Nuclear Instruments and Methods in Physics Research A250 (1986) 534–538
North-Holland, Amsterdam

AN AUTOMATIC FAULT MONITORING SYSTEM USING A MICROCOMPUTER.

R. BESOLD, R. BARAN, A HOFMANN, P. HOLLECZEK and R. MUELLER

Physikalisches Institut der Universität Erlangen-Nürnberg, 8520 Erlangen, FRG

Received 11 April 1986

This paper describes an automatic fault monitoring system using a microcomputer. The hardware is based on the Z80 processor and the software on the programming language PEARL. The microcomputer system is able to detect faults in the equipment during data acquisition and provides useful information for the subsequent data evaluation. The results of two applications are discussed.

1. Introduction

In nuclear and particle physics a trend towards more complicated apparatus is apparent. This usually contains several detectors which provide various types of information such as energy, time of flight, and tracks. This information is processed in complex electronic equipment and then recorded by computers. The growing complexity of this equipment makes the supervision of the experiment much more difficult. Its supervision is only partially practicable for the physicist and even then requires a large amount of manual recording by the experimenter. Furthermore there are often difficulties in obtaining information about the quality of the data extracted directly from the apparatus. These data require further processing to show that there is no defect.

Inaccuracies and defects in detectors or electronic components are difficult to recognize especially during data acquisition. They often appear later in the data evaluation or remain undetected. The consequences are wrong measurements or wrong interpretation of the results and loss of expensive experimental time.

One way to implement automatic supervisory tasks is the use of microcomputers. These can now perform applications which were previously performed by process control computers.

In refs. [1] and [2] a general overview of the use of computers in laboratory automation is given. They do not explicitly refer to the use of automation in fault monitoring, but if computers can be used for measurement they can also be used for fault monitoring.

In this paper a fault monitoring system is described which detects certain failures in the apparatus and helps the physicist to locate them. The system is based on a Z80 microcomputer and a real-time software package. The microcomputer obtains the supervisory data from the experiment and transfers them in a reduced form to the on-line computer. This supervisory information could not be recorded or analyzed additionally by the on-line computer without the dead time for the experimental data becoming excessively large. Moreover the microcomputer obtains its data at points in the electronics at which the data have not yet been reduced and therefore contain much more information for the recognition of defects than the reduced data which reach the on-line computer.

In ref. [3] a microcomputer-based system for measurement and control is presented. This also states the need for using microcomputers for control, not only for the automation of measurement tasks. The system has some functional similarity in the hardware modules to the system presented here, but it is not based on modules which can be purchased. The main difference between the system of ref. [3] and that presented here is in the software. Our microcomputer is a stand-alone system, whereas the former needs a host computer. This severely decreases its flexibility.

2. General testing method

In many experiments fault monitoring was normally performed manually by the physicists. They record, for example, count rates, and compare them with older values. Another typical task is checking the centre of some peaks in a spectrum. A failure is present if the value of one of the test points varies. Therefore fault monitoring means the continuous or sampled checking of differences from a good initial value. This initial value is defined by the experimenters after the test and start phase when the apparatus works properly. An automatic fault monitoring system then looks periodically at the values from various test points. If there is a significant difference from the correct value the system performs the "defect reaction". The equipment to be monitored usually consists of several components such as amplifiers, discriminators, or detectors. The fault monitoring system can check these components for proper functioning. Due to the flexibility of the fault monitoring system the defect reaction is often simply an output message for the experimenters. The interpretation of the fault and its cause can only be made by the physicist, because only he has the background information on which the correct value was based.

3. Possibilities of testing

As mentioned above the electronic equipment consists of several modules. Possible failures of these modules are

- change of an amplifier gain
- change of a discriminator threshold,
- change of the timing between some components,
- losses due to changing load (e.g. beam intensity).

Tests can be made with two types of pulses. First there are pulses due to the reaction in the target – the "natural pulses". They are mainly used for manual tests. However, a second type of pulses is often necessary to complete the tests. These are "test pulses" which are additionally fed to the electronics.

The natural pulses have no influence on the electronics, but test pulses lead to a higher load on the electronics and also their height and/or timing have to be selected so as to be separable from the significant data in the subsequent evaluation. Also the fault monitoring microcomputer has to generate these test pulses. This could, for example, be achieved by a programmable function generator.

To minimize the influence of the test pulses, measuring could be turned off during the test periods (later called off-line). This means that the pulse registration is cut off at the detectors and the test pulses replace the "normal" detector signals. But then it would not be possible to test losses due to changing load and also expensive experimental time would be wasted.

The use of natural pulses in experiments with accelerator beams requires the normalization to the current beam intensity and effects such as increasing activity in the target or scattering at slits must be taken into account.

A programmable system should be used to meet the requirements of a wide range of experiments and to provide optimal performance for particular equipment.

In the following section the procedure using test pulses is discussed in detail first and then the use of natural pulses. In both cases a microcomputer performs the tests.

3.1. Test procedure with test pulses

Controlling of the attributes mentioned above means a comparison between the current and the initial values. In particular this means:

(a) The gain of an amplifier can be checked for constancy and linearity. Off-line this could be done by a continuous variation of the pulse height within the amplification range. On-line one has to select regions in a spectrum where the additionally created peaks belonging to the test pulses do not strongly influence the original spectrum with the peaks of interest. Normally there are only a few regions where these test peaks do not disturb the spectrum. In any case the adequate pulse heights must be selected interactively.

(b) The discriminator thresholds can be checked for constancy and width of the window. The pulse heights must be selected around the thresholds. The microcomputer has to count the number of pulses at the output of the module, to check whether their height is above the lower and below the upper threshold.

(c) The test of the timing checks the conversion range and the setting of a time converter (TPCs in older setups and TDCs in newer ones). This requires not only suitable heights of the pulses but also the correct timing. The microcomputer has to analyze the number of pulses for example in the TPC peak and the position of the peak within the spectrum. Variations from the initial value lead to a drift in position and/or a broadening of the peak.

(d) In principle a test of losses due to varying load can be performed, but the additional test pulse rate influences the actual load and therefore the measuring procedure influences the measured value.

3.2. Test procedure with natural pulses

The test procedures using the natural pulses are as follows:

(a) The gain of an amplifier can be checked for constancy and linearity. There is no difficulty in spectra with proper peaks. Some of them must be marked at the beginning of the test. The controlling computer then checks the position and width of the peaks.

(b) The thresholds of discriminators can be checked for constancy. The computer measures the normalized count rate at the output. A change in the threshold setting leads to a change in the normalized count rate.

(c) The test of the timing is analogous to the test of the amplifier gain. In this case testing by combining test pulses and natural pulses is also possible. If a moderate fraction of the start pulses are split and fed into a programmable delay, and the delayed pulses are added to the normal stop pulses, one new peak will appear in the spectrum. The recoupling of the delayed pulses could be performed anywhere in the electronics where a check of one module or a whole detector arm is possible.

(d) The test of losses due to load could be performed if a defined number of pulses of the right energy and timing are recorded before and after the module.

3.3. Comparison of the test procedures

There is one big difference between test pulses and natural pulses. The test pulses are an electronic imitation of the natural pulses. As a consequence the range of checking with test pulses is limited to electronic devices, whereas the natural pulses allow a check of the complete experiment (including detectors, accelerator, etc.). Another difficulty is the generation of the test pulses. It is not simple to adjust the test pulses to a real detector signal, or to generate a random frequency rather than a fixed pulse distance. Furthermore the signal to noise ratio is different. Detectors always show a certain noise level but a pulse generator has no noise. These problems must be taken into account to obtain a suitable test result.

4. Requirements for automatic fault monitoring

In this section a list of requirements which seem to be necessary for an automatic control system is given.

(1) For a given experiment not much test equipment should be necessary.

(2) The fault monitoring system must be able to detect defects automatically and it is valuable if it can localize them. The data must be presented to the experimenter in a format he is used to e.g. as spectra on a graphics display.

(3) To achieve flexibility the hard- and software components of the test equipment should be modular.

(4) When using microcomputers, the danger of complicated software should be avoided. As a single program cannot serve all purposes, the experimenter must have tools such as high level languages with modular structure. Furthermore, sufficient utility software should be available.

(5) The fault monitoring software should be portable.

(6) The microcomputer used must not become the bottleneck of the experiment.

(7) The microcomputer must have flexible communication interfaces. The data which are recorded by the on-line computer belong to the reaction which is the object of the experiment. The data from the equipment checks describe the conditions under which this reaction takes place. These conditions do not, in general, vary very fast, e.g. in minutes, whereas the reaction data rate is much higher, but they are correlated. Both these data sets should be archived on the same storage medium for later evaluation. In general, the on-line computer is unable to record the additional data from the fault monitor because of increasing dead time. Therefore the fault monitoring computer should be able to transfer the results of its checks to the on-line computer upon request.

(8) The installation of the fault monitoring computer must be simple.

(9) A failure of the fault monitoring system should not interrupt the experiment.

5. Components of the fault monitoring system

5.1. Hardware

For the implementation of the fault monitoring system a microcomputer with a Z80 processor was selected. For this processor the ECB bus is available. The reasons which led to this decision are:

- The microcomputer is built of individual Eurocard modules and therefore the fault monitoring device could be simply configured for a particular experiment.
- The processor and the process peripherals are set up on the same bus in the same frame and no further controllers or interfaces are necessary.
- The computing capacity of an 8-bit processor seems to be sufficient for this purpose.

The kernel of the microcomputer consists of three cards. One contains the CPU and 64 KB memory, one the floppy disk interface and the third the arithmetic processor unit. These cards can drive a graphics terminal, a serial link to a host computer, a printer, and two floppy disks with 1 MB storage capacity each. In the frame remain 15 slots for the process peripheral cards. The directly accessible memory of the 8-bit processor is limited to 64 KB, but for fast data buffering purposes a memory of up to 1 MB is addressed as a RAM-disk unit.

The wide range of process interface cards for the ECB bus system needs only simple adaptations to the devices normally used in physical experiments such as translators from NIM signals to TTL logic, etc.

5.2. Software

One of the central tasks of the design of the fault monitoring system was the selection of the programming language used in the Z80 processor. The real time process control language PEARL [4] was chosen because:

- it supports (quasi-)parallel processing,
- it supports access to the process interfaces at language level,
- it is possible to divide the program into modules,

- it can handle interrupts,
- the software is portable,
- it is easy to learn and therefore the experimenters themselves can write the fault monitoring programs,
- maintenance of these programs is relatively easy.

A package of graphics display functions is available for PEARL. It is a subset of the GKS (level 6.2) standard [5]. The operating system which provides a large variety of utilities (ranging from screen oriented editors to file transfer programs) is CP/M 2.2 [6]. A special method [7] for specifying the user programs can be used to describe the synchronization, parallel processing, interprocess messages, hard- and software, in a highly effective manner.

6. Components of the controlling system

The processes of the fault monitoring system can be divided into four parts:

(1) Processes dealing with the experiment such as
- generation of the test pulses,
- registration of the (test) pulses,
- switching of data paths and gates.

(2) Processes communicating with the experimenter such as
- adjustment of the test parameters,
- process management,
- display of the test results.

(3) Internal processes:
- periodic start of the test cycles,
- saving the test data,
- processing the test data.

(4) Communication processes:
- transfer of the test results to the on-line computer.

The analysis of these tasks and the requirements listed above produce the following structure for the automation software. First, tasks should be identified that can be implemented as individual program modules. This means that there will be programs for fault monitoring of the digital parts of the experiment and others for fault monitoring of the analog parts. Each of these programs is divided once more into a section for input and editing of the necessary test parameter set which is stored on the floppy disk, and a second section to process the tests using the stored parameters as input. This further division has the following advantages:
- It is possible to test intensively the parameter input for plausibility. This leads to the exclusion of many mistakes before testing.
- The set of parameters can be changed faster because only switching to another parameter file is necessary. No new editing is necessary.
- It is possible to create a new set of parameters while the control program is running on a different microcomputer.
- After a longer interruption the test parameters are still available and the system is usable at once.

The microcomputer presented in ref. [3] differs from the Z80 configuration described here mainly in its software. In ref. [3] no hint is given of a high level language for application programs. The authors even have to develop their firmware with cross-software. Therefore we suppose that their system has to be programmed in assembler language which requires an enormous effort to write the necessary software and is the central problem in the acceptance and performance of the equipment.

7. Applications

One way to classify the components of an experiment is to distinguish between devices whose signals carry digital or analog information (see sect. 6). This classification is based on two different ways of recording the signals. Digital signals are counted, whereas analog signals require ADCs.

Some control programs have already been developed and results from their prototype versions, one dealing with analog signals and the other with digital signals, are reported here.

First there is a program which is able to check the position of a peak within a spectrum and sends the actually calculated peak position to the on-line computer. This program was used in a (π, π', γ) angular correlation experiment [8] at SIN (Swiss Institute for Nuclear Research). In this experiment some 5 in.$\times$5 in. NaI scintillation detectors have been used to detect the γ's. In these detectors a light emitting diode (LED) is installed for amplification checks. Fig. 1 shows the gain, i.e. the position of the LED peak, of such a detector in a time interval of 350 min. This leads to a LED peak, which is displayed in the upper part of fig. 2. The results of the fault monitoring system, the varying peak

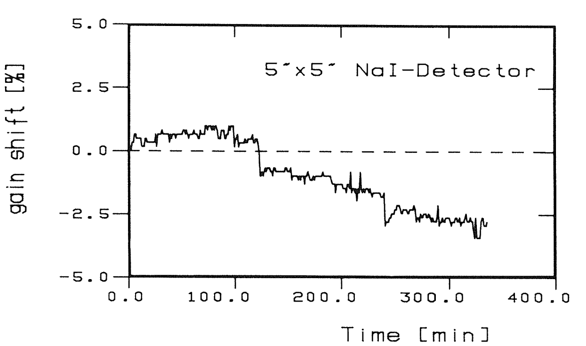

Fig. 1. Gain shift of a 5 in.$\times$5 in. NaI scintillation detector system, as a function of time, measured via the position of a LED peak.

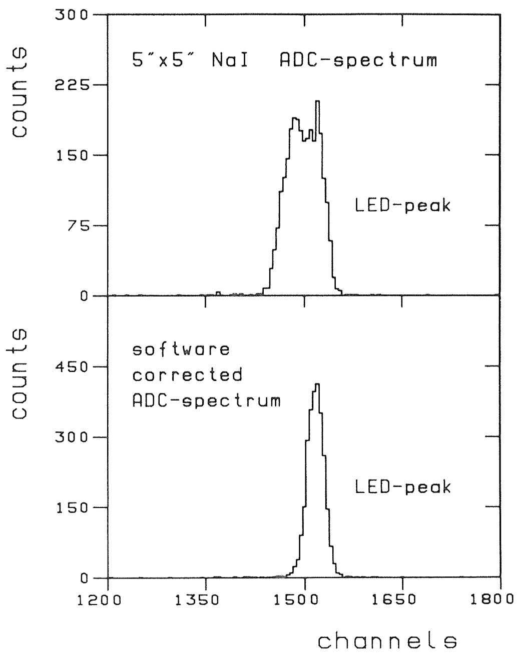

Fig. 2. Shape of the LED peak due to the gain shift displayed in fig. 1 (upper part). The same peak after a software correction of the gain shift (lower part).

position due to the varying amplification of the detector, was used for a software correction in the subsequent evaluation. The resulting peak after this correction is shown in the lower part of fig. 2, where the effects of the amplification instabilities are eliminated.

In addition a second program was developed, which is able to monitor count rates at 160 points in the electronics. A prototype version has been used in a (d, d′, γ) angular correlation experiment at the Erlangen Tandem accelerator [9]. Here failures of different components of the experiment have been detected by checking the constancy of the count rates. Some of the detected failures were:

- One of the discriminators lowered its threshold and it came to lie in the noise range of the detector signals causing an increased count rate.
- One detector became defect causing a count rate of nearly zero.
- The power supply of a crate broke down causing a count rate of zero at several points.
- The thickness of the target decreased by sublimation of the target material causing steadily decreasing count rates.
- The beam position on the target varied causing varying count rates with the same phases.

These two application examples show that the fault monitoring system can detect failures during the experiment and produces helpful information for the data evaluation.

8. Summary and outlook

The fault monitoring system described is based on a microcomputer and performs an automatic test of components of the experiment. This leads to nearly complete fault monitoring during data acquisition, and manual testing becomes unnecessary. The high level programming language and modular hardware used make the system simple to adapt to specific experiments. The test results and the recording conditions are not only present in a logbook, but also on the data storage medium and can be used in the evaluation. There are some more programs for fault monitoring as well as the two examples described in sect. 6. Prototype versions have been used in various experiments and further details of the planned versions will be reported later.

Acknowledgements

The language PEARL was implemented on the Z80 processor within a project of the Deutsche Forschungsgemeinschaft.

Furthermore we would like to thank the computing center of the Erlangen university for their help, which ensured fast development of the software.

References

[1] F. Hofmann and P. Holleczek, GIT-Z. Laboratorium 12 (1979) 1103.
[2] M.W. Sachs, Nucl. Instr. and Meth. 162 (1979) 719.
[3] M. Brendle, Nucl. Instr. and Meth. 219 (1984) 347.
[4] Deutsches Institut für Normung, Programmiersprache PEARL, Teil 1, DIN 66253 (Beuth, Berlin, 1981).
[5] Deutsches Institut für Normung, Proposal of standard DIN 006252 Information processing, Version 6.2, DIN-NI/UA 5.9/27-80 (Beuth, Berlin, 1980).
[6] Digital Research Company, CP/M operating system manual, Pacific Grove (1982).
[7] C. Andres, A. Fleischmann, P. Holleczek, U. Hillmer and R. Kummer, in: Kommunikation in verteilten Systemen I, Informatik Fachberichte Band 95, eds., D. Heger et al. (Springer, Berlin, Heidelberg, New York, Tokio, 1985) p. 588.
[8] R. Olszewski, Dissertation, Erlangen (1986) to be published in Phys. Rev. C.
[9] H. Ortner, Dissertation, Erlangen (1984) unpublished.