T. R. Lusebrink
C. H. Sederholm

Computer Facilities for the Laboratory

Abstract: Digital computers have become indispensible aids for many laboratory disciplines, allowing the performance of experiments which would be infeasible without the aid of a computer. Until recently, these computations have usually been carried out off-line, i.e., experimental data has been acquired in real time and subsequently processed on a large central computer. Small, relatively inexpensive digital computers first entered the laboratory as a substantial aid in real-time acquisition of data and control of experiments. Such computers, however, suffer severe limitations with regard to ultimate processing of the data. Hence the data processing must still generally be done on a large computer.

To realize the full potential of computer-instrument interaction, one should use the computer to: acquire data while performing requisite control of an experiment; calibrate, reduce, and compare the data with files of known physical parameters or theoretical calculations; and finally, produce desired reports and documentations of the experiment. In order to obtain results soon enough to effect the next experiment, i.e., in minutes, or at most, tens of minutes, all of the foregoing steps should preferably be carried out in a single computer.

Fulfilling all of these needs requires a computer too large and expensive to be dedicated to most single experiments. Therefore, a computer shared among several instruments is required. A computer system which simulates the independence of small dedicated computers, but which is also capable of performing medium- to large-scale computations when they are required, is most desirable.

Of course, many of these statements are controversial, and in order to investigate the validity of this approach an experimental computer system has been designed and implemented specifically for automation of multiple laboratory instruments in a time-shared mode. Experience gained from the simultaneous operation of a mass spectrometer, a nuclear magnetic resonance spectrometer, and two gas chromatographs is presented. The application programs and some proposed augmentation of these programs are also discussed.

Introduction

Computer augmentation of the functions of an instrument or a group of instruments usually takes place in a series of stages, each stage being somewhat more sophisticated than the previous ones. Let us consider, in particular, the automation of the broad class of instruments known as spectrometers or spectrographs. A characteristic of this group of instruments is that their output is a plot of one or a series of dependent variables as a function of an independent variable. With this definition, this classification encompasses a wide variety of instruments measuring properties such as electromagnetic radiation, tensile strength, ionic concentration, and numbers of particles.

The first stage in this sequence is usually data logging, in which the instrument continues to operate in the conventional manner except that the output is digitized, in addition to being recorded on a strip-chart recorder which is still retained as a primary record of the data. The digitized data is then processed in batch mode on a large computer. The main benefits of data logging are those of better precision and calibration, and in some cases a large reduction of labor. If all of the data from an instrument are used, rather than just that small fraction which is usually retrieved by "reading" a spectrum, the precision of the results can be increased substantially.[1] Using a computer, it is also easy to routinely incorporate statistical treatments which are usually too time-consuming to be done manually. By recording calibration spectra periodically, the computer-aided system can do a far more complete and thorough job of calibration of sample spectra than is normally done by a human operator.

The next stage of automation consists of open-loop control of the instrument during the data acquisition process. In open-loop control, the control parameters are determined prior to the start of the experiment and are not dependent on the data being acquired. This procedure provides for the unattended execution of a

The authors are located at the IBM Scientific Center, 2670 Hanover Street, Palo Alto, California

series of data acquisition steps, e.g., multiple scans over certain spectral areas for signal-to-noise ratio improvement, or spectral acquisition at a series of previously calculated crystal orientations of the sample.

The third stage of automation consists of closed-loop control of the instrument during the data acquisition step. In this mode of operation, control of the instrument is dependent on the data that are being acquired. For example, the scan rate of a spectrometer might be reduced as the output of the instrument moves away from the base line. The major benefit of closed-loop control is that the rate at which useful data is acquired is increased, i.e., the instrument is controlled in such a way that minimum time is spent acquiring nonsignificant data (baseline).

The fourth stage of automation is one of "on-line" (although not real-time) data reduction and interpretation. Very often, in the first stages, the data is acquired on one computer and is then transported in some intermediate form such as punched paper cards to a second computer operating in batch mode for actual processing. Since the turnaround time on such batch computers is usually several if not many hours, and since the data acquisition computer is often not kept busy, some data reduction and report generation in a background mode is often attempted on the computer used for the data acquisition steps.

This step immediately escalates the size of the computer, thereby presenting a conflict with the *small* dedicated computer concept. Nontrivial data reduction almost necessitates using more than 4K core memory and/or some bulk file capability; input/output on a typewriter begins to require large amounts of actual or elapsed time. Increased core and input/output capabilities quickly move one out of the low-cost computer class since such devices are expensive. These considerations lead one to consider sharing of a single computer for the automation of several laboratory instruments.

Instrument or laboratory automation

Let us first distinguish between instrument automation and laboratory automation. In the former case a single computer, usually small, is devoted to a single instrument, or at least to one instrument at a time; whereas in laboratory automation, a group of instruments in a laboratory is automated using one central computer system. A discussion of the relative advantages of these two modes of operation follows.

There are several advantages to a dedicated computer. The most important of these is that of isolation. Each individual user prefers to concern himself with his own problem. He does not want malfunctions of other instruments or their interfaces to jeopardize his experiment. Programming considerations associated with his own instrument are sufficiently complex that he does not want to worry about other users' programming problems too. He is rightfully afraid of being forced to factor parts of his programming requirements into general purpose programs that serve the entire laboratory (programming by committee).

Immediate accessibility to the computer system can be guaranteed in a devoted computer configuration. This fact also encourages one to favor multiple computers, one per instrument; however, if a time-shared computer system could almost always be made available in a period of a few seconds, or at most one or two minutes, this would not be intolerable. The prospects, however, of having to schedule one's experiments and schedule the use of the computer facilities long in advance is very unpalatable to most users. In general, users prefer fewer facilities which are routinely available to larger facilities available by appointment only. Hence, the desire for immediate computer access tends to favor multiple instrument automation over laboratory automation.

Relative costs should also be considered. A computer system which is capable of expanding to handle the requirements of an entire laboratory will, of necessity, be more expensive, even in a minimal configuration, than a computer system capable of automating a single laboratory instrument. Therefore, when one is taking the first step toward laboratory automation, that is, the automation of a single instrument, there is a very strong tendency to automate that instrument using one small computer, since the initial cost is considerably lower. Quite understandably, an organization is reluctant to commit large capital and manpower resources to a project in which it has little or no experience. However, if the ultimate goal of completely automating the laboratory is considered, many of the considerations discussed below imply that a single, shared laboratory computer would provide more performance per dollar.

To realize all of the potential of automation, one must not only acquire data but also control the instrument during the data acquisition step, process the data which have been acquired, standardize them, compare them against known parameters (e.g., compare an unknown spectrum against a file of spectra of known compounds for identification), and finally present the results in a form which is usable to the experimenter. Data acquisition and control steps can very often be adequately performed by a small computer. However, the data reduction steps, the comparison with data files, and the presentation of results in usable form often require much larger computer capabilities. One solution to this is to record the raw data, which have been acquired with a small stand-alone computer, on a recording medium such as magnetic tape or punched paper tape. These data may then be processed on a large computer when time is available. However, if the raw data are at all voluminous, and magnetic tape must be used, the cost of the magnetic tape drive relative to the

cost of the small computer can become very high. Hence, this approach tends to encourage one to minimize the quantity of data taken, often reducing the precision of results. Furthermore, the turnaround time on very large computer facilities still is much longer than the individual investigator would like to wait between epochs of his experiment. That is, if he could have the data from one epoch back quickly, these results could be used more efficiently to determine conditions for his next experiment.

A larger shared computer has several advantages for the automation of a laboratory. It can have sufficient core and processing capabilities to do a large portion of the data reduction required in most laboratories without resorting to a large central computer. By dynamic allocation of system facilities one may take advantage of the fact that most analytical and spectroscopic instruments have low duty cycles (i.e., much of the time is spent in sample preparation or with the instrument completely idle). This dynamic allocation of system facilities will allow one to reduce the total size of the required system below the sum of each experiment's individual requirements. In addition, under such a system, background data reduction tasks may be carried out by utilizing excess capacity that exists at any moment.

Another advantage of a shared computer is that more sophisticated input/output devices are available to all the users. The advantages of having access to large disk files, line printer, card reader/punch, and magnetic tape units are obvious. Only the largest devoted computer/instrument combination could justify the most modest of such devices.

Environment

Laboratory automation environments vary drastically according to their mission.[2] They range from laboratories serving production facilities to those associated with research and development. The research laboratory environment is far more dynamic in its computer requirements than is the production facility. In a research environment the user is continually modifying his experimental apparatus and computer programs. He wants complete independence and flexibility in his use of the computer, and he wants to be able to sign-on and sign-off the system at random. In the production laboratory, the application programs and instruments are less dynamic and the entire computer system load can usually be scheduled. In the production laboratory, instrument duty cycle (and hence computer system) demands may reach nearly 100 percent of the time, whereas in a research laboratory an instrument duty cycle of 10 percent is not unreasonable. It should perhaps be pointed out that a system which is capable of meeting the requirements of the research environment will have sufficient flexibility to meet the needs of most other laboratory environments, although the system overhead of an R-and-D oriented system may be greater than the system overhead associated with a production system. This paper deals specifically with automation in an R and D environment.

Interfacing philosophy

Many instruments have as their primary output an analog signal which conventionally drives a strip-chart recorder. As a first step in digital data acquisition, one often considers connecting this analog output to an analog multiplexor and shared analog-to-digital converter associated with the laboratory automation computer. (This is true only when multiple instruments are automated using a single automation facility.) This philosophy is easily implemented. Two possible problem areas should be considered before deciding upon this scheme of analog-to-digital conversion. They are signal-to-noise ratio and calibration.

Signal-to-noise ratio: In general, the automated instrument will be several hundred feet removed from the computer. When analog signals are transmitted over such distances, one increases the problem of noise pickup in the analog lines. This can be further aggravated if both computer and the instrument are locally grounded, introducing a ground loop.

In addition, one also finds a loss in signal-to-noise ratio inherent in any shared analog-to-digital converter, since the shared analog-to-digital converter will, in general, sample a signal only for a short period of time, typically ten microseconds. If the signal from one instrument is being sampled every ten milliseconds, this implies that the signal is being "observed" by the converter for 0.1% of the time, resulting in a theoretical reduction of the signal-to-noise ratio by a factor of the square root of 1000 under the signal-to-noise ratio attainable using an integrating analog-to-digital converter. This situation can be improved by analog averaging of the signal before it goes to the converter.

Calibration: A shared A/D converter can be triggered to convert a signal in one of two ways. It can be triggered at more or less equal intervals of time, or it can be triggered at equal increments of the independent axis of the instrument by the device providing the analog signal. This independent axis is not necessarily time. In either case, if the A/D converter is being shared, it will not always be available for an immediate conversion when a demand is made upon it. As a result, there will be an indeterminancy in the time at which the A/D conversion was made. If the instrument sweep is controlled by the computer so that the analog signal can be held indefinitely, this will cause no error; but if the instrument is scanning continuously, or if time is the independent variable of the instrument, such as in gas chromatography, this uncertainty in the A/D conversion time will introduce an error.

As already stated, time is not the independent variable of many instruments. An infrared spectrometer, for instance, has as its independent variable, wave length or wave number, or at a lower level, the position of the lead screw which turns the grating or Littrow mirror. In any case, processing of the acquired data requires that both the value of the independent variable and the dependent variable be known. The easiest method of determining the value of the independent variable is to take values of the dependent variable at equal increments of the independent variable. However, if time is not the independent variable, one must try to scan the instrument linearly in time so analog readings may be taken at equal increments in time, or one must allow the A/D converter to be signaled to make its conversion by the instrument itself rather than by the computer. The first alternative is somewhat difficult because of mechanical problems and also prohibits such useful devices as automatic scan suppression. (Automatic scan suppression is built into many scanning spectrometers and reduces the rate of scan as the first derivative of the signal increases, slowing down the scan as the peaks in a spectrum are passed.) Triggering the A/D converter from the instrument itself circumvents both of these problems.

The above considerations cause us to favor interfacing an instrument to a remote computer by using a devoted A/D converter at the instrument. (Due to the declining cost of good A/D converters, this approach is becoming economically feasible.) Furthermore, we believe that the A/D converter should be triggered by the instrument rather than by the computer. In many applications, the A/D converter should be an integrating converter rather than a sampling type. Transmission of the digital signal from the A/D converter to the computer should be in a demand/response mode. (Demand/response mode implies that the A/D converter makes a demand to the computer when it has digital data to present and the computer responds when it has taken the data.)

By devoting A/D converters to instruments, one is able to tailor them to the specific requirements of an instrument. For instance, when interfacing to a mass spectrometer which has a dynamic range of 2^{20} but a required accuracy of 2^{10}, a possible interfacing philosophy is to use three inexpensive, ten-bit A/D converters operating simultaneously on three different outputs with three different gains, hence providing both the precision and dynamic range required. This is a competitive approach to A/D converters with automatic range switching. Likewise, when instruments are limited in their data rates, less expensive digital voltmeters may be applicable.

On those instruments where time is not the independent variable of the instrument, the most flexible interfacing approach is to place the sweep of the independent variable under computer control. This approach allows nonlinear, even nonsequential sweeps which can, as will be demonstrated later, speed up the useful data acquisition rate substantially.

It should be noted that an instrument interface usually contains a substantial amount of digital logic in addition to the A/D and D/A converters. This additional logic allows the programmer to sense and control conditions in the instrument. It also allows the instrument operator to communicate with the system. The complexity and function of this portion of the interface is dependent on the type of instrument, the modes of operation of the instrument, and the amount of money available.

1800 Palo Alto Laboratory System

It has been pointed out here that the computer requirements of a spectroscopist may vary from one moment to the next depending upon the task he wishes to perform. Ideally, one would like to have access to a large computer, but one would like to pay for only the facilities required on a minute-by-minute basis, implying a shared system. Each user of the system would not wish to be concerned with any other user, either at the time he is using the system or at the time he is writing and debugging his programs.

Such a computer system was developed for the IBM 1800, a medium-sized computer. The system was specifically designed to service laboratory automation applications in the research environment described above. The major features of this system, the 1800 Palo Alto Laboratory System (PALS), are listed below.[3]

1. Flexibility is achieved by dynamic allocation of all of the computer's facilities. These include core, central processing unit, disk storage space, and use of the standard input/output devices. When a person signs on the system he is allocated only those facilities which he requires, as he requires them. When his requirements change, the system changes his allocation of facilities as needed. When the user signs off, all facilities he has been using are returned to the system for reassignment.

2. An individual user's program is isolated in that it is not necessary to consider other instruments which may be in the system, either at the time a program is written or when it is executed. In addition, programs need not be modified as a result of changes in the total instrument configuration. Programs are protected from each other by having the loader implement hardware storage protection for all words which are not changed during execution, e.g., data buffers and variables. In addition, the first and last words on each page are storage-protected to prevent overruns from adjacent programs.

3. All data acquisition and control is done via real or programmed data channels. The user's program need only specify a task to the system to acquire a block of data from the experiment. The system then handles the acquisition of the data, and the user's program is brought back into

play only when that block of data has been acquired by the system. Thus the user's program is not responsible for gathering datum by datum and hence need not be in execution while data are being acquired.

4. PALS offers a guaranteed turnaround time, nominally 50 milliseconds, assuming a limit of five real-time programs in simultaneous execution. That is, from the time that a data block is completed until the application program demanding that data block can gain control of the computer, and gain 5 milliseconds of processing time (usually more than sufficient to process the data block and return some control commands to the instrument), is guaranteed to be no more than 50 milliseconds. (Note that data blocks may be one datum large.)

5. All data stored on the disk are stored in the form of logical tapes. This means that the size of the data file need not be specified, but expands or contracts automatically. A given application program may also access multiple logical tapes.

6. The programming language supported by PALS is a macro assembly language which offers many statements very similar to FORTRAN.[3] However, it offers in addition many statements which are far more flexible than FORTRAN statements, in particular those dealing with the operation of the laboratory instrument and those dealing with data input and output.

7. Users' programs are assembled in background mode. When an assembly is required, a partition is reserved in core by not reassigning the 12 pages at the high end of core as they become free. When this core block becomes available, the assembler is loaded into this partition. When the assembler has completed its operation, this partition is returned to variable core. This means that several minutes may be required from the time it is requested to load the assembler until the assembler is actually loaded. However, since this system considers that assembly should be a background operation, this allocation scheme was chosen.

8. Machine errors that occur during the execution of an application program, e.g., storage protection errors or operation code checks, are detected by the monitor system. The monitor determines which program has caused the error and terminates that program. The operating system is then restored to its previous mode of operation.

Observations of an operating system

A joint study was carried out with Varian Associates in an effort to investigate and demonstrate the usefulness of a time-shared, laboratory automation computer system. A laboratory was established which simulated a research environment. It contained an M-66 medium resolution mass spectrometer, an A-60 NMR spectrometer and two Aerograph gas chromatography columns. Each of these instruments was interfaced with the computer via prototype Varian Spectrosystem 400 interfaces. These interfaces were designed to incorporate most of the interfacing criteria discussed above.

Each instrument interface provided facilities which allowed the spectroscopist to have complete control over his use of the computer from his spectrometer, which was remote from the computer. These facilities consisted of pushbuttons and thumb switches. A set of prompting lights, operated under program control, served to indicate the present status of the operating program. Functions executable from the remote console included loading and aborting application programs, controlling branch points within the application programs, and entering parameters during the execution of programs. In general, the recorder associated with a particular instrument was used as the graphical output device, while tabular reports were printed out on a central, shared line printer.

The computer was equipped with 32,768 words of two microsecond core. All of the instrument interfaces were attached to the computer system through a digital multiplexor channel. Two or more subchannels were associated with each interface. The system was equipped with three disk files (each with a capacity of 500,000 words), a line printer, a card reader, an incremental plotter, and a Varian STATOS electrostatic plotter. All of these instruments could make independent, simultaneous use of the computer system without taxing the computer's capacity. In fact, simultaneous operation of all four instruments plus card reader, line printer and incremental plotter did not overload the system. Following is an outline of the automated experiments implemented on each of these instruments, with comments as to additional capabilities which could easily have been added.

Each instrument had associated with it a master program resident on the disk. When an experiment was to be performed, the master program was loaded into the core memory by depressing a pushbutton on the interface. The function of the master program was to initialize the spectrometer and to provide a choice of the available application programs to the user. After the master program was loaded, it armed appropriate pushbuttons for program selection. This was indicated by lights behind the several pushbuttons. When one of these programs was selected, it was loaded and the master program would exit (release its core). After the subprogram completed its function and before it exited, it reloaded the master program. This mode of operation minimized the core requirements of a single user. Specific program functions are described here.

M-66 Mass Spectrometer

The interface between the computer and the mass spectrometer enabled the computer to control the magnetic field, the *x-y* plotter incorporated in the spectrometer, and

the period of the local interval timer which controlled the rate of data acquisition from a 14-bit A/D converter. The application programs could be loaded from the interface. In addition to the master program, five additional programs were written for the mass spectrometer. These were (1) mass calibration, (2) data acquisition under computer control, (3) data re-presentation, (4) identification of pure compounds from their spectrum, and (5) compound identification from manual input of the five most intense peaks. These programs are described next.

Mass calibration

This program was used to adjust the gain and offset of the D/A converter which drove the mass scan (magnetic field) of the spectrometer. It was convenient to equate a digital value of 0 to the mass presented on the sweep offset dial (initial mass for the sweep) and a digital value of 16,000 to a mass equal to the offset plus the sweep width (the maximum mass to be swept). This program was used only occasionally and therefore was not called by the general master program but rather as a stand-alone master program. The calibration program generated a sawtooth scan over two peaks in a reference compound (PFA). The peaks were selected at the extremes of the possible sweep. To perform a calibration, the program was first loaded from disk and set into execution. A repetitive sweep over the low mass reference peak was started. One could switch to the other reference peak and back by pushing one of two buttons on the interface. Thus, with a calibration compound in the spectrometer, it could be calibrated by setting the sweep width and offset controls to appropriate values and then adjusting two potentiometers until the reference peaks appeared in the middle of their respective sawtooth scans. After the mass calibration had been completed, the program was caused to exit by pushing a button at the remote interface.

The remaining programs were called by loading the master program and subsequently selecting the desired program.

Data acquisition

Parameters defining the initial mass, mass range, and sampling rate were entered by the spectroscopist via the remote interface. A linear scan table of 16,000 points was generated and stored on a logical tape on the disk. Upon command from the interface, the scan was started and data points taken and stored on another logical tape for subsequent processing. Options which could be exercised from the interface allowed the user to start a scan (with or without concurrent plotting), stop the scan at will, set the limits of the scan, and set the speed at which the scan was made.

Computer control of the magnetic scan over the mass range, of course, allows alternate modes which have certain advantages. For example, in low-resolution mass spectrometry only those peaks at integral masses are of interest. Thus, if the scan is perfectly linear over the mass range, one could calculate a scan table which would scan the magnetic field only over the integral mass values plus or minus five percent. Assuming the magnet can rapidly follow the jumps (which is the case for the present spectrometer over one mass unit), a saving of 90 percent of the scan time can be realized by selective, controlled scanning in this open-loop control mode. Even if one could not assume a linear, or time invariant relationship between mass and D/A converter reading, the same effect could be accomplished under closed-loop operation, in the time-sharing mode, although at a higher cost in computer overhead. Starting at a known mass, e.g., N_2 at $m/e = 28$, a small scan table could be generated assuming a linear relationship, to bracket the $m/e = 29$ by $\pm n$ percent. After the scan of $m/e = 29$ had been made, the real-time application program could search the acquired data to determine if a peak was present in that interval. If no peak was detected, a scan of $m/e = 30$ could be calculated and executed, again assuming a linear (or other) relationship between m/e and D/A value. This loop could be continued indefinitely until a peak was discovered. As the extrapolation became longer and longer, the scan window might be increased in size. When a peak was finally detected within a scan window, a correction could be made to the assumed relationship between m/e and D/A value, the scan window width could be reduced to $\pm n$ percent again and the process could be continued. Such a scheme was not actually implemented to control the data acquisition during the study; however, a modification of this scheme was successfully used to speed up data handling in the compound identification program.

Let us consider the timing associated with these methods of scanning. Let us take as a standard a linear scan at a rate of one mass per second. At this rate, an individual mass peak would be traversed in 0.1 seconds, since under normal conditions the mass peaks are 10 percent of the intermass distance. Under this linear scan, a scan of 500 mass units would take 500 seconds = 8+ minutes. Using the open-loop incremental scan mode, 100 milliseconds would have been spent on each of 500 masses, yielding a total of 50 seconds, an improvement of a factor of ten in scanning time, while the resolution and signal-to-noise ratio remain constant. Using the closed-loop control function of the system, nominally 100 milliseconds is still spent on each mass window, but in addition a maximum of 50 milliseconds will be used waiting to gain control of the computer for the closed-loop calculation before the next scan window can be initiated. Hence, under this mode of operation, scan time would approximate 150×500 milliseconds = 75 seconds. If the mass spectrometer were the only user of the system at that time, the scan would be

completed in a nominal 53 seconds. In either case, the self-calibrating scan is still much faster by nearly an order of magnitude than the linear scan, and in addition, self-calibration is incorporated in the scan. This latter approach could not be implemented without closed-loop control capabilities.

Data presentation

This program allows the presentation of any portion of a previously acquired spectrum to be expanded to full scale in both the x and y directions and replotted on the mass spectrometer recorder. This playback program is initiated from the instrument interface; all parameters, such as the mass range limits of the spectrum to be represented, are entered through the interface.

Compound identification from the spectrum

This program processes a spectrum which has been previously acquired and stored on the disk. The processing consists of searching the spectrum for the mass numbers and intensities of the highest peaks, followed by a file lookup program which identifies, if possible, the pure compound on the basis of its five highest peaks.

This program searches a spectrum which had been recorded with a linear scan and stored on the disk; however, the search was carried out in the incremental manner described above under the section entitled Data Acquisition. The program searched only the data points in regions around those positions where a predetermined calibration curve predicted that the integral masses would occur. If a mass peak was found, the calibration curve was dynamically corrected for the remainder of the search. As was pointed out earlier, this scheme could have been used during the actual scan to speed up the scan, rather than being used simply as a dynamic calibration technique at the peak searching time.

Compound identification from the five most intense peaks

For those cases where spectra had not been taken under the present system, a program was written which allowed one to enter the masses and intensities of the five most intense peaks in a spectrum. After the intensities of the peaks were normalized, the same subprogram used for the previous compound identification performed the table lookup and report generation.

After the five most intense peaks were determined, either from the spectrum or from manual entry, a table consisting of a subset of the ASTM Index of mass spectral data was searched to determine if the unknown compound matched any compounds in the table. All compounds in the table which had the same five highest peaks as the unknown were printed out. The experimental and tabulated values of the individual intensities were listed, and a single parameter, equal to the root-mean-square deviation of the observed intensities from the file intensities, was also listed. The file lookup program was designed in such a way that a file of 5000 entries would not take an unreasonable length of time to process.

Throughout this entire set of programs, except for the generation of the final report, the only access to the computer which the experimenter needed was through the instrument interface, or through the x-y plotter of the instrument. The report was printed on the centralized, time-shared line printer. Even with a few entries in the table of possible compounds, the report was sufficiently voluminous that it would have been quicker to walk several hundred feet to the central computer and pick up the report than it would have been to wait for the entire report to have been generated on a remote typewriter. Obviously, the value of remote report generators (typewriters) depends upon the distance of the instrument from the computer facility.

• *A-60 NMR spectrometer*

The interfaces for the A-60 NMR spectrometer and M-66 mass spectrometer were very similar. The A-60 interface contained an A/D converter, an interval timer, and a D/A converter for the scan control. In addition, two more D/A converters were provided for control of the magnetic field homogeneity. The x-y recorder of the instrument was under control of the computer, and lighted pushbuttons were provided as part of the instrument interface, as described above. The NMR master program armed four pushbuttons on the interface which could then be used to select any one of the programs described below.

Time-averaging data acquisition program

This program controlled the data acquisition and allowed several options regarding the mode of recording the spectrum. Parameters entered via the interface consisted of the initial and final sweep positions of the magnetic field in ppm from a standard position, sampling rate, number of sweeps, and number of samples to be taken at each value of the magnetic field. This allowed signal-to-noise ratio improvement by digital averaging (taking multiple readings the of NMR signal at given values of the magnetic field), or by "catting" (taking multiple sweeps), or both. The spectrometer scan was controlled by calculating the prescribed table for a particular scan and transmitting these values to the interface. The D/A converter at the interface converted these values to their corresponding magnetic field in the spectrometer. The resultant signal was then digitized by the interface and sent back to the computer.

Several options were available for data presentation during the data acquisition phase, all controlled through the interface. If only a single point per scan position was

taken during a single scan, the NMR recorder simultaneously recorded the spectrum. If time averaging was being performed, the plot was not recorded until the end of the time averaging; however, it was possible to stop the data acquisition function at any point and have the averaged cumulative curve plotted in order to see whether the data had been sufficiently improved by the time-averaging technique. If the improvement was adequate, the program could be terminated; if not, the data acquisition could be resumed.

Digital smoothing

In addition to the time averaging described here, it was also possible to use digital smoothing methods to improve signal-to-noise ratio. This program read data from a logical tape, smoothed it, and returned the smoothed data to another logical tape on the disk.

Magnetic field homogeneity adjustment

This program optimized the curvature and y-gradient of the magnetic field in essentially the same manner described by Ernst.[4] Its operation was completely automatic after the program was loaded into core. The program operated by calculating a coarse grid of values to cover ranges of curvature and y-gradient settings for the magnet. The computer then positioned the controls at these settings and recorded the height of a standard peak for each setting. The position which yielded the maximum intensity was used to calculate a finer grid of points about that point. The points in this second grid which yielded the best signal were then used as the basis for yet another finer grid, and so on until no further improvement was noted. The user could execute this program at any time he chose, or the program could be called into execution periodically by the "catting" program in order to correct the field automatically at any desired interval.

Spectrum simulation

Sometimes it is desired to simulate a spectrum by calculating a sum of known peak positions. This program used a series of line positions and intensities together with a line half-width to calculate a sum of Lorentzian peaks. The resulting "spectrum" was then plotted out on the NMR recorder.

• *Gas chromatography*

Two gas chromatographs, a flame ionization and a thermal detection type, were coupled to the computer through a single interface. The interface had the standard set of back-lighted pushbuttons and contained one A/D converter capable of automatic ranging. When the master program was loaded into core, the interface was activated so that initial parameters for the total system, i.e., both chromatographs, could be entered. These parameters included the total sampling rate for both columns and sensitivity factors which controlled peak detection using first and second derivatives as criteria. If it was not desired to run both chromatographs under the same conditions to sense peak maxima, minima, and inflection points, operating parameters for either chromatograph could be changed by activating a sense switch that identified data sent from the interface as belonging to that instrument only.

Signals from both chromatographs were continuously digitized by the interface once the master program had been loaded into core. The data were merely cycled into a buffer, however, until the sample was injected and the start button for that particular chromatograph was activated. When one or both of the chromatographs were started, the data acquisition program then sorted the data and stored it on separate logical tapes for each instrument and run. Pressing the end button terminated the scan for an instrument. After a sample run had been terminated, analysis was carried out by the two programs described below.

Peak definition and integrated areas

This program used first and second derivatives to sense the beginning, maximum or inflection point, and end of each peak. The peaks were further defined as to type, i.e., single peak or one of an unresolved group of peaks. The areas of the peaks were integrated to a base line of zero, so that if the actual base line was above zero, corrections could easily be made by subtracting the trapezoidal area from the uncorrected peak area. These uncorrected areas and peak positions were stored on a logical tape on the disk for additional processing by the following program. These calculation programs could operate simultaneously under time sharing with each other and with the data acquisition program. For example, both chromatographs could be running samples simultaneously with the data being stored on separate logical tapes. The names of these tapes were passed on to the peak definition program to be processed as rapidly as it was able to, with its output being written onto another set of logical tapes whose names were passed on to the third program. Each program worked on its input set of tapes at its own speed so that even if the real-time data acquisition program were to get ahead of the processing programs, they would automatically process the data as time was available until the entire set was completed. With only four instruments on the operating system, of course, the computer processing was easily able to keep up with the data acquisition programs.

Calculation of base-line-corrected peak areas

In the present study, only two options for the calculation of areas of unresolved peaks were implemented.[5] The first method was to drop a vertical line from the minimum

between two peaks; a reasonably satisfactory estimate if the peaks are moderately well resolved and not too asymmetric. When relatively small peaks follow a large asymmetric peak, the user could select the option of an area "slice" for the smaller peak. This slice is roughly equivalent to smoothing out the larger peak and equating the area of the smaller peak to the area above this smoothed line.

This program calculated the net peak area by subtracting the trapezoidal area between the actual base line and zero. The total area was then normalized to 100 percent and the area of each peak was calculated to give nominal weight percent of the sample. The output report was printed on the line printer and contained identification by job and instrument number. All samples were considered as unknowns, so the peaks were listed in order of their emergence. Peak numbers, elution times in seconds, and weight percents were presented in the report.

Additional programs

Since the previous application programs did not require all of the 1800's time even when all devices were operating simultaneously, additional programs could also be run. These could consist of programs called by the real-time data acquisition programs to further process the data, or they could be programs which used data from other sources. Examples of such programs are: a spooling program which read a deck of cards and printed a listing on the line printer; a plotting program which plotted a previously prepared logical tape on a digital plotter.

Another example involved a series of linked programs to process data from a Varian C-1024 time-averaging device. A spectrum was dumped from the C-1024 to punched cards in an octal format. The first program read the octal data from cards to disk in order to free the card reader as soon as possible. These data were then converted to decimal integers and stored on a logical tape. This program then called a second program and relinquished its space in core. The second program searched the logical tape for maximum and minimum points and shifted the base line to zero by subtracting the minimum from all points. This program then called a smoothing program and relinquished its space in core. The smoothing program carried out a nine-point digital smoothing calculation on the data in the logical tape and called a plotting program before it relinquished its space in core. The last program scaled the smoothed data, "plotted" the data on the line printer and then relinquished its core space. A minimum of core was required by this linking process because each program gave up its core as soon as it had called its succeeding program.

Conclusions

In general, the attachment of a digital computer to spectroscopic instruments can substantially augment the capabilities of the instruments. Operation of the instrument and acquisition of data can be automatically controlled to obtain the greatest amount of data in minimum time under optimum operating conditions. These functions can be carried out under open- or closed-loop control of the instrument by the computer. Much tedious labor formerly done by hand can be done with the computer in the areas of data acquisition and data reduction. This allocation of tedious jobs to the computer makes it reasonable to consider doing experiments formerly assumed to involve too much data manipulation. Even more exciting is the potential of being able to perform experiments which would be infeasible without the aid of a computer.

Large-scale data reduction and file capabilities available on the data acquisition computer can result in rapid turnaround time from data acquisition to final report generation. Ideally this time will be short enough to allow the experimenter to get his results back in time to affect the course of his next experiment.

A medium-sized time-shared computer system which allocates its facilities among several instruments on a dynamic basis has been developed and demonstrated. This system can provide very flexible and useful laboratory automation support to multiple users. The application programs for the various instruments are completely independent of each other; that is, each application program can be written and executed as if it were the only program running in the computer. Each user may take advantage of all of the high-level input/output devices, such as card readers and line printers, while sharing the cost of them with other users. Sufficiently rapid closed-loop turnaround time for most spectrographic applications can be provided to multiple simultaneous users (e.g., with five users, each user can be guaranteed 5 milliseconds of processing every 50 milliseconds). A high-level, automation-oriented language is available, allowing the users to modify present application programs or generate new application programs with far greater ease. Typical data acquisition and control programs for this time-sharing system can be written and debugged with no more effort than is presently needed to write a data reduction program for execution on a centralized batch processing computer.

Acknowledgments

The authors wish to express their thanks to Varian Associates for providing the instruments and prototype instrument interfaces used in this investigation. These devices were provided as part of a joint study by IBM and Varian Associates. We wish to further thank William Beaver, Hanspeter Benz and Fred Lee of Varian for all of their efforts in the joint study. Finally, we wish to thank our associates at IBM including Paul Friedl, Chris Jenny, Diana Bloch, Donna Nelson, Michael Bell and Jon

Petersen for their work in providing the PALS monitor system and in coding parts of the application programs.

Footnotes and references

1. Consider the problem of calculating the composition of a binary mixture from an infrared absorption band containing substantial absorption from both compounds. Measurement of the transmission at two wavelengths would provide two equations in two unknowns, a completely determined system assuming no noise in the measurements. However the statistical precision of the answers is approximately proportional to the square root of the number of simultaneous equations minus the number of unknowns. Hence if 102 transmission measurements were made at different wavelengths, the precision of the answer would increase by roughly an order of magnitude.
2. J. W. Frazer, *Anal. Chem.* **40,** 26A (1968); J. W. Frazer, *Sci. and Technol.,* p. 71 (July 1968); and D. Secrest, *Ind. Eng. Chem.* **60,** No. 6, 75 (1968).
3. Details of the PALSystem are given in IBM Scientific Center Report No. 320-3244.
4. R. R. Ernst, *Abstracts of Pittsburgh Conference on Analytical Chemistry and Applied Spectroscopy,* Pittsburgh, Penna., March 5-10, 1967.
5. In order to provide significant improvement in peak resolution over these two options, it would be necessary to analyze the line shapes in detail. This procedure is quite complicated because the line shapes in gas chromatography are not easily defined mathematically. Several programs for carrying out such analyses are available in the literature; however, for the gas chromatography application, a certain amount of empiricism cannot be avoided.

Received June 21, 1968