Development of a Fourier Transform NMR Instrument Simulator†

Kenneth L. Ratzlaff*

Instrumentation Design Laboratory, Department of Chemistry, The University of Kansas, Lawrence, Kansas 66045

David VanderVelde

NMR Laboratory, Department of Medicinal Chemistry, The University of Kansas, Lawrence, Kansas 66045

Received July 28, 1992

A software simulator for a Fourier transform NMR instrument is described which provides a realistic experience and context-sensitive help for the user. Real-time programming techniques are required in order to achieve accuracy so that asynchronous activities in the instrument may be simulated. The help support allows an instructor to tailor the messages to be displayed for any desired level of rigor.

INTRODUCTION

A modern chemistry laboratory curriculum should include techniques involving major instruments such as Fourier transform (FT) NMR. However, these instruments are not universally available, or their availability for instruction may be sharply limited. Schools with a strictly undergraduate educational mission usually cannot justify the cost and maintenance of a medium-to-high-field instrument. On the other hand, research-oriented institutions possess instruments purchased with research funds and usually cannot release sufficient time to laboratory classes for the students to train on and use the instruments in lab courses. Furthermore, funding agencies do not have the resources to provide these instruments as educational tools to all deserving institutions. At most institutions, the students generally learn spectral interpretation from low-field continuous wave (CW) spectra which are now seldom used in academic or industrial research. Only a passing glance is given to instrument operation; when given the usual "lab tour" it is difficult for them to understand the operation of an instrument when every aspect is new.

Consequently, educators often must find alternatives to dedicating time on expensive instruments for instructional use. A computer-based instrument simulator can be a partial solution, and in this paper one such approach is presented.

Although simulators have not been used extensively, one can point to at least three different ways in which they are used, both pedagogically and in the development of intelligent instrumentation.

The *method trainer* is the first simulator category. In one example, simulation of a generalized high-performance liquid chromatograph (HPLC)[1] places attention on the method, not on any particular instrument. Success in using the HPLC software depends on making correct choices of operational variables including solvent, column, and detector. The general method trainer allows the programmer to compromise between the needs of the method and the ease of programming. Standard linear programming techniques can be used.

In the second category, *development simulators* have been devised to simulate the instrument's output for use with other instruments. One type generates simulated analog signals. For example, an electronic peak generator[2] and an electronic general instrument signal simulator for studying computer interfacing[3] have been described. A novel approach is a software driver, which acts as a data acquisition simulator for both the software development process and the training of students before the instrument is used. In one case, the data acquisition driver for flow injection analysis (FIA) software is a separately installed module.[4] As a convenience to the programmer developing the main program, a simulator was developed which mimics an actual data acquisition/control system connected to FIA equipment. While this module was invaluable to the software developer, it also can be used to convert the actual FIA program to a teaching simulator.

The last category is a *specific instrument simulator*. For example, a simulator for the Varian EM360,[5] a low-field continuous-wave instrument, has been used in preparation for students using the actual instrument. The computer emulates the electronic console: the display, the homogeneity and phase controls, and the plotter. Samples can be selected from a database; the software also simulates the effect of incorrect adjustments.

At the University of Kansas, QSIM,[6] a simulator for a user-friendly, computer-based, routine instrument, the General Electric QE-Plus 300 MHz NMR,[7] has been developed. Our goals were to support several groups of students (typically 20 in a laboratory section) in a short time (typically 30–45 min) such that they will have the skills needed to operate the instrument effectively, and they receive an orientation to practical pulsed Fourier transform NMR spectroscopy.

The development goals can be classified into two groups:

(1) to be *faithful* to the "look and feel" of the actual instrument in all operations so that the student will be confident that the simulation experience will provide preparation for actual instrument use

(2) to be *gracious* to the user by providing appropriate help whenever the student desires and by being tolerant of user errors (even those to which the QE responds by aborting the current operation)

THE INSTRUMENT

The QE is a fully computer-driven, pulsed Fourier transform instrument. All user interaction (except for sample insertion) takes place through a computer. A single knob (simply a shaft encoder detected by the computer) has replaced all the

† Presented in part at the Xth ICCRE Meeting, Jerusalem, Israel, July 1992.

0095-2338/93/1633-0303$04.00/0

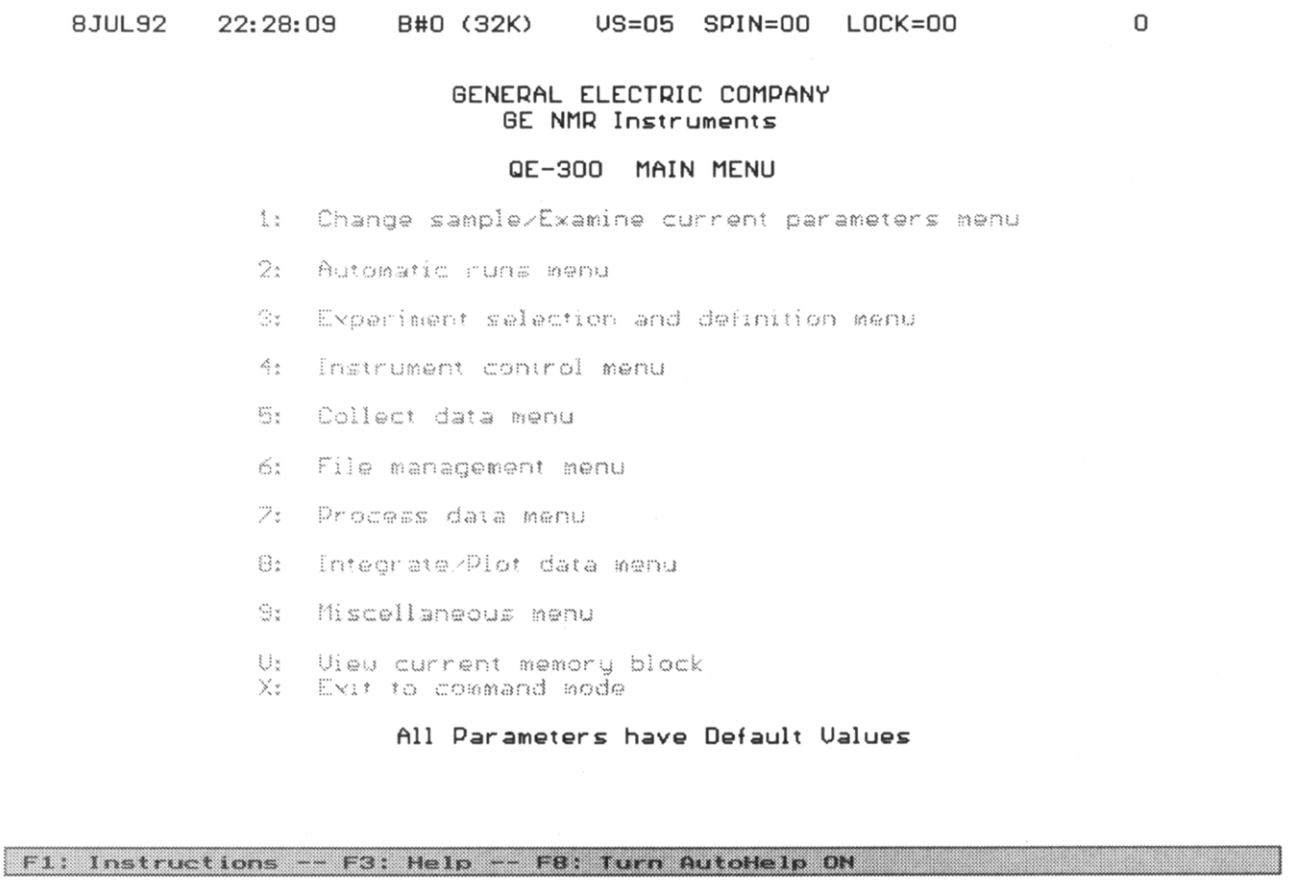

Figure 1. Menu typical of both the QE and QSIM.

old front-panel knobs; software toggles have replaced on–off switches or buttons.

Operations are most conveniently controlled through a sequence of menus. An entry in one menu usually leads to another menu of further options. Other entries lead to operations on the instrument including data acquisition and homogeneity adjustment. The menus may be edited by the manager of the instrument, modifying wording, sequences, and functions. For this instrument, the contemporary paradigm of windows is not used; its software does not include on-line help or manuals. The absence of these features is more or less typical of the current generation of major turnkey instruments.

Samples are exchanged pneumatically, either manually or by a 100-position automatic sample changer.

The QE's computer interacts continuously with the instrument, updating data areas on the screen to show the time, temperature, gain, spin rate, and lock level in real time. During the actual acquisition sequence, the instrument displays the accumulating FIDs, and then goes through processing and a standard plotting sequence followed by a display for interactively viewing and plotting the data.

THE SIMULATOR PLATFORM

The QSIM simulator executes on a common computer, the IBM PC computer or clone thereof. The images are most accurate when a color VGA display is employed although the software will run with an EGA display or a Hercules monochrome display. Obviously, all color is lost in the case of the latter. Some operations will be noticeably faster if a math co-processor is available, and the display speed only equals that of the QE if the computer's CPU is a 25-MHz 80386 or better. Extended or expanded memory are not required. QSIM will run very effectively over a network with several simultaneous users (as we do with laboratory sections).

Spectra generated by QSIM may be plotted on an HPGL-compatible (Hewlett-Packard graphics language) plotter (such as Hewlett-Packard plotters) or a Laserjet printer (Laserjet II with an HPGL cartridge or Laserjet III). A mouse can be useful although it is not required.

In the classroom, we use a group of 80386-based computers networked via a local area network (Lantastic, Artisoft, Tucson, AZ) to a server with the only hard drive. QSIM will arbitrate concurrent requests so that all computers can share the same files, data input files, and output files.

THE SOFTWARE

Simulator software must provide a very accurate "look and feel" of the actual instrument. That emulation encompasses four areas:

- Realistic simulation of real-time instrument responses
- Accurate screen images
- Accurate FID and spectral data
- Adequate representation of noncomputer instrument functions

In addition, there are functions that are not included in the instrument:

- Context-sensitive help
- Gracious error handling.

Real-Time Instrument Responses. The simulation of instrument monitoring functions utilizes real-time programming techniques, a significant break from conventional linear programming techniques. Within the program, concurrent operations (multitasking) must be supported in order to simulate the asynchronous activities which take place within the instrument, and those operations are able to pass messages

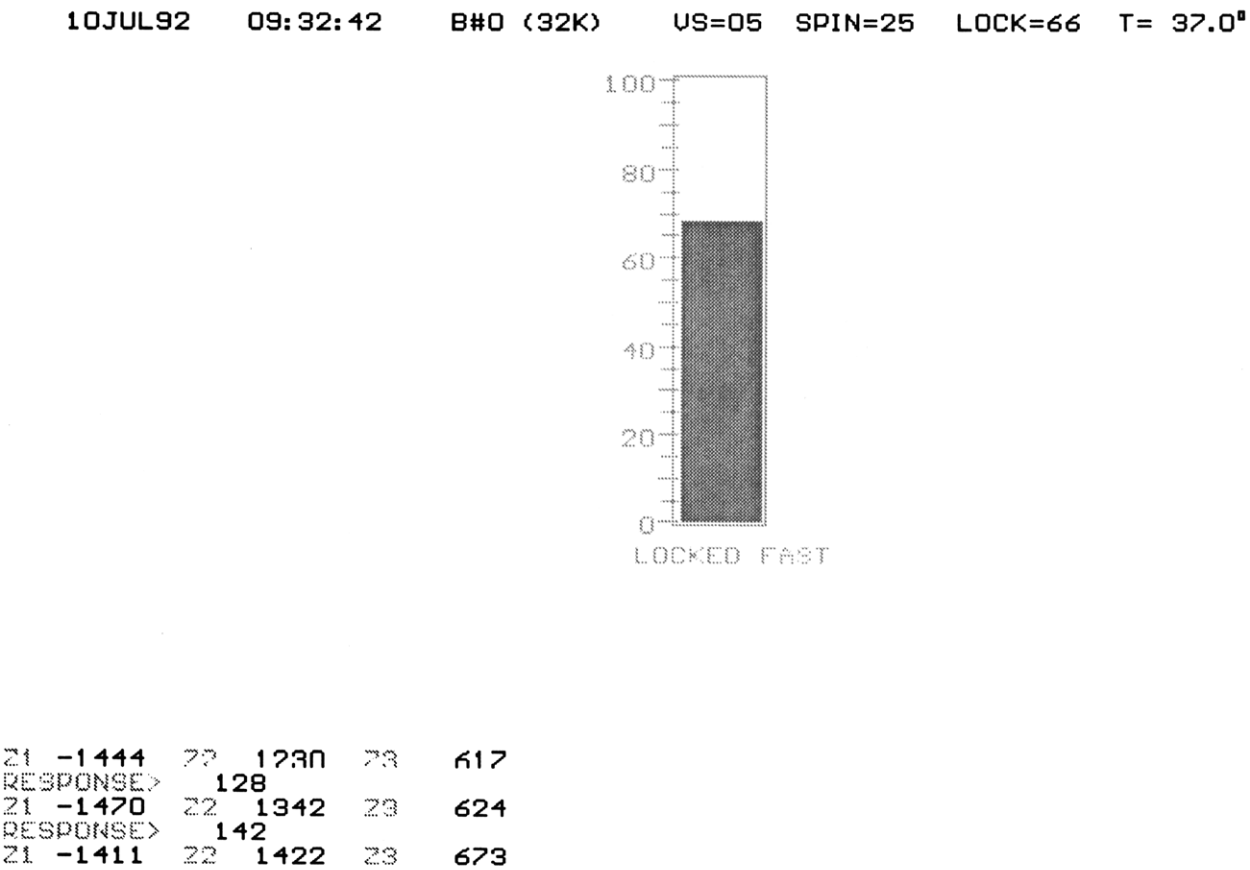

Figure 2. Illustration of the Compu-Shim process screen.

back and forth. While keyboard or graphical (mouse or encoder) data entry takes place, numerical values in data areas on the screen must be updated with simulated values and variations in those values.

To accomplish these concurrent operations, an interrupt-service routine (ISR) is attached to the time-of-day interrupt and made responsible for all simulated monitoring operations. Consequently, operational variables such as temperature, spin rate, time, and lock level can be updated on the status line (the top line, shown in Figure 1) or elsewhere on the screen without interfering with the foreground function of interacting with the user. For example, if the operator changes the air flow rate, that datum is passed to the ISR, which can appropriately ramp the sample spin rate, display that rate, and illustrate the simulated moving knob in an "action window" described below.

Communication between the two processes occurs via a block of variables that can be accessed by both processes. At each interrupt (about 18 Hz in standard PC operation), the ISR examines the communication block to determine if the main program has updated any variables that affect the ISR. The ISR also calculates other changes that are triggered by messages from the communication block.

The real-time bar graph that displays the status of the deuterium field-frequency lock level is another monitoring function which takes place while the shims and other parameters are being adjusted. It is used during the "Compu-Shim" procedure in which an optimization routine is used to maximize the homogeneity by adjusting three or four shim coils. In QSIM, the Simplex algorithm[8] is employed. During the process, illustrated in Figure 2, the user will see the variables change and the lock, displayed as a bar, optimized. The shim coil variables are passed to the ISR, which computes the lock value, includes a small amount of random noise, and displays that value continuously both as a bar and in the status line, thereby giving the look of real-time acquisition.

Screen Images. The menus appear identical to those on the QE. One such menu screen is reproduced in Figure 1. As a requirement of accuracy, menus on the QE can be edited by the system manager. Consequently, the simulated menus can be edited by an instructor to match any changes made to those on the QE; each menu contains a file with the menu entries and a technique for encoding the operational flow from one menu or operation to the next.

PC graphics systems have both a text mode and graphics (bit-mapped) mode. To provide seamless transitions between screens requiring graphics and those that do not, all screens use the bit-mapped mode.

FID and Spectral Data. It is essential that the acquisition of realistic data is simulated. The data are "acquired" from disk files of data derived from actual QE data; both the actual FID and the actual spectrum are used, and both proton and carbon data can be included in a single file. (A procedure for copying and transferring spectra from the proprietary format used on the QE's $3^1/_2$-in. diskettes has also been developed.[9]) The screen image of a typical proton FID is illustrated in Figure 6. For proton spectra on the simulator, as in the instrument, a fixed number of FIDs are acquired and averaged until a preset number of repetitions is reached; for carbon, the acquisition continues until a satisfactory signal-to-noise ratio is attained. Since random noise is added to the FID, the user can detect the improvement in the averaged FID during the course of the acquisition, particularly for carbon-13 measurement.

The data are transformed and displayed at the conclusion of the acquisition. The QE goes through numerous automated operations to complete the phasing and integration, many of which are skipped in the simulation since they require no operator interaction. The spectrum is plotted on any HPGL

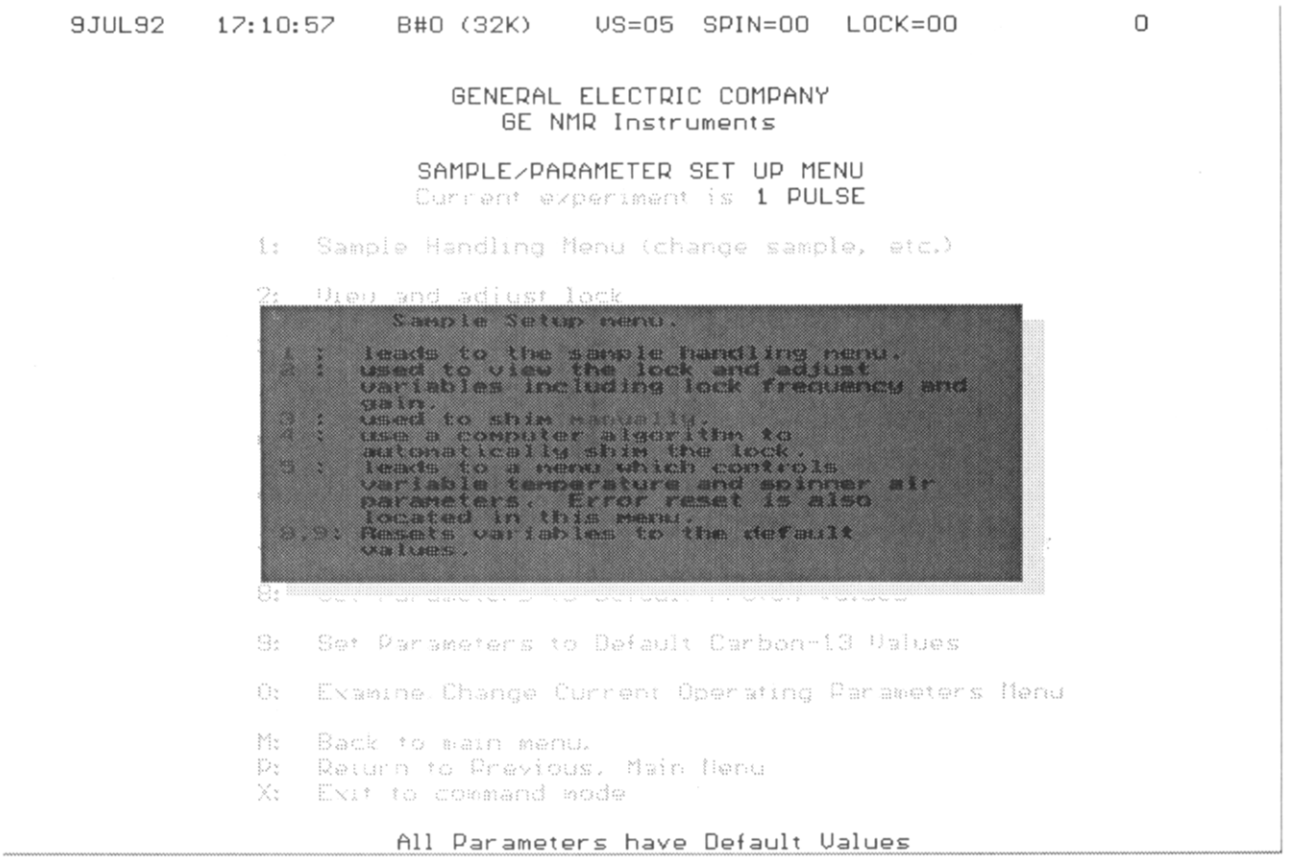

Figure 3. Context-sensitive help message in a window superimposed on a menu. The position is movable so that the menu will not be completely obscured.

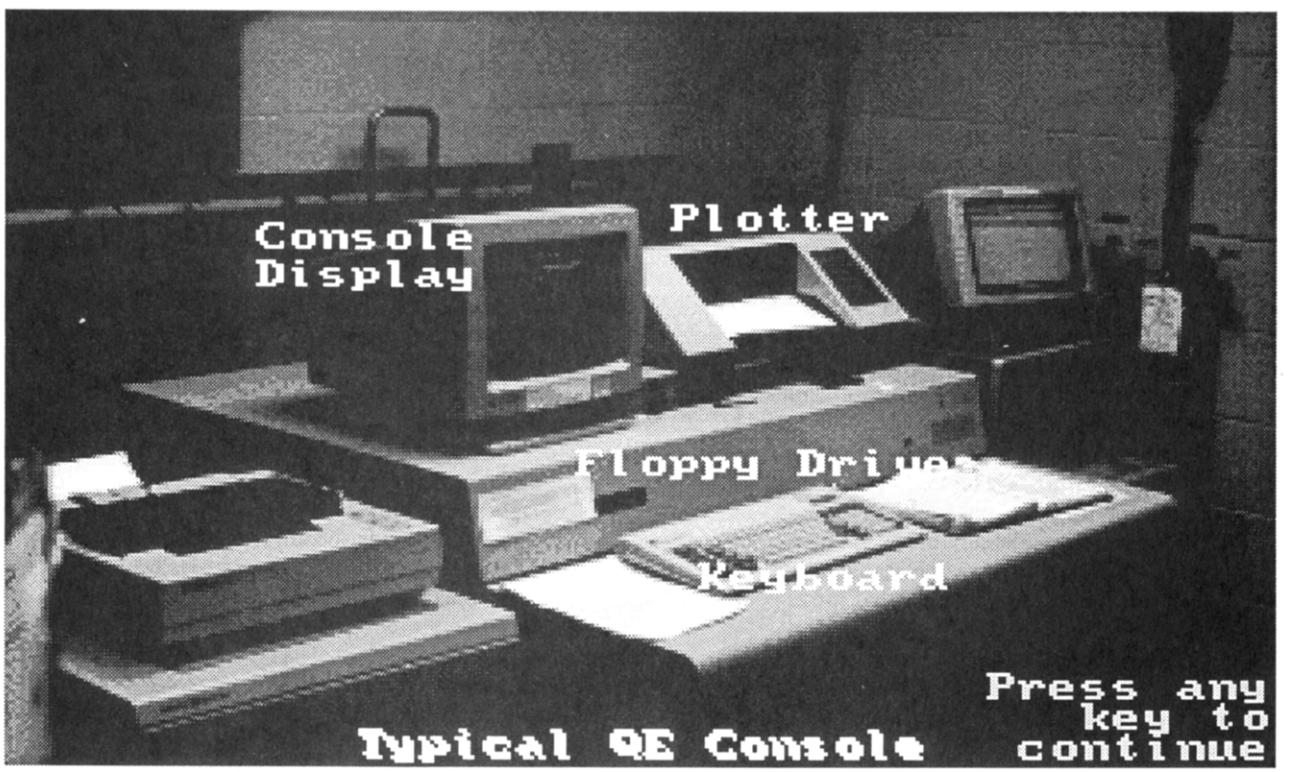

Figure 4. Screen image of the QE console, accessed as help information.

plotting device. In our laboratory, we find LaserJet printers to be practical. (Using additional software developed in our laboratories,[10] we have also been able to replace the standard plotter on the QE with a LaserJet with great success.)

After the automatic run is completed, a "View" menu allows the student to examine the spectrum interactively. A portion of the spectrum may be selected (the shaded portion in Figure 8) and expanded to full scale, horizontally and vertically. Since the data are real, artifacts do appear as expansion increases.

Although the QE is highly automated, operators can make errors in parameter entry, and these should be realistically simulated in the results. In particular, the effects of incorrect

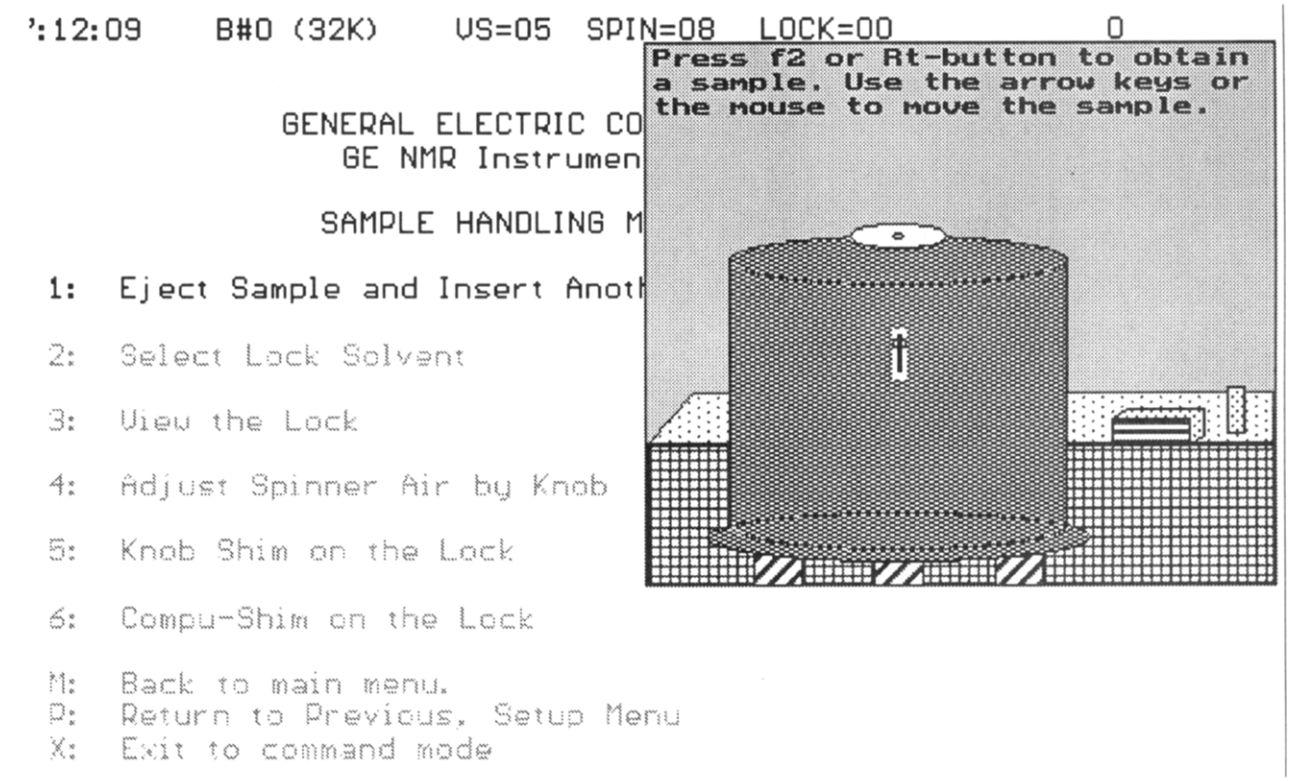

Figure 5. QSIM screen containing the magnet action window. The sample (with the spinner attached) is being moved within the magnet bore.

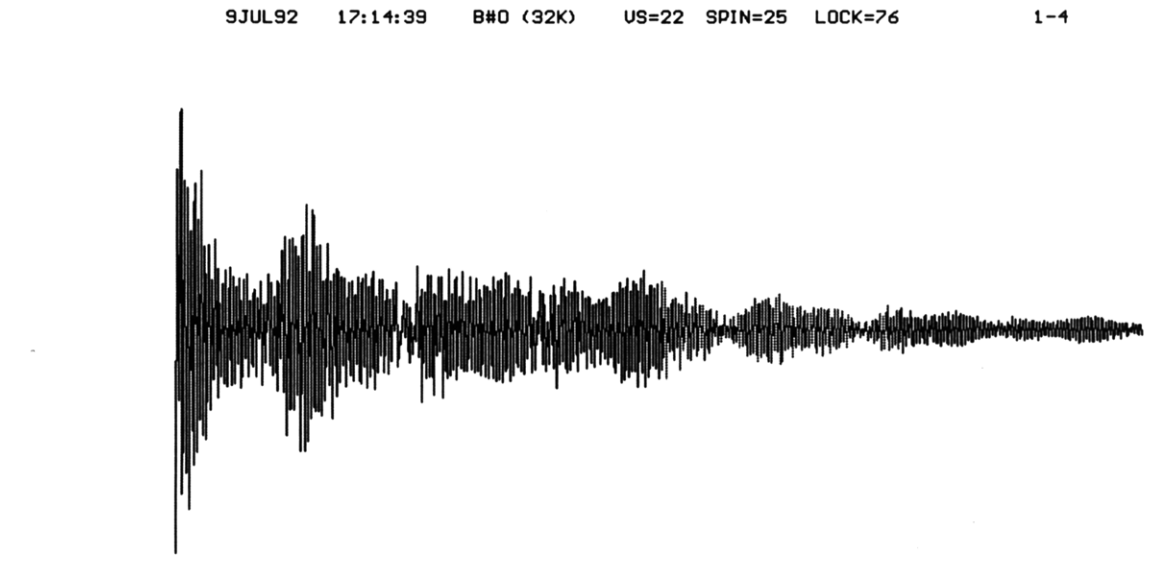

Figure 6. Typical free induction decay (FID) trace.

solvent entry will result in the proper chemical shift errors, and failure to shim properly leads to broadening of the peak. The simulation of these effects requires that the system starts with accurate data that is modified. These errors both are reflected in the results and are flagged by on-screen messages.

Noncomputer Instrument Functions. In addition to emulating the QE, the simulator must support the simulation of non-keyboard operations, the foremost of which is the introduction of the sample into the magnet. Activities which are not seen on the QE screen are easily distinguished by placing them in shaded boxes or windows which are tiled. Those windows that contain simulation of these activities are *action windows*.

The action window is used for an animated simulation of an activity involving a component that is not part of the computer. For example, the user must select a sample and transfer it to the magnet. The action window for sample introduction is illustrated in Figure 5. A superconducting magnet without a sample changer is illustrated. With this window open, the sample is selected, placed in a spinner turbine, and moved to the top of the magnet using the mouse or keyboard arrow keys.

Help Windows. The first type of help window is that already visible in Figure 1: the bottom line. That message provides some guidance to the available options: which key to use to obtain help screens, etc. The second help window provides context-sensitive help. At any time the user, by pressing a function key, can bring up a help window containing a message pertaining to the current operation; such a message is shown in Figure 3. Currently, there are over 50 such messages

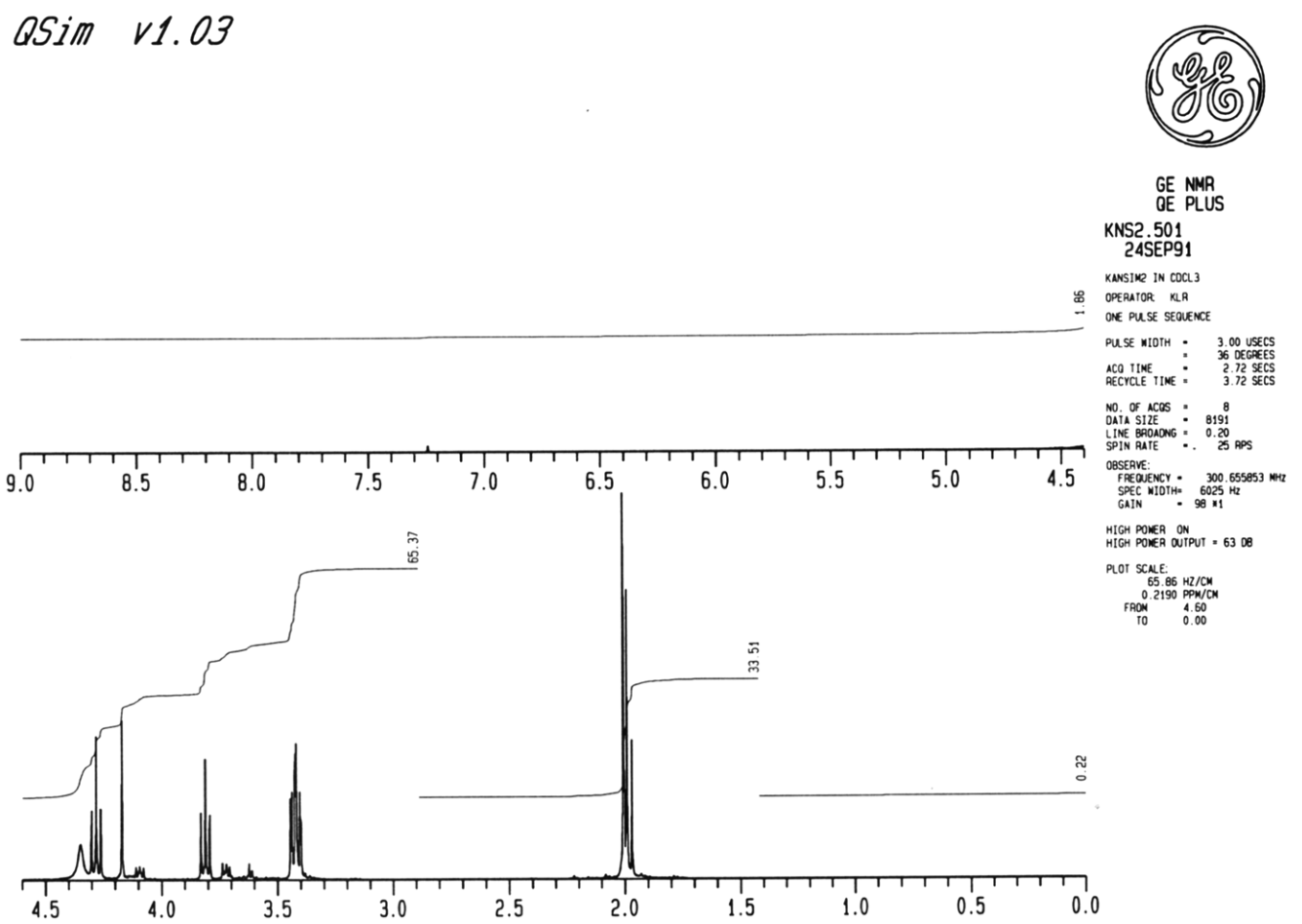

Figure 7. Typical spectrum plotted by QSIM, the proton spectrum of 2-methyl-3-pentanone.

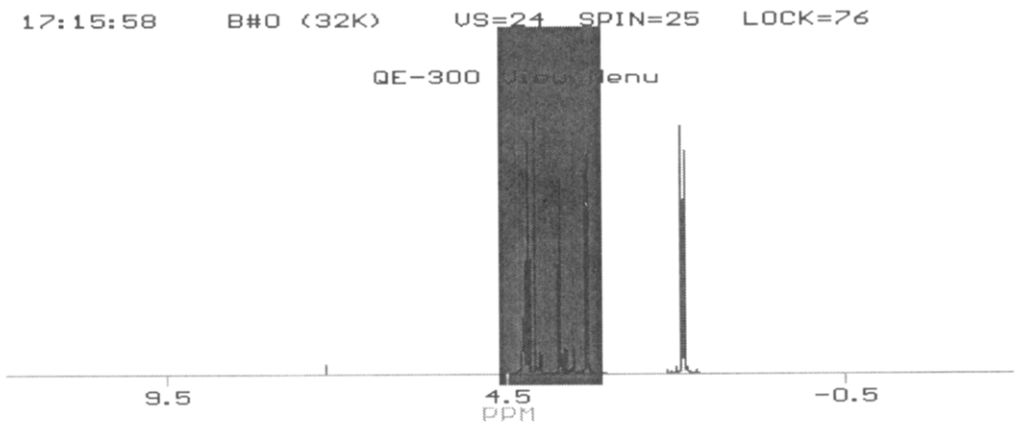

Figure 8. "View" screen. A shaded area can be selected and expanded in both dimensions.

available. If the message covers an important part of the simulator screen, it can be moved. Similarly, general instructions for QSIM can be summoned by pressing F1. Additionally, after the help screen has been displayed, one can exhibit a photograph of the instrument console. Like the menus, the help messages can be edited by the instructor who may wish to change the information or explanation presented.

Error Messages. When errors in data entry or keyboard sequences are made on some instruments (including the QE), the user is at best provided with a cryptic message or message number; often the system will crash or hang. In the simulator, the student may be "let off the hook" graciously after receiving the appropriate error message.

RESULTS AND CONCLUSIONS

QSIM has been in use at The University of Kansas and other sites for about 1 year. Instructors have described students as being attentive and eager to use the software, sometimes returning to "brush-up" before they receive actual console time. New versions will make greater use of the ability to embed photographs of the instrument and its components. Further studies of student use are underway.

ACKNOWLEDGMENT

The authors wish to make the following acknowledgements: Professor Jack Landgrebe helped define the project and test the software; Tom Peters consulted on the programming. This work was supported in part by a grant from the

NSF Instrumentation and Laboratory Improvement program. The libraries for the installation of the real-time event simulator and for the context-sensitive help manager are available from K.L.R.

REFERENCES AND NOTES

(1) Rittenhouse, R. HPLC—An Instrument Simulator. *J. Chem. Educ.: Software* **1989**, *1B*, 2.
(2) McWilliam, I. G. An Instrument Simulator. *Chem. Aust.* **1991**, *58*, 118.
(3) Wohltjen, H.; Dessy, R. An instrument simulator for use in computer interfacing laboratories. *J. Chem. Educ.* **1979**, *56*, 153–6.
(4) Ratzlaff, K. L.; Peters, T.; Hiebert, A., manuscript in preparation.
(5) Schatz, P. "NMR Simulator" and "IR Simulator"; Falcon Software, Campton, NH.
(6) Available from the Instrumentation Design Laboratory, Department of Chemistry, The University of Kansas, Lawrence, KS 66045.
(7) GE NMR Instruments, Fremont, CA.
(8) Ratzlaff, K. L. *Introduction to Computer-Assisted Experimentation*; Wiley: New York, 1988; pp 378–84.
(9) QXC, available from the Instrumentation Design Laboratory.
(10) QZ2HP, available from the Instrumentation Design Laboratory.