

all and published in a table. That is, it was implicit that all methods would be distribution-free. A good example is the χ^2 test for goodness of fit of a smooth curve to a data histogram. The significance level can be read from a pre-calculated table of χ^2 , depending only on the number of histogram bins (the "degrees of freedom"), but not on the functional form fitted. Very nice, except that real data have a nasty habit of failing to conform to the requirements of the standard test: The bin contents may not be independent, or not Poisson distributed, or not large enough (the test is only *asymptotically* distribution-free). This is no obstacle to the physicist armed with a computer since he needs no table, and need not even worry about whether his test is distribution-free; he simply determines "empirically" the distribution of the test statistic under the actual experimental conditions, by generating random data according to the null hypothesis.

An even more radical departure from traditional statistics is represented by a class of methods known as *bootstrap*. The word comes from the expression "to pull yourself up by your own bootstraps", and refers to the use of the data sample itself to calculate the significance of the test. This technique is especially useful for comparing samples whose underlying distribution is unknown, which is nearly always the case with biological data, and often is true also in the more mathematical sciences.

A typical bootstrap situation would be to test whether the very highest energy cosmic rays observed in the northern hemisphere are significantly more energetic than those observed in the southern. Suppose we have set the energy cutoff so high that the sample consists of only 13 northern and 6 southern events. The straightforward approach would be to calculate the difference between the mean energies of the two samples (call this Δ_r) and try to determine whether this value is significantly different from zero. Traditionally one would apply the Student t-test, which is exact and optimal, but only when the underlying energy distributions are Gaussian. In the case at hand, the distributions are not known, but there is no reason for them to be Gaussian, so the traditional test would not be very meaningful.

Using the bootstrap, first calculate Δ_r , as before, then forget which rays came from which hemisphere, pool the 19 events together and consider all the possible ways of dividing these 19 energy values into two samples of 13 and 6. For

each of the 27 132 combinations, calculate Δ_r , the difference between mean energies, and let $g(\Delta)$ be the number of combinations with energy difference Δ . Under the null hypothesis (no difference between hemispheres), each combination is equally probable. This means that the sum of $g(\Delta)$ from Δ_r to infinity, divided by 27 132, is directly the confidence level in the usual statistical sense: the probability, under the null hypothesis, of observing a difference Δ greater than Δ_r . The method is immediately understood intuitively, takes account of all the observed quantities, and does not require any additional unjustified assumptions. It is also possible to use a distance measure other than the difference of mean energies, if such a measure is more sensitive to the anisotropy predicted by a theory. And if the samples are so large that even a computer cannot enumerate all possible combinations, the method can be modified to sample randomly the different combinations until a sufficiently accurate approximation for $g(\Delta)$ is obtained.

Applications of computation-intensive techniques in theoretical physics are too numerous to discuss here in any detail, but some mention must be made of what is perhaps the most spectacular such application, *QCD on the lattice* [2]. Pioneered by Ken Wilson, this allows in principle calculation of the most fun-

damental properties of matter, such as the masses of the elementary particles, from first principles of Quantum Chromo Dynamics. The lattice is a discretization of four-dimensional space-time, and must have enough points to approximate the continuum. One then uses a Monte Carlo method to find states of statistical equilibrium with respect to fluctuations of the complex quantum field. Even the least ambitious QCD lattice calculations strain the resources (both memory space and computing speed) of the world's most powerful computers, a situation recognized very early on, and leading many of the researchers in the field (including Wilson himself) to spend a large part of their time designing computer hardware specially adapted to solving this problem.

A cynic might remark that all this is just another manifestation of Parkinson's law: that as computers get bigger, physicists think up new ways to keep them busy. I prefer to look at these applications as exciting new approaches to the solution of physical problems previously considered intractable.

REFERENCES

- [1] James F., 'Monte Carlo Theory and Practice', *Rep. Prog. Phys.* **43** (1980) 1145-1189.
- [2] Creutz M., Rebbi C. and Jacobs L., 'Monte Carlo computations in lattice gauge theories', *Phys. Rep.* **95** (1983) 4, 201.

Experimental Physics Control Systems

Inter-Divisional Group

From 28 September to 2 October 1987, Villars-sur-Ollon (Switzerland) hosted the international conference on control systems for experimental physics organized by the EPS Interdivisional Group on Experimental Physics Control Systems (EPCS). Over a full week, scientists from all over the world discussed the problematics of controlling processes as complex as particle accelerators, nuclear fusion devices, large telescopes and high energy physics detectors.

Close to 80 contributed papers were accepted; most of which were displayed in poster sessions. Invited papers gave overviews of major control systems, supplemented by contributed papers, workshops, tutorials and exhibits. Panel workshops covered a substantial part of the programme. Authors with different

opinions were invited to the podium to defend their points of view, often leading to lively discussions. There were workshops on Architecture, Interfacing and Intelligent Process Equipment, Applications Software, Operational Aspects, Automation, Software Engineering. A final brain storming on Contemporary and Future Main Issues, terminated this series of highly animated debates.

The trend towards more complex machines leads to more complex controls. On the other hand, data processing technology progresses rapidly and the cost of hardware decreases steeply. No surprise that the current trend in controls is towards fully distributed systems with more and more local intelligence at the device level. In broad lines, the control architectures are based on two level networks: a backbone, token ring or

Ethernet, connecting workstations to, mostly, VME based front end microprocessors, or concentrators, which in turn drive local distribution networks (e.g. 1553B) to individual devices. In such environments, software is becoming increasingly complex and expensive. Although no well integrated software for large projects exists as yet on the market, a number of recipes and techniques were presented which should be followed when developing large systems. Structured engineering methods have proved their usefulness in a number of control projects, and, although still partly in a theoretical research stage, object oriented concepts are now showing promising signs of success.

Control systems should be designed not only to control but also, and to an equal extent, to maximise reliability, ease of development, integration and enhancement. Maintenance draws resources and precious operation time: it is thus essential to take this aspect into account at design time and to include facilities to trace back the behaviour, or misbehaviour, of the system. This implies that software should be as much as possible, not to say entirely, data driven. On-line data, needed for real time activities, should relate to off-line data describing the operational constraints of the process or providing information for managerial purposes. The data base is thus an essential part of the control system and demands appropriate management systems.

In such a context, management becomes of course of paramount importance and control systems implementation should be based on solid planning with identified milestones, realistic schedules and defined budgets. Gone are the days where ingenious engineers or physicists could improvise some *ad hoc* ancillary controls in their spare time. As a corollary, smaller installations with scarce manpower run into problems. There was a clear call from the smaller institutes for help from their bigger brothers. They wish to benefit from development done in larger organization e.g. by taking over entire parts of control systems. Process independent parts should be identified and designed so as to constitute a basic package or "kit" of facilities which can be transported to other environments.

Against this background, application software, which takes a major fraction of the resources available for the development of modern controls, was looked at in more details. It is structured in layers of modules each catering for a specific type of activity: device control,

physics variable control, specific process application and operator interaction at the console level. For the reasons mentioned in previous paragraphs (data driven software, maintenance, etc.) code should be separated from data: the realm of the code constitutes a library of reusable routines and the realm of the data constitutes the on-line data base. All housekeeping, scheduling, etc. activities should be in standard modules. This should lead to basic packages that are process independent and into which process specific application can be embedded at a "minor" cost by filling data tables and writing short, simple special purpose procedures or tasks. Although this is the general tendency, more sustained research for appropriate structures and standardization are still needed.

For higher efficiency, operation and exploitation need to be carried out from a single control room with non-dedicated, standard consoles. Traditional operator's consoles with alphanumeric, and graphical displays, touch panel, tracker balls, etc. are giving way to workstations. Current experience indicates that they are over-flexible for everyday operation and rules for a consistent use have to be defined. Questions that have arisen are: can alarms be included on the main console workstation? Can a single workstation display be modified so that more than one person can use it at a time? The workshop discussing this also defined a number of facilities which should be provided as standard by any control system: summary status display, chronological recording of events including hardware faults and operator request, etc.

At the workshop on Automation, a number of levels of controls were proposed as a standard to define the successive degrees of complexity achieved by control systems. In an economically tight environment, such definition should help clarify what one is talking about when it comes to the often painful matter of money and manpower. The lowest level is pure data acquisition, where data is collected through the system and used to monitor and study the operation of the process as well as to develop control algorithms. The second level is more concerned with supervisory controls. Next level up incorporates modelling programs to test proposed changes against the theoretical behaviour. The fourth level performs continuous controls: the setpoints of the control loops are available to the operator, but they are maintained by the system to assure steady operation. At

the next higher level again the system is able to change the state of the process by transiting from one state to the next according to a master sequence. This level involves automatic start-up and shut-down and requires sequential control algorithms of all process devices. The last but one level extends this sequential control to automate fault diagnostics and possible recovery: the process is returned in a fully, or perhaps degraded, operational state. The highest level of sophistication is when automatic optimization is achieved. Here the behaviour of the process is optimized by setpoint adjustment and state changes to reach optimal performance. This is a field where artificial intelligence could be applied. As to the question why accelerators are not fully automated yet, it was felt that the data from the measurements was not reliable enough, that there was not enough time allowed for automation, not sufficient control parameters and that their settings are in general not sufficiently reproducible.

By special request an extra session was improvised on the topic VMS versus UNIX. The session highlighted the differences between a highly flexible proprietary operating system and an open one, in areas important to controls. It led to an interesting exchange of information and experience as most people were only familiar with one of the systems. But, as it turned out, there was no definite conclusion as to which was to be preferred.

Conclusion

To summarize: the conference was most successful as measured by the number and quality of the participants. It was agreed that it should be "institutionalized" at the rate of one every two years, and should be complemented by workshops and study groups on more specific subjects.

In view of the increasing complexity of controls the need for closer collaboration between laboratories and standardization at all levels, was often emphasized. In particular smaller laboratories expect to benefit from the control developments made in the larger ones.

Along these lines, a first working group has been started with the aim of standardizing the operational control protocols of power converters. Seven institutes are currently participating in this first collective EPCS endeavour.

A. Daneels Chairman EPS-EPCS

B. Kuiper Chairman Scientific Programme Committee