LBL-5577

A Control System Oriented Human Interface

P. Barale, V. Jacobson, R. Kilgore,
D. Rondeau

Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

November, 1976

This work was done with support from the U.S.
Energy Research and Development Administration.

Phenomena observed in collisions of beams of high energy heavy ions caused both nuclear physicists and medical researchers to desire a better source of these particles. Two separate particle accelerators were in use at U.C.'s Lawrence Berkeley Laboratory. One, the Super Hilac, was capable of accelerating a broad range of heavy ions with good intensity but inadequate energy. The other, the Bevatron, could accelerate beams of moderately heavy ions to very high energy but only at very low intensity. It was decided to marry these two machines in order to achieve the desired high energy, high intensity beam. The complexity of controlling the two machines simultaneously made the use of computer control mandatory.

As we began the initial design we ran into considerable opposition from the existing operating staff. Some of the objections arose from normal resistance to change. Others, however, had very valid bases. Sorting the sour grapes from the constructive criticism was not always easy.

We soon realized that an interface between the human operator and the computer would have to be designed, which would keep the computer as transparent as possible. The Bevatron had been under partial computer control for several years. We, therefore, had already gained some experience in this field. The first objection we encountered was the control device.

In charged particle accelerators the predominate tuning device is an electromagnet. The Bevatron alone presently uses over 80 magnets to steer the beam to the users. Another 80 are used to guide the beam from the Hilac into the Bevatron. Each time a different ion is desired the operator must begin at the upstream end of the beam line and adjust the current in each of these magnets for proper beam steering and focus. Once the line is "tuned", the current

levels of each magnet are stored on a disc so the next time that particle is needed the line can be brought up almost instantly. Fine tuning must go on continuously, however, due to minor parameter changes. When we first considered computerizing the beam line, we thought it should only be necessary to supply the operators with a normal terminal. After all, all the operator would have to do would be: type in the number of the magnet he desired to adjust, and the computer would reply with the present value of that magnet's current; the operator would then type in the new value and check the results on the beam position display! It didn't take long for us to realize that this was too cumbersome.

Our first attempt at human interface consisted of four thumbwheel switches, four push buttons and a roto-switch knob. The thumbwheel switches were used to select the magnet to be adjusted. The four push buttons were labelled active, save, restore, and zero. The "active" button opened a channel to the computer which allowed the operator to adjust the selected magnet with the roto-switch. The roto-switch was, of course, actually interfaced to vary the value of a location in memory. This value was periodically transmitted to a DAC which in turn controlled the magnet current. The computer's speed was fast enough that the operator felt he was controlling the magnet directly, thus achieving the computer transparency we were seeking. The remaining three buttons gave the operator a tool he had not had prior to the computer installation. At the instant the active button was pushed, the computer program copied the active list of magnet current values into a save list. If, while tuning, the operator inadvertantly went too far and lost track of the beam

completely, he had only to push the "restore" button. and the computer would fetch the original list. If, however, he improved the beam, he had only to push the "save" button and the adjusted value became a part of the permanent list. The "zero" button simply cleared the entry,driving the magnet to zero current.

The operators accepted the roto-switch feature quite readily. It gave them the same general "feel" they had become used to in using a hellipot to adjust currents. The activate, save, restore features were also readily accepted. The selection of magnet via thumbwheel switch, however, drew considerable criticism. It simply took too long to change from one magnet to another. It is often necessary to switch back and forth between several magnets to achieve optimum steering, and they had been used to a separate potentiometer for each magnet. The controller had general overall approval despite this drawback, and it is, in fact, still used to control the older portion of the Bevatron.

When design for control of the new beam line connecting the Hilac to the Bevatron was started, we attempted to improve even further on human interface. The first area we tackled was obviously the magnet selection. This problem was further compounded by the fact that not only was the beam line to be put under computer control, but also the entire Hilac. This meant that the new device must not only be fast but also very versatile. We finally decided on the control consol shown in figure 1. A 'menu" list, consisting of the names of sixteen possible display patterns, is written on the right hand edge of the monitor. The operator can call up any one of the listed displays by depressing

the pushbutton adjacent to it. Since the displays are determined by software, they can easily be adjusted to match any control configuration.

We divided the accelerator down into major sections; each display lists the controllable devices in a particular section. The device selected is intensified on the monitor screen. One of the top two knobs shown on the control panel just below the monitor in figure 1 allows the operator to step this intensified line through the list of devices. When the intensified line is on the device he wishes to adjust, he can push one of the "A" (activate) buttons on the left keypad. The computer will then vary the parameter of the selected device. Should the operator wish to have two devices active at once, he moves the intensified line over the second device and pushes the other "A" button. The remaining bottom knob will then "attach" to the second device. The save and restore features of the earlier controller are called in with the "S" and "R" buttons respectively. An "attached" knob can be disconnected with the respective "D" button. A section may contain more controllable devices than can be displayed on the available space on the screen. In this case, all of the devices are listed on the program but only enough to fill the alloted screen space are printed. The remaining knob on the panel allows the operator to scroll through the entire list. The right hand keypad allows an operator to directly punch in a device number and the new parameter value desired.

The preceding paragraph is intended only as an example. The possible ways of making displays are limitless. The four knobs each send a string of ASCII characters to the computer every 100 msecs while they are tuned. The string is formatted as follows: 1. A delimeter indicating start of string;

2. The number of the knob; 3. Three digits of magnitude information, showing the extent of rotation in the last 100 msecs; 4. A delimeter indicating end of string. Each button on the keypads and on the monitor sends a similar string identifying itself when it is pushed. It can be seen then that any knob or any button can be assigned any compatible task. Further, the control functions of both these devices (keypads and knobs) can be simulated using a standard keyboard. No fixed format, with the possible exception of the "menu" list, is dictated by the hardware.

Most standard computer terminals are too slow to give satisfactory results in this type of service. When a new list from the "menu" is called, the new data must appear instantly if the computer is to maintain its transparency. We selected an Aydin model 5205B display generator. This display generator will allow the computer to completely rewrite a monitor screen in less than 50 msecs which to the human eye is instantly.

The 5205B is also capable of controlling a color monitor. Seven colors can be displayed. Each controller can control up to four display monitors. We designed the interface to accommodate this feature, but, as will be explained later, we were forced to abandon the color monitors.

The Aydin 5205B requires a message string in the following format:

SOH	Start of heading
CA	Channel address
STX	Start of text
COMAND	Command as to type of transmission
↑	
TEXT	Text (variable length)
↓	
ETX	End of text
LRC	Longitude redundancy check
EOT	End of transmission

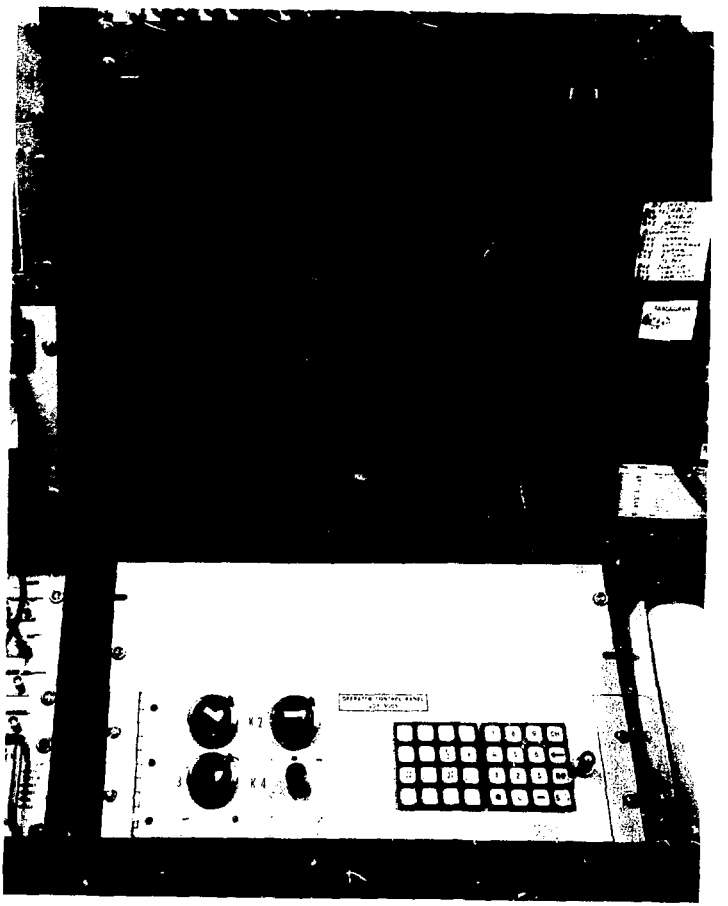
The controller assembles any messages from the keyboard in this same format before sending it to the computer. Our programmers did not want to waste computer time and space generating the LRC so this was built into the hardware in the interface. The hardware generates the LRC and the EOT signal for transmission to the controller. On transmissions from the controller, the hardware checks the LRC and generates a service interrupt, setting a bit in the status if an error is detected. Neither the LRC or the EOT from the controller is sent to the computer. The message string is in 8 bit ASCII with the MSB being a parity bit. Our interface also adds this to the computer's output and checks and deletes it from the controller's input to the computer. We also accept words from the computer packed two characters per word and pack the characters from the controller before sending them to the computer. We managed to squeeze all this onto a Mod Comp 4801 general purpose controller card. Either DMP transfer or register I/O can therefore be used. We, however, do everything DMP.

Actually, the interface was designed and built to control 2 Aydin controllers. We are thus able to have 8 different display stations operating through a DMP channel. In practice we found that so far we have needed no more than two stations at one time. The entire system has not as yet been converted over to the Mod Comp, but I now feel that four channels of display is all we will ever use.

The selection of Aydin controllers was partially influenced by the fact that they are capable of controlling color monitors as well as black and white. We had originally planned to make use of this feature to make graphic displays

of the controlled areas. We had also planned to have as many as three knobs active at once, distinguishing which knob controlled which parameter by use of colored text. When the colored monitors were installed, however, we were besieged with complaints that the definition of the text was so bad it was impossible to read. Several different color monitors were then brought in for evaluation but none proved satisfactory. The Aydin controller allows a command which will blink a section and also a command which will write a section in reverse background. We then built a simple circuit to allow adding the three color outputs from the controller. In this way we can get up to seven shades of gray for display. From a practical standpoint, however, it's hard to distinguish much more than three unless they are immediately adjacent.

The Aydin controller also accepts a keyboard input. In our original design we transmitted the output of the knob panel through the Aydin keyboard. The operators, who were, at first, quite impressed with the control consol, soon began to complain that they often would turn a knob or punch a button and get no or improper response. The problem turned out to be lack of "handshaking" between the Aydin controller and keyboard. Perhaps in normal use this would not be too objectionable. In our use, however, we are constantly updating the screen to make sure that all of the displayed parameters are current. If a knob was turned or button pushed just as the computer started an update transmission, the control panel data was lost. We found no easy fix for this and have since abandoned the use of the Aydin keyboards. We now use the Aydin controller strictly to drive the displays. The knob panel is now sent to the computer via an asynchronous control card. As the Aydin keyboard requires the controller, we found it necessary to purchase new keyboards.



CBB 760-10765

Figure 1