FEEDBACK SYSTEMS IN THE SLC*

K.A. Thompson, R.K. Jobe, R. Johnson, N. Phinney
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

Abstract

Two classes of computer-controlled feedback have been implemented to stabilize parameters in subsystems of the SLC: (a) "slow" (time scales ~ minutes) feedback, and (b) "fast", i.e., pulse-to-pulse, feedback. The slow loops run in a single FEEDBACK process in the SLC host VAX, which acquires signals and sets control parameters via communication with the database and the network of normal SLC microprocessors. Slow loops exist to stabilize beam energy and energy spread, beam position and angle, and timing of kicker magnets, and to compensate for changes in the phase length of the rf drive line. The fast loops run in dedicated microprocessors, and may sample and/or feedback on particular parameters as often as every pulse of the SLC beam. The first implementations of fast feedback are to control transverse beam blow-up and to stabilize the energy and energy spread of bunches going into the SLC arcs. The overall architecture of the feedback software and the operator interface for controlling loops are discussed.

1. Introduction

The successful operation of the SLC requires stabilization of such quantities as beam transverse position and angle, beam energy and energy spread, and kicker timing. In many regions of the machine, these quantities vary significantly on a relatively slow time-scale (~ minutes) and can be stabilized by loops controlled from a FEEDBACK process residing in the SLC host VAX. There are a total of approximately fifty such loops presently commissioned in the SLC control system.

In addition, in some regions of the machine it is necessary to stabilize some of these quantities on a pulse-to-pulse basis. For this purpose, several dedicated feedback microprocessors (at present with the same Intel 8086 architecture as in the "normal" SLC micros) are being commissioned to run fast feedback loops. The first instances of such micros are to control transverse beam blow-up and to stabilize the energy and energy spread of bunches at the entrance to the SLC arcs.

The locations and types of existing SLC feedback loops are shown in Fig. 1.

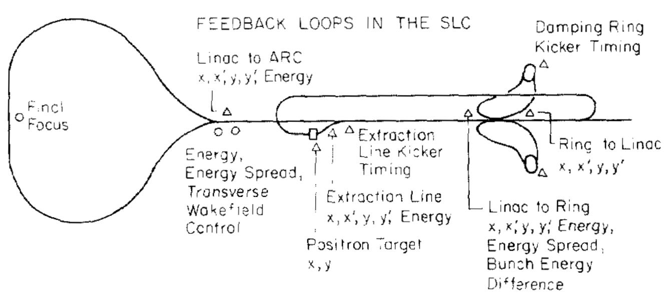

Fig. 1. Existing fast and slow feedback loops in the SLC. X, X′, Y, and Y′ denote horizontal position and angle, and vertical position and angle.

2. System Architecture for Slow Feedback

The FEEDBACK process is a standalone, batch process, which communicates with the SLC Control Program (SCP) through the SLC message service and the database. It has a group structure, with one group allotted to each set of related loops. Each group has its own application-specific driver linked into the FEEDBACK process, containing the data-acquisition and control routines used by the loops in that group. Each loop may be scheduled to run periodically, where the period is a database parameter controllable from a SCP. A group is scheduled to execute whenever one or more of its loops is due to execute. In many cases, it is desirable to have all the loops in a group be scheduled at approximately the same time, so that they can share a data acquisition and/or be cascaded (for example, the four loops stabilizing horizontal and vertical beam position and angle at some point in the machine). In other cases (e.g. the group of loops that stabilize the timing of kicker magnets) there is no reason to synchronize the loops and it may be desirable to run them with different periods.

Two important assumptions are made regarding the structure of the feedback loops. The first is that a loop may be regarded as having a single "input" and a single "output" variable. In general, a loop may acquire data from more than one source (we will sometimes refer to this data as "monitored variables") and change the state of more than one device ("control variables"). However, for the purpose of simplifying and unifying the operator interface and displays, each loop is represented in terms of two variables (having the same physical units):

1. Signal variable: There is one such variable for each loop, and it is calculated from the values of the monitored variables. The loop has a setpoint, and the FEEDBACK process tries to keep the value of the signal variable equal to the setpoint, to within some tolerance.
2. Command variable: There is one such variable for each loop, and the control variables are derived from it. Changing the value of the command variable is the means by which FEEDBACK tries to keep the signal variable near the setpoint.

The second important assumption is that all loops are designed to be as "orthogonal" to each other as possible. If one loop can significantly affect the value of the signal variable for another loop, the algorithm for the first loop is required to feed-forward to try to compensate for any change that would have been produced in the second loop's signal variable. In addition, the second loop can be scheduled to run immediately after the first, in case the feed-forward was not perfect. FEEDBACK is designed such that this "cascading" of loops is easy to implement in those cases where it is needed.

For details regarding the data acquisition and control software for slow feedback on beam position, angle, and energy, see Refs. 1 and 2.

* Work supported by the Department of Energy, contract DE-AC03-76SF00515.

CH2387-9/87/0000-0748 $1.00 © IEEE

3. System Architecture for Pulse-to-Pulse Feedback

The dedicated pulse-to-pulse feedback micros have an architecture that is as similar as possible to that of the standard SLC micros. However, in addition to some of the standard jobs (e.g., beam position monitor and timing jobs) these micros each have a feedback job that is specialized to the particular micro. The feedback job consists of a general framework common to all the micros and pieces which are customized to the data acquisition and control requirements of the particular micro.

The basic feedback job contains the following:

1. Two interrupt handlers, which the user must specialize to do calculations, CAMAC operations, etc. on a pulse-to-pulse basis.
2. A ring buffer, in which data from successive pulses may be stored by the interrupt handler. The nature and the quantity of data per pulse and the total number of pulses to be kept in the ring buffer are specified by the user. Buffer interface routines are provided to the user, to set up the buffer, add a point of data to the buffer, and send a block of data points back to the VAX.
3. A routine to receive a set of constants from the VAX, for use in the interrupt handler's calculations. These constants are loaded into a common block, the size and structure of which must be specified by the user.
4. General initialization routines, including one which the user may specialize to do initialization unique to the particular application.

The first interrupt handler is invoked at NMI (Non Maskable Interrupt) level on every pulse of the machine. It checks to see if beam code for the pulse is the one on which feedback is to be done and may do certain other things (such as send out a CAMAC package to read beam position monitors) depending on the specific application. It then triggers a normal interrupt handler to do the rest of the work (e.g., floating point calculations, which cannot be done at NMI level).

For details of the algorithms used in the feedback micros, see Refs. 3 and 4.

4. Operator Interface

4.1 Slow Feedback

The user interface for control and monitoring of feedback loops resides in the SCP. After selecting the loop of interest, the user may go to a touch panel on which information about the loop is displayed, from which loop parameters may be changed. Additional displays may be invoked on the SCP color monitor. An example of this panel in its present form is shown in Fig. 2. The loop parameters that are displayed on and may be changed from the panel include:

1. The signal and command variables
2. The setpoint.
3. The tolerances above and below the setpoint. If the signal variable is within the tolerances, the loop does not attempt to change the command variable.
4. Limits on the signal and command variables.
5. The loop gain.
6. The beam code on which the loop is to run.

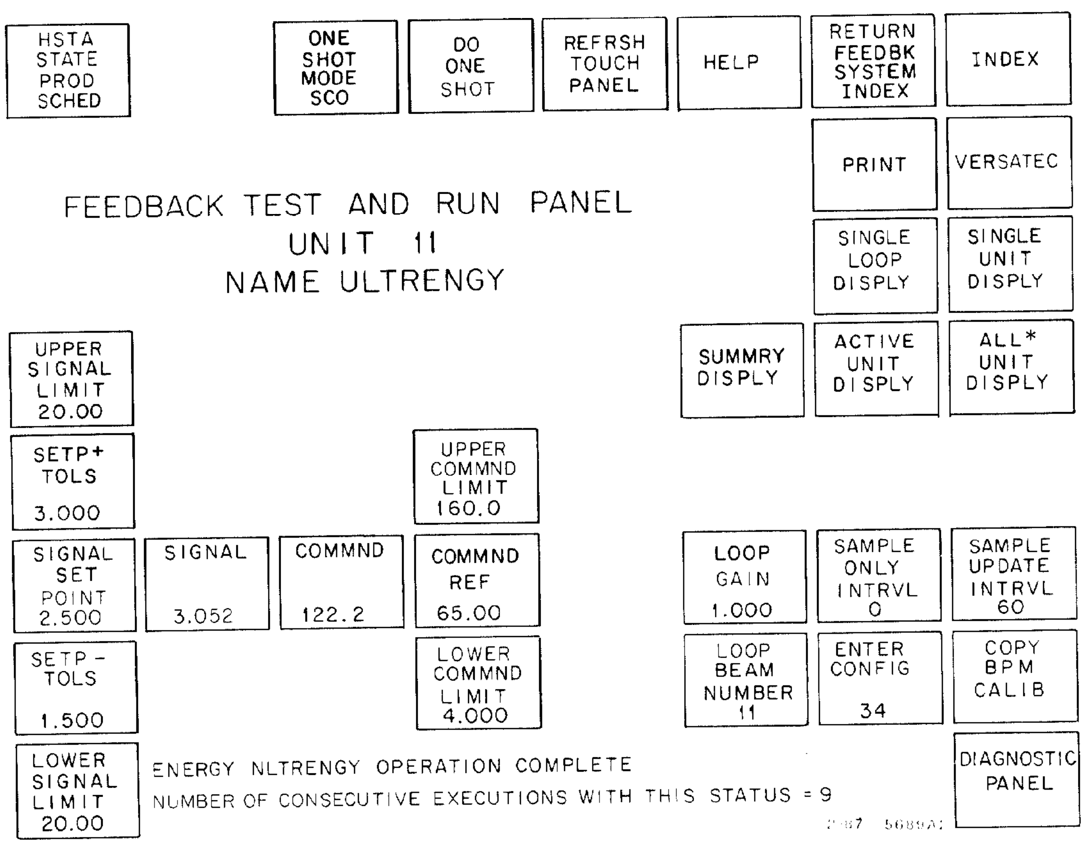

Fig. 2. The panel for controlling slow feedback loops.

7. The mode of operation of the loop. The loop can be in one of three states: (1) Scheduled: The loop runs automatically on a periodic basis, (2) Request-only: The loop only executes when the "one-shot" button is pushed, (3) Off.
8. The period on which the loop is to run if it is in scheduled mode. There is also an option to do a sample-only (data acquisition) on a periodic basis.

There are additional diagnostic tools available. One-shot execution of the loop may be done, selecting all or only a subset of the three steps: (1) Sample, (2) Compute command variable, (3) Output new control variables. Furthermore, although the production FEEDBACK process runs in batch, a second development FEEDBACK process may be run on a terminal and any loop may be assigned to run in either one of the two processes.

From this panel, the user may also call up several displays. There is a single-loop graphical display of the signal and command variables showing limits, setpoint on signal, tolerances on signal, etc. (see Fig. 3). There are also displays giving summary information about all the loops in the system (signal, setpoint, and command values, time that the loop last changed the value of the command variable, mode of loop operation, etc.)

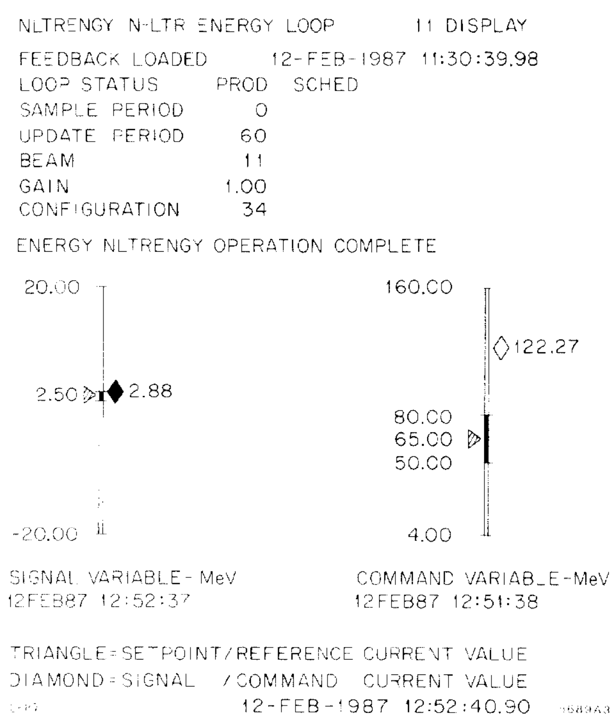

Fig. 3. The single loop signal and command variables display.

4.2 Pulse-to-Pulse Feedback

There exists a user interface program on the VAX to control the pulse-to-pulse feedback micros, i.e., enable and disable data acquisition and feedback, select beam code on which feedback is to be done, obtain a set of data from the ring buffer in the micro, and process and display the data. This interface will eventually be moved into the SCP and integrated with SCP control of slow feedback.

Acknowledgments

I. Almog, L. Sanchez-Chopitea, and H. Shoaee are responsible for the injection software used for data acquisition by the energy, position and angle slow feedback loops. Thanks are also due to J. Bogart and T. Gromme for additions to the beam position monitor software needed to implement feedback loops, and to J. Tolmasoff for help in coding the user interface.

References

1. R.K. Jobe, K.A. Thompson, I. Almog, N. Phinney, J.T. Seeman, J.C. Sheppard and H. Shoaee, "Position, Angle, and Energy Stabilization for the SLC Positron Target and Arcs," Proceedings of this Conference.
2. J.C. Sheppard, I. Almog, P.S. Bambade, J.E. Clendenin, R.K. Jobe, N. Phinney, H. Shoaee, R.F. Stiening and K.A. Thompson, "Three Bunch Energy Stabilization for the SLC Injector," Proceedings of this Conference.
3. G. Abrams, J. Seeman, I. Campisi, W. Herrmannsfeldt, M. Lee, A. Petersen, N. Phinney, M. Ross, K. Thompson, D. Tsang, C. Adolphsen and E. Soderstrom, "Fast Energy and Energy Spectrum Feedback in the SLC Linac," Proceedings of this Conference.
4. J. Seeman, I. Campisi, W. Herrmannsfeldt, M. Lee, A. Petersen, N. Phinney, M. Ross, R. Stiening, K. Thompson and D. Tsang, "Transverse Wakefield Control and Feedback in the SLC Linac," Proceedings of this Conference.