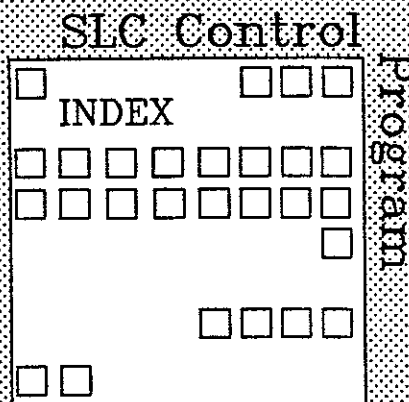


Index Panel

SLAC's Software Engineering Newsletter



March 29, 1991

All that Fits is News to Print

Vol. 5, No. 8

New Fast Feedback Release 0.97

March 11, 1991

Author: *Fast Feedback Team*
Panel Changes: *Many*

Subsystem: *Feedback*
Documents: *Many*

User Impact: *Large*
Help File: *Lots*

This article heralds the release of the long awaited and much anticipated new fast feedback system. It will be used to stabilize the beams in many places (about a dozen) in the SLC. As it is database driven and uses existing instrumentation (BPMs and correctors,) it will be relatively easy to add new loops. The feedback processes run on the 80386 microprocessors and hence can execute fairly frequently.

You get to the new fast feedback panels by going from the INDEX to the FEEDBACK SYSTEM INDEX to the FFBK SELECT PANEL. The old fast feedback panels are still available from the FEEDBACK SYSTEM INDEX via the OLD FFBK PANEL button. Note that the new and old fast feedbacks are two completely separate systems and are not at all integrated. We intend to eventually phase out the old fast feedback software.

Release 1.0 represents about 7 person-years of work. About 97% of this is completed. This includes all the essential parts. To make this article a useful reference document, it and the touch-panel help describe version 1.0. Parts which are not done yet are so noted in parentheses. We expect to finish the remainder in the next month or two. After that we will gradually add extra functionality to bring this up to version 2.0. This will take over six months. Fast feedback was brought to you by the following people: Stephanie Allison, Susan Castillo, Bob Fuller, Tony Gromme, Phyllis Grossberg, Linda Hendrickson, Tom Himel, Karey Krauter, Dave Nelson, Lee Patmore, Forest Rouse, Bob Sass, and Hamid Shoaee.

This article only gives a brief introduction to fast feedback. Extensive documentation is available elsewhere.

1. Every button on all 9 fast feedback touch panels has online help. The main help for some of the touch panels include descriptions of methods used and definitions of some terms.
2. The **System Requirements for Fast Feedback** document tells what feedback is intended to do. To print a copy when logged on to the SLC VAX,
IBMSERVE DOC\$FUNC_REQ:FBCK_SYSTEM_REQ.TEX IMMCC1 /SLACNET /TEX
3. The **Fast Feedback System Design** document gives a top level view of how fast feedback works. To print a copy when logged on to the SLC VAX,

IBMSERVE DOC\$DESIGN:FBCK.SYSTEM.DESIGN.TEX IMMCC1 /SLACNET /TEX

4. The Fast Feedback Users Guide gives an overview of fast feedback and gives some specifics about how to operate it. At present it just contains this index panel article. We will gradually update and expand it as needed. To print a copy when logged on to the SLC VAX,

IBMSERVE DOC\$USER:FBCK.USERS.GUIDE.LATEX IMMCC1 /SLACNET /LATEX

5. **Fast Feedback: The First Eight Loops and Fast Feedback: The Next Four Loops** give a slightly out-of-date description of the first loops which will be implemented. A more up-to-date (but less detailed) list is given later in this article. To print a copies when logged on to the SLC VAX,

IBMSERVE DOC\$FUNC.REQ:FBCK.FIRST.LOOPS.TEX IMMCC1 /SLACNET /TEX

IBMSERVE DOC\$FUNC.REQ:FBCK.SECOND.LOOPS.TEX IMMCC1 /SLACNET /TEX

6. Many more documents giving the detailed design have been written. Few people will be interested in these. A list of these documents can be obtained with the following command.

IBMSERVE DOC\$FEEDBACK:.CONTENTS IMMCC1 /SLACNET

An Introduction to Fast Feedback

The general purpose of feedback is to obtain some MEASUREMENTS using SENSORS, to calculate some STATES of the system, and then to stabilize these states to a desired SET-POINT by setting some ACTUATORS. For the most common SLC application, the measurements are of the beam position at several points along the beam line; the sensors are Beam Position Monitors (BPMs); the states are the position and angle of the beam; and the actuators are correction magnets.

A secondary purpose of fast feedback is to provide information to the operator about how well it is working. The main mechanism for this is the display of RING BUFFERS. A ring buffer is a record of the value of a vector element (e.g. a BPM reading or corrector setting) for the last N cycles of feedback. Typically about the last 1000 pulses are recorded. You can then display one reading versus another or one versus time. You can also plot projections and Fourier transforms.

A single feedback loop uses many measurements which are grouped together into a MEASUREMENT VECTOR to estimate and control several states (e.g. position and angle) which are grouped together into a STATE VECTOR. Several actuator settings which are grouped together into an ACTUATOR VECTOR are then calculated and applied to the actuators (e.g. correctors) to try to move the STATE values to the desired set-points. In short, a single fast feedback loop acts like a group of slow feedback loops. For example there is a single fast feedback loop in LI03 which reads 13 BPM's for both e^+ and e^- in x and y . It controls 8 states: electron x position, electron x angle, electron y position, electron y angle, and the same for positrons.

Fast feedback is implemented using the standard SLC hardware. The correctors and BPMs it uses are shared with other SLC users. Since it executes at a high rate (20-60 Hz) most of the work is done on the 80386 microprocessors. The VAX is only used to monitor and control (turn on and off and set limits and set-points) the loops.

In case there are problems, it helps to understand the basic structure of the micro software which is illustrated in the figure below. The BPMJOB reads BPMs at up to 120 Hz. Data from a beam pulse can both be sent to the VAX for use in a BPM display and it can now be passed to the FMES micro task. FMES checks the measurements, saves them in ring buffers and maintains a running average for possible later display on the VAX and then passes the MEASUREMENT VECTOR to the controller task, FCTL. FCTL uses the measurements to calculate the STATE VECTOR which contains its best estimate of the present values of the STATES (e.g. the position and angle of the beam). It saves these states in ring buffers and maintains a running average for possible later display on the VAX. FCTL next uses the STATE VECTOR and the previous ACTUATOR VECTOR to calculate the new ACTUATOR VECTOR. It sends this to the actuator task, FACT. FACT uses the actuator settings in the vector and does the CAMAC operations necessary to set the actuators. It also saves the actuator settings in ring buffers and maintains a running average for possible later display on the VAX. It sends a status back to FCTL.

The hardware used by a feedback loop is likely to reside on more than one micro. For example the LI02 electron loop has BPMs in LI02 and correctors in DR13. To handle this we must send data (e.g. the ACTUATOR VECTOR) from one micro to another at up to 120 Hz. This is handled with a new network dubbed KISNET (for Keep It Simple NETwork). This consists of simple point-to-point links which run on twisted pair for short distances (< 150 meters) or fiber optic cable for longer distances. The electronics for KISNET consists of a 2" x 3" PC board which attaches to the SBX connector on the 80386 microprocessor board. If a micro has more than one link, then a new (commercially made) multibus board with 6 SBX connectors is inserted in the micro's multibus crate. The DCL command KISTEST can be used to test a KISNET link. Note that this will interfere with the running of fast feedback.

Description of the Panels

There are a total of 9 panels associated with the fast feedback system. For ease of operation some buttons appear on several panels. In general if two buttons have the same label then they do exactly the same thing. Short descriptions of the panels follow.

The FEEDBACK SELECT PANEL allows you to select a feedback loop, turn it on and off, change the gain, and look at a few predefined plots and summary displays.

The CANNED PLOT PANEL allows you to look at many predefined (in the data base) plots to check the performance of fast feedback.

The STATE PANEL allows you to change the setpoint and tolerances for the states (e.g. position, angle and energy) controlled by a loop. The setpoints can be assigned to knobs (not done yet). It allows history buffer displays.

The GOLD ORBIT PANEL allows you to save a gold orbit to disk in a configuration file and to load a gold orbit from disk for feedback to use.

The MAGNET PANELS PANEL just gives quick access to the regular magnet panels that have magnets used by fast feedback. For example you can get to the NRTL corrector panel from there.

The CUSTOM PLOT PANEL allows one to plot anything versus anything by typing in the desired variables for each axis. Hopefully plots which are needed frequently will be available on the CANNED PLOT PANEL so this panel will not be used much.

The following panels will not be used as frequently and are not directly accessible from this main fast feedback panel. They can be reached from the STATE PANEL among others.

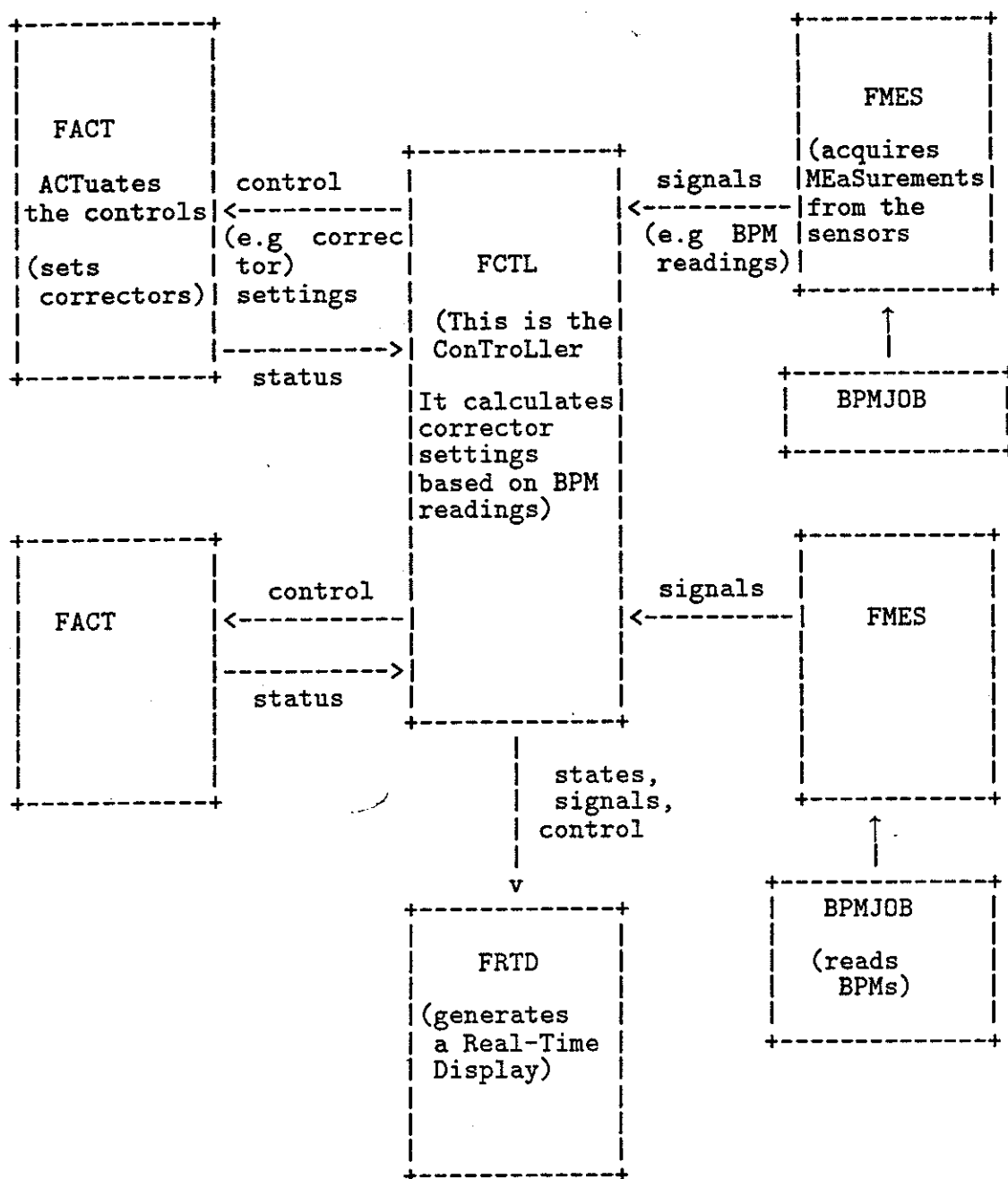


Figure 1. Block diagram of the micro part of feedback

The DIAG + CALIB PANEL allows one to run feedback in various diagnostic modes. The most important of these is the CALIBRATION, which varies an actuator and measures how the states change.

The ACTUATOR PANEL allows one to change tolerances and limits for the actuators (e.g. the maximum magnetic field that is allowed for a corrector magnet).

The MEASUREMENT PANEL allows one to change tolerances and limits for the measurement devices (e.g. the limits for which a BPM reading will be considered nonsense and hence not used).

Functional Description

In this section a bit more detail is given about what feedback does and how it does it. The emphasis is on the control methods. Information about the displays and user interface are best obtained from the online help.

1. The basic action of fast feedback is to read sensors (*e.g.* BPMs), calculate states (*e.g.* x , x' , y , y' , *Energy*) and then calculate and implement new actuator (*e.g.* corrector) settings to try to stabilize the states at desired setpoints. The basic feedback algorithms assume everything is linear (*e.g.* the reading of a BPM will change linearly as a function of corrector setting).

There are many details though, which actually turn out to be most of the work.

2. Feedback can make use of the present hardware (*e.g.* BPMs and correctors) and is integrated into the present control system. Old fast feedback required all of the control hardware to be duplicated.
3. When a feedback loop is on, the actuators it uses are flagged. They will show up white (like no-control magnets) and be labelled **FDBK CTRL** on the magnet displays. Also, the magnet job will not allow them to be trimmed etc.
4. New loops which are similar to old ones can be added without adding new code. This can be done because fast feedback is database driven.
5. In Version 2 (V2) loops will be able to minimize a state (*e.g.* energy spread). Another way to look at this is that they will be able to force the derivative of a state with respect to an actuator setting to a desired setpoint. (Not done yet)
6. An actuator can be any magnet-like device except for movers such as an XCOR, YCOR, LGPS, AMPL, and PHAS. In V2 timing devices (to allow adjustment of the PSK time for the positron energy loop) will also be controlled.
7. A sensor can be any BPM-like device such as a BPMS, DUGADC, ARRY, or TORO. In V2 the read out of profile monitor and wire scanner data will be available.
8. Feedback can use multiplexed BPMs. This allows us to use fast feedback in many places without having to install new BPMP's. Note that the LINAC BPMs are effectively multiplexed between e^+ , e^- , and the scavenger bunch. Where high speed is essential, we can of course still demultiplex signals.
9. The typical launch (controls beam position, angle and sometimes energy) loop will have two to three times as many BPMs as needed to determine the states. This provides some redundancy. The loop continues to work with some fraction of the signals missing. The maximum allowable fraction is set in the database.
10. Measurements from beam pulses which are vetoed by the klystron veto system are not used by feedback.
11. Saturation of an actuator (*e.g.* a corrector is at its maximum value and the controller wants it to go higher) is gracefully handled. In particular, the operator is notified if the condition persists for more than a 8 seconds and feedback still properly controls those actuators which are not saturated.

12. GOLD ORBIT CONFIGURATIONs can be saved. This works much like it does in slow feedback. That is the present readings of all the relevant measurements (*e.g.* BPM readings) are saved to a configuration disk file. Old versions of this file can be used to restore the loop to a previous state.
13. When a loop is controlling two beams (*e.g.* most of the linac loops control both the electrons and positrons) and only one of the two beams is present, feedback will still control the beam that is present, leaving the other one alone. An allowed exception to this is when an actuator is saturated. In this case feedback may change the unmeasured beam in order to correct the measured beam.
14. Most of the control mechanisms (filtering, transfer matrices, gains, etc.) are set up offline by filling in matrices used by the controller. Just in case that setup isn't perfect, a single gain constant per loop is available for change from the SCP. This determines the fraction of the calculated actuator change which will actually be applied.
15. A loop will continue to work properly (without operator intervention) as the beam rate varies from 1 Hz to 120 Hz. Obviously its bandwidth will be less at the lower beam rate but it should remain stable and not need to have its gain changed.
16. The system can do a *calibration*. This involves varying each actuator one at a time and noting how the sensor measurements change. This data is then used to determine some of the control matrices. The part which allows feedback to use the measured transfer matrices is not done yet. The measurement and display part is done and can be used as a diagnostic.
17. The structure of the program allows for a clean way to implement "special cases" such as the following.
 - (a) The scavenger energy loop changes the phases of two sectors of the linac in order to change the energy. The calculation of these phases is nonlinear and hence will require a special purpose routine. In addition some special initialization is needed, namely the total energy gain of each of the two sectors. (This is not done yet. It is started and is proceeding at high priority.)
 - (b) The interaction point loop tries to keep the beams colliding by measuring the deflection angle. The conversion from measured deflection angle to beam separation involves a division by the beam intensity. This is nonlinear and hence presumably a special case. This is not done yet and is not considered high priority as FB69 feedback works pretty well.)
 - (c) Separate control of the linac positron and electron energies involves changing the PSK time for all the klystrons in the linac. The relation between the PSK time and the energies is nonlinear and hence is presumably a special case. (This is not done yet. This control will probably remain under slow feedback for quite some time.)
18. The micros accumulate ring buffers of relevant data such as BPM readings, actuator settings, and state variables (x , x' , E). The ring buffer contains the value of that variable for the last N beam pulses. The value of N and which variables should be accumulated are changeable with a DBEDIT. The VAX can request and display these buffers. The ring buffer data is synchronized (Exact synchronization is not done yet). That is the M^{th} point shown in a sensor reading plot will be the number that was used to calculate the M^{th} point in a state ring buffer plot. These ring buffers can be used for debugging and diagnostic purposes. They are available even while a calibration is in progress.

19. Ring buffer data can be saved to disk in a form that MATLAB can read (not done yet).
20. When a micro is IPLed, the necessary data is automatically down-loaded to it, and if its feedback is on it is automatically started up.
21. When a BPM is turned on or offline from the BPM diagnostic panel feedback will be automatically informed and will start or stop using that BPM.
22. We can add a limited number of states, actuators, or sensors to an existing loop with less hassle than involved in a DBGEN. This is done by leaving spare units in the database which can be turned on with a DBEDIT. Then the controller micro is IPLed and a COLD START is done for the loop to all the micros to re-read their part of the database and re-initialize their structures. Note that turning a loop off and on again does not completely re-initialize a loop; the COLD START button can be used for this.
23. The micros make periodic (every 60 seconds or so) asynchronous database updates of the loop status, measurements, states, actuator settings and so on. This is done using the standard asynchronous database tools (that is instigated by the micro, not the VAX). The data is used by the history buffer program (not done yet,) and SDS display.
24. Most of the feedback loops assume that correctors within their range of BPMs are not changed. If they are changed, one will need to save a new GOLD ORBIT. For a few loops this will be too inconvenient (NFF and SFF have correctors used for the CCS eta correction bump in the range of BPMs used by the feedback loop). For these cases, feedback can be set up to get the field strengths (as read by the regular magnet job) and use them in the calculation of the states (e.g. x , x' , energy). This is done in a manner similar to slow feedback. It is not done yet.
25. The loop setpoints (for states) can be set from a touch panel. For example this allows one to request that a loop maintain the beam energy 50 MeV higher than it was when the gold orbit was saved.
26. The setpoints can be assigned to a knob and adjusted. This may be useful in minimizing backgrounds or tuning some other parameters not directly controlled by a feedback loop. This is not done yet.
27. The necessary subroutines to interface to correlation plots have been written. This includes the ability to change a setpoint, "one shot" a loop (let it run for long enough to converge), and attain the value of a state averaged over several pulses. The modification to the correlation plots user interface to take advantage of these routines is not done yet.
28. Interface routines are provided to allow other SCP software to attain the present HSTA (ON, OFF, Sample Only) of a loop and to change the HSTA. This interface looks very much like the existing interface routine used for slow feedback and V0 fast feedback.

Loops Being Implemented

We intend to implement many loops using this new fast feedback system. At this time, none of them have been fully commissioned. Here we list the loops roughly in the order that we expect to commission them.

1. LI03 e^+ and e^- : This reads BPMs in the last three quarters of LI03 and adjusts the first few correctors in LI03 to correct the launch positions and angles. All hardware is in the same micro which makes this loop particularly simple to test. That is why it will be done first. This loop was already tested fairly well in November 1990.

2. LI02 e^- : This reads BPMs in LI02 and adjusts correctors in the NRTL to correct the launch positions and angles of the electron beam. While loop 4 is more urgent as far as machine operation is concerned, this loop is simpler to implement and testing will be much less destructive since if loop 4 goes astray, the scavenger line MPS will trip off the accelerator.
3. LI02 e^+ : This reads BPMs in LI02 and adjusts correctors in the SRTL to correct the launch positions and angles of the positron beam.
4. Scavenger energy: This loop adjusts the energy going into the scavenger extraction line in sector 19. This is the loop most needed by operations. It needs some special software for the phase shifter control which is not done yet. Note that this loop is intended to work hand-in-hand with the new fast feed forward system.
5. NLTR and SLTR launch and energy: These are very important loops but also very complicated. They control three beam energies (electron, positron, and scavenger) and eight positions and angles (position and angle for electrons and positrons in the vertical and horizontal). Everything is highly coupled. In contrast to all of the above loops, some of these have never been run in slow feedback. We intend to first implement just the electron energy loop and then gradually add control of other parameters.
6. LI04 e^+ and e^- : This will be just like the LI03 loop put one sector down-stream.
7. LI06 e^+, e^- : This reads all the BPMs in LI06 and LI07 and adjusts correctors at the beginning of LI05 to correct the launch positions and angles of the luminosity electron and positron beams.
8. LI11 e^+, e^- : This reads all the BPMs in LI11 and LI12 and adjusts correctors at the end of LI10 to correct the launch positions and angles of the luminosity electron and positron beams.
9. LI18 e^+, e^- : This reads all the BPMs in LI18 and LI19 and adjusts correctors at the end of LI17 to correct the launch positions and angles of the luminosity electron and positron beams.
10. LI23 e^+, e^- : This reads all the BPMs in LI23 and LI24 and adjusts correctors at the end of LI22 to correct the launch positions and angles of the luminosity electron and positron beams.
11. Replace FB29: Hopefully the new feedback system will have a faster response time than the FB29 software. It also should be more robust. Hence it will be worthwhile to replace this important loop.
12. NFF: this reads BPMs in the north final focus and adjusts correctors at the beginning of the NFF to correct the launch positions and angles of the electron beam into the rest of the NFF. It also will measure, but not adjust, the energy.
13. SFF: Same as the NFF but for the south final focus.