The Selection and Installation of Plant-Wide Process Automation Systems
A Practical Point of View

Prokopios Belezinis

Titan Cement Company S.A.
Drepanou Achaias Plant
GR-26500 Drepanou Achaias
GR-26004 Patras
GREECE

Abstract: The introduction of Process Automation Systems has been an important trend in the industrial world for more than a decade.
In response to what is evidently a high and persisting demand, several manufacturers have developed such systems for a wide variety of applications. The characteristics of these systems are most pronounced in the case of plant-wide installations.
This paper will therefore focus on exactly such systems, which will be discussed and compared from the user's point of view.

Dr Prokopios Belezinis received the Diploma in Electrical Engineering from the University of Patras, Patras, Greece in 1980, the degree of Master of Engineering from Carleton University, Ottawa , Canada in 1982, the Ph.D from Imperial College, University of London, England in 1986, and the MBA from the Athens Laboratory of Business Administration, Athens, Greece in 1994. He has been with the Titan Cement Company since 1988 and he is currently the Production Manager of the Patras Plant. Published work includes papers on Spread-Spectrum Communications and Process Automation Systems.

Introduction

The critical review attempted in this paper does not aim at highlighting the advantages or disadvantages of any specific product. On the contrary, the characteristics of a wide range of equipment from various suppliers have been grouped together to provide models suitable for the purpose of a comprehensive introduction.

While it is true that there are no uniformly excellent or disappointing products, the sheer size and complexity of a system often result in some parts of it being better than others. Indeed there are systems which combine, within the same structure, the state-of-the-art components with remnants of a control philosophy now almost twenty years old.

The paper is divided into three parts.

In Part I the Centralized and Distributed Database architectures are presented and compared. New versions of both architectures are also presented. It is shown that in the process of improving their performance, the two architectures are gradually converging to a common pattern. In the course of system development, it is possible that a new version be released, which may in certain respects be actually worse than its predecessor. Two examples are given to illustrate this point.

In Part II the programming requirements of the systems are compared. The trade-off between ease of programming and flexibility is discussed for both system configuration and application programming. Again possible weaknesses of design are noted and used as pointers to problems, which the user may face with in the on-line programming of the system.

In Part III the ambient requirements are outlined, with emphasis on system protection against electromagnetic interference and voltage surges.

I .System Architectures

We discriminate between the available types of system architecture according to the design of the system database. Two options are available, the Centralized and the Distributed Database.

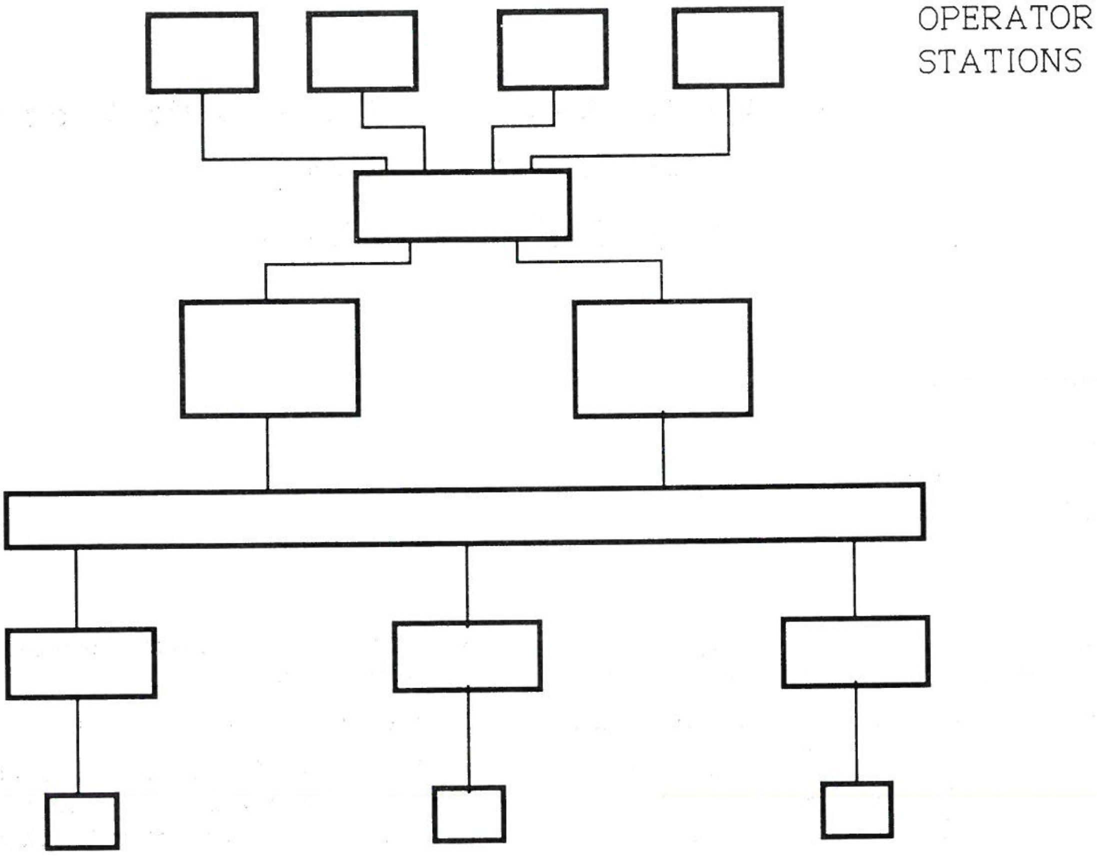

Figure 1. The Centralized Database Architecture

This is a three-layer hierarchical system. The Lower Level (Input/Output-I/O) provides the interface between the system and the process. The Intermediate Level (Process Coordination-P/C) realizes the whole of the automation program. Finally, the Upper Level Computer (Operation & Database-O/D) realizes the Man-Machine-Interface (MMI) and also holds the database of the entire system. It down-line loads each of the subordinate computers with the corresponding software.

This is not a hierarchical system, because each computer has its own database and needs not be down-line loaded. The two lower levels merge into one (P/C & I/O), while the role of the upper level is restricted to the support of the MMI.

Comparison

The two architectures are compared below according to the following criteria : processor redundancy, bus and data line redundancy, computer hardware requirements, computer software capabilities, MMI interface hardware requirements, availability of local operation panels, and system administration requirements.

a) Processor Redundancy:

The user's demand for problem free performance, may only be met by systems offering redundancy for all processors. In essence this means having a second system to be used only sporadically, and then only in part. Considering that the system cost may be 7.5% of the cost of the whole plant, it is immediately obvious that designs offering considerably less than 100% redundancy must be employed.

P/C and I/O processors are very rarely doubled due to the relatively simple tuning of the running program. Should a processor fail the corresponding plant section stopped. Only in cases where such failures cause intolerable disturbance is a redundant processor incorporated into the design. In general, one finds such

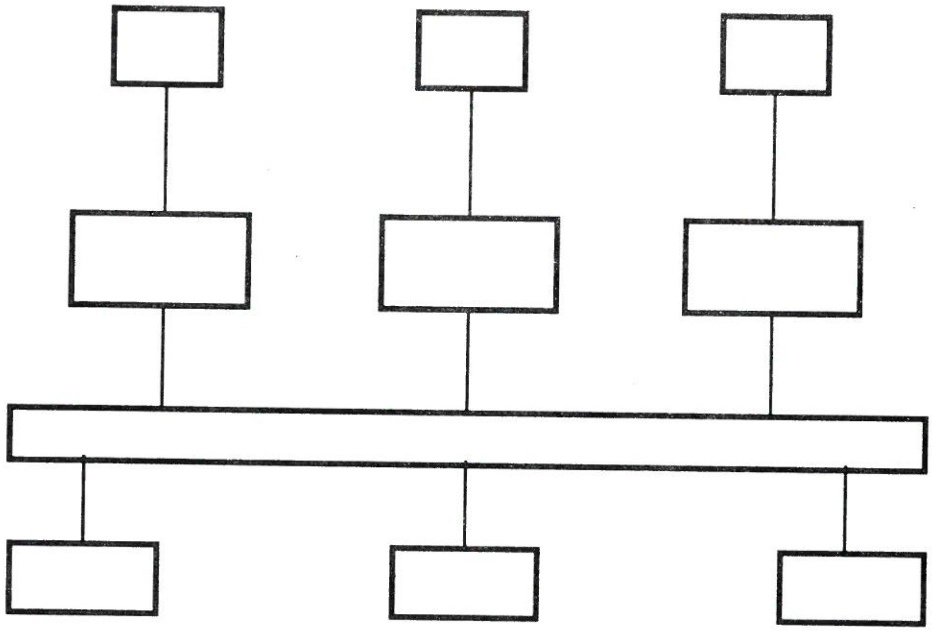

Figure 2. The Distributed Database Architecture

redundant processors in the distributed rather than in the centralized systems.

Contrary to Process Coordination processors, MMI processors require a higher degree of redundancy, because their failure results in the loss of an entire MMI. In the distributed systems this is provided in the form of multiple identical MMI processors, each of which is configured to run the whole installation. In the centralized systems there are usually two O/D computers in Master/Standby configuration monitoring each other with the use of a watchdog utility. In case of failure the switch of the standby to the master state is automatic.

b) Bus and data line redundancy :

Distributed systems are designed on the assumption that each P/C and I/O computer combination should be located in the center of the corresponding plant section. The communication between these combinations and the operation computers is supported by a plant-wide bus, usually a thick-wire Ethernet. Should the bus be damaged all communication would be disrupted and the whole plant stopped. It is, therefore, necessary to have a redundant bus.

The switchover from the damaged bus to the standby is in many systems not automatic, thus asking for a re-loading of the whole network. Having a redundant bus implies the sometimes unacknowledged necessity for a redundant path, one which follows an entirely different route.

Contrary to the distributed systems, centralized systems locate only the I/O processors in the corresponding plant sections and connect each of them with its P/C computer via a dedicated line.

All coordination processors are located in the central computer room and communicate over a thin-wire Ethernet which needs be no longer than three meters. It is highly unlikely that this bus gets damaged, while any damage to an external line affects only one plant section.

c) Computer hardware requirements:

The main difference in hardware requirements between the two architectures is to be found at the Upper Level. In the Centralized architecture it supports the MMI and at the same time contains the database of the entire system. The application software and the tasks to be down-line loaded are built on the O/D master computer. This means that a processor which will perform the editing and building, while simultaneously meeting the I/O requirements of plant operation, is required. In the distributed architecture the support of the MMI is the only task of the Upper Level computers, and the hardware requirements are correspondingly simpler.

d) Computer software capabilities:

Centralized systems usually employ real-time computers which are suitable for a variety of applications. These are multi-user, multi-tasking machines, which offer a lot of utilities not available on the special purpose machines used in most distributed systems.

e) Man-Machine Interface hardware requirements:

Certain centralized designs require a switching unit which connects the operator stations to the current master O/D computer.

f) Local Operation Panels :

These are simplified operator stations which may be supported only by a distributed architecture. They may be useful if the main control room is destroyed but they mainly serve as cheaper operator stations in smaller systems.

g) System administration requirements :

The difference between the two architectures outlined above is also reflected in the administration requirements of the corresponding systems. The centralized system demands a considerably greater amount of administrative work due to the relative complexity of the centralized database as compared to the straightforward simplicity of the distributed database.

The two architectures described above serve as models of systems which have been available on the market since the 1980s. Their chief characteristics from the user's point of view are further listed.

Centralized systems

a)require additional hardware for the MMI switching unit

b)employ throughout their structure expensive families of real-time processors and are for this better integrated

c)the utilities supported by these processors allow for more ease of programming, better MMI and more flexible logging of variables and events.

Distributed systems

a)do not require a MMI switching unit

b)consist of special purpose computers at the P/C level and simple PC like computers at the operation level.

c)this simplicity, though desirable in principle, usually causes relative programming difficulty and MMI unattractiveness.

It was with these concerns in mind that the system manufacturers took advantage of recent developments in computer hardware and software and launched new versions of both architectures.

New types of centralized database systems

The new centralized database systems make use of workstations at the upper level. This eliminates the switching unit, as each computer is itself an operator station. The configuration is shown in Figure 3. For a system with N computers (N workstations) failure of one results in N-1 workstations. In the older configuration failure of one computer results in absolutely no loss of operator stations, but failure of the switching unit results in loss of all operator stations.

Notice that at all times there is one master computer, while the rest of computers have standby status. Each process coordination computer monitors the heartbeat of the master, and transfers the process status information to the standby if the master fails.

The configuration includes a Development computer, which carries both the off-line and the on-line database, and should be able to support program development while serving as a standby.

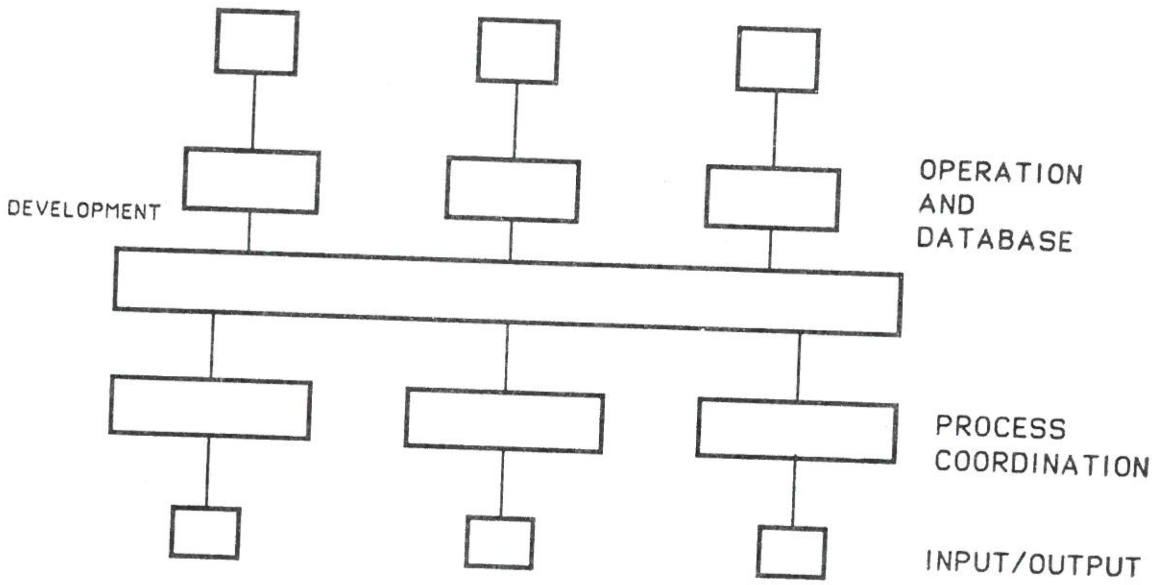

Figure 3. New Type of Centralized Database Architecture

Such systems usually evolve gradually from their predecessors, and are not always available in perfected form. It may happen that a new system appears to be inferior to its predecessor with respect to certain characteristics. For example, the formation of pictures on the screen and the task-building of database and application software may be slower. Both of these problems could be due to the gradual replacement of one type of operating system by another. A new operating system could be selected because it promises to be the standard in the coming years, but it may not at first be available on machines with satisfactory I/O capabilities. Or it may be that the database is generated under the new operating system and the application software under the old one. The discrepancy between operating systems dictates that the old operating system be emulated on the new one, thus causing considerable delay.

As new powerful single board computers appear on the market, it becomes attractive to use them at the process coordination level. This allows for a new type of redundancy, which covers failure of any one of the P/C computers with a single backup.

New types of distributed database systems

The latest versions of distributed database systems aim at improving their user-friendliness at the application programming level. They include a development computer (see Fig.4) from which the task images may be down-line loaded to the respective computers. This is an attempt to mould into an integrated system a collection of loosely connected processors. It would appear that distributed systems are being transformed into centralized database systems.

The introduction of the development computer is accompanied by a host of utilities which uses graphic and linguistic interfaces to facilitate application programming. This seems to put considerable power into the hands of people who may not have the required understanding of the installation, and should be regarded with reservation.

A relatively recent requirement is for the system to support a PC-based MIS. At an elementary level this means the logging of variables and the

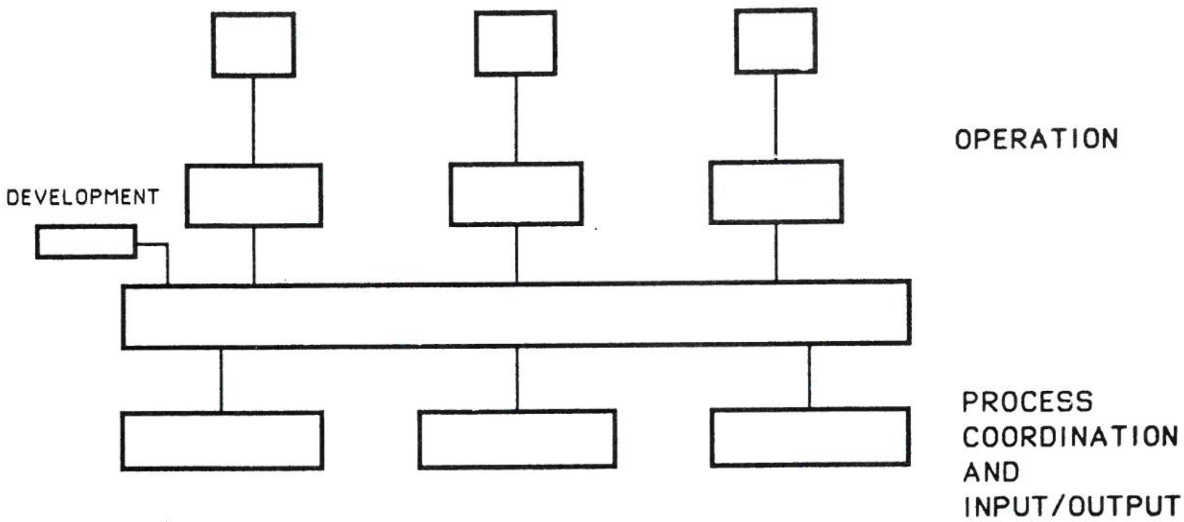

Figure 4. New Type of Distributed Database Architecture

tracing of trend curves. The data are then processed using a suitable package. Frequently the usefulness of these particular tools to the operator is ignored. Full trend displays are reserved for the MIS computer located in the manager's office, and the operator is left to work with crude trend displays.

2 . Programming Requirements

a) System programming

In general we discriminate between the off-line and the on-line programming phase.

The off-line phase comprises the setting-up and building of the database, the writing of the application software (MMI interface, control loops and protocols) and the building of the appropriate task images with respect to the already built database. The newly built database and the application software must be loaded for the on-line phase to begin.

The on-line phase comprises the modification (not the re-sizing) of the database, and the modification of the application software.

There appears to be the following general trend among manufacturers. Centralized database systems, perhaps because they are almost always based on general- purpose real-time computers, leave the off-line definition of the database to the user, under the sole constraint of the number of available addresses. The user is therefore free to allocate more resources to a given plant section if he so chooses. This freedom of configuration is gained at the expense of simplicity, as the user is obliged to build the desired database, and later to build the application software on it.

Distributed database systems have different database configuration requirements. They are based on independent process coordination computers (PLCs), which derive their independence from their preconfigured database. For example if a PLC has 256 registers evenly divided between timer and counter functions, the user cannot reconfigure it to obtain 200 timers and 56 counters if the application demands it. Such limitations do not apply in the case of the user configurable database. It follows that the distributed system offers an advantage over the centralized one, in not requiring any database configuration and building, thus saving time and effort, but sacrificing flexibility.

The re-sizing of the database implies a restart of the computer. This is a major concern in the case of upper level computers. Restarting them means a complete loss of the MMI for as long as it is required for the system to restart and for the new database and application programs to be loaded and, if necessary, down-line loaded to the process coordination computers.

Distributed systems have, as already mentioned, a number of smaller databases, and their restart is faster, other things being equal. The fact remains, however, that a system restart means the loss of the MMI. Some manufacturers automatically stop their process coordination computers whenever there is a loss of the MMI. However, most manufacturers allow for the process coordination computers to run, presumably on the optimistic assumption that the interlocking program offers sufficient protection to personnel and equipment.

Supporting continuous operation and software modifications at the same time is a useful feature, especially for control loops programming, yet not one offered by all manufacturers as one might be led to believe. Ideally, program changes to the process coordination computers which do not require a new database, should be handled on-line, without any interruption. The fact that a large number of relatively cheaper systems offer this basic utility should not make someone consider it a universal feature.

A new and desirable feature is the utility of verification between the application programs in the development and any process coordination computer, as well as programming with symbolic names at all levels of the system.

There is a significant number of systems which do not fully support on-line programming. This is sometimes the case with systems which are expansions of older DDAC (Digitally Directed Analog Control) designs. Originally these systems dealt only with closed -loop control, and sequential control was added later as an expansion. Some manufacturers even use two computers, one for closed-loop and one for sequential control. Others have combined both types of control in one computer. Even so, close scrutiny of the system software reveals the joint between the two parts.

As it is to be expected such systems offer much better modules for closed- loop than for sequential control. The user finds that the modification of difficult closed-loop control programs, even sophisticated adaptive ones, turns out to be easier than rudimentary sequential programming. Indeed such an imbalance of the programming utilities may alert the new user to further problems with the on-line programming of the system.

Some manufacturers side-step the problem by offering redundancy for each coordination computer. The new program is written on the standby computer, which is then switched to master state. Given the generally high reliability of the computers themselves, this kind of redundancy is an expensive and, in most applications, unnecessary feature.

Application programming

Application programming includes the writing of the control loops, the MMI and the protocol programs.

a) **Control Loops Programming** : this part of the programming work sets up the control loops (open and closed), and may therefore be counted among the responsibilities of the electrician. Recognizing this requirement a large number of manufacturers take care of making their systems accessible to electricians and non- specialist programmers in general. Several utilities have been developed in the form of friendly programming "languages". These aim at presenting the program as a virtual circuit or block diagram. The conventional list of statements is also offered, but not by all manufacturers. The availability of such utilities, though desirable in principle, is not free of drawbacks. These are revealed in the phase of on-line programming, which is the work most frequently done by the electrician.

In order to do debugging or diagnostic work the user needs the display of as great a segment of the program as possible on the screen of the programming device. The screen must be automatically refreshed so as to reflect the true state of the signals at all times. Moreover the dumping of new program segments on the screen must be instantaneous. Obviously all these requirements are much more difficult to satisfy when the processor is simultaneously supporting a "language".

In view of this it is surprising that certain manufacturers do not offer the option of statement list programming, and at the same time ignore the above concerns. This results in real programming speed and flexibility being traded-off for an ultimately limited user-friendliness.

b) **MMI programming** : the quality of the MMI depends on the characteristics of the graphics editor supplied with the system. Fully graphic editors are now the norm for better systems at least. However, semi-graphic editors are maintained in many designs, usually to supplement a fully graphic editor. The reason for this is that semi-graphic pictures are dumped on the screen considerably quicker than fully graphic ones. The user must then judiciously combine the two types in each mimic diagram so as to achieve satisfactory response times.

Drawing a picture with a semi-graphic editor will take longer and produce a diagram of poorer quality than with a fully graphic one. However, not all fullygraphic editors are equal. This may be easily demonstrated when modifications on an existing graph are carried out.

c) **Protocols** : monitoring plant performance is greatly helped by the ability of the system to produce protocols with production variables, important statistics, etc. Writing the necessary programs is often beyond the capabilities of the user. In order to solve this problem many manufacturers offer their own preconfigured protocols which, though meant to be helpful, are ultimately an impediment.

3 . Hardware Problems

This section focuses on practical hardware related problems which may occur in the course of the installation or operation of a plant-wide system. Such problems usually arise due to unfavourable ambient conditions and electromagnetic interference.

Ambient requirements:

For the system and the operators to properly function, certain environmental requirements must be observed. These include air quality, temperature, humidity, lighting and vibration.

a) air quality : in many cases industrial processes contain particulate, gaseous or liquid contaminants, the effect of which may be accelerated by temperature or humidity. They accumulate on surfaces and cause electrical short circuits and heat buildups. Components most likely to suffer are those with moving parts, such as hard disks, relays, printers and switches. It follows that the equipment must be installed in rooms free of corrosive and conductive contaminants, and that a cleaning program must be consistently appli ·d.

b) temperature and humidity : the temperature range must not exceed the characteristics of the equipment. The humidity level is particularly important because it must not allow for the accumulation of static electric charges on the equipment or the operators. Condensation should also be avoided

c) lighting : this requirement concerns only the operators. In most cases it should range from 730 lux (operator stations) to 1000 lux (data preparation) on horizontal surfaces (JEAC 5001)

d) vibration : the operation of many plants causes substantial levels of vibration. These may be continuous (process related) or completely random (due to maintenance work). The vibration level allowed varies with frequency. Typically, for frequencies below 14 Hz its amplitude should be lower than 0.5mm. For higher frequencies up to 100 Hz, the acceleration limit is 2G.

Electromagnetic Interference

The frequency of electromagnetic disturbances increases with the degree of miniaturization and packing density of the electronic components. The problem is compounded by voltage levels induced on long cables and by random strikes of lightning which cause voltage surges along the data lines.

In general the manufacturers design systems with good interference rejection performance and give detailed instructions for their correct installation and protection from voltage surges. They are, however, only partially successful, as the increasing range of interference fighting devices testifies.

Although it is difficult to verify the performance of a system before purchase, there are two generally reliable tests of its interference rejection capability. The first is to examine whether metal shielded boards are deployed in the I/O cabinets. The second is to verify the operation of the equipment, especially of the I/O cabinets, when a walkie-talkie is being used very close to them.

Surge protection is more complicated and cannot easily be tested. Although the user can choose among a variety of anti-surge devices, there is no guarantee that the problem will be solved. Indeed, it has been shown in practice that the decoupling offered is not always satisfactory.

4. Conclusions

This paper covered the area of plant-wide process automation systems from the point of view of the user. The centralized and the distributed architectures were discussed and compared both in their original form and in their latest versions. It was shown that the two are converging to a common pattern. The issues of reliability and versatility are of primary importance to the user, and their proxies redundancy and programming freedom received special attention. It was shown that sometimes redundancy is employed to compensate for low programming freedom.

The trade-off between user-friendliness and programming flexibility was also discussed. Finally, the ambient requirements were briefly presented, with emphasis on protection from electromagnetic interference and voltage surges.