THE FEASIBILITY AND ADVANTAGES OF COMMERCIAL PROCESS I/O SYSTEMS FOR ACCELERATOR CONTROL*

Robert A. Belshe, Victor P. Elischer, and Van Jacobson
Real Time Systems Group
Lawrence Berkeley Laboratory
University of California
Berkeley, California

Summary

Control systems for large particle accelerators must be able to handle analog and digital signals and timing coordination for devices which are spread over a large physical area. Many signals must be converted and transmitted to and from a central control area during each accelerator cycle. Digital transmission is often used to combat common mode and RF interference.

Most accelerators in use today have met these requirements with custom process I/O hardware, data transmission systems, and computer interfaces. In-house development of hardware and software has been a very costly and time consuming process, but due to the lack of available commercial equipment, there was often no other alternative.

Today, a large portion of these development costs can be avoided. Small control computers are now available off the shelf which have extensive process control I/O hardware and software capability. Computer control should be designed into accelerator systems from the beginning, using operating systems available from manufacturer. With most of the systems programming done, the designers can begin immediately on the applications software.

Historical Development

The early accelerator control systems consisted of small minicomputers replacing complex manual control functions. Advances in computer hardware and software technology allowed later systems to take advantage of the computational power of the computerized control system to automate and simplify operational sequences. Recent control systems have taken advantage of the steadily decreasing cost of computers and computer hardware to separate the computation and control problems. This separation has given the systems designer the ability to create modular and extensible systems. The three accelerator control systems at Berkeley, LAMPF, and NAL exhibit the evolution in accelerator control technology. A summary of the system characteristics is shown in Table 1.

The Berkeley Bevatron Control System

The Bevatron control system at Berkeley (fig. 1) is possibly the earliest example of an accelerator control system. The systems evolution to digital computer control began when the manual controls of one subsystem (Injector Inflection Control[1]) were replaced by a small mini-computer. The initial success led to the installation[2] of several more mini-computers each dedicated to the control of a particular subsystem. Virtually all of the hardware for these systems was developed in-house. Both the low-level process I/O hardware and the data transmission system have gone through several evolutionary phases resulting in a product that matches the requirements of the accelerator control environment.

The system exhibits a very high control bandwidth because of the dedicated processor per subsystem architecture. However, the information in one processor is not shared by the others, requiring the operator to coordinate the control parameter changes for different experimental setups. The system was not designed to supply the computational capability to support this requirement.

All the programs are written in assembly language using a CDC 6600 cross assembler. The real time control algorithms are for the most part core resident. The operator interface code, however, often consists of many intricate overlays due to the (4k) core restrictions. Consequently, the initial creation of the control pro-

grams and any significant changes require a programmer/engineer experienced on the system. The learning time to gain this experience is excessive by today's standards and has limited applicability in other areas. This has restricted the program development to a few experienced personnel.

The LAMPF Control System

The LAMPF control system (fig. 2) was developed after the Berkeley system[3]. It was built as an integral part of the accelerator. Preliminary design studies of the control system revealed that the instrumentation tended to "cluster" around the RF tanks. A remote acquisition and control terminal and the associated data transmission system were designed in-house. The serial data transmission system links the (55) remote cluster terminals to a central site[4].

In contrast to Berkeley's approach of putting the control function of each subsystem into a separate processor, the LAMPF system designers chose to put the control functions for all the accelerator subsystems into one central processor. This approach has an advantage in that it provides centralized operator control and data access allowing complex operating parameter changes to be accomplished by single requests from the operator. However, the single central processor imposes a limitation on the number of real time control functions that can be accomplished in a given interval. (The LAMPF designers have already addressed this deficiency by adding a satellite computer to perform the real time data acquisition and control function for the LAMPF injector.[5])

Like Berkeley, the majority of the software was written in-house. Although the vendor supplied FORTRAN, the control bandwidth problem mentioned above limits its application. The operating system and all the real time software were written in assembly language at LAMPF. Unlike Berkeley, the presence of FORTRAN transforms the computer control system into a viable tool for the accelerator control physicists and operators. However, the necessity of performing real time control in the same computer limits the utilization of this tool.

The NAL Computer Control System

The NAL control system (fig. 3) addressed both Berkeley's centralized information problem and LAMPF's control bandwidth limitations. From the very beginning they adopted the concept of a distributed control system[6]. Like LAMPF, they recognized the need for centralized operator control. In addition, they recognized, like Berkeley, the need for a system which could easily expand its control bandwidth.

At the time the design was started, there was no commercially available distributed control hardware. So, the designers were forced to implement their own intercomputer links.[7] They chose to develop most of their instrumentation in CAMAC, where the CAMAC crate served as the base of their instrumentation cluster.[8] A serial link controller, a data transmission system and an interface to the real time control computer were developed in-house.

The in-house hardware required in-house software to be developed to handle it. NAL personnel wrote operating systems for the MAC-16 remote computers. In addition, they wrote all of the distributed network software and all of the CAMAC data acquisition and control software.

In the central computers, the NAL designers took advantage of the high level facilities (FORTRAN, editors, loaders, files, etc.) available in commercial operating systems. A good deal of emphasis was placed on providing a software environment where engineers and physicists could develop applications programs involving the control of the accelerator.

Commercial Process I/O Systems

All of these accelerators faced the same basic control problems. They all needed to interface to a large number of widely distributed signals. Some means had to be provided for the real time monitoring and control of these signals. And finally, the operator had to be given the means to exercise control over the accelerator. Each group of designers solved these problems in essentially the same way:

1. They all designed some low-level process I/O hardware and some form of a remote data acquisition terminal.

2. They all built a digital transmission system.

3. They all incorporated mini-computers to perform some real time control and monitor functions.

4. They all implemented some form of operator interface.

As the state of the art of commercial control technology advanced, it became possible to purchase integrated subsystems which could be incorporated into the control system saving design and development costs. However, these subsystems still had to be integrated into the overall design.

Today, systems like the one at NAL can be purchased as fully integrated off-the-shelf units. The pressures of industrial automation coupled with the rapidly decreasing cost of mini-computers have resulted in: the integration of previously available low-level process I/O subsystems with mini-computers; implementation of remote data acquisition terminals and digital transmission systems; and the development of viable mini-computer networks. The potential performance of these systems can exceed that of any of the systems we have discussed.

Commercial Low-Level Process I/O Hardware

Commercial A/D converters fall into the major classes shown in Table 2. It is sufficient to measure most accelerator signals to an accuracy of 0.1% to 0.05%. Most vendors offer a number of A/D subsystems in this range to solve a variety of environmental problems. They almost all isolate the analog system from the digital system. One vendor offers a 12 bit* A/D with programmable gain and a zero suppression option. The 12 full scale input ranges vary from ± 5 mV to ± 10.24 V. The programmable gain coupled with the zero suppression can provide an overall measurement resolution of 15 bits.[9]

Commercial D/A's fall into the same accuracy and resolution range as the A/D's. However, there is usually no provision for individual channel isolation or remote ground sensing. (This can be compensated for at the signal receiver.) D/A's are the most expensive hardware component within an integrated system ranging in cost from $75.00 to $250.00 per channel.

Signal conditioning requirements separate digital signals into many different classes. Once conditioned, however, all signals are handled in the same manner. Practically all of the signal conditioning requirements in the accelerator control system can be solved by commercial process control systems. Those that are exceptions require only a signal conditioning interface to integrate them into the system. Typically, costs range from $50.00 to $250.00 per 16 bit word (depending on conditioning requirements).

Remote Process I/O Terminals

Several commercial vendors supply remote process I/O terminal hardware.[16,17,18] Data transmission rates vary from 10^4 to 1.5×10^6 bits/second and transmission distances of one mile are not uncommon. The terminal types range from passive data and control busses to fully programmable micro-computers. Their cost ranges from $1000.00 to $3000.00.

Distributed Real Time Control Processors

A number of computer vendors have networked process control systems. Since they are marketed to solve the general distributed control problem, they provide the following capabilities:

1. Multi-program real time operating systems for central and remote computers with the same external specifications, including inter-task protection mechanisms.

2. The ability to generate an operating system for a remote at the central.

3. The ability to down load an operating system into a dead remote computer and activate it.

4. The ability to dynamically down load tasks into a running remote.

5. The ability to control and debug a task running in a remote from a console on the central.

6. The provision for inter-task communications between tasks in the central and tasks in the remotes.

7. A general provision for transferring files between the central and the remotes.

The man-years of effort required to implement and document these general capabilities are usually not allotted for in an accelerator design and development budget. In addition, the cost of this development is typically an order of magnitude above the cost of the hardware itself.

The hardware to support these networks varies from low speed (9600 bit/sec) serial lines to high speed serial or parallel links (750k - 2,000k bit/sec) complete with hardware error detection and error correction facilities.

A Specific Example

In July of 1974, we submitted specifications and requested bids on a process I/O system for the control of the SuperHILAC at LBL.[15] We received replies from three companies all of whom met the specifications to a satisfactory degree. A description of the system that we selected is an example of what is available in an integrated form from commercial vendors. The system, shown in fig. 4, consists almost entirely of the vendor's standard product line, not hardware and software developed especially for LBL. It includes a remote data acquisition system, a network of three mini-computers, a medium scale central computer and several operator consoles.

Remote Data Acquisition System

The physical layout of the six SuperHILAC instrumentation clusters is shown in fig. 5. The vendor supplied his standard remote data acquisition system as shown in fig. 6.[9] Each remote link consists of two major assemblies: a link controller and a link terminal. A controller is capable of driving from one to four serially connected terminals in full duplex mode, over a pair of RG58U coax lines at a rate of 50 kwrds/sec. The terminals can be remoted as far as a mile away from the computer (although the maximum transmission rate degrades to 25 kwrds/sec). The terminals are electrically isolated from the controller and each other. Each remote terminal reproduces the computers parallel I/O bus and is capable of supporting the standard process I/O products of the vendor.

The remote data acquisition system's functional capabilities are outlined in Table 3. The system can be operated either under programmed I/O control or using direct memory access channels. The input and output channels are independent and can be operated simultaneously.

A fully loaded terminal in our particular configuration could support 224 analog input channels and 96 analog output channels. Digital I/O devices (16 signals/device) could be substituted for some or all of these channels.

The remote acquisition system is fully software supported under the vendors standard operating system. The higher level language support includes the ISA (S61.1) standard FORTRAN I/O calls for process I/O hardware. A set of stand alone diagnostics is also included.

Remote Computer Network

The SuperHILAC operates on a 25 ms cycle time. It will be run in a pulse-to-pulse time shared mode with a worst case duty cycle of 50%. Following the earlier Berkeley example at the Bevatron, a network of three mini-computers will supply the control bandwidth for the system. Each computer drives two full duplex data acquisition links, providing an aggregate process I/O bandwidth of 450 kwrds/sec. A programmable timing generator, designed in-house, will distribute control timing signals to all the real time processors. Each remote processor is linked to the central over a 125 kwrds/sec serial link.

The instruction set and operating systems in the remote processors are a subset of those in the central; the network software allows the remotes to treat peripherals on the central as if they were local and vice-versa. For example, a disc I/O call in the remote can be directed to a file on the central or a process I/O call in the central can be directed to a data acquisition link in the remote.

Central Computer

The central computer is a medium scale processor with floating point and assorted peripherals. It supports a multi-user multi-programmed real time environment. There are sufficient resources so that all the operator interface and higher level control functions can be written in FORTRAN. Its single real time function is to coordinate the radially distributed control processors and synchronize their actions with the accelerator machine cycle. The network software has the seven general capabilities outlined in the earlier section on distributed real time control processors.

Control Consoles

The control consoles were the only major system components which were specified separately and will be integrated in-house. The human interface is the most difficult to define and invariably goes through the longest evolution in both hardware and software.[10] This contention is supported by all the examples we researched. A study of the Berkeley control programs revealed that almost 80% of all the original applications code and about 90% of all the changes involved the interface to the human operator. The LAMPF control console and supporting software have gone through a major evolution after a careful human factors study.[10] The designers at SLAC also modified their original operator interface. They introduced a clever touch panel using crossed wires as push buttons on the face of a CRT containing a picture of a control panel.[11] These examples all indicate that large sums of money and time were spent on the operator interface. We expect these same changes to occur at the SuperHILAC. In order to minimize their impact we have set down some basic constraints:

1. To simplify the interface to the operating systems and higher level languages we have required that all operator input and output devices communicate in ASCII strings.

2. Wherever possible, we will use the standard communications interfaces and protocols.[12,13,14]

3. The operator interface code will be partitioned into applications and systems code so that the applications code can be made device independent.

4. The initial configuration will be as open ended as possible.

The primary control terminals we selected were color alpha-numeric CRT's. (This decision was influenced by LAMPF's success with a similar device.) The terminals we picked were specifically designed for control applications. The screen has 48 lines by 72 characters and can be completely rewritten in 7.2 ms. The terminal also has a limited graphics capability. A number of input devices are available with it; we selected a control keyboard (standard keyboard, numeric pad, cursor keys, control keys, and special function keys). The vendor can also supply joysticks, track balls, light pens, etc.

One auxiliary input device was designed in-house; a panel containing four knobs. Each knob contains a digital integrator which transmits its accumulated count into the computer in the form of an ASCII digit string.

The Advantages of Commercial Systems

The eventual success of an accelerator control system is strongly coupled to the participation of accelerator operators, technicians, engineers and physicists in its development. One way to help insure this participation is to eliminate all the mysteries surrounding computer control systems. This is a major requirement in commercial systems because they want to attract as wide a user base as possible. They spend a great deal of money providing a high level of documentation, good hardware diagnostics, and a software system that is easy to use.

In addition to the long-range benefits mentioned, there are some important short-term advantages of commercial systems including a well defined price and delivery schedule. In our example, the best in-house estimate was 50% above the purchase price of the system and the delivery of the complete system occurred two and one-half months after the vendor received the purchase order.

Hardware and Software Documentation

The commercial market forces vendors to provide a high level of documentation with their systems. A product is not generally marketable unless the buyer has sufficient documentation to use and maintain it without excessive interaction with the vendor. The vendor can amortize this documentation over many systems. In contrast, it is not economically feasible for a research institution to document a one-of-a-kind system to the same level.

The typical documentation for an A/D subsystem in the example consists of:

1. A Hardware Technical Manual, with programming notes, logic flow charts, block diagrams, signal and state descriptions and interface requirements. (These manuals are typically 100 pages long.)

2. A set of schematics including board and component layouts, and wire-lists.

3. A set of documentation on the diagnostic software. This consists of an operations manual and a discussion on errors. (Typically 50 to 100 pages.)

4. Documentation on the operating system interface to the device. (Typically 25 pages.)

5. Documentation for a set of ISA (S61.1) standard FORTRAN calls to read and control the device.

Hardware Diagnostics

The market requirements that force the vendor to provide documentation also force him to provide diagnostics. The generation of diagnostics is non-trivial, requiring both an engineer and a programmer. The diagnostics for the A/D mentioned above consists of 500 lines of code. (Integrated with a general diagnostic interpreter whose length is about 1200 lines of code.) Based on previous experience and documented cases[3] it would require at least two man-months to develop the A/D diagnostic by itself and at least another month to document it. In-house hardware seldom has this level of diagnostics.

Software

The advantages of multi-program real time executives are best expounded by the various vendors in their glossy sales brochures. However, one significant effect of operating systems in general is often overlooked: The presence of an operating system requires all the programs to use a common set of systems services to get something done. The use of these services imposes a de facto standard on their implementation. This commonality then makes it easier for one person to read another's code. It also forces him to think of the system in terms of these service capabilities, providing a common language for communication. When this is coupled with a high level of systems software documentation, the mysteries of the system are eroded and the level of participation in the solution of the control problem begins to encompass the general accelerator community.

Some Disadvantages of Commercial Systems

Any commercial system has built in constraints in both hardware and software. In general, one can overcome these constraints by minor modifications and still retain all the benefits of the overall systems capabilities.

In our example case the vendor could not supply D/A's with individual output isolation. With his cooperation, however, it was a relatively simple matter to re-do the analog output circuit to provide remote ground sensing. This was done within the packaging constraints of the system preserving all the benefits of the systems integration (documentation, diagnostics, software, etc.).

The common problem with most operating systems is that the systems overhead limits its real time response. In most cases this constraint has no effect. In the cases where it does the operating system can be circumvented. In our system the worst case time to recognize an interrupt, start a task, and queue an I/O request took about 1.6 ms. In one situation we required a faster response. So with the help of the operating systems sources and

technical documentation we added a facility to dynamically connect to an interrupt source, and to queue I/O requests at the interrupt level cutting the response time down to an acceptable 100 μsec. In general, one can get around most deficiencies in an operating system without impairing its general utility.

Another disadvantage of integrated process I/O systems is that the vendor typically won't support nuclear instrumentation. However, CAMAC interfaces are commercially available for all three of the systems which met the SuperHILAC specification.[15] We know of no vendor that has CAMAC software support.

Conclusion

Commercial process I/O systems can be applied in accelerator control applications, saving money and time. In addition, good documentation and software support allow a larger community of accelerator operators and physicists to participate in the development of the control system's software.

Bibliography

1. R. W. Allison, et al., "Digital Control of the Bevatron-Injector Inflection Trajectory," *Proceedings of the 1966 Linear Accelerator Conference*, p. 483 (Oct., 1966).

2. J. R. Guggemos and D. J. Rondeau, *Modern Control Technology with Model T Computers*, LBL-2136 (Dec., 1973).

3. T. M. Putnam, et al., "Computer Control of the LAMPF Accelerator," *Proceedings of the 1966 Linear Accelerator Conference*, p. 471 (Oct., 1966).

4. D. R. Machen, et al., "A Compact Data Acquisition and Control Terminal," 1969 Particle Accelerator Conference, IEEE NS-16, no. 3, p. 883 (June, 1969).

5. D. R. Machen and J. M. Potter, "The Satellite Minicomputer-A Practical Solution to Accelerator Control," 1973 Particle Accelerator Conference, IEEE NS-20, no. 3, p. 648 (June, 1973).

6. R. E. Daniels, et al., "The NAL Computer Control System," 1973 Particle Accelerator Conference, IEEE NS-20, p. 505 (June, 1973).

7. S. R. Smith, et al., "Intercomputer Communications in Real Time Control Systems," *ibid*, p. 536.

8. L. J. Hepinstall, et al., "CAMAC Experimental Beam Line Control System," *ibid*, p. 514.

9. Modular Computer Products *System Design Handbook*, 1974.

10. B. L. Hartway, et al., "The Central Control Room Man Machine Interface at the Clinton P. Anderson Meson Physics Facility," 1973 Particle Accelerator Conference, IEEE NS-20, no. 3, p. 633 (June, 1973).

11. W. C. Struven, "Experience with Touch Panel Control at SLAC," *ibid*, p. 636.

12. *Code for Information Interchange*, ANSI Standard X3.4-1968.

13. *Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange*, EIA Standard RS-232C (Aug., 1969).

14. *Procedures for the Use of the Communication Control Characters of ASCII in Specified Data Communication Links*, ANSI Standard X3.28-1971.

15. V. P. Elischer, R. A. Belshe, "Specifications for the SuperHILAC Computer Control System LBL-3691.

16. PDM70 Product Specification, Digital Equipment Corporation (1974).

17. Series 2000 process I/O system specifications (1974). Micro-Pac 80 process I/O system specifications (1975). Process Computer Systems. Inc.

18. RTP 7400 Product Description. Computer Products Inc. (CPI), (1974).

Table 1. Control System Comparison

	BEVATRON	LAMPF	NAL
Low-Level Process I/O Hardware	Developed in-house; different hardware for each application.	Developed in-house. Remote clusters communicating over serial links.	Primarily CAMAC or CAMAC standard hardware. Much developed in-house. Remote clusters communicating over serial links.
Data Acquisition Terminal and Data Transmission System	Developed in-house. Different hardware and protocol for each application.	Developed in-house. One type of high speed serial transmission link for all devices.	Developed in-house. Several types of high speed serial and high speed parallel transmission links.
Control Computer(s)	Individual processors, each dedicated to a single function or subsystem. No central control or shared data. One operator console for each processor (teletype).	One central processor for all control. Multiple operator consoles, any one capable of controlling entire accelerator.	Large central processors handling operator communication and control. Multiple remote computers for device handling and real time control.
Software	Software developed independently for each processor. No operating system. No high-level language. Little commonality of code or data structures between processors.	Majority of software, including operating system, developed in-house. Majority of code done in machine language (due to processor bandwidth limitations). Several complex setup and beam optimization programs written in FORTRAN.	Central processors use vendor supplied operating system. Majority of central (i.e. operator communication code) written in FORTRAN. Network software to talk to real time control computers and real time control software developed in-house, and written in machine language.

Table 2. Commercial A/D Systems

	NON-INTEGRATING	INTEGRATING
SINGLE-ENDED MULTIPLEXED		
No. of Channels	16 to 512	
Conversion Rate	100 hz to 2 mhz	
Full Scale Output	5 mv to 100 v	
Resolution	8 to 12 bits	
Accuracy	.1%	
DC Common Mode	10 v	
AC Common Mode	-	
Normal Mode	Filters available	
Base Price	$2,000 to $3,500	
Cost/Channel	$40 - $60	
DIFFERENTIAL, MULTIPLEXED		
No. of Channels	8 to 256	8 to 512
Conversion Rate	100 hz to 2 mhz	5 to 100 hz
Full Scale Output	5 mv to 100 v	5 mv to 100 v
Resolution	8 to 12 bits	12 bits
Accuracy	.05%	.01%
DC Common Mode	10 v	500 v
AC Common Mode	60 - 80 db	110 db
Normal Mode	Filters available	
Base Price	$2,000 to $5,000	$3,000 to $5,000
Cost/Channel	$50 - $80	$60 - $80
DIFFERENTIAL, NOT MULTIPLEXED		
No. of Channels		1
Conversion Rate		<40
Full Scale Output		5 mv to 1000 v
Resolution		12 to 14 bits
Accuracy		.01%
DC Common Mode		500 v
AC Common Mode		60 to 120 db
Normal Mode		
Price		$30 - $5,000

A/D converters available with standard process I/O systems

Table 3. REMAC* Command Description
*(trademark of Modular Computer Products, Inc.)

TERMINAL TRANSMITTER SERIAL WORD FORMAT
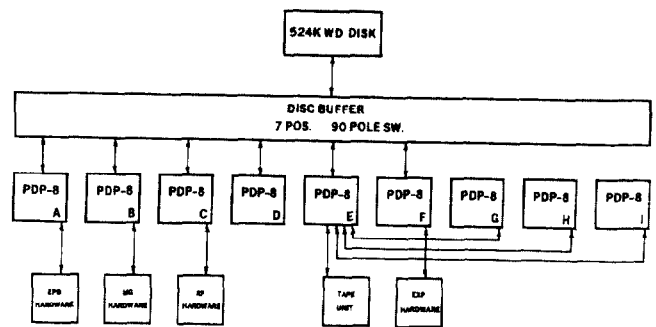
TERMINAL RECEIVER SERIAL WORD FORMAT

Fig. 1. Berkeley Bevatron Control System

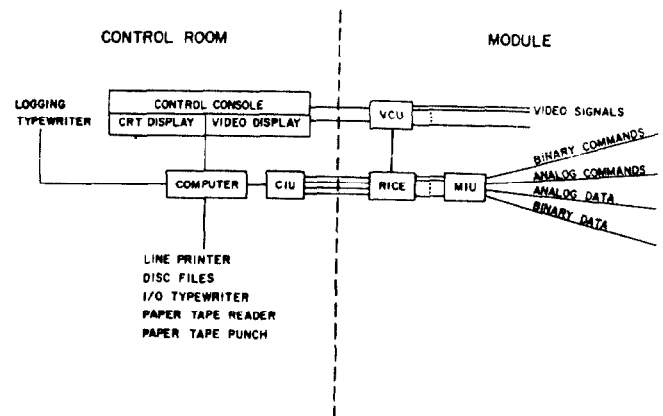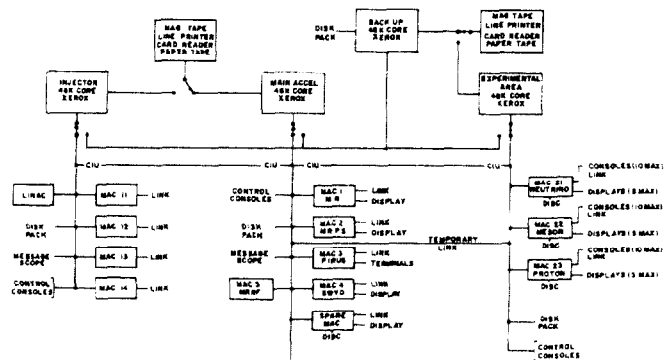
Fig. 2. LAMPF Control System
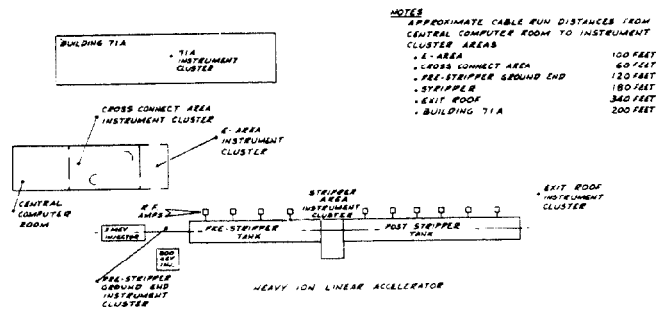

Final Configuration of NAL Control Computers

Fig. 3. NAL Control System

Fig. 4. SuperHILAC Control System

Fig. 5. SuperHILAC Physical Layout

Fig. 6. Remote Acquisition System