added to examine relationships between process inputs and sequences of events,

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

creating a new variable that reflects some attribute of that relationship. The data collection system can store the results of that variable, such as the sum of the number of times it occurred in some larger time interval, allowing it to be low enough in volume to be mass stored for later analysis.

5.2 File Storage

An easy way to store data is in a file, often a comma-delimited file. This is easy to program and can easily be imported into analysis tools such as spreadsheets. It is not well suited to the compression techniques mentioned above because of the complexity of storing and interpreting data. However, for storing records of multiple variables collected at a time interval this can be a very useful technique. An example is shown in Table 4.

In the above example, the filename was chosen so that it would be unique. The date and time were concatenated together, eliminating every invalid character with an underscore. This helps to prevent files from being overwritten accidentally and facilitates store-and-forward techniques described below. Files can be sorted by date or name. Often, they are stored in a directory structure so the number of files in any one directory does not get too great. This speeds up file search in a given directory but can make programs more complex that search for files across directories. An example directory structure is the following:

```
C:
 DataDirectory
  2003
   11—files created during November 2003 are stored here
   12
  2004
   01
   02
   03
```

Archiving files to backup media is easy in this file organization because one only needs to reference the particular directory for the time frame desired. Files can fill up mass media and so either a manual process to check for file limits, back up old files, and delete them must be instituted or a program to provide the same functions would need to be created.

A major deficiency with a file-based system is that when the time range of a search is larger than one file, the analysis can become very difficult. One may be searching for events,

Table 4 Example of Storing Multiple Variables in Files

Filename	ProcessData20040312142305
Data in file:	Timestamp,TemperatureA,TemperatureB,TemperatureC,Rate1,Rate2,Rate3
	2004-03-20 12:34,15,14,15,12,13,13
	2004-03-20 12:35,15,14,14,12,13,13
	2004-03-20 12:36,16,14,13,12,13,13
	2004-03-20 12:37,15,14,12,12,13,13
	2004-03-20 12:38,15,14,11,12,13,13
	2004-03-20 12:39,14,14,11,12,13,13
	2004-03-20 12:40,15,14,10,12,13,13
	2004-03-20 12:41,15,14,10,12,13,13
	2004-03-20 12:42,16,14,10,12,13,13
	2004-03-20 12:43,15,14,09,12,13,13

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

specific time frames that go across file boundaries, or relationships between variables that may not be effectively evaluated within the time frame of one file. The analysis task usually consists of importing a number of files into some analysis tool and then using the analysis tool to look for relationships. This means that the importation process has to include the organizing of data and identifying relationships between events, a difficult task at best. A common tactic is to import data via a script or macro for a given time range, so the user only has to specify a beginning and ending time.

5.3 Database Storage

Database technology has been improving for many years, resulting in database management systems being increasingly the storage tool of choice for data acquisition systems. Database management systems provide organization tools, compression of data, access aids in the form of indexes, and easy access for analysis tools. A special benefit of database management systems is that they allow the combination of discrete data and time-based data collected on different time intervals. Relational databases are now the dominant database management system type. Data are organized in tables. Each table is composed of a set of rows, each row having a fixed set of columns. Indexes are provided to speed access to data. In data acquisition systems, the designer often adds a time stamp column to each row to facilitate retrieval and analysis of data. An example of a simple database is given in Table 5.

The TimeData table contains data that are sampled every second, while the BatchEvent table contains a record for each batch that has occurred. The SQL language is a common language used to examine and extract data in the tables. An example of its power is that a query can be constructed to use the BeginBatchTime and EndBatchTime to extract data from the TimeData table and combine it with related batch events in the BatchEvent table. A sample query to combine event- and time-based data is as follows:

```
Select BatchEvent.Timestamp, Batchevent.BatchNumber,
Batchevent.MixPercent,
Min(TimeData.TemperatureA),Max(TimeData.TemperatureA)
From BatchEvent, TimeData
Where BatchEvent.OutputProductType='BENCH' and TimeData.Timestamp
between BatchEvent.BeginBatchTime and BatchEvent.EndBatchTime
Group by BatchEvent.BatchNumber
Order by BatchEvent.BatchNumber
```

The above query searches for batches that created a certain output product type (BENCH) and reports the maximum and minimum temperatures from those batches. This

Table 5 Example Database Structure

Time Data Table		Batch Event Table	
Timestamp	datetime	Timestamp	datetime
TemperatureA	float	BatchNumber	integer
TemperatureB	float	MixPercent	float
TemperatureC	float	InputMaterialAQty	integer
Rate1	integer	InputMaterialBQty	integer
Rate2	integer	OutputProductType	varchar(50)
Rate3	integer	OutputProductQty	integer
		BeginBatchTime	datetime
		EndBatchTime	datetime

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

can facilitate research, for example, on what conditions lead to the best yield of a particular product.

The ease of performing this operation is a particular advantage of relational databases. There are some disadvantages, including overhead due to the access methods, extra space requirements due to the creation of indexes, and costs and complexity associated with the database management system. Indexes may add as much or more than 100% to the size of a database. Old data must be managed and removed as with any other storage system. This typically is via an automated program, since the structure is not as simple as just looking for the file creation date of a file.

5.4 Using Third-Party Data Acquisition Systems

When data storage is fairly simple, it is not hard to store data in the above-mentioned methods, but when one is using sophisticated methods of data compression, it is recommended that systems that have robust implementations of those methods be used rather than attempting to reinvent the wheel. They can be quite complex to implement reliably. Transfer of data to those systems can occur through a variety of methods, including creation of files that are captured by the other systems, insertion of data into standard interfaces such as OPC or message buses, or insertion into database tables which are monitored by the other systems (often ODBC links). Third-party systems often have software development kit (SDK) interfaces that allow the engineer with some programming skills to store data directly into the system.

Process historians are optimized for storing time-based data. A technique used to provide some relational capability to the data is the following:

- Store related data at exactly the same time stamp (time stored in the database for when the data elements were collected).
- Treat data stored with the same time stamp as being part of the same record.
- Select a set of these "records" based on a time range.
- Search a variable for some attribute value (e.g., having some value or range of values).
- Provide to the display system the values of some related variable in the same record having the same time stamp as the desired attribute variables.

This is functionally the same as performing a relational database query on a set of records in a table, with criteria based on values in some columns.

6 DATA DISPLAY AND REPORTING

There are a variety of ways to reference and display data acquired from a sensor and stored in suitable media. The current value can be inspected, values can be stored for inspection later, values can be trended, alarm conditions can be detected and reported, or some output back to the process can be performed.

6.1 Current-Value Inspection

Often, one wants to see the data as they are being collected. This can be of critical importance in experiments which are hard or costly to repeat, allowing the researcher to react to situations as they occur. As it is collected, each data item will be called the current value for that

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

sensor. Current data are usually stored in high-speed storage (the computer main memory). As new values are obtained, they replace the value from the last reading. The collection rate can vary widely (Table 6).

For instance, detecting the profile at 10-mm intervals for a log moving at 100 m/min requires values to be obtained for each sensor 167 times per second. In continuous processes, data may only need to be acquired once per minute, as in monitoring the level of a large vat. It is useful to remember that human reaction time is in terms of tenths of a second, so displaying data at a faster rate would only be useful if it were easier to program the data. Do not waste time and energy attempting to record data at a high frequency if the only reason is for display to an operator, even if the operator must immediately react to an alert. Human–machine interfaces often show data changes at the time the new value arrives from a sensor. They have display elements that are tied to sensing points. Process historians (data acquisition systems architected for the long-term storage of process data) provide tools to extract data and present the data to the analyst. Their display systems normally provide update tools that automatically refresh the user's display at some display refresh rate (often in terms of seconds, such as 10 s). The data being collected by the historian may be changing faster, and the data may be stored at a faster rate, but the display normally is still refreshed at the standard refresh rate. It is useful to have the time displayed when the value was collected, as there is often a time delay between the collection and the display of data values. This is especially true where the data collection system may be disconnected temporarily from the display system. The data may come back in a rush when the link is reconnected and is useful for the user (and systems performing analysis) to provide a context for what time the data represent.

6.2 Display of Individual Data Points

Display of the data is normally in text or some simple bar graph representation. Other techniques include button or light indicators where color may represent the state of some value or range of values of the current value. Coded values may take the current value and translate it into some form that provides more value to the user such as zero being translated to the string ''FALSE'' on the display.

Often, one can better understand the data being obtained by using an analog representation (Ref. 9, pp. 243–254, and Ref. 10). This involves representing the measured quantity by some other continuously variable quantity such as position, intensity, or rotation. A common example is the traditional wristwatch. The hours and minutes are determined by the position of a line indicator on a circle. A common analog representation for data acquisition

Table 6 Data Collection Rates: Examples

Type of Operation	Time per Event
Discrete manufacturing operations	
Assembly line manufacturing/assembly	0.01 to multiple seconds
Video image processing	0.001 s
Parts machining	0.002–0.02 s
Continuous processes	
Paper machine	1–60 s
Boiler	Several seconds to several minutes
Refinery	Seconds to minutes
Dissolving operations	1 s–20 min

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

is the faceplate. This is a bar graph where the height of the bar corresponds to the value being measured. Often, lines or symbols may be overlaid on the bar to indicate high or low ranges. A frequent indicator is the meter. A needle rotates in a circle with the degrees of movement corresponding to the value obtained from a sensor. Many voltmeters use this technique (Fig. 6). Increasingly sophisticated calculations can be established to translate a flow rate, for example, into a cost number, providing the user with immediate feedback on the costs being incurred by the current process rate.

A common technique for representing trends of current value is to create a simple array and plot it as a trend line on the display. As new values are gathered, the array values are shifted through the array, with old values shifted out at one end of the array while the new values are shifted in at the other. This is a simple technique that provides some of the benefits of data storage without requiring the complexity of actually storing data in mass storage and managing it.

6.3 Display of Historical Data

There are two main issues with display of historical data:

- Selection of the data
- The representation the data will have

Selection of Historical Data

Selection of historical data involves several factors, including time frame and attributes of the data. Identifying a time frame is probably the most common activity in selecting historical data. How the data are updated can be an important consideration when comparing third-party historians.

The time frame is often referenced by a span (such as a number of hours) and a starting point which can be absolute time (e.g., 2004-12-14 16:22:03) or relative time (e.g., −4H for starting 4 h in the past). Another option is to provide an absolute start time and an absolute end time. It is common to have the time frame updating (moving forward with time) if the start point is relative (but check your particular vendor's software for their practice) and to be fixed if the start point is absolute. For example, if the span is 2 h and the start point is

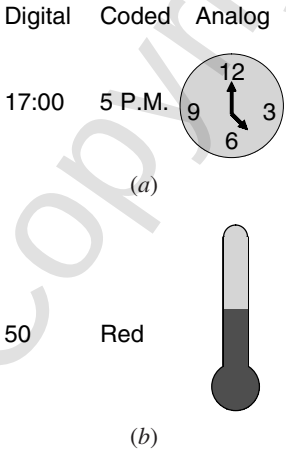


Figure 6 Comparison of digital, coded, and analog data representation: (*a*) time; (*b*) temperature, °F.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

−2 h at 15:00, the starting point on a trend line would be 13:00 and the ending point would be 15:00. Ten minutes later the starting point on the trend line would be 13:10 and the ending point would be 15:10.

For relational data, there is often a desired time frame as described above (to restrict the size of the data to be searched) and some selection criteria based on characteristics of the data itself or of related data, including events. For example, one may wish to find those manufactured units for the past month that were for product *X* and see how many had quality defects. As described above under third-party data acquisition systems, there are techniques for selecting data from process history databases that approach (but do not equal) relational capability.

Representation of Historical Data

The primary difference between current data and historical data is that there are multiple data points, normally with an order defined by the time they were acquired (for process history data) and/or by their relationship to other variables and events (for relational data). Once the time frame and relationships are selected, the data must have some method of representation on a display.

Historical data have a number of potential representation techniques. Multiple data values can be combined into a single number such as average, standard deviation, mode, maximum, minimum, range, variance, and so on. These can then be represented by techniques for individual data elements as described above. The equations in Section 4.4 are examples of summary statistics formulas.

The simplest form to represent historical data is the list. Create a column for each variable of interest. If they are all collected at the same time (the ''records'' described above), then each time data were collected can be used as the first value on the left in each row. The data for each variable of interest can be placed on the row corresponding to its time stamp, similar to that for files shown in Table 4.

Where data were collected with different time stamps, a new row can be created whenever a new value is obtained and values only placed in the column–row combinations where there is a corresponding time stamp between the row's time stamp and the time stamp of the variable in question. While useful, the problem with this is that there are now holes in the data. This list is very useful when viewing a trend line or graphical tool to validate numbers, to verify time stamps, and diagnose problems with collection.

Often, one desires to view the relationships between data and time or other variables. Trend plots typically are used to show the relationship between variables and time. The *X* axis is typically represents the time range and the *Y* axis represents the value of the desired variable. The range of data on the *Y* axis can vary depending on how one wishes to view the data. The maximum and minimum of the data can be used to set the top and bottom of the range of the *Y* axis, respectively. This can have two undesired effects. It may make small movements in data appear to be very large when the range is small. In cases where there are spikes in the data where a value is disproportionately high or low, representing the *Y* axis based on the maximum and minimum could make it difficult to view normal variation in the data. One has to understand the potential use of the data to choose the *Y*-axis scale appropriately.

When more than one variable is shown on a trend chart, the selection of scale of the *Y* axis becomes more complex. If all the variables are representative of the same domain, such as all temperatures, then perhaps the same *Y* axis can be used for all of them. Often, however, the viewer is attempting to compare relative variations in data, sort of a poor man's correlation analysis. In this case it may be useful to have multiple *Y* axes and select the range of

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

each of them such that they represent the range of one or more of the variables being viewed (Fig. 7).

Another approach to compare variation between two variables is to use one variable for the *X* value and the other variable for the *Y* value (an *X*–*Y* chart). This is useful when two variables are related by sample time or some other selection technique that results in a paired relationship between the two variables. The correlation function in Section 4.4 represents the mathematical correlation between two variables and can be used to determine the strength of that relationship. Chapter 1 discusses correlation and the calculation of the line through a distribution of data.

7 DATA ANALYSIS

7.1 Distributed Systems

Distributed systems are a powerful approach to data acquisition systems because they combine some of the best of both stand-alone and host-based systems. The data acquisition portion is located on a small processor that has communication capability to a host computer system. The small system collects the data, possibly reducing some to a more compact form, and then sends the data to the host systems for analysis. The host system can analyze the data when it has the available time to do so. Only the data acquisition portion needs to be very responsive to the process. If the data acquisition task gets too big for the small system, the cost of expansion is limited to moving the data acquisition software to a new computer or splitting it up over several computers and changes to the host computer portion are not required. The major disadvantage of distributed systems is that they suffer from a more complex overall architecture even though the individual parts are simple. This leads to problems with understanding error sources and increases the potential errors because of more parts. Unless the communications are designed carefully, the messages sent between the small systems and the host system may be inflexible, causing increased effort when one wants to change the type of data being collected. Distributed systems may be expensive because of the number of individual components and the complexity required but often fit well with environments where one already has a host computer.

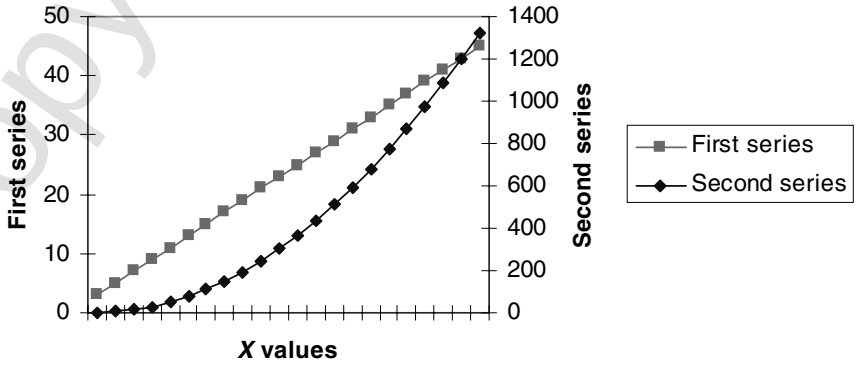


Figure 7 Multiple-axis chart.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

7.2 System Error Analysis

The errors that can occur at different stages in the data acquisition process must be analyzed, as they can add up to make the data meaningless. For instance, one may have very accurate sensors, but by the time the data reach the host computer they might have been converted to integer data or from real to integer and back to real again. This can lead to dangerous assumptions about the accuracy of the received numbers, because each conversion can cause rounding or other errors. It is the responsibility of the person setting up the acquisition system and the analyst to examine each source of potential error, discover its magnitude, and reduce it to the point where it will not have a significant impact on the conclusions to be derived from the data. Use of the filtering techniques described above under data collection can be of use to eliminate random error. It is not within the scope of this chapter to cover system error analysis, but Chapter 1 gives some foundation.

8 DATA COMMUNICATIONS

Data communications are involved in many aspects of data acquisition systems. The communications between the sensing and control elements and data acquisition devices, as well as the communications between the data acquisition system and other computer systems, can be carried out in many ways. This section will cover some aspects of communications, especially as they pertain to computer systems.

8.1 Serial Communications

A serial communication link means that data sent over a communications line is spread out over time on one physical data path. For instance, if a character is sent from a sensor to a computer, each bit making up the character (normally eight bits) will be sent one after the other (Table 7). This is often useful for low-cost, low-speed (usually less than 10,000 cps) rates of data transfer.

Table 7 Time Sequence of Bits Sent over Serial Communication Line

Character "A" is (in bit form)
Bit number 6 5 4 3 2 1 0
Bit value 1 0 0 0 0 0 1
The communications are using the RS232C communications standard and sending the ASCII character A.

Bit Value	Time
Start bit	0 s after start
1 (bit 0 of A)	1/9600 s after start
0 (bit 1 of A)	2/9600 s after start
0 (bit 2 of A)	3/9600 s after start
0 (bit 3 of A)	4/9600 s after start
0 (bit 4 of A)	5/9600 s after start
0 (bit 5 of A)	6/9600 s after start
1 (bit 6 of A)	7/9600 s after start
parity bit	8/9600 s after start
stop bit	9/9600 s after start

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

8.2 Parallel Communications

A serial communication link may not require very many wires, but the time spent to transfer data can add up. A way to improve the speed of communications is to use parallel communication links. This is done by having a number of wires to carry data. For instance, sending the same "A" over a nine-wire bus would only require one transfer (Table 8).

8.3 Networks

Ethernet with Transmission Control Protocol/Internet Protocol (TCP/IP) has become the dominant communications network protocol for data collection. There are proprietary process control and data acquisition networks that serve special purposes, but Ethernet has proven to be versatile for everything from office communications to data collection from smart sensors. Many computers can be connected to the same network segments. The use of switches and routers provides ways to isolate and limit communications to improve performance and security. Firewalls provide filters and protection for entire classes of messages and sources.

While Ethernet cannot guarantee delivery (being based on a collision detection and retransmit strategy), it has been shown to provide excellent response to moderate network activity. Communications speeds are regularly being improved to provide an even greater range of applicability.

8.4 OSI Standard

The International Standards Organization has developed a set of standards for discussing communications between cooperating systems called the Open Systems Interconnect (OSI) model (see Table 9). This defines communications protocols in terms of seven layers.[11] While

Table 8 Time Sequence of Bits Sent over Parallel Communications Interface

Character "A" is (in bit form)
Bit number 6 5 4 3 2 1 0
Bit value 1 0 0 0 0 0 1
The communications are using a hypothetical nine-line parallel communications bus sending the ASCII character A.

Bit Value	Time
Start bit	0 s after start
1 (bit 0 of A)	0/9600 s after start
0 (bit 1 of A)	0/9600 s after start
0 (bit 2 of A)	0/9600 s after start
0 (bit 3 of A)	0/9600 s after start
0 (bit 4 of A)	0/9600 s after start
0 (bit 5 of A)	0/9600 s after start
1 (bit 6 of A)	0/9600 s after start
parity bit	0/9600 s after start

If the bus could handle the same rate of change of bits as the serial interface, then the next character could be sent 1/9600 s after the first character (the A).

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

Table 9 Open Systems Interconnect Model

Layer	Principle	Example
7. Application	Application	Millwide reporting
6. Presentation	Display, format, edit	Convert ASCII to EBCDIC
5. Session	Establish communications	Log onto remote computer
4. Transport	Virtual circuits	Make sure all message parts got there in order
3. Network	Route to other networks	Talk to Internet
2. Data link	Correct errors	Send character downline
	Synchronize communications	Send Ack-Nak
1. Physical	Electrical interface	Wire and voltages

not providing for specific interface protocols, the OSI model has had a significant impact on communications because it has provided a framework for compartmentalizing aspects of communications to allow the handoff of information from one device to another in a standard way. For instance, the transmission of data from one media type to another (such as copper wire to fiber to satellite to copper wire and then to wireless) is a result of standards enabling the seamless transfer of messages in a way that is transparent to the user.

8.5 OPC Standard

A recent standard of use in manufacturing is the OPC (OLE for process control) standard, which provides for a standard way of communicating with process equipment. It is sponsored by the OPC Foundation and originated as an extension for process control from the Microsoft OLE functionality.[4]

Functions provided by OPC include ability to browse the variable database of a device and monitor data on demand or when events occur. The capability of OPC has been expanded to work with Web communication methods such as XML and cover complex data such as record structures. The power of OPC is that data from an instrument can be available via a standard network interface so that any data acquisition program that uses the OPC interface can gain access to any OPC device. The need to know the protocol of each device or adhere to the wiring of specialized communications or use custom database access methods is eliminated through the use of a standard protocol. Multiple programs can be simultaneously monitoring the same piece of data, performing different functions at different time intervals, as events occur. The last point is particularly significant, because much of the work of data acquisition systems is spent in polling for changes in data or otherwise attempting to determine when an event has taken place. A program can subscribe to an OPC item and it will be notified when the item changes value, reducing the complexity of monitoring data dramatically.

As an example of the power of this approach, consider the following example (Fig. 8). A device can collect the identification from a unit of material such as unit number, color, and manufacturing date and make it available via OPC. As the unit is processed in a manufacturing center, another device collects defect counts and makes the unit number available via OPC. A human–machine interface program can monitor both sources with the same interface protocol and software and display it live for an operator to see. Simultaneously, another program can collect the defect counts and summarize them into totals. Yet another program can monitor the totals and wait for the unit number to change, triggering a transaction to a database or an email if there was a problem. The power of the OPC interface is that it provides real-time access to the data from each of the sources and multiple programs

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

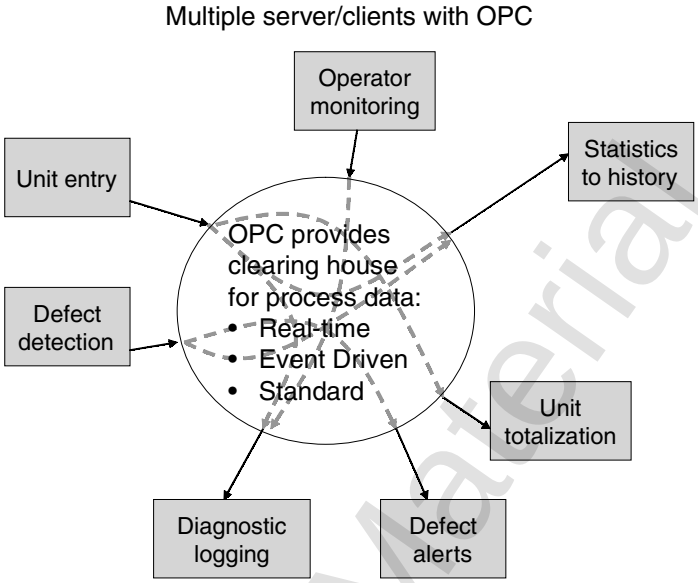


Figure 8 Example use of OPC communications.

can monitor the same OPC sources to perform work. Diagnostics can monitor the same data to evaluate system processing, downtime, or quality issues. Other programs can sample and store data to log files or diagnostic files for further analysis. All can be operating in parallel with no need to understand the internals of the other programs, via a standard interface and standard protocols that support asynchronous delivery of data.

8.6 Benefits of Standard Communications

When implementing data acquisition and display systems, the ability to communicate in a standard fashion can play a large role in the cost of the system. This is realized in a variety of ways:

1. Different sensors can be connected to a system without having to buy a whole new system.
2. Data can be sent to other systems as needed for further processing.
3. As technology or need changes, portions of a system can be mixed and matched.
4. Increased competition from vendors tends to bring prices down.
5. A standard will have many people using products based on the standard, resulting in more vendors and greater availability of parts with a greater variety of options.

9 OTHER DATA ACQUISITION AND DISPLAY TOPICS

9.1 Data Chain

As materials and parts move through a manufacturing operation, the data collected are separated by time and type of data. Combining that data together presents a number of chal-

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

lenges. One can consider the first piece of data collected about some object to be the beginning of a chain of data and each successive data acquisition point in the manufacturing process to be a link in that chain until the end of the chain, where the last piece of data about a manufactured item is collected. Often, the steps of the manufacturing process may proceed from raw material to some intermediate work-in-process unit, to some other step that may be time based, then to some other step that may be finished-unit based. Many varieties of the above exist. Each link can represent one of the following:

- Data collected from a start time to an end time
- A set of attributes about a particular manufacturing unit with an associated time of processing

Often referred to as the genealogy, the steps can be linked together through:

- Some assumptions based on time stamp relationship of one link back to the previous one
- Recording of units that entered or left the time-based portion of the process and the beginning and ending of the entry time
- Some assumptions about the mixing of elements of the manufactured item

Using the techniques described above for combination of time-based and event data, a set of data for the whole life cycle of a manufactured item can be created. Where mixing occurs, the data will be less accurate but may provide clues as to the factors that went into the final product, such as proportions of additives. As an example, Table 10 shows various queries that can be combined to provide one picture of all data sources for a manufactured unit.

Starting at the finished-good item, the batch ID of the previous step acts as a link into the range of time data in the previous step. If the list of items broken down is retained from the prior step, then those can be used to link back to the previous time frame.

Depending on the amount of mixing, the results will be more or less indicative of what actually happened. The smaller the batch sizes, the easier the tracking back to source data will be. Time lags between process steps can dramatically impact the ability to assume when the raw materials for a particular item were processed.

9.2 Web Programs and Interfaces

Web interfaces have improved to the point that user interfaces can be written in Web browser screens. This eliminates the effort and organization required to deploy code across a company. The application is written for a Web interface. When the user uses a Web browser to access the page, functionality is downloaded to the user's page or is executed in such a fashion as to obtain the results of a query and transmit a data page to the user. The developer does not have to get involved in the process of manually installing software on the user's machine. This reduces the demand on the desktop computer and allows the developer to make a change and have it proliferated to all users when they next reference the Web.

9.3 Configuration versus Implementation

As a general rule of thumb, third-party data acquisition and storage systems provide configurable tools for acquisition and display. For simple applications which do not require great flexibility in program functions, such as generating alarms, unusual graphics types, control of the process, or integration into larger systems, it is appropriate to use these, often simple,

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

Table 10 Data Chain Sample Queries

State in Process	Important Data (Typical)	Example Data	Simplified Query to Combine Data with Previous Step
Raw material inventory	Identifiers: raw material batch ID Data: supplier, quality, time in inventory	Lot 1: 5% rejects; lot 2: 7% rejects	
Intermediate goods processing	Identifiers: raw material lots consumed; raw material batch ID; start and end time of entry to process; time delay through process Data: piece count, rejects, downgrades, process characteristics such as rate, temperature, modifications to materials	3:00–4:00 raw materials from lot 1; 1000 pieces; 10 rejects; 200 degrees 4:00–5:00 raw materials from lot 2; 1045 pieces; 14 rejects; 205 degrees; :10 residence time in process	Use raw material batch ID to match to raw materials characteristics; material run at 4:00–5:00 had 7% rejects when delivered to plant
Intermediate package creation	Identifiers: raw package ID, start/end time of creation Data: piece count, package dimensions	Package PR1 created: 3:30–4:10; 100 pieces; package PR2 created: 4:10–5:10; 150 pieces	Use time of manufacture, time lag through process to identify characteristics from intermediate process and raw materials lots; package PR1 was processed at 200 degrees, was created from lot 1 and was from a lot that had 5% rejects detected when delivered to the plant
Intermediate inventory	Identifier: intermediate package ID Data: location in inventory	Package PR1 location warehouse 1	Use ID of intermediate package to match to package at intermediate package creation; package PR1 had 100 pieces
Finished goods processing	Identifer: finishing batch ID, intermediate package consumed, start/end time of consumption; time delay through process Data: piece count, rejects, downgrades, process characteristics such as rate, temperature, modification to materials	Intermediate packages consumed at breakdown: 5:30–6:00 PR1; 95 pieces; 400 degrees 6:00–6:30 PR2; 147 pieces; 390 degrees 6:30–6:40 PR3; 25 pieces; 395 degrees 6:40–7:00 PR4; 40 pieces; 410 degrees Typical residence time :15	Use ID of intermediate package to match to intermediate inventory; in this example, package PR1 came from warehouse 1, lost 5 pieces in consumption
Finished goods package creation	Identifiers: finished package ID; start/end time of creation Data: piece count, grade, package dimensions, customer order	Finished package PF1 created from 6:00 to 6:30; 40 pieces; prime grade; order PO5670	Use time of manufacture, time lag through process to identify characteristics from breakdown, and which raw packages sourced this finished package. There may be significant mixing. In this example, PR1 and PR2 would be sources for PF1. PF1 was possibly created at 400 degrees, stored in warehouse 1, lost 5 pieces when loading into finished goods process, was probably processed at 200 degrees, and was probably from lot 1

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

question–answer or menu-type systems. When the system must be very flexible or customized, it may be more appropriate to write a custom program. When considering this approach, be cautious, for the cost of implementing a program is often much higher than expected. For instance, if one wanted to perform simple data acquisition and storage from a sensor, the cost to write a program would probably be higher than buying a small off-the-shelf system and entering the parameters for data collection. Writing a program involves analysis, design, development, debugging the program, and testing of results. The cost of documenting a program is often a large unplanned cost. If the results are intended to be used to make economic or process-related decisions, then the program must be tested carefully. Additionally, maintenance of the program can be quite expensive. Someone must be trained in the technologies used to build the program, the logic of the program, and the installation of the program. Another factor to consider is that costs of improvement of third-party software are borne by many customers and driven by many customers. The net result to the user of this software is that it is normally constantly improving, constantly tested, and maintained by a group of developers whose primary job is software development. One reason to build and maintain software internally is that a company can keep special knowledge within the company and thus maintain competitive advantage.

9.4 Store and Forward

When data acquisition and data storage are on two separate machines, it is important to provide methods to retain data in case the link between systems is broken. Message buses provide automated methods of maintaining a link between data acquisition systems. The developer inserts data into the message bus. If the link between the two systems is broken, the message bus queues up the data messages on the collection machine. When the data storage machine connection is reestablished, the message bus passes on the data to the data storage machine.

When a message bus is not available or feasible, a simplified mechanism can be created where a file representing each sample of data is created. If the data collection and data storage system are linked, then the data storage system monitors the directory of the collector for a new data file. If the data storage system detects one or more files on the data collection computer, then it will process them into storage. If the link is broken, then the files build up until the link is reestablished. A related technique is to store data in a database or similar mechanism on the data collection computer and scan it periodically for missing data from the storage computer. This is particularly useful when connection to the data collection computer is unreliable.

9.5 Additional Communications Topics

When considering transmission media, some points may provide value to the engineer. Fiber-optic cabling is less sensitive to noise than other transmission media. Wireless access points provide increased flexibility in positioning of sensors and greatly reduce wiring costs. Particularly, if one wishes to collect data from sites that may move, such as environmental sampling sites, the costs of wiring and rewiring can be quite significant. Using wireless transmission technology eliminates much of the wiring costs and facilitates moving the sensors from one location to another. Wireless transmission has a set of concerns that must be taken into account by the engineer, including security, since other units can monitor signals (still evolving) and ability to be jammed.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

10 SUMMARY

The tremendous change in technology for data acquisition and display systems since this chapter was first written has driven us to take a different approach than with the first edition. The technologies for data acquisition and display have become more standardized. Engineers are increasingly reliant upon and versed in computing technologies. The combination of data from various sources into an integrated view of the process has facilitated process improvement and leads to competitive advantage.

This chapter has attempted to provide tools and techniques to aid in the acquisition, storage, and manipulation of process data, expanding from the previous edition into techniques to aid in the manipulation of data for integration and analysis.

REFERENCES

1. C. D. Johnson, *Microprocessor-Based Process Control,* Prentice-Hall, Englewood Cliffs, NJ, 1984.
2. P. W. Murrill, *Fundamentals of Process Control Theory,* Instrument Society of America, Research Triangle Park, NC, 1981.
3. B. G. Liptak, ''System Accuracy,'' in B. G. Liptak ed in chief, *Instrument Engineer's Handbook,* 4th ed., Vol. 1: *Process Measurement and Analysis,* CRC, Boca Raton, FL, 2003.
4. http://www.opcfoundation.org.
5. C. J. Date, *An Introduction to Database Systems,* 5th ed., Addison-Wesley, 1990.
6. W. H. Beyer (ed.), *CRC Standard Mathematical Tables,* 24th ed., CRC, Boca Raton, FL, 1976.
7. M. L. Crossley, *The Desk Reference of Statistical Quality Methods,* ASQ Quality Press, Milwaukee, WI, 2000.
8. http://www.aspentech.com/publication_files/White_Paper_for_IP_21.pdf
9. R. W. Bailey, *Human Performance Engineering: A Guide for Systems Designers,* Prentice-Hall, Englewood Cliffs, NJ, 1982.
10. E. R. Tufte, *The Visual Display of Quantitative Information,* Graphics, Cheshire, England, 1983.
11. ''Open Systems Interconnection—Basic Reference Model,'' Draft Proposal 7498, 97/16 N719, American National Standards Institute, New York, 1981.
12. J. D. Wright and T. F. Edgar, ''Digital Computer Control and Signal Processing Algorithms,'' in *Real-Time Computing,* D. A. Mellichamp (ed.), Van Nostrand Reinhold, New York, 1983.

MAGAZINES THAT CARRY RELEVANT INFORMATION

Control Engineering International: http://www.controleng.com/.
Design Engineering: http://www.designengineering.co.uk/.
IEEE Control Systems Magazine: http://www.ieee.org/organizations/pubs/magazines/cs.htm.
Industrial Technology: http://www.industrialtechnology.co.uk/.
Instrumentation and Automation News: http://www.ianmag.com/.
Pollution Engineering Online: http://www.pollutionengineering.com/.
Scientific Computing and Instrumentation: http://www.scamag.com/.
Sensors Magazine: www.sensorsmag.com.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com