Introduction

The SPS control system has a number of features which at the time of its inception were new in the accelerator field. The most original of these features were a completely new distribution of the responsibility for instrumentation, an attempt to perform control entirely from general-purpose control desks, and the almost exclusive use of a high-level language (NODAL) for application programming.

The SPS was commissioned and improved using this control system, and whilst it is not generally agreed whether success has been because of or in spite of the control system, the success itself is nowhere doubted. A 400 GeV accelerator with an 8.5 second cycle time can be controlled and developed with a control system of this kind.

Questions are now being asked which are of a more general nature. Are the parameters of the SPS particularly suitable for this kind of instrumentation? Can this kind of project only succeed on a newly-built machine? Are the principles of this control scheme obsolescent in the light of the foreseeable development in computer techniques? This paper will try to answer these questions, and there is no point in concealing the fact that the answer to each of them will be 'No'. We shall start by defining the new methods and showing how they differ from the old, and then go on to show that all conceivable problems of central and closed-loop control can be solved by the new methods.
It is important to note that while the new methods solve the same problems as the old, they may differ enormously in the implementation. This may well mean that the people operating and using the new complex need to rethink control problems they have long considered solved. Some of them will be capable of living with the new methods and even using them to their advantage. The others will fight a rearguard action against the new methods.

1. What is meant by 'Computer Control'?

There are two distinct kinds of control. Closed-loop control, and operator control and surveillance. Figures 1 and 2 explain the distinction in pictures; the arrows representing information flow:

![Fig. 1: Closed-loop control](image1)

![Fig. 2: Operator control and surveillance](image2)

The first process is classical, and has its roots in the clock-making revolution of the Renaissance. The Cam, the steam governor and the electrical servo-system have fulfilled the requirements for this kind of control, and the microprocessor has recently started to take over the low-speed end of the market. When the application is sufficiently banal for a mechanical linkage an analogue circuit or a computer to close the control loop, closed-loop control is possible and sufficient, and the result is a completely automatic process.
When the process to be controlled is too complex for complete automaticity, or when the problems to be solved are ill-defined, it may be necessary to incorporate an intelligent human being in the loop. It should be noticed that the difference between figures 1 and 2 can be ignored if it is agreed to consider the operator as a part of the control equipment. One feels instinctively, however, that there is a difference between automatic and manually operated machinery, and the distinction will therefore be sustained. This is important, since a human operator has a reaction time three orders of magnitude lower than the best modern digital systems and six orders lower than the best analogue ones. The arrangement of fig. 2 gives the advantage of human intelligence, but only renders it necessary to pay the speed cost in the left-hand part of the loop.

We are now in a position to define total computer control. A computer controlled system may have analogue loops with microsecond responses, digital loops with millisecond responses, and manually controlled loops in the one-second range, but all operator communications with the three kinds of process will be via the computer. Figure 3 should make this clear.

Fig. 3: Three kinds of control
The top loop is very fast: it goes via specialized hardware (small box in the diagram) whose parameters can be observed or changed from the computer. It finds its application in certain analogue loop control systems like the RF control of the SPS beam. The middle loop is quite fast, and has the advantage that the feedback mechanism is made of software and therefore under operator control. The lowest loop works at operator speeds, but gives a feedback mechanism with human flexibility and intelligence.

A fourth possibility exists, which would be represented in diagrammatic form by a pair of control lines between the operator and the process, not involving a computer at all. The existence of even a single sub-process which is controlled in this way implies that the process is not under total computer control, and the implications of this will be discussed later.

2. Two useful tricks

Given that the slow control loop is the most desirable from the point of view of flexibility, it is useful to seek methods of performing time-critical operations on a leisurely scale. Two such methods are available: the use of computer-independent timing and the installation of computer-controlled information multiplexers.

Computer specifications for real-time applications often require a very high speed of response to an external stimulus. Such high speeds are clearly needed if the operation to be performed has not been foreseen and if major decisions must be taken at interrupt time for which the data were previously not available. In practice, however, real-time operations can often be pre-planned. The operation is divided into three phases:

(a) A planning phase, determining what will be done at the critical time, and when that time will occur,

(b) The time-critical (or speed-critical) phase,

(c) A checking, or data collection phase (which may not be necessary).
Only phase (b) is time critical, and if the computer is not to be under pressure at interrupt time, phase (b) must be entirely performed by hardware independent of the computer system. This hardware may involve an independent timing system, external memory, or even a free-standing dedicated computer, but it is vital that the independent hardware itself be under total computer control at the slower level. A concrete example of this at the SPS is the various function generators, which can be set up at leisure and then triggered with great precision by a timing system which has itself been pre-programmed from the computer.

It should be noticed that, in principle, all time critical operations can be performed by autonomous hardware pre-parametrised at leisure. If the parametrisation itself is time-critical (as in the case where all function generators must be re-loaded before the beginning of the next cycle), then a small gain can be won by double-buffering in external memory. If this is insufficient, special hardware must be supplied which sets up parameters from a small number of super-parameters set by the control computer.

The second trick for allowing more leisurely operation is available because some analogue information is processed directly by the operator's eye and brain, and there is therefore no need for it to pass through the computer system. Analogue signals presented on operator displays could be considered by a purist as implying a loss of total computer control, and indeed such information cannot be used directly by the computer, but there are many cases where a display is used only as a part of a low-speed operator-controlled loop. It is a pity to clutter the computer with displays of this kind when cables can be used, and some computer control can be retained by multiplexing, routing and choosing the signals under computer control. Concrete examples at the SPS are TV pictures and analogue signals.

![Fig. 3: Analogue or TV bypass](image-url)
4. The advantages of total computer control

If total computer control can be achieved, two enormous advantages accrue. The first is the ability to build generalised consoles, and the second is that all changes to the control system can be performed in software. We shall discuss these advantages in detail.

If the computer control is total, the operator may be said to be connected, not to the process, but only to the computer. Figure 3 shows that a console containing only computer peripherals and general-purpose displays will perform all functions of the process which are computerised, and hence by definition all functions of the process.

Given a general-purpose console with controls and displays connected to the computer system and direct-viewing devices multiplexed under computer control, it is clear that this console can be configured without direct consideration of the process being controlled. The console and its interface computer are arranged to suit an operator, and the computer (or computer system) is interfaced to the process as a separate operation. One advantage of this arrangement is that the building of the operator interface can precede the completion of the project, or better still, one can start with an unsuitable interface and build an improved one subsequently. The operator interface of the SPS is of just such a general-purpose type.

The second advantage of total computer control is that, since all process hardware is controllable by the computer system and all operator functions are equally connected to it, all possible programs defining the interaction of the operator with the process can be written in software. Those parts of this programming which work with the operator will be slow, and those which work with the process may, if necessary, be fast and even reside in special-purpose peripheral computers or microprocessors, but all of them can be changed.

This brings us to high-level languages. Ideally all programs should be in as high a level of language as possible, for comprehensibility and ease of use. Lower levels must be used close to the process where speed
is at a premium, and special hardware may be needed at the process interface where the speed is critical. Nothing is lost by using low-level languages and special hardware so long as the modules so implemented are constructed and debugged once for all time. By using the pre-parametrisation techniques described earlier, this can usually be achieved. The SPS control system uses the interpretive language NODAL at the operator level and for local programs, the machine code of the control computer for interface subroutines, and hardware connection or microprocessor code below that. Everything below the NODAL level changes rarely. This is an advantage, but is not an essential feature for a working system.

5. **Administration of a control scheme**

The method of administration and the allocation of responsibility used at the SPS can be used anywhere though the programming of the operator interface from generalized consoles in a high-level interactive language helps. Briefly, the method is to separate the hardware interface subroutines, the operator-oriented programs and the system services into separate well-defined modules. The operator-oriented programs are written by the operations personnel who will use them. The hardware interface subroutines give generalized program control of each element of the process without taking decisions about how these are to be combined later by the operations staff: these subroutines are written by the staff responsible for the individual hardware elements. Finally, the channels of communication in and between computers, the system software and other facilities needed by the other two teams are supplied by a computer controls group.

The administrative structure therefore corresponds to a breaking-down of responsibility along lines parallel to those of the control scheme. Just as there is a simple interface between a general-purpose console and the computer system and between that system and process, so there are well-defined interfaces between the responsibilities of the various groups. People who are unsatisfied with the resources on the other side of their interface have the usual recourses of complaining, paying for better resources and so on.
6. User programs

Established control-schemes which use dedicated consoles tend to have very large user programs. We have found at the SPS that the best user programs are small, simple and quick in execution. A major operation (such as setting-up and trimming a particle beam line) may consist of several elemental programs each of which performs part of the process. No single program represents an enormous investment, and so the improvement of any program by rewriting it can be contemplated. Sequences of such programs can be used by an operator to do his work in a much more flexible manner than would be possible with a monolithic control program. Furthermore, if the control process is executed in small modules, the general-purpose console can be used between modules for subsidiary enquiries or operations.

If an established process is subjected to the new form of total control from a single general-purpose console, there will be a tendency to impose the old operations methods on the new structure, and to make the new consoles behave like the old, dedicated ones. This is usually possible, often at great expense, and is almost never recommended. The control problem must be re-thought in the light of the new resources available, and a suitable solution devised. This solution, once found, is often easier to operate, more informative and will lend itself to modification or re-design. In most cases no-one can find the solution except the person with the problem, and it is therefore vital to get the future users of the control system to understand it and develop enthusiasm, so that they can give their unique skills to the design of the system which they themselves will use.

This human problem is at the core of the matter. Commitment to total computer control must be total. All indispensable users of the process to be controlled must therefore agree to collaborate in a system free of exceptions and special cases, even if this means thinking very hard to find solutions to problems which were non-existent in less structured control systems. If there is any user of the process who thinks his sub-system cannot be controlled by the methods of total computer control, an attempt must be made to convince him. If this is not possible, the computerisation project is destined to failure.
7. Conversion of an existing system to total control

An existing process will usually consist partly of manually-controlled elements, partly of computer surveillance projects covering sub-systems, and often, in addition, of the debris of abandoned attempts at more completely integrated computer control.

Usually, the process cannot be stopped for the months required for the installation of a new system. The new system must be installed on-line in a piecemeal fashion. It is clear that computer control may not be total in the interim, and for a large process this may be months or years.

The procedure which must be followed in a case like this is to design a general-purpose human interface and a method of joining up any number of such interface with a computer, as well as a method of joining computers with the process. A small sub-process is then removed from the main process and moved to the new complex during a short break in operation, and with the possibility of returning to the old sub-system in case of trouble. Other processes are then moved to the new scheme, the quantum being determined by what was considered to be monolithic in the old scheme and the time and risk at each installation being functions of the size of the quantum. While some benefits are gained as soon as one installation is commissioned, the considerable advantages of total control are not available until the last installation is completed. No essentially new features may be introduced during the entire installation without considering their effect on all the systems already installed. This may require considerable self-discipline in the rejection of clever, cheap or attractive techniques which become available during the installation period.

There is therefore, in principle, nothing harder about installing total computer control in an existing process than commissioning a control system for a new device. Each new sub-system added to the complex will have its own teething troubles, but these will be sub-system dependent since after the first few installations the computer system and the operator interface will be well-established.

It should be well understood that a drop in reliability is unavoidable during the installation of a new control system. The decision to go to total computer control will have taken into account many factors, of which a certain loss of reliability during installation is only one.
8. The view from the SPS

The preceding paragraphs are an attempt to show that there is nothing magic about the control methods being used at the SPS. A system of total computer control, itself under the control of its users and not of some well-intentioned but less involved third party, is rendered accessible from a number of general-purpose consoles. Expansion of capacity, both with respect to speed and size, is available at all times, and the cost of such expansion is for each case only the irreducible theoretically necessary down time.

The SPS has collected, made, or bought the following elements in order to implement its scheme:

A) A computer, for which all our software is written. This comes in a number of forms:

- **NORD10**: obsolescent
- **NORD1OS**: $1 \frac{1}{2} \times 2$ x speed of Nord 10
- **NORD1OM**: same speed as NORD1OS: portable and newly-designed in 1978
- **NORD1OL**: planned: same format as NORD1OM but faster and probably much more expensive.

Further versions in the NORD range cannot, from current knowledge, be relied upon to run our software or to drive our hardware interface. It is not excluded, of course, to build a compatible computer to protect the software and interfacing investments, but this is a dangerous course, as the lead-time on the design and construction of a computer renders the completed item very liable to be behind the state of the art.

B) Message-transfer hardware, designed by an industrial firm, which is compatible with all the above hardware, and which drives full-duplex serial links at many times the speed available with conventional equipment.

C) An operating system (SYNTRON) scheduling real-time programs at several interrupt levels, managing a compatible filing scheme, and allowing multi-computer tasking from NODAL.
D) NODAL and a data-module and function management scheme. A NODILER is being written, giving pre-compilation facilities for speed.

E) Message-handling software which runs under SYNTRON in a message-handling computer. This software is configurable so that the system may contain up to 256 computers, any of which may be a message-handling computer, in any configuration not involving alternative paths for the blocks of a message.

F) CAMAC interfacing to the computer, with NODAL-driven software for both serial loops and crate-controller systems. Properly arbitrated local microprocessor control is possible without prejudice to the host-computer's control of the same CAMAC units.

G) A general-purpose multiplexer system, available to the software system from debugged routines, which gives control and surveillance at a distance and at all but the highest speeds, and which is demonstrably cheaper and more reliable than the current generation of CAMAC.

H) A microprocessor framework, compatible with the multiplexer system mentioned above, software supported, and able to deal with the need for specially designed autonomous equipment with local intelligence, which can be added without a requirement for specialized central software to be written in each case.

I) Excellent general-purpose consoles each interfaced to a computer.

These are the elements which the SPS has produced. On top of these is an enormous super-structure of applications programs, ever-changing, ever increasing, ever being superseded by new versions. The elements described are certainly (by definition) capable of controlling the SPS. The extensions to faster systems, more complex systems and the conversion of existing systems have been discussed. It is difficult to envisage an arrangement for which total computer control is impossible, using the components available at the SPS. An exhaustive demonstration of this contention entails the construction
of a number of control systems, and is therefore not possible outside the field of the thought-experiment.

The question arises, whether it is possible to attain total computer control with other components. The answer is almost certainly yes. The thing which is important is the philosophy of design. It is a pity to redesign things which work perfectly well, but that in itself will not sink a control project.

Conclusion

The design principles used at the SPS can be used for the total computer control of any large process from general-purpose consoles.

No special feature of the SPS has been used to render the process simple, and neither speed nor complexity will be a bar to the inclusion of any further sub-system. The principle is susceptible to continuous on-line extension, and there are already existing examples of the fact that new technologies can be incorporated progressively.

The understanding, collaboration and participation of prospective users were vital to the success of the SPS computer control project and would be equally indispensible to the success of any other project for total computer control. If there is anything special about the SPS which has allowed this method of control to be installed successfully it may be the enthusiastic participation of the operations group in the implementation.